Panda Inc

System Requirements Specification

Quinton Swanepoel	15245510
Azhar Patel	15052592
Tshepo Macebo Malesela	14211582
Monkeli Fred Dilapisho	15074260
Keaton Pennels	14373018

Stakeholders

Multiply: Philip Kruger

Contents

1 Introduction **2**
1.1 Purpose for Test . . . 2
1.1.1 Reducing bugs in new features . . . 2
1.1.2 Reducing bugs in existing features . . . 2
1.1.3 Tests improve design . . . 2
1.1.4 Testing makes development faster . . . 2
1.2 Project Outline . . . 2
1.3 Scope . . . 2
1.4 Test Environment . . . 2
1.5 Assumption and Dependencies . . . 3
1.5.1 Dependencies . . . 3

2 Test Items **3**

3 Functional Features to be Tested **3**

4 Test Cases **4**
4.1 Test Case 1 . . . 4
4.2 Test Case 2 . . . 4

5 Item Pass Criteria **4**

6 Test Deliverables **4**

7 Detailed Test Results **4**
7.1 Overview of Test Results . . . 4
7.2 Functional Requirements Test Results . . . 4
7.2.1 Test Case 1 (4.1) . . . 4
7.2.2 Test Case 2 (4.2) . . . 4

8 Conclusions and Recommendations **4**

1 Introduction

1.1 Purpose for Test

1.1.1 Reducing bugs in new features

New tests are developed as we write the code. As a team, believe that doing testing does not result in a fully bug proof system, but does in most cases drastically reduce the number of bugs as we add new code.

1.1.2 Reducing bugs in existing features

With the use of quality tests, the addition of new components or features hardly disturbs the existing features. If a new feature breaks existing functionality, the existing tests fail, which makes it very easy to pinpoint where the errors occurred.

1.1.3 Tests improve design

When writing tests, one is forced to have testable code. We have used a strategy known as TDD(Test driven development) which ensures that you write efficient code that fulfills its basic functionality.

1.1.4 Testing makes development faster

Testing slows you down on a class-by-class basis, however with experience, your overall velocity increases because you need not fear breaking existing code when new features are added. With TDD, we realised that no extra code is written which saves coding hours and increases efficiency.

1.2 Project Outline

The main objective of this system is to allow a delivery person into a demarcated area of your house when you are not there. You should be able to give access remotely and monitor the delivery person while you they are in the area.

This document will demonstrate how the functionality of this system was tested by team codeBlox

1.3 Scope

The scope of this document is structured as follows. The components that have been identified for testing will be discussed. The tests that have been identified have been drawn from the requirements that have been constructed by the group and the Panda Inc Team. Furthermore, this document aims to outlines the test environment and the risks involved in the testing approaches that will be followed. Assumptions and dependencies of the testing will be discussed due to the environment chosen.

1.4 Test Environment

- Programming Language
  - node js
  - HTML5 web-components, Polymer
  - Java(server)
  - Java (android)
- Coding Environment
  - Node package envrionment (npm)
  - JAVA EE (jdk8)
- Operating system
  - Linux
- Hardware
  - iBeacons

1.5 Assumption and Dependencies

- assume that user has android device and is able to use it
- assume that the beacons are on and they are lully functional

1.5.1 Dependencies

-

all npm node-modules

2 Test Items

3 Functional Features to be Tested

- User registration
- User authentication
- Beacon Registration
- Server-port rgistration
- Push notification (android)
- Get active day

4 Test Cases

4.1 Test Case 1

Test case 1: connection to server
Condition: open android application
Objective: check if mobile device connects to server
Input: web URL
Outcome: 200 status

4.2 Test Case 2

Test case 2: add user
Condition: user should exist in the Multiply User Database
Objective: check if user details match those on the database
Input: first name, last name, id, email, password1, password2
Outcome: 200 status and user added to database

5 Item Pass Criteria

-

6 Test Deliverables

7 Detailed Test Results

7.1 Overview of Test Results

7.2 Functional Requirements Test Results

7.2.1 Test Case 1 (4.1)

- The website opened and the video stream started
- Result: Pass

7.2.2 Test Case 2 (4.2)

- New user was added to the system
- server returned status code of 200
- Result: Pass

8 Conclusions and Recommendations