Software Test Plan:

RentO

Prepared by:

The Underbots

Version 1.0

CSUN Senior Design Class

Comp 490 Fall 2022
12/16/2022

Revisions

Version	Primary Author	Description	Date
1.0	The Underbots	Primary Version	12/16/2022

Table of Contents:

Revisions 2

1.0 INTRODUCTION 4

2.0 OBJECTIVES AND TASKS 4

2.1 Objectives 4

2.2 Tasks 5

3.0 SCOPE 5

General 5

4.0 TESTING STRATEGY 6

4.1 Unit Testing Definition: 6

4.2 System and Integration Testing Definition: 7

4.3 Performance and Stress Testing 8

5.0 ENVIRONMENT REQUIREMENTS (TOOLS) 9

6.0 TEST SCHEDULE 9

7.0 CONTROL PROCEDURES 9

Problem Reporting 9

Change Requests 9

8.0 RESOURCES/ROLES & RESPONSIBILITIES 10

9.0 RISKS/ASSUMPTIONS 10

Appendix 11

1.0 INTRODUCTION

The purpose of this document is to describe the tests that will be done to the pages of RentO, as well as the JavaScript used to handle data input by the user. We want to see if the pages are properly loading once clicked and to see if the website can accept the user's login credentials or account details when they are creating an account. If there are no issues, the data will be passed to the database for processing and storage (once we have our backend up and running). RentO is a ecommerce for item rental so we want to make sure our users are able to access the pages they need to go to with no issues, as well as not have any problem when they enter information, even sensitive ones like payment details.

2.0 OBJECTIVES AND TASKS

2.1 Objectives

The objective for this Software Testing Plan is to set up the tasks that need to be tested and ensure it is working, the responsibilities given to the team, including creating the test cases and executing the tests, as well as who deals with the results. Documenting the proper results via a standard form and giving it to the proper member to evaluate it. The document will also talk about the testing schedule, necessary environmental tools and proper documentation.

2.2 Tasks

The first group features that will be tested are the web pages and the ability to go to a different page via a link. We will explore the webpage and see if we get to a certain page with no issue, such as starting from Home and working our way to Account page. This will include the current webpages we have already created, and they are:

- Account page
- Home page
- Sign In page
- Sign Up page
- Forgot Password page
- Terms and Conditions page
- Checkout Page

The second group feature will involve the JavaScript of certain web pages and how they handle data input by the user. These include:

- Sign In page
    - Email and Password verification
- Sign Up
    - First and Last Name, Email, and Password verification
- Forgot Password
    - Email verification

3.0 SCOPE

General

The scope of this test is to verify that our pages are working properly, in the sense that there are no broken links or lead users to a dead end, causing annoyance to them. We want to make sure the places the user can go are up and running. We also want to test the JavaScript part of our website and see if what we have coded is working right, being able to accept the user's login credentials, check to see if they are good and allow the user to continue. And, to see to prevent the user from entering bad information or nothing at all when creating an account.

4.0 TESTING STRATEGY

The overall approach for the testing includes two major features of our website, in its current version. One will focus on the web pages that we currently have, which includes home, sign in, sign up, forgot password, and account details. We will make sure that the user is able to go back and forth between pages by clicking on the appropriate link and see that everything is running ok. The second group of features will focus on the JavaScript codes found in the sign in, sign up and forgot webpages, and see that it is able to verify whatever the user has entered and either reject or accept the user's information, such as password, email, names and having a strong password. For the first group, we will just do basic website exploring and click on all the working links to see if they are functional while, in the second group, we will create a test case of several passwords, emails, and names and see if the JS functions are able to verify and see what to do with them. All these will use a device that has an internet connection and is able to access the RentO website.

4.1 Unit Testing Definition:

The minimum degree of comprehension for the first major group would be to see if all links are working properly. For the second group we will considered complete once we are able to process and verify 20 different sets of account objects, including first and last name, email and password, without having any issues, such as a bad email by passing the verification and is being accepted. We will do a Requirements Traceability Matrix to make sure that each test case was able to test certain requirements for the testing activity.

Participants:

While the team will be involved in the Unit Testing, which includes Aaron, Ravindu, Mohammed, Talha, Cristian, and Vince, only certain members will do the execution of the test and others that will setup the test cases, activities, and procedures. Cristian and Vince will be responsible for the test cases and setting up everything while the rest would do the testing and report any bugs or errors that should not be there.

Methodology:

The unit testing will be conducted by the team, at their choice of location and using their personal computer. While Cristian and Vince will write the test scripts for the unit testing, the rest of the team will perform the testing and report any issues they find. We will work our way up, testing a major group of features and see if any issues have been located. Depending on the major group, the activity should take at least an hour but the whole testing activity will last for a week, since our members have different schedules for school and work. It will begin on a Monday and will finish on a Sunday, with everyone turning their report to Cristian and Vince for evaluation and seeing if any proper steps need to be taken. If there is, fix the issue and repeat that specific test and if not, continue with the next major group of features.

