CS691 – Computer Science, Spring 2020

Pace University



SYSTEM TEST PLAN

Handy Helper

Authors: Yixuan Ma, Veenus Thakkar

Project Manager: Jack Brody

Date: 04/14/2020

**Table of Contents**

[**INTRODUCTION**](#_30j0zll) **3**

[**1. TESTING SCOPE**](#_3znysh7) **3**

[**2. TESTING OBJECTIVES**](#_tyjcwt) **3**

[2.1 Core Features to be Tested](#_1t3h5sf) 4

[2.2 Non-Functional Features to be Tested](#_2s8eyo1) 5

[2.3 Features not to be Tested](#_3rdcrjn) 6

[**3. TEST PROCESS DEFINITION**](#_lnxbz9) **6**

[3.1 Test Process Phases](#_1ksv4uv) 6

[3.2 Testing Tasks and Deliverables](#_2jxsxqh) 7

[System Test Plan document](#_z337ya) 7

[Role and Responsibility Table](#_3j2qqm3) 7

[**4. APPROACH TO SYSTEM TESTING**](#_4i7ojhp) **8**

[4.1 Approach to Functional Testing](#_1ci93xb) 8

[4.2 Approach to Non-Functional Testing](#_2bn6wsx) 8

[**5. ENTRY/EXIT CRITERIA**](#_3as4poj) **9**

[5.1 Entry Criteria](#_1pxezwc) 9

[5.2 Exit Criteria](#_49x2ik5) 9

[**6. SYSTEM TEST ENVIRONMENT**](#_147n2zr) **9**

[**7. ROLES AND RESPONSIBILITIES**](#_23ckvvd) **10**

[**8. TEST CYCLES AND SCHEDULE**](#_j1syh5u3hif0) **1**3

[**9. RISKS AND CONTINGENCIES**](#_41mghml) **1**3

# INTRODUCTION

This document is for the establishment of the System Test Plan that provides a common understanding among the “Handy Helper” project stakeholders on the following aspects, including the scope, objectives, and approach to performing the system testing. Moreover, 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 mainly consists of two aspects, which are the functional scope and technical scope.

In line with our expected functions of our system, the scope of system testing includes following functional aspects:

* Build Profile
* User’s Account
* Searching
* Listing a job
* Messaging
* Booking
* Reviewing

The corresponding modules in the Handy Helper system are: user account, build profile, and booking modules.

The technical scope includes the following components that are associated with the expected functions:

* Web browser
* Application server
* Database server
* Content server

# 2. TESTING OBJECTIVES

This testing objective is to validate the implementation of the system in order to measure whether both functional and non-functional performance can match the designed requirements. In this statement of objective for system testing, the target testing objectives will be clearly stated in order to distinguish them from those objectives that will not be tested.

A number of project documents will be involved and used in testing cases. Main documents include:

* Business Requirements Document (BRD)
* User Stories
* Requirements Composition Table (RCT)
* Requirements Traceability Matrix (RTM)
* Feature Short Descriptions Table (for modules)
* Entitlements Specification (ES) table
* Responsibility Assignment Matrix – RACI table
* Project Initiation Document (function descriptions)

## *2.1 Core Features to be Tested*

This section lists all core features that will be tested. The presentation is organized by an order of modules below.

* Build Profile
  + To test whether a user can be identified as a Helper or Customer
  + To test whether a user can label profile from an array of skills or trades and describe experience and expertise
  + To test whether a user can add professional photos/identify location/toggle profile serachability/set availability times (for Helper)
* User’s Account
  + To test whether a user can register for/sign into/deactivate an Account
  + To test whether a user can reset password/manage email preference
  + To test whether a user can see history of previous bookings and the corresponding financial transactions
  + To test whether a user can contact customer support
* Searching
  + To test whether a user can view/filter/sort Helpers by trade/location/fee/experience/rating/availability
  + To test whether a user can view/filter Customer jobs by location/trade/fee
  + To test whether a user can favorite Helpers
* Listing a job
  + To test whether a Customer can list a job/set an initial fee/upload images
  + To test whether a user can categorize post by trade
  + To test whether a Customer can cancel a job
* Messaging
  + To test whether a user can request a direct messaging channel and exchange messages with another user in real time
  + To test whether a user can view archived messages
* Booking
  + To test whether a Customer can assign a Helper an existing job listing
  + To test whether a user can set a price, either hourly or by project completion
  + To test whether a user can view all current and historical bookings
  + To test whether a user can cancel a booking
* Reviewing
  + To test whether a user can leave a star rating/written feedback for a user upon completion of a booking
  + To test whether a user can view feedback and star rating

Besides the core features requiring tests mentioned above, the function testing also will cover crosscutting concerns that are applicable to the context of the individual core features (refer to RCT).

