Test Plan and Cases (TPC)

Hangry Mobile Food Truck Locator

<Team Number>

Team Members and Roles

John Humlick ---------- Architect
Amir Radman ---------- Project Manager/UI Designer
Aaron Escoto ---------- UI Designer
Daniel Tam ---------- Back-End Engineer
Natalia Zarubin ---------- Back-End Engineer

Version History

Date	Author	Version	Changes made	Rationale
02/08/06	PP	1.0	· Original template for use with LeanMBASE v1.5	· Initial draft for use with LeanMBASE v1.5
11/14/07	SK	1.9	· Removed section 3.2 agile test first	· Feedback from Class project
01/02/09	SK	2.0	· Removed section 1.2 References · Added section 3.1.3 Test completion criteria · Modified section 4, added schedule · Removed section 5, Test completion criteria · Added expected information in each section	· To comply with Instructional ICM-Sw standard

Table of Contents

Table of Tables

Table of Figures

1. Introduction

<< This section should include the following information
- *Provide the purpose, background of testing within this project.*
- *Provide scope of testing within this project*
- *Provide the focus of this testing*
- *Provide information about type of testing you are going to implement in this project >>*

This is a living document that will detail the test strategies regarding the primary use cases for Hangry Mobile Food Truck application. The majority of the focus will test the core features of this application which will primarily be the usability of the application, and secondary will be the functionality of supplemental features.

2. Test Strategy and Preparation

<< Describe the project's test strategy with respect to such options as:
- *Agile test-first*
- *Agile continuous integration*
- *Boundary and exit/entry criteria between unit testing and integration testing*
- *Integration order-of-build*

- *Value-based test prioritization*
- *Test environment preparation*
- *Requirements-test traceability >>*

We are going to use continuous agile development in the development for this application. All members will attend weekly standups every saturday. We will maintain bi-weekly sprints to ensure continuous integration.

2.1 Hardware preparation

<<Describe the procedures needed to prepare the hardware for the test, including support hardware (e.g., test equipment). Reference any operating manuals, if applicable.
Provide the following as applicable:
· *Specification of each hardware that will be used in testing and number (if applicable)*
· *Purpose of each hardware item, security and privacy considerations. >>*

The hardware that will be used for this application for testing will be our personal cellular phones.

2.2 Software preparation

<< Describe the procedures needed to prepare the software for the test, including test scaffolding and support software (e.g., input generators, simulators, data recording/reduction software, test output orales). Reference any software manuals, if applicable.
Provide the following as applicable:
· *The specific software to be used, identified by name and, if applicable, version number*
· *Purpose of each software item, security and privacy considerations*
· *Type and description of test data>>*

The software that will be used for this application for testing will be android emulators to test basic functionality of the application. We do require to use two emulators, one providing its current location, stored in AWS and the other would work as our interface to retrieve the coordinates.

2.3 Other pre-test preparations

<< Describe any other pre-test personnel actions, preparations, or procedures needed to perform the test not accounted for in 2.1 or 2.2 above. >>

There are no further pre-test preparations at this moment.

2.4 Requirements Traceability

<< Include a Requirements Verification Matrix specifying how each requirement from SSRD will be verified either by Testing, Demonstration, Analysis, Simulation, Inspection, etc. You can use the following table to illustrate the matrix:

Table 1: Requirements Traceability Matrix

Requirement ID	***Verification***	***Test Case ID (if applicable)***

	Type	
Mobile Application Interface	Demonstration	
Generate and store location coordinates	Testing	TC-01 - Provide current geolocation
Use the stored coordinate locations to find trucks	Testing	TC-02 - Locate food trucks

3. Test Identification

3.1 Test Identifier

<< This identifies the test by a project unique identifier and test case name. It shall provide a brief description of the test.
For example: TC-01 Generate summary report. >>
TC-01 - Provide Current geo-location
TC-02 - Locate Food Trucks

3.1.1 Test Level

System performance level: We will perform tests on low-end phones that run on inferior processors and see how well does our application operates.