4.2 System and Integration Testing Definition:

Our understanding of system and integration testing is a type of software testing that is carried out in an integrated hardware and software environment to verify the behavior of the complete system. In our case, the system would involve the front end of the website, the back end of it and see if the servers we choose is able to work with no issues. The System Integration Test also verifies the interactions between the modules of a software system. Which in our case, would be the webpages and data being passed between via the database.

Participants:

For the System and Integration Testing, the team responsible would include Cristian, Vince, and Mohammed. We will create the test cases and test the system to see if it is running with minimal issues. Execution of the testing will include the rest of the team.

Methodology:

Like the unit testing, the system and integration testing will be conducted by the team, at their choice of location and using their personal computer. While Cristian, Vince, and Mohammed will write the test scripts for the unit testing, the rest of the team will perform the testing and report any issues they find. We will work our way up, testing key features, especially areas that use database, interacts with other webpages, and needs the user data input, and see if any issues have been located. Depending on the key features, the activity might take from a few hours to a whole day, making sure all the parts are covered in the testing. The whole testing activity will last for a week, starting on a Monday and will finish on a Sunday, with everyone turning their report to Cristian, Vince, and Mohammed for evaluation and seeing if any proper steps need to be taken. If there is, fix the issue and repeat that specific test and if not, continue with the next major group of features, until the website is considered error free or complete to a certain perspective.

4.3 Performance and Stress Testing

As of 12/16/2022, we do not have the necessary features to perform a stress test, but at a later version, we will have one done.

Definition:

N/A

Participants:

N/A

Methodology:

N/A

5.0 ENVIRONMENT REQUIREMENTS (TOOLS)

The tools that the user needs to have for our test is to have access to a smartphone, tablet, or PC and have internet connection to access our website.

6.0 TEST SCHEDULE

Testing will occur every time we add a major feature, such as having a backend database up and running, having the mock payment system added, creating, and implementing the two-way review system, etc. Every time we add a major requirement, we will update this document with a new revision. But for now, we will focus on the pages, and JavaScript. The next major feature to be tested would be the back-end server.

The estimated time required to do each testing task will be a maximum of one hour, as checking the links of the pages would not take long. JavaScript will take most of the time, since we will be adding various data and see if the system is able to verify and reject/accept them. Each testing period, we will have staff work on it and report if there are any errors and they will be using their personal computers so that we can see if there's any issues with various hardware's that has different specs. Example, user A can use his login credentials and access his account page, but User B is not able, despite entering valid information.

7.0 CONTROL PROCEDURES

Problem Reporting
We will use a standard form procedure, to properly document the problems and try to fix it as quickly as possible. If any bug has been found, we will have the user try to replicate and if that is not possible, document the situation, such as which page they were in, what did they type in, if applicable, and what have they done previously to see if any previous actions might have caused the issue. We have not finalized a standard form template, but in the Appendix, we have a screenshot of what it might look like, seen from the Klariti website, a place that offers free and paid MS Word templates.

Change Requests
Since Git can create a history of who modifies what in the project, it would be best for certain team members to have the ability to push or pull the project and make the changes. To make sure that no changes are made that might break the website and it will be difficult to see who caused the issue. We will have a main branch where the updates are added, and we will also have a separate branch where the code is being worked on by the team and we will see what works or not. With Git, any minor step, such as adding a file or updating one without properly pushing and committing will cause issue and might be deemed too risky. As of now, half the team knows and are comfortable with Git.

8.0 RESOURCES/ROLES & RESPONSIBILITIES

As of now the, the staff members that are involved with the test project are Cristian Ramirez and Vince Fearing. They will compile test cases for the Acceptance Tests as well as make the changes or verify to see where the issue might lie. While they will be managing, designing, and preparing the test activities, the execution and resolving would be done by the whole team as to have many different points of views and increase our chance of finding any bugs, if any. The team is also responsible for the test environment, using their own choice of device and accessing the website to test the features.

9.0 RISKS/ASSUMPTIONS

The risk-assumption of the test plan is the possibility of overlooking an error and possibly letting it linger until it might become a major issue, as to which we will need to go back several versions to fix and start again. If one of the deliveries is delayed due to some reason, the contingency plan would be to work during the weekends to try to have it delivered in a reasonable time. An example would be to perform some form of crunching, working all day on a Sat or Sun to achieve the goal of the deliverables. Or if that is not possible, then to work longer in the night and complete as much as possible, but at the same time to get some rest and not overwork oneself.

Appendix

Standard form for problem reporting: