

CS455 Software Engineering

Software Requirements Specification (SRS)

Version: 1.0.0

Date: 9/22/17

Created by: Danny Rivera

Esther Tapia

Date: 9/22/17

Date: 9/22/17

Reviewed by: Sergio Miranda

Date: 9/18/17

Approved by: Elitanya De La Cruz

Date: 9/18/2017

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms, and Abbreviations
- 1.4 References
- 1.5 Overview

2. The Overall Description

- 2.1 Product Perspective
 - 2.1.1 System Interfaces
 - 2.1.2 Interfaces
 - 2.1.3 Hardware Interfaces
 - 2.1.4 Software Interfaces
 - 2.1.5 Communications Interfaces
 - 2.1.6 Memory Constraints
 - 2.1.7 Operations
 - 2.1.8 Site Adaptation Requirements
- 2.2 Product Functions
- 2.3 User Characteristics
- 2.4 Constraints
- 2.5 Assumptions and Dependencies
- 2.6 Apportioning of Requirements

3. Specific Requirements

- 3.1 External interfaces
- 3.2 Functions
- 3.3 Performance Requirements
- 3.4 Logical Database Requirements

3.5 Design Constraints

3.5.1 Standards Compliance

3.6 Software System Attributes

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

3.7 Organizing the Specific Requirements

3.7.1 System Mode

3.7.2 User Class

3.7.3 Objects

3.7.4 Feature

3.7.5 Stimulus

3.7.6 Response

3.7.7 Functional Hierarchy

3.8 Additional Comments

4. Change Management Process

5. Document Approvals

1. Introduction

1.1 Purpose

The purpose of this document is to explain and propose a mobile application for local businesses. It will delve into the features and user interface for both local business owners and users/customers, and most importantly explain the purpose and the functionality of the mobile application.

1.2 Scope

The mobile application will be designed for local business owners to showcase their business/products to potential customers. They will have the ability to design their own profile specifically for their business, a marketplace for just them. Local business owners will have all the control on how they want to represent their brand and we will give them that tool. They will be able to customize and input important information, such as address, phone number, business description, products, links and live feature.

Users/customers will be prompted to input a zip code or opt to GPS location and within how many miles they want to find a local business. Then browse through the various local business.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
Local Business	Any type of local business, such as clothing stores, shoe stores, thrift stores, ice cream shops, food, cafes, electronic stores, music stores, flower shops, online stores, etc. We accept all.
User/Customer	Person looking for a local business in their area and potential customer.
Database	Collection of local business owner accounts, profiles,
Android Studio	Software/IDE we used to create the core/frame of the mobile application
Adobe Illustrator	Tool used to create user interface

UI	User interface
IDE	Integrated development environment

1.4 Reference

Google. (2017). Google Maps for every platform. [Online]. Available:
<https://developers.google.com/maps/>

Firebase. (April 18, 2017). Firebase Cloud Messaging. [Online]. Available:
<https://firebase.google.com/docs/cloud-messaging/>

Firebase. (May 17, 2017). Firebase Realtime Database. [Online]. Available:
<https://firebase.google.com/docs/database/>

Firebase. (August 9, 2017). Firebase Cloud Functions. [Online]. Available:
<https://firebase.google.com/docs/functions/>

Firebase (May 17, 2017) Firebase Cloud Storage. [Online]. Available:
<https://firebase.google.com/docs/storage/>

1.5 Overview

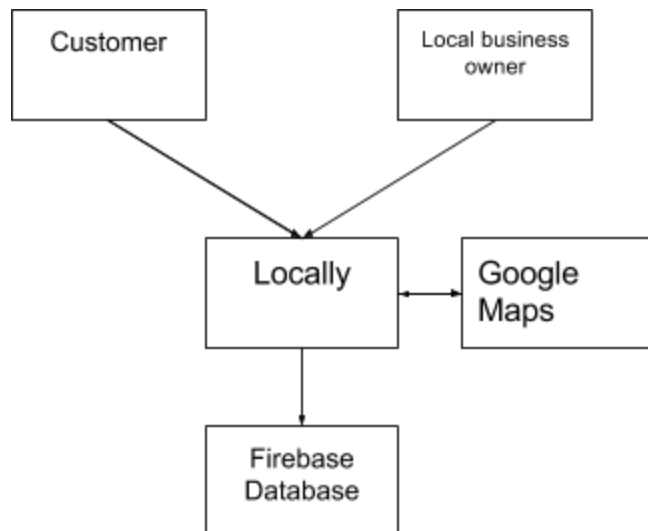
Section two, the Overall Description, will thoroughly explain the functionality of the mobile application and provide a step-by-step guide of what type of user interfaces a local business owner and user will interact with.