Software item level: At this level, we will split the testing into unit testing and integration testing. At the unit testing level, each individual functionality will be tested. At the integration level, we will test the whole application as a whole.
<< This section shall describe the level at which the testing will be performed, for example, software item level or system performance (level of service) level.
For example: Software item level >>

3.1.2 Test Class

Erroneous test: This test will have the developers try to break the application in every way that we can think of. E.G Invalid inputs.

Maximum capacity test: We will have a class that will simultaneously push and request data at the same time to see how our scalability will fare.
<< This section shall describe the type or class of the test that will be performed, for example, timing tests, erroneous tests, maximum capacity tests, etc.
For example: Erroneous test >>

3.1.3 Test Completion Criteria

<< In this section, list of completion criteria specific for this test identifier
For example: The test for generate summary report will be completed when
- The system generates a complete and correct summary report
- The generated summary report is correctly displayed on the monitor and printed through the printer
- Etc.,
>>

3.1.4 Test Cases

<< A test case specification specifies inputs, expected results, and a set of execution conditions for a test item. For each test case, create a sub-heading using the following structure:
Identify a test (one of the tests in the test set comprising the application testing addressed by this test description) by a project-unique identifier and provide the information specified below for the test. The name includes the identification of the applicable unit. ***There may be several test cases for one test identifier.*** *Use the following table to provide test case information.*

Table 2: TC-01-01 Provide Current geo-location

Test Case Number	*TC-01* - Provide Current geolocation
Test Item	Testing android geolocation is operational and sending data/populating in AWS
Test Priority	*Must Have*
Pre-conditions	*Location services must be turned on both user and truck applications and both must have the application running.*
Post-conditions	*DynamoDB will be populated with latitude and longitude of users and food trucks.*
Input Specifications	*Latitude and longitude of users and food trucks*
Expected Output Specifications	*DynamoDB Tables will be populated with current locations*
Pass/Fail Criteria	*The data is in AWS DynamoDB*
Assumptions and Constraints	*User has a data connection and geolocation capabilities on their device.*
Dependencies	A mobile phone with a data connection and geolocation capabilities

	AWS
Traceability	*< < Provide mapping to requirement(s) from SSRD >>*

Table 3: TC-02 Locate Food Trucks

Test Case Number	TC-02
Test Item	The Android application and its interaction with AWS / Lambda will be tested.
Test Priority	Must have
Pre-conditions	We need to ensure TC-01 has successfully been accomplished, and we have the mobile phone's geo location.
Post-conditions	We have a list of mobile food trucks nearby or an empty list if none is available.
Input Specifications	User selects an action button labeled "Find nearby food trucks."
Expected Output Specifications	*The user is presented with a list of nearby food trucks, or a message stating that no food trucks could be found.*
Pass/Fail Criteria	Pass: Successfully retrieve all available food truck locations within a proximity. Fail: not returning valid output/results. Unexpected output
Assumptions and Constraints	*User has a data connection and geolocation capabilities on their device.*
Dependencies	A mobile phone with a data connection and geolocation capabilities AWS
Traceability	*< < Provide mapping to requirement(s) from SSRD >>*

3.2 Test Identifier

<< Repeat the same structure for the next test identifier.
For example: TC-02 Submit job requests >>

3.2.1 Test Level

3.2.2 Test Class

3.2.3 Test Completion Criteria

3.2.4 Test Cases

4. Resources and schedule

2 cellphones, 2 people, a working application

4.1 Resources

We will have five developers: Amir Radman, John Humlick, Aaron Escoto, Natalia Zarubin, and Daniel Tam. Our time constraint for this project will be until the end of the semester -- around middle of May. Our only budget will be for the potential AWS fees if our application gets enough usage.

<< Identify all resources need for testing, such as test data set, software, budget, and etc >>

4.2 Staffing and Training Needs

People who know how to press buttons on a mobile device.

4.3 Schedule

Table 4: Testing Schedule

Date	Test Identifier	Responsible person	Resources	Training needs
01/02/09	*TC-01-01 to TC-01-04*	*John Smith*	*Report test data sets, JUnit*	*N/A*