Requirements Document Version 1.0

Team: Shortcuts

Team Members: Stephen Bowder and John Hibbert

CSC305 Spring 2015

University of Rhode Island

Dr. Mello-Stark

Table of Contents

[Preface 3](#__RefHeading__1139_647031393)

[1: Introduction 4](#__RefHeading__10800_1086536616)

[2: Glossary 5](#__RefHeading__10802_1086536616)

[3: User Requirements Definition 7](#__RefHeading__10804_1086536616)

[3.1 Add Account (High) 7](#__RefHeading__10806_1086536616)

[3.2 Edit Account (High) 7](#__RefHeading__10808_1086536616)

[3.3 Delete Account (High) 7](#__RefHeading__10810_1086536616)

[3.4 Add Photograph to Account (High) 7](#__RefHeading__10812_1086536616)

[3.5 Delete Photograph (High) 8](#__RefHeading__10814_1086536616)

[3.6 Search Accounts (High) 8](#__RefHeading__10816_1086536616)

[3.7 List Accounts (High) 8](#__RefHeading__10818_1086536616)

[3.8 View Account (High) 8](#__RefHeading__10820_1086536616)

[4: System Architecture 9](#__RefHeading__10826_1086536616)

[4.1 Activities 9](#__RefHeading__10828_1086536616)

[4.2 Services 10](#__RefHeading__10830_1086536616)

[4.3: Content Providers 10](#__RefHeading__10832_1086536616)

[4.4 Data 10](#__RefHeading__10834_1086536616)

[4.5 Broadcast Receivers 10](#__RefHeading__10836_1086536616)

[5: System Requirements Specification 11](#__RefHeading__10838_1086536616)

[5.1: Functional Requirements 11](#__RefHeading__10840_1086536616)

[5.2: Non-Functional Requirements 13](#__RefHeading__10858_1086536616)

[6: System Evolution 15](#__RefHeading__11572_1086536616)

[7: Detailed Managerial Review 17](#__RefHeading__10882_1086536616)

# Preface

This document is a requirements document. It is an official statement of what the system developers will implement.

This document is to be made available for review of the customer, the development team, test engineers, project lead, and the senior VP of development.

Version 1.0: Assorted revisions for final submission. (5/4/15)

Version 0.2: Assorted revisions by J. Hibbert (3/25/15)

Version 0.1: Initial Version (2/17/15)

# 1: Introduction

The application ‘Shortcuts’ is a customer management application for Android tablets designed to increase customer satisfaction by facilitating customer input. It is primarily intended for salons, barbers, and other grooming services establishments. The user (in this case the proprietor rather than the customer of the business) will be able to take pictures of his or her results (haircuts, nails, etc.) with the device’s integrated camera. The application will store them in an internal account system, associating each picture with an account for that customer for convenient reference and searching.

The benefit to the user is the ability to record the result of his or her work during return visits without the hassle of manually managing hundreds or thousands of photographs. The customer can request to see the style they received at a previous service, and there will be a record of it. If they want to describe a desired change, they can point right to the picture on the tablet. The user can show his or her customer pictures to let him or her decide what worked and what didn’t work. This increased customer input means increased customer satisfaction. Increased customer satisfaction means return business, and that is always good for the bottom line.

## 

# 2: Glossary

**Account:** A record of a customer.

**Activities:** A component of Android applications that provides the user with a GUI with which to interact with the system.

**Application:** A computer program designed to run on a smartphone or tablets.

**Backup:** To create a copy of computer data that is stored separately to protect against loss.

**Character:** A single element that makes up a text file or string, such as a single digit or letter.

**Content Providers:** A component of Android applications that manipulates data.

**Data:** Pieces of information.

**Functional Requirements:** Requirements that are directly related to the services the system provides.

**GUI:** Graphical User Interface. The part of a computer that a user interacts with by manipulating graphical objects.

**Hardware:** Physical elements of computer systems.

**Ice Cream Sandwich:** A version of the Android Operating system. Versions are given confectionery-themed code names in alphabetical order.

**Integrated:** Describes a piece of hardware built into the device.

**Operating System:** The base software that manages hardware and software resources of a device.

**Non-Functional Requirements:** Requirements that are not directly related to the system’s function.

**Phablet:** A mobile device of a size between a smartphone and a tablet. A blend of the words “phone” and “tablet.”

**Preconditions:** Statements that must be true before a system activity has enacted.

**Postconditions:** Statements that must be true after a system activity has enacted.

**Services:**

1: In android architecture, a component of applications that commits operations in the background.

2: Functions provided by a program or machine.

**Smartphone:** A phone with an enhanced operating system.

**Software:** Computer instructions that operate on hardware.

**String:** A series of characters.

**System Architecture:** The way in which components of the system interact with each other and external devices.

**System Evolution:** A prediction of how the system may change over time for many reasons, such as changing customer needs or changes in hardware.

**Tablet:** An all-in-one flat computer operated by a touchscreen.

**Use Case:** A diagram that expresses the services the system will provide.

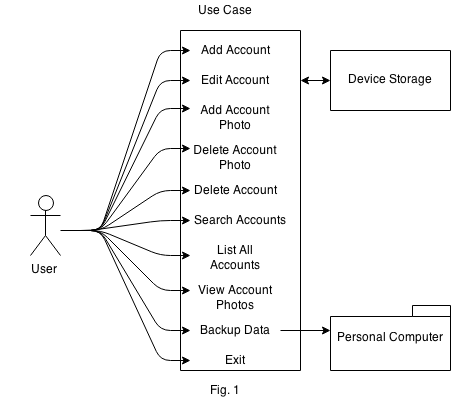
**User:** The person who is using the system. For this system, it will be the business proprietor or employee.

**User Requirements:** A description of the services the system must offer to the user.

**Wildcard Search:** A method of computer searching that allows for searching by incomplete strings rather than exact matches of whole strings. For instance, searching for “SMI” will bring up “Smith” and “Smithson” rather than just entries with the exact name “Smi”.

# 3: User Requirements Definition

A user requirements definition is a description of the services the system will provide to the user. It include use case diagrams and user requirements. A use case diagram shows the actors and how they relate to the system. User requirements are natural language statements of services the system is expected to provide and constraints the system will be subjected to.

This Android application is intended to allow the user to create and maintain a searchable catalog of clients that captures and stores pictures specific to each individual client. To maintain that catalog, they have the ability to add and delete accounts, and edit information in those accounts, including photographs. To use the catalog, the user can search for specific accounts or peruse an alphabetical list.

The application also will include a way to backup the data to a personal computer, to safeguard against damage or loss.

## 3.1 Add Account (High)

The user shall be able to create accounts for their clients within the system, under the precondition that the user has the necessary information from the client. That information is first name, last name, and email address,

## 3.2 Edit Account (High)

The user shall be able to edit the information of existing accounts within the system.

## 3.3 Delete Account (High)

The user shall be able to delete accounts from the system, under the postcondition that they enter their password.

## 3.4 Add Photograph to Account (High)

The user shall be able to take photographs with the device’s integrated camera, and associate those photographs with an account, under the precondition that an account exists.

## 3.5 Delete Photograph (High)

The user shall be able to delete photographs from accounts.

## 3.6 Search Accounts (High)

The user shall be able to search existing accounts by name or email address.

## 3.7 List Accounts (High)

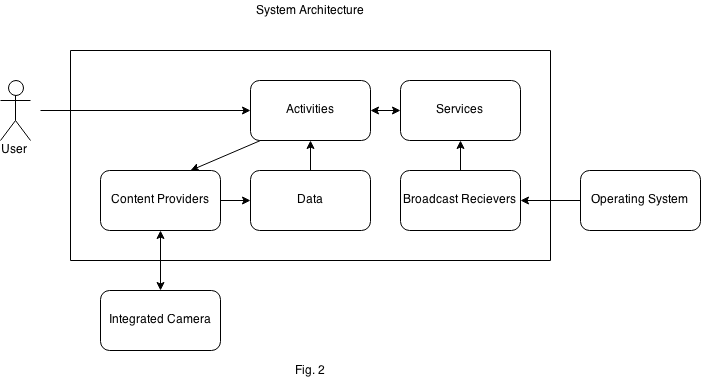
The user shall be able to view an alphabetical list of all accounts.