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

The system test will cover following testing objectives according to the non-functional requirements:

* Usability
  + To test whether the application has a clear interface.
  + To test whether users are able to use the application without a high level of computer experience.
  + To test whether the application requires users to take less than 5 minutes to figure out a feature.
* Performance
  + To test whether the response time of the application is not exceeding 4 seconds depending on the user's connection condition.
  + To test whether the application is available for users 24 hours a day, 365 days per year.
  + To test whether the response is fast enough to avoid users’ response collisions.
  + To test whether the application can simultaneously support several users.
* Security
  + To test whether the application can be protected from hacking.
  + To test whether the system can encrypt all user information and communications.

## *2.3 Features not to be Tested*

There are a few features that will not be covered in the testing based on the requirements of the project. A non-test features list is given as follows:

* An international-oriented service will not be tested. This project has a specific target area for customers; therefore, international users will not be considered at the current testing stage.
* Impacts created by mobility will not be tested. Mobile computing always has various performance due to distinct networking connections and hardware capabilities. Current focus of this project is to develop a system that can successfully achieve designed functions. Networking issues will not be concerned by this testing.

# 3. TEST PROCESS DEFINITION

## *3.1 Test Process Phases*

The test process phases of this system testing consists of five phases, which are test planning, design, preparation, execution, and reporting. Each phase has a few purposes/tasks/goals, which are given as follows:

* Test Planning
  + Define scope and objectives of testing
  + Define roles and responsibilities
  + Define testing approach
* Test Design
  + Determine test design logic
  + Design test case specifications
  + Determine requirements for test data
* Test Preparation
  + Setup a test environment
  + Provision test data
  + Install the software in a proper environment
* Test Execution
  + Execute all test cases
  + Find and report software defects
  + Evaluate the system stability
  + Validate all target features
* Test Reporting
  + Create a testing report for stakeholders
  + Clearly state the testing process
  + Summarize the test execution
  + Report defect metrics and execution status
  + Evaluate the test exit criteria
  + Signing off the system testing by providing the approval of the final report

## *3.2 Testing Tasks and Deliverables*

The following table further provides details of the test process phase in terms of the statements given in Section 3.1.

|  |  |  |
| --- | --- | --- |
| **Process Phase** | **Tasks** | **Deliverables** |
| Test Planning | * Define scope and objectives of testing * Define roles and responsibilities * Define testing approach | System Test Plan documentRole and Responsibility Table |
| Test Design | * Determine test design logic * Design test case specifications * Determine requirements for test data | * Test Design Specification * Test-Case Specifications * Test Management System |
| Test Preparation | * Setup a test environment * Provision test data * Install the 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 test cases * Find and report software defects * Evaluate the system stability * Validate all target features | * Defect reports reported in the defect tracking system |
| Test Reporting | * Create a testing report for stakeholders * Clearly state the testing process * Summarize the test execution * Report defect metrics and execution status * Evaluate the test exit criteria * Signing off the system testing by providing the approval of the final 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 purpose of this section is to present the method of the functional testing for examining the developed system. Since this project rarely concerns the optimization issue, a black-box testing technique will be mainly utilized in examining whether functionality reaches the designed goal and whether the application follows business requirements (refer to Section 2.1). There are a few reasons for using a black-box testing approach. First, using a black-box testing can effectively examine the functionality of an application without the knowledge of the internal structure. This manner is close to real-world application scenarios as customers do not know internal workings. Next, a black-box testing approach ignores the internal mechanism and directly touches inputs and execution conditions. All testing objectives’ functionalities are observable based on this approach. Finally, a black-box testing is suitable for functional and user acceptance testing, which matches the tests of the business requirements.

A few black-box techniques that will be utilized in the testing include boundary-value analysis, cause-effect graphing, decision table testing, and state-transition testing.

## *4.2 Approach to Non-Functional Testing*

Both black-box testing and white-box testing will be used for non-functional testing. A black-box testing will address three aspects of the non-functional testing, which are usability, performance (refer to Section 2.2). The features and advantages of a black-box testing have been discussed in Section 4.1.

Moreover, a white-box testing also will be applied to a few non-functional testing operations, including security (refer to Section 2.2). As an internal-oriented testing approach, a white-box testing emphasizes the mechanism and skills deployed in the system. There are a number of reasons for supporting this white-box testing strategy. First, a white-box testing considers the testing object from an internal perspective of the system, such that programming skills will be investigated. Thus, security issues and service efficiency aspects can be fully addressed during a white-box testing. The other reason is that a white-box approach is good for unit testing, so that some non-functional testing can be processed, such as security investigation.

Main white-box techniques that will be used in the testing include control flow testing, data flow testing, statement coverage, and decision coverage.

