**School of Engineering Technology and Applied Science**

**ICET Department**

**Mobile Application Development Project MAPD-726**

**Test Plan - Sprint 6**

Wind Warriors

1. **Introduction**

## Goals

This document is a high-level overview defining our testing strategy for the Appetite application. Its objective is to communicate project-wide quality standards and procedures. It portrays a snapshot of the project as of the end of the sixth sprint.

## Assumptions

The aim of this document is to determine the testing scope for the current sprint. No automation tests will be in scope.

## Risks and Assets

Due to testing just on emulators, users may face some issues with the user interface when using the application on real devices. Tests will not be done on phones on this sprint.

1. **Scope**

It is aspired to be able to achieve a successful layout and functionality of the following Pivotal Tracker IDs:

* #156345201 – Github integration;
* #156344254 - As a user, I want to sort;
* #156344191 - As a user, I want to search by terms; and
* #156331212- As a user, I want to filter by price.

**2.1 Features to be tested**

The following modules will be tested in this release:

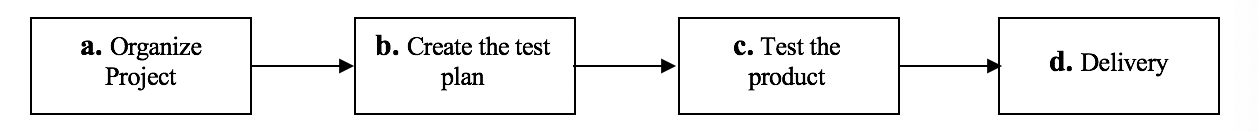
* Filter in the list of Restaurants using best\_match, rating, review\_count or distance;
* Filter in the list of Restaurants using terms; and
* Filter in the list of Restaurants by price.

**2.2 Features not to be tested**

It is out of scope the modules not mentioned on item 2.1 above due to further implementations are still under development.

1. **Testing Procedures**

The figure below describes the testing procedures adopted for this sprint:

**

***Figure 1: Testing Procedure***

* **Organize Project** involves creating checking and understanding the project requirements, assigning responsibilities, creating test plan and creating a timeline;
* **Create the test plan** involves creating and identifying Test Cases, Expected Results, etc. Test conditions/expected results will be identified by the team. The team will then identify Test Cases and the Data required;
* **Test the product:** The tests identified in the previous step will be executed;
* **Delivery**: Delivery happens when all tests are successfully executed with expected results.

## 3.1 Test Objectives

The objective of the test is to find and report as many bugs as possible to improve the integrity and quality of Appetite. We will exercise a broad range of tests to achieve our goal. Appetite which collects and shows restaurants that are nearby to the user. Our test plan is solely based to ensure quality of the application for the user.

## 3.2 Types of Testing

### 3.2.1 Unit testing

The strategy for unit testing of individual subsystems is described. This includes an indication of the subsystems that will undergo unit tests or the criteria to be used to select subsystems for unit test. Test cases are NOT included here.

### 3.2.2 Integration testing

In this sprint, the application will be integrated with YELP API which will provide the data for the user filter restaurants using specific parameters. Filters will be tested checking the results.

### 3.2.3 Acceptance testing

The acceptance testing will be conducted after Sprint 6, throughout a demo to the customer.

### 3.2.3 Stress testing

Not applicable for sprint 6.

### 3.2.4 Performance testing

Not applicable for sprint 6.

**3.3 Testing Tools**

Not applicable for sprint 6.

1. **Schedule and Deliverables**

Test deliverables are scheduled as follows:

|  |  |  |
| --- | --- | --- |
| **Key Test Milestone** | **Start** | **Finish** |
| Organize Project | 03/28/2018 | 03/30/2018 |
| Create Test Plan | 03/28/2018 | 03/29/2018 |
| Test the Product | 04/02/2018 | 04/03/2018 |
| Delivery | 04/03/2018 | 04/03/2018 |

|  |  |
| --- | --- |
| **Key Test Deliverable** | **Description** |
| Test Plan | This document |
| Test Case spreadsheet | Dashboard which contains all test cases, test conditions, expected results and test results |

The excel sheet created for the test cases identifies each of the test case problem statement and the result. This helps the developer to see the positive and the negative results and will help to generate robust results. We do not aim at specific number of test cases but definitely at a higher range in order to overcome any existing or overseen bug and obtain optimum functionality.