

# **Test Plan and Cases (TPC)**

**Fooder**

**QWERTY**

**Jake Motta - Co-Lead Programmer, Graphics Designer**

**Alex Le - Project Manager, Co-Lead Programmer**

**Brandy Kao - Co-Lead Programmer, Tester**

**Jason Springer - Co-Lead Programmer, Quality Assurance**

**3/8/2017**

# Version History

Date	Author	Version	Changes made	Rationale
3/08/17	JS	1.0	Initial app prototype created	· Created without any dependencies
3/14/17	JS	1.0	Menu icons updated, swiping capabilities added	·

## Table of Contents

Test Plan and Cases (TPC) i

Version History. 2

Table of Contents.. 2

Table of Tables.. 3

Table of Figures.. 3

1. Introduction. 4

2. Test Strategy and Preparation. 4

2.1 Hardware preparation. 4

2.2 Software preparation. 5

2.3 Other pre-test preparations 5

2.4 Requirements Traceability. 5

3. Test Identification. 5

3.1 Test Identifier 5-6

4. Resources and schedule 7

4.1 Resources 7

4.2 Staffing and Training Needs 7

4.3 Schedule 7

# Table of Tables

Table 1: Requirements Traceability Matrix 5

Table 2: TC-01-01 Check report completeness 6

Table 3: Testing Schedule 7

# Table of Figures

## 1. Introduction

Initial barebones testing of Fooder conducted through the use of similar style apps, such as Tinder, Grinder, and Yelp. The focus of this is to see the functionality, style, color scheme, and audience of similar existing successful apps, and understand why these characteristics work together in the app.

As Fooder develops, we will begin to implement test comparisons between style and response time to that of Tinder and Yelp, ensuring that we are of comparable (if not better) rate of app use.

## 2. Test Strategy and Preparation

Fooder test strategy follows an environment similar to that of “value-based test prioritization”, where we’ll be testing based off several priority steps:

1. Determine Risk Exposure for Each Test Case (RE)
2. Calculate Risk Reduction Leverage for Each Test Case (RRL)
3. Prioritize the Test Cases According to their RRL Value

### 2.1 Hardware preparation

Hardware preparation for Fooder includes:

- 2 x Laptops
  - Laptop 1: MacBook Pro Retina 13-inch
    - 2.8GHz, Intel Core i5, 8GB RAM
    - OS X 10.12.3 (Sierra)
  - Laptop 2: Lenovo
    - Windows 10 Home (10.0.14393 Build 14393)
    - Intel Core i7-6700HQ, 2.60GHz, 8GB RAM
- 2 x Cell Phones
  - Phone 1: Iphone 7 Plus
    - iOS 10.2.1
  - Phone 2: LG V20
    - Android 7.0

Laptops are used within our testing environment to research any web-based applications that we deem similar style to our ideal application for Fooder, and for virtual application building and testing. Both phones are used to run Tinder, Grinder, and Yelp applications, ensuring that any noticeable in-app differences between Android and iOS are noted.

## 2.2 Software preparation

Software preparation needed included:

- Tinder
  - iOS: Version 7.1.0
  - Android: Version 6.8.3
- Grinder
  - iOS: Version 3.4.1
  - Android: Version 3.2.0
- Yelp
  - iOS: Version 11.10.1
  - Android: Version 9.6

Tinder and Grinder are used for similar app design and functionality reference. These two apps provide profile privacy and security provided through Facebook Login API. These apps also inform the user that no associations between the app and Facebook are made, so that the user has no fear of their apps being linked, besides login functionality.

Yelp was used for context on how the app's database functions, how complex the menus could be, and how in-depth users could define search criteria.

## 2.3 Other pre-test preparations

Apps described in 2.2 needed to be downloaded to all phones described in 2.1.

## 2.4 Requirements Traceability

*Table 1: Requirements Traceability Matrix*

<b>Requirement ID</b>	<b>Verification Type</b>	<b>Test Case ID (if applicable)</b>
<i>AR-0 Application Reference</i>	<i>Testing</i>	<i>TC-01: Application Reference</i>

## 3. Test Identification

### 3.1 Test Identifier

TC-01 Application Reference: Testing and observation of various applications described in 2.2

### 3.1.1 Test Level

TC-01 Application Reference:

- Unit Testing - validate how each unit of the software performs

### 3.1.2 Test Class

TC-01 Application Reference will employ timing tests and functional tests

### 3.1.3 Test Completion Criteria

The test for TC-01 Application Reference will be completed when:

- All apps defined in 2.2 have been installed
- The following app features have been tested and documented:
  - General app functionality
  - Shapes and color scheme
  - User-enrollment
  - User-friendliness
  - Response time
  - User in-and-out time

### 3.1.4 Test Cases

Table 2: TC-01-01 Application Reference completeness

Test Case Number	<i>TC-01-01 Application Reference completeness</i>
Test Item	<i>-General app functionality:</i> How does the app work? <i>-Shapes and color scheme:</i> What kinds of shapes does the app employ? Are things rounded? What colors do they use? <i>-User-enrollment:</i> How can user sign up for the app? <i>-User-friendliness:</i> Is the app hard to use? <i>-Response time:</i> Does the app lag? How quickly do menus appear and actions perform? <i>-User in-and-out time:</i> How long does it generally take for a normal user to open the app, use the app, and close the app?
Test Priority	TC-01 test is very important in determining what Fooder's MSCW, allowing us examples to play with to determine what kinds of features we would want to integrate into our app, or remove.
Pre-conditions	Apps described in 2.2 must be installed

Post-conditions	Must establish a list which entails features that Fooder will incorporate that other apps detailed in 2.2 use
Input Specifications	Not Applicable
Expected Output Specifications	Not Applicable
Pass/Fail Criteria	Not Applicable
Assumptions and Constraints	Not Applicable
Dependencies	Not Applicable
Traceability	AR-0

#### 4. Resources and schedule

TC-01 Application Reference will be split among two individuals: Jake Motta, Jason Springer

Time to be allocated to testing and observation is one week of continued app use. No budget will be used for this, as these apps can be used for free. No resources needed, as the phones used for testing are personal phones held by Jake and Jason.

##### 4.1 Resources

Resources needed defined in 2.2

##### 4.2 Staffing and Training Needs

All developers, as well as Ramin Moazeni, are stakeholders. All developers are responsible for managing, designing, preparing, executing, witnessing, inspecting and resolving test items. No training for TC-01 necessary.

##### 4.3 Schedule

Table 4: Testing Schedule

Date	Test Identifier	Responsible person	Resources	Training needs
3/10/17	TC-01-01	Jake Motta Jason Springer	Defined in 2.2 or 2.1	N/A