# 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

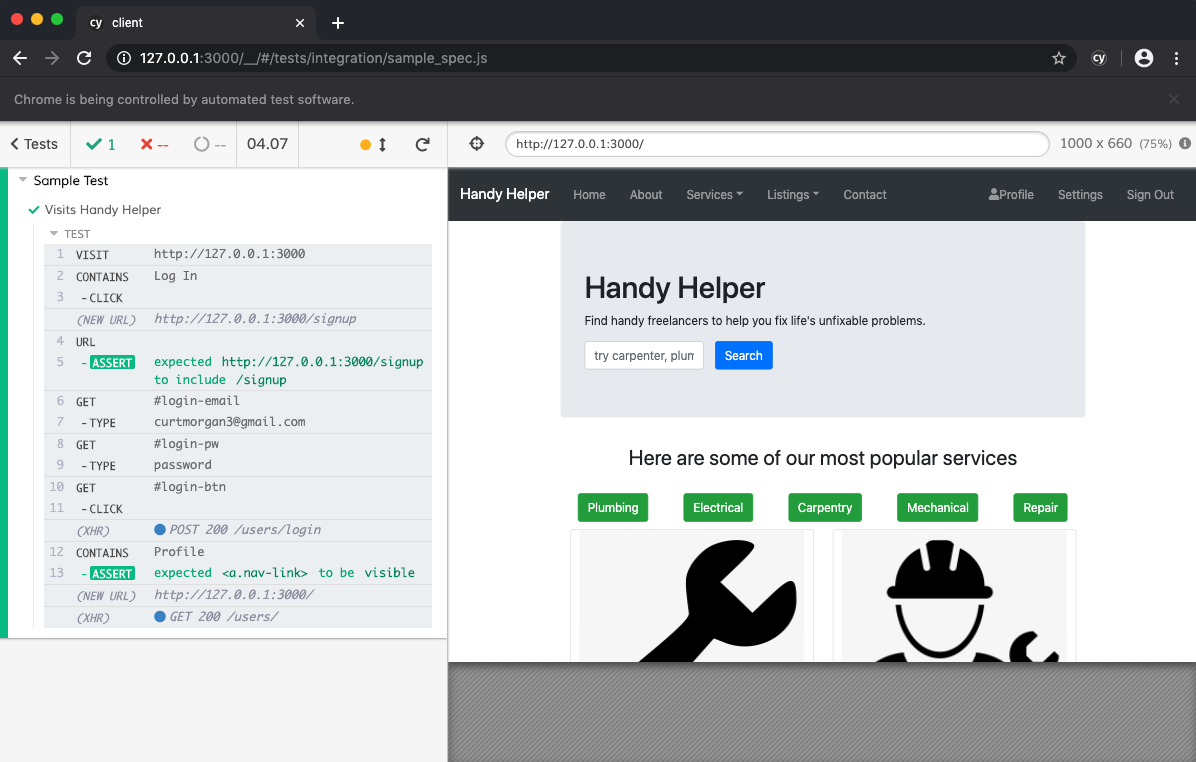
## *5.2 Exit Criteria*

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 High-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

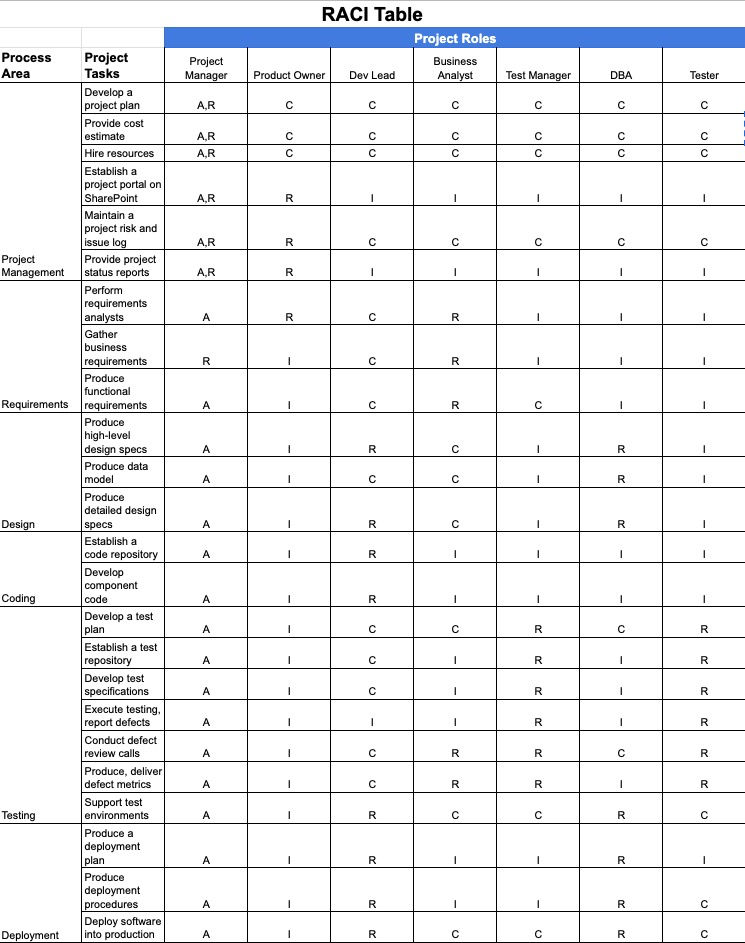
Handy Helper will utilize a system of automated, integrated testing. The testing suite will consist of an instance of Chromium to mimic a user’s browser which programmatically follows business logic and checks results against a set of expected behavior. The tests will check for HTML elements appearing on the page, HTTP status codes, correct URL paths and timing of network requests.