## 3.8 View Account (High)

The user shall be able to view the information of a specific existing account, including name, email address, and photographs.

# 4: System Architecture

System architecture is the way in which the components of the system interact. They include both modules within a piece of software and outside applications, and devices with which the system communicates.

 This application will be intended for tablets running Android version 4.0.3 (Ice Cream Sandwich) and other compatible versions. This application will be developed for the Android Application Framework. Therefore, this application will have an architecture similar to most common Android applications. They are generally composed of five different kinds of components: Activities, Services, Content Providers, Data, and Broadcast Receivers.

## 4.1 Activities

Activities are single screens containing a user interface. While proper design will make these activities flow into one cohesive experience, each one is actually a separate activity.

For this application, activities will comprise of the graphical user interface (GUI). That may include the account list screen, a screen for viewing a particular account, or a screen for viewing the photos attached to a particular account.

## 4.2 Services

Services are components that run in the background and do not provide a user interface. This is for when the application must do something without changing the current activity.

For this application, services include searching the accounts, retrieving data from storage, and remembering which account is currently selected.

## 4.3: Content Providers

Content Providers are the components that manage an app's data and allow it to be queried and altered by the application or other applications.

For this app, content providers will be utilized by activities to alter the user's stored customer data and connect pictures with accounts.

## 4.4 Data

Data are pieces of information that a content provider stores, manipulates, alters and queries. While the application itself is technically data, in this context, it is referring to the data that the application is manipulating.

For this application, the data are the accounts and their contents: names, email addresses, and photographs.

## 4.5 Broadcast Receivers

Broadcast receivers are components that respond to system-wide announcements, and create status bar notifications.

For this application, there will probably not be much use for broadcast receivers.

# 5: System Requirements Specification

System requirements are descriptions of a system’s expected functions and anticipated operational constraints. They are divided into two types: functional and non-functional requirements. Functional requirements are statements regarding the system’s necessary functions: services it provides and how it behaves in particular situations. Non-functional requirements are other requirements that are not directly related to the system’s function.

## 5.1: Functional Requirements

### 

### 5.1.1 The system shall have the ability to create a stored account for each individual client. (High)

This is the most critical functionality of the app. If the user cannot create accounts for their clients, then the application cannot access that information through any other means. The data included in each account will be the customer’s first name, last name, email address, and the associated photographs.

### 5.1.2 The system shall have the ability to display stored information pertaining to a particular account. (High)

This is the other critical functionality of this application. The client has to be able to retrieve the information within a given account for it to be useful to them. The information that needs to be available are the first name, last name and email address.

### 5.1.3 The system shall sort all stored accounts alphabetically by last name. (High)

Organization is critical. Without a logical organization, the information will not be easilly accessible. Last name is a commonly understood means of organization.

### 5.1.4 The system shall search the stored accounts of the application via filtering by name or email for quicker access. (High)

If the number of accounts grows large enough, alphabetical search becomes less practical. A ‘wildcard’ search will allow the user to find existing accounts no matter how large the number of clients is.

### 5.1.5 The system shall have the ability to modify client information including name and e-mail address. (High)

Though names and email addresses do not change often, it is important that the user have the option to change existing accounts when necessary,

### 5.1.6 The system shall have the ability to remove client accounts as necessary. (High)

Even if deleting accounts is rarely if ever necessary, the user may wish to do that.

### 5.1.7 The system shall have the ability to capture and attach photos to accounts for later access. (High)

The most logical time for a user to want to add pictures to an accounts is when they have found the account within the system.

### 5.1.8 The system shall have the ability to remove any unwanted photos attached to an account. (High)

There must be a means to remove undesired photographs.

## 5.2: Non-Functional Requirements

### 5.2.1 (Usability) The system shall be able to be used after 30 minutes of practice or training. (High)

This application could attract users of all levels of computer literacy. Therefore, the application should be very user friendly, requiring only basic computer skills to operate.

### 5.2.2 (Efficiency) The system shall allow searching for accounts, which should take no longer than a second. (Medium)

A fast, clean user experience will be necessary for a satisfactory experience.

### 5.2.3 (Efficiency) The system shall load photos for a given account should in less than two seconds. (Medium)

A fast, clean user experience will be necessary for a satisfactory experience.

### 5.2.4 (Dependability) The system shall crash no more than once a year. (Medium)

This system should be as sturdy and reliable as is practicable.

### 5.2.5 (Space) The system shall be no larger than 500MBs, including account data and pictures. (Medium)

As the user can take an unlimited number of pictures, there is a chance that the application could grow to an unwieldy size. A limit on the number of accounts will be necessary.

### 5.2.6 (Platform Compatibility) The system needs to be small enough to be published on the Android Market, and therefore cannot exceed 4 GB in initial size. (HIgh)

As of the most recent chance (3/5/12), the maximum size for an Android application download is 4 GB. Considering the requirement that the application be less than 500 MB, this shouldn’t be a problem unless these specifications change.

### 5.2.7 (Robustness) The system shall not corrupt its data corruption upon crash. (High)

The user’s data should not be put at risk.

## 

# 6: System Evolution

A system evolution is an attempt to describe how the system will change over time. It describes the assumptions on which this system is based, so they are understood if those assumptions should change. Also, it is an attempt to anticipate changes to the system that arise from hardware evolution or changes in the user’s needs.

First, this project will operate under the assumption that the general paradigm of the haircutting and grooming industries will not significantly change over any expected lifespan of this product. Even if times and trends change, it is exceptionally unlikely that people will stop needing haircuts.

Currently, the system will be made for Android. Android offers feature-rich development and a high number of supported devices. Android 4.0.3, aka Ice Cream Sandwich, is the version that has been selected for its high level of compatibility with various devices. The application is expected to be simple enough to be ported to later versions of Android if 4.0.3’s life cycle comes to an end. If the market changes such that tablets are no longer in use, the application could be adapted for smartphones and “phablets.”

The application could theoretically hold thousands of accounts and each could have dozens of pictures. Therefore, it needs to be built in such a way that it cannot grow so large as to negatively affect its performance or space limitations. A means to accomplish this would be to limit the number of accounts to an arbitrarily large number.

A future consideration to keep in mind is that the application could also one day be expanded to use automatic cloud backup in lieu of manual backups to a personal computer. This would greatly increase the convenience and slightly increase the overall productivity of the app, so this would be a desirable improvement. As of now, this is not a feature that is slated for inclusion, but this should direct early design, and could serve as an alternative solution to the space issue listed above.

Some users may wish to search for accounts in different ways, such as by number of pictures. The search method should be flexible to allow for expansion of miscellaneous types.

Some consideration should be given to allow the application appeal to other unanticipated businesses that may wish to use it, so its branding and iconography should be generic.

# 7: Detailed Managerial Review

I don't imagine I'm supposed to redo this. So, I removed the original commentary here to avoid confusion. Things have changed since that was written. Obviously, the day after that was written, the team lost a member. Communication has been improved and things are running relatively smoothly.

I know Mello-Stark suggested at some point that Stephen should do the revised review to make up for the problems with the original, but the way things happened to fall this week (it is 3/25/15) and the lab assignments and where we both were, it was just more straightforward if I (John) just did it. I had to wait around in the library for a different class (WRT104) to meet up with students who weren’t going to show, so I just figured I'd do it while I waited. It was barely 90 minutes of changes, give or take.

Also, another thanks to Dave and Stephen. I somehow lost the version with the notes on it, but they had it. No idea how I managed to do that, but at least someone had it.