Section three, Specific Requirements, lists the technical aspect of each function/feature in the mobile application.

2. The Overall Description

2.1 Product Perspective

LOCALLY is an application developed for Android devices. LOCALLY is related to Yelp and FourSquare. Two different types of users will interact with LOCALLY; customers and local business owners. LOCALLY will use Google Maps API to provide nearby local businesses to the customer. Firebase Realtime Database will be used to store and sync data of the user such as name, email, password, address, and phone number. Firebase Cloud Messaging will be used to send notification to users and local business owners when they have received a message from each other. It should also notify the user of activities that were posted on the feed tab. Firebase Cloud Storage will be used to upload image files to the application so that any user nearby can have access to it.



2.1.1 System Interfaces

Google Maps API – Provides visual tracking, route planning, and marker identification through its clean map interface.

Firebase Cloud Messaging – Allows notifications to be sent to user

Firebase Realtime Database – Stores information and updates.

Firebase Cloud Storage – Upload image files to the feed page.

2.1.2 Interfaces

LOCALLY will interact with its users in two ways. The primary way to interact with this application is through its Graphical User Interface (GUI). Local businesses will be prompted to create an account and profile. They will be able to update their profile page. Customers will be prompted to sign in via email, and input location information such as zip code.

2.1.3 Hardware Interfaces

The primary hardware interface will be an Android-capable device, such as an Android smartphone or an Android tablet. Both user local business owners and customer must be GPS capable.

2.1.4 Software Interfaces

In order to obtain all the necessary functionality, customers and local businesses are required to have a minimum of Android version 4.0.3, "Ice Cream Sandwich." Any version greater than 4.0.3 will be supported.

2.1.5 Communications Interfaces

Locally must have internet access in order to load the content of the app. The mobile application must have GPS access to notify the user of activities happening nearby.

2.1.6 Memory Constraints

There are no specific memory constraints. Any device capable of running the specified software interfaces will be able to use LOCALLY.

2.1.7 Operations

The first operation will have a user interface that provides the customer the ability to browse on a feed that display images of activities or products happening nearby.

The second operation will have a user interface for the local business owners. The user interface will have a feed page, a profile, and the ability to take a photo and publish it.

Periods of interactive operations will consist of local business profile page updates, customer clicking on notifications, customer searching for local businesses via zip code, customer and local business communicating via message board, and customer viewing local business profile page.

Periods of unattended operations will consist of customer and local business receiving notifications.

2.1.8 Site Adaptation Requirements

There are no site adaptation requirements beyond installation of LOCALLY mobile application on a user's Android device and enabling gps location.

2.2 Product Functions

The following is a high-level overview of LOCALLY functionalities to its users.

1. The mobile app should provide a way for a new user to register with a valid email address, password, first name, and last name.
2. The mobile app should authenticate an attempt to login. If authentication fails, user shall be instructed to reattempt login.
3. The mobile app should provide a way for a user to reset their password.
4. The mobile app should provide an option for a user to logout, at which point that user should be taken to the launch page.
5. The mobile app should have an UI screen that ask for input if they are creating a customer or a local business page.
6. If the user has a local business or a product and they have just registered an account then customize their profile page by adding a photo, description text, phone number, address, link, or special offer description. The profile page can be customizable by adding different colors for the background screen. It can also have different type of fonts.
7. If the user does not have an account then the user can only browse through Google Maps.
8. If the user has a local business or a product, and they have already customize their profile page then the user should have 3 main tabs: a google map tab, a capture photo tab, and a profile tab.
9. If the user has a local business or a product, they should be able to post a photo by taking a photo or uploading it from their gallery with a description text.
10. Both customer and local business owners should be able to log out in the profile tab.
11. The customer should receive a notification if a local business owners published a post nearby the customers area.
12. Both customer and local business owners should have access to GPS through Google Maps API.
13. The customer and local business owner should be able to communicate through Firebase Cloud Messaging.
14. The customer and local business will have a Google Maps in order to interact.
15. Local businesses would have the option to go live through Google Maps.

2.3 User Characteristics

A user who is using the LOCALLY mobile application must be able to know how to use a mobile device such as installing a new mobile application, and navigate around a basic UI. The UI will focus on simplicity, so it can be access for most Android users

2.4 Constraints

Users must have an Android-capable device, such as an Android smartphone or an Android tablet. The user devices must be GPS-capable in order to get notification from local business owners to customers.