# 7. ROLES AND RESPONSIBILITIES

The project team has Eight members, including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, Lead QA Analyst, Test Manager 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 assisting 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. |
| Test Manager | Responsible for assisting the QA Analyst with developing and managing test specifications, executing testing and report defects, and keeping the team informed throughout the testing period. |
| Professor | Guide/advise the project team. |



# 

# 8. TEST CYCLES AND SCHEDULE

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

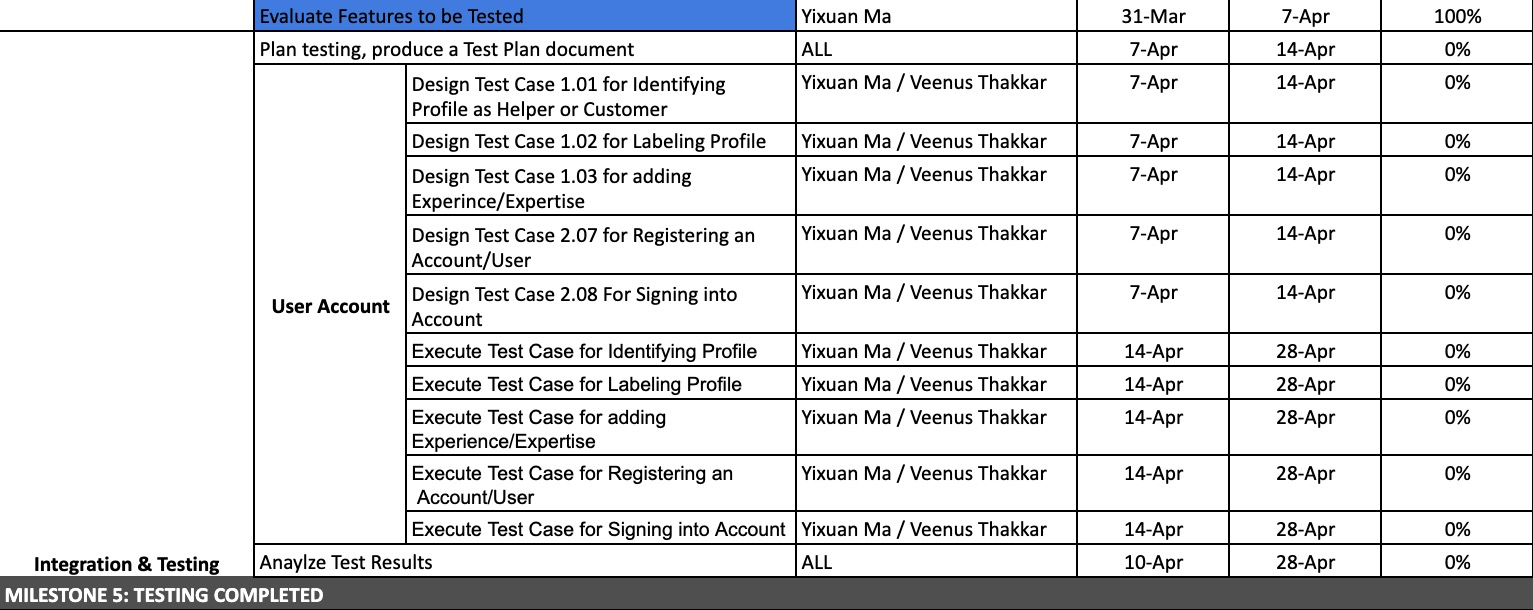
Cycle 1 User Account Module

This cycle concentrates on testing the User Account Module.

Cycle 2 Build Profile Module

This cycle concentrates on testing the Build Profile Module.

For the timeline of the project schedule and all testers’ activities please refer to the Project Plan Document.



# 9. RISKS AND CONTINGENCIES

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

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