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

## 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 Panda

## 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 for Admin Interface
  - Node package envrionment (npm)
  - JAVA EE (jdk8)

- Coding Environment for Android Interface
  - Android Studio
  - JAVA EE (jdk8)

- Coding Environment for Server
  - Eclipse.
  - 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

-

## 2 Test Items

## 3 Functional Features to be Tested

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

- Device connection to beacons
- Mobile connection to server
- Web interface connection to server
- Add User
- Edit User details
- Remove user
- Add beacons.
- Remove Beacons.
- Edit beacon location.
- Get user communities.
- Get global communities
- Get user leader-board
- Get global leader-board
- Get communities leader-board.
- User notification upon location entry.
- Prevention of Login screen, after login
- Update user details in database
- Prevention of multiple ActiveDay requests. (on the application)

# 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

### 4.3 Test Case 3

**Test case 3**: authenticate user
**Condition**: username and password must match relative records in user database
**Objective**: check if user details match those on the database
**Input**: username, password
**Outcome**: 200 status and user logged in to application

### 4.4 Test Case 4

**Test case 4**: notify via push notifications
**Condition**: push notifications should be enabled on user device
**Objective**: Notify user of event
**Input**:
**Outcome**: notifications are pushed to users phone

### 4.5 Test Case 5

**Test case 5**:User authentication
**Condition**: User must have filled in the user name and password
**Objective**: Authenticate the user
**Input**: User name and password
**Outcome**: Status 200

### 4.6 Test Case 6

**Test case 6**: Beacon Registration
**Condition**: Beacon field properties
**Objective**: add beacon to database
**Input**: Beacon database fields
**Outcome**: Status 200

### 4.7 Test Case 7

**Test case 6**: Server-port registration
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.8 Test Case 8

**Test case 6**: Device connection to beacons
**Condition**: Mobile device within beacon range, blue-tooth

**Objective**:<br>
**Input**:
**Outcome**:

### 4.9 Test Case 9

**Test case 6**: Mobile connection to server
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.10 Test Case 10

**Test case 6**: Web interface connection to server
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.11 Test Case 11

**Test case 6**: Edit User details
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.12 Test Case 12

**Test case 6**: Remove user
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.13 Test Case 13

**Test case 6**: Add beacons.
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.14 Test Case 14

**Test case 6**:Remove Beacons.
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.15 Test Case 15

**Test case 6**: Edit beacon location
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.16 Test Case 16

**Test case 6**: Get user communities.
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.17 Test Case 17

**Test case 6**: Get global communities
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.18 Test Case 18

**Test case 6**: Get user leader-board
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.19 Test Case 19

**Test case 6**: Get global leader-board
**Condition**:

**Objective**:
**Input**:
**Outcome**:

### 4.20 Test Case 20

**Test case 6**: Get communities leader-board.
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.21 Test Case 21

**Test case 6**: User notification upon location entry.
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.22 Test Case 22

**Test case 6**: Prevention of Login screen, after login
**Condition**:
**Objective**:
**Input**:
**Outcome**:

### 4.23 Test Case 23

**Test case 6**: Update user details in database
**Condition**: User record has been modified
**Objective**: Change user data
**Input**: User details
**Outcome**: status 200

### 4.24 Test Case 24

**Test case 6**: Prevention of multiple ActiveDay requests. (on the application)
**Condition**: User has already requested an active day
**Objective**: Block multiple get active day requests
**Input**: User must have sent initial request
**Outcome**: Disabled get active days

# 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 app should open
- 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

### 7.2.3 Test Case 3 (4.3)

- App send time spent at location
- The server should receive an object containing date, time, user and location.

### 7.2.4 Test Case 4 (4.4)

- Get Active day

### 7.2.5 Test Case 5 (4.5)

- Adding new beacons to a new location

### 7.2.6 Test Case 6 (4.6)

- Add multiple beacons to the same location.

### 7.2.7 Test Case 7 (4.7)

- Remove beacon from a location.

### 7.2.8 Test Case 8 (4.8)

- Remove user from system.

## 8 Conclusions and Recommendations