In order to obtain all the necessary functionality, users and are required to have a minimum of Android version 4.0.3, "Ice Cream Sandwich." However, anything from version 4.0.3 and up will be supported.

A device must be connected to the internet in order to receive notifications and sync data in real time. An interruption in internet signal or service may lead to the mobile application to not work properly.

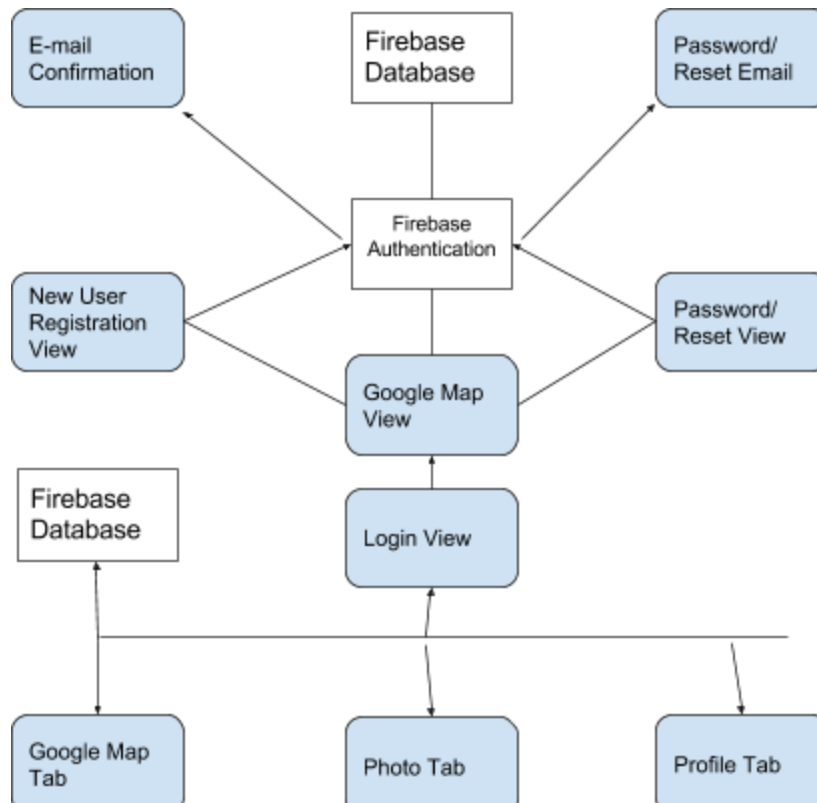
2.5 Assumptions and Dependencies

LOCALLY will be dependable on third party from Google in order to continue operating the service. A few of the services are Firebase Cloud Messaging, Firebase Realtime Database, Firebase Cloud Storage, and Firebase Cloud Function. LOCALLY should also be dependable on Google Maps API.

2.6 Apportioning of Requirements

All of the currently planned requirements are accomplishable within the initial development cycle. If there are any additional requirements in the future, this SRS will be updated to account for such changes.

3. Specific Requirements



3.1 External Interface

3.1.1 User Interface

It is recommended for users to use a touchscreen-Android device. Users are also allowed to interact with the messaging interface of the system using a keyboard-driven mobile device.

3.1.2 Hardware Interfaces

Android mobile device - required for both users customer and local business owners.

3.1.3 Software Interfaces

Android Operating System – Minimum version 4.0.3, “Ice Cream Sandwich.”

Firebase Realtime Database – Stores information and sync updates.

Cloud Functions for Firebase – Triggers notifications based on updates.

3.1.4 Communications interfaces

Google Map API - establish user's location.

Internet Service – Gets the user's cell phone carrier or Wi-Fi through the Android mobile device.

Firebase Cloud Messaging – Allows notifications to be sent to from client to local business owners.

Firebase Cloud Storage - Able to upload photos to the cloud so the user and local business owners can see them from around their local area.

3.2 Functions

Locally provides many interactive functions and a minimal amount of unattended operations.

3.2.1 Interactive Operations

There are many interactive functions a local business owner and customer engage with while active on Locally.

3.2.1.1 Functional Requirement 1.1

Locally should provide a way for a new user to register with a valid email address, password, first name, and last name.

3.2.1.2 Functional Requirement 1.2

The mobile app should authenticate an attempt to login. If authentication fails, user shall be instructed to reattempt login.

3.2.1.3 Functional Requirement 1.3

The mobile app should provide a way for a local business owner and customer to reset their password.

3.2.1.4 Functional Requirement 1.4

The mobile app should provide an option for a user to logout, at which point that user should be taken to the login screen.

3.2.1.5 Functional Requirement 1.5

Separate UI screen for local business and customer, determined by asking yes or no question if user has business or product.

3.2.1.6 Functional Requirement 1.6

Upon registering, local business then is directed to customize profile page by entering a business photo, phone number, address, website link, or special offer information. If desired to customize profile page further, local business has the ability to add a background color and change font.

3.2.1.7 Functional Requirement 1.7

Customer UI is built with a Google Map tab.

3.2.1.8 Functional Requirement 1.8

Local business UI is built with a Google Map tab, capture photo tab, and profile tab.

3.2.1.9 Functional Requirement 1.9

Local business capture tab must ask for permission to access camera to post a photo by either taking a photo or uploading a photo from gallery. Photo is accompanied with description text.

3.2.1.10 Functional Requirement 1.10

Local business owner should be able to logout in the profile tab.

3.2.1.11 Functional Requirement 1.11

Customer should have access to GPS through Google Maps API.

3.2.1.12 Functional Requirement 1.12

Customer and local business owner have the option to communicate through Firebase Cloud Messaging.

3.2.1.13 Functional Requirement 1.13

Customer and local business owner should be able to see their history of messages in the feed tab.

3.2.1.14 Functional Requirement 1.14

Customer Google Maps tab should generate nearby local businesses by GPS location.

3.2.2 Unattended Operations

Unattended operations include functions occurring while screen is in sleep mode or while mobile app, Locally, is not currently in use.

3.2.2.1 Functional Requirement 2.1

Customer should receive notifications when local business are nearby their local area.

3.2.2.2 Functional Requirement 2.2

Notifications clicked on should open up mobile app, Locally, and direct to local business profile page to view published post.

3.3 Performance Requirements

The system does not have specific performance requirements

3.4 Logical Database Requirements

Firebase Realtime Database shall store the following data:

1. Local business owner's email address, password, business name, address, and phone number upon registration. If online business, email address, password, and link to website upon registration.
2. Customer's email address (Gmail) and password
3. Customer's location using Google Maps API
4. Local business photos, and description text when publishing a post.
5. Local business location using Google Maps API

3.5 Design Constraints

System must be developed on a minimum Android version 4.0.3, "Ice Cream Sandwich."

3.5.1 Standards Compliance

Users' personal information such as first name, last name, email address, and location shall be stored securely.

3.6 Software System Attributes

3.6.1 Reliability

At the time of release, less than 50% of devices should have experienced technical difficulties. Such difficulties should be experience not to neglect in proper testing in code, but simply due to the variety of devices available.

3.6.2 Availability

Due to the fact that users may be using this application at any time of the day or night, it should be available 24/7. If a system is scheduled to go down for an emergency repair, users should be notified, if feasible.

3.6.3 Security

All local business and customer account information should be securely stored on the Firebase Realtime Database. Location will be restricted to LOCALLY app clients.

3.6.4 Maintainability

Comments shall be written in code where names of classes, fields, methods or variables do not readily provide all of the necessary information about their operations. A method shall only execute one action. This means that a method should only have one responsibility. If a method verifies user credentials, it should only do that and nothing else. This will ease the process of testing and maintainability

3.6.5 Portability

Any device operating Android version 4.0.3 and a GPS location Google Map will meet the requirements of operating Locally.

3.7 Additional Comments

Due to the fact that this is an initial version of the document, changes are going to be expected as the development progresses. Individuals reading this document must be open to a possibility of changes as new options are continuously explored.

4. Change Management Process

Should a customer request a change, the Team Leader should determine if the change is viable and if so, discuss with the customer further of the change. Once the Team Leader has reached an understanding of the change, the Team Leader shall then present the change to the team. The team will then come to an agreement that the change is doable and can be completed within a reasonable time frame. If more time is needed, then the team should discuss and determine a new time frame for development. Once the change has been implemented, an SRS update should be made in the rightful section(s).

5. Document Preparation, Review and Approval

This document was prepared by:

Danny Rivera (Lead Programmer)

Signature: *Danny Rivera*

Date: 9/22/17

Esther Tapia (Design Architect)

Signature: *Esther Tapia*

Date: 9/22/17

This document was reviewed by:

Sergio Miranda (QA Engineer)

Signature: Sergio Miranda

Date: 9/18/17

This document was approved by:

Elitanya De La Cruz (Team Leader)

Signature: *Elitanya De La Cruz*

Date: 09/22/2017