
Software Requirements Specification

for

Trackoholic

Version 1.0 approved

Prepared by:

Kyle Chan

Yasmeen Diego

David Eisenbaum

Neal Layumas

Joshua Rogers

Storm Eagles

4-29-2020

Table of Contents

| | |
|---|-----------|
| Table of Contents | ii |
| 1. Introduction | 1 |
| 1.1 Purpose | 1 |
| 1.2 Document Conventions | 1 |
| 1.3 Intended Audience and Reading Suggestions | 1 |
| 2. Overall Description | 2 |
| 2.1 Product Perspective | 2 |
| 2.2 Product Functions | 2 |
| 2.3 User Classes and Characteristics | 2 |
| 2.4 Operating Environment | 2 |
| 2.5 User Documentation | 2 |
| 2.6 Assumptions and Dependencies | 3 |
| 3. External Interface Requirements | 3 |
| 3.1 User Interfaces | 3 |
| 3.2 Hardware Interfaces | 3 |
| 3.3 Software Interfaces | 3 |
| 3.4 Communications Interfaces | 3 |
| 4. System Features | 4 |
| 4.1 System Feature 1 | 4 |
| 4.2 System Feature 2 (and so on) | 4 |
| 5. Other Nonfunctional Requirements | 4 |
| 5.1 Safety Requirements | 5 |
| 5.2 Security Requirements | 5 |
| 5.3 Software Quality Attributes | 5 |
| 5.4 Business Rules | 5 |

1. Introduction

1.1 Purpose

The product of this document being discussed is the Trackoholic ver. 1.0 application. It has the capability to track inventory for both personal and business purposes. This document covers the application as a whole and describes the design decisions made for the application.

1.2 Document Conventions

Categories are bolded and typed in a large 18 size font, while subcategories are typed in a smaller 14 size font to differentiate between the two. Details in those subcategories are indented. Each requirement statement made in this document is to have its own priority.

1.3 Intended Audience and Reading Suggestions

This document is designed for current and future developers of this product, as well as any well informed users. This document is organized to be read in order, reading through the overviews of each section before proceeding through each subcategory.

2. Overall Description

2.1 Product Perspective

While there are many different types of inventory trackers that exist, the Trackoholic application adds new innovative technology that allows a person to track the inventory of both personal and business related items in one app. This application was created for android devices so users may track their items from the comfort and portability of their phones.

2.2 Product Functions

The product allows the user to track items and document important information about each item. There are set fields such as price, model, serial number, and more, which allows for a full and complete description that fits the user's needs.

2.3 User Classes and Characteristics

The app is free so anyone with an android has the ability to obtain it. This application is very simple to use, not requiring much education or experience with other apps. It is perfect for any person or company that wishes to track their inventory and manage the data pertaining to it.

2.4 Operating Environment

Trackoholic runs on mobile devices that run android ice cream sandwich or earlier. Other apps that Trackoholic will need access to includes the built in image gallery application.

2.5 User Documentation

User documentation can be found in the following link https://github.com/kylechan19/380_app The github includes our application's README file and has a general description of the system.

2.6 Assumptions and Dependencies

The requirements stated in this document may be affected by any changes in third party services that our application requires. Our database utilizes sqLite and the application itself is built on Android studio and any changes in those applications may affect Trackoholic as well.

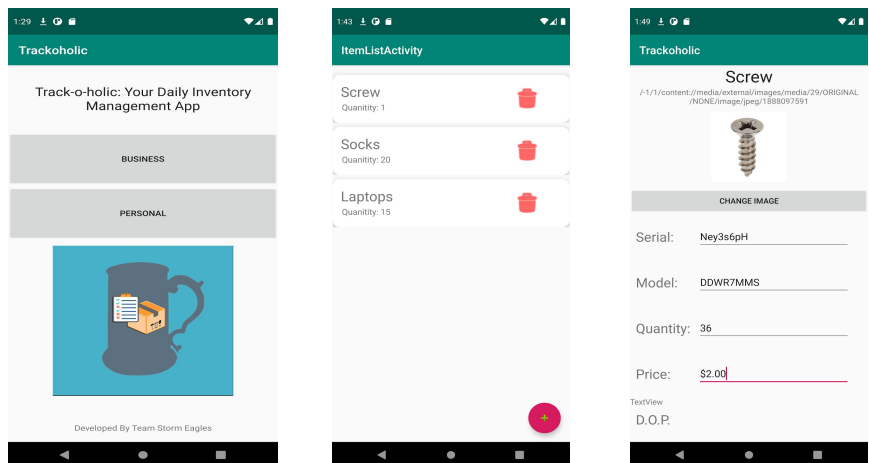
3. External Interface Requirements

3.1 User Interfaces

At the start of the app the user will be greeted with two options to select from: business and personal.

After selecting one of the options a vertical linear layout will appear that will provide each item row by row. In each row there is a trash can icon that will delete the item when clicked. There is also an add button at the bottom right of the screen that will allow you to add a new item.

When the user clicks on the item they will be taken to the item screen which will show the name and picture of the item at the top. Below the image there will be information about the item such as: serial number, model number, price, quantity, and date of purchase.



3.2 Interfaces

This product utilizes the current version of SQLite(version 3.31.1) as our primary database. This application is also dependent on Android Studio's implementation of their development tools. Data is inputted by the user, usually into the item page, and is then placed into an arraylist that is then serialized and placed in our database, ready to be accessed again. Data shared across software components include the names of items as well as its main description.

4. System Features

4.1 Item Description

4.1.1 Description and Priority

This is one of the key components of this app and therefore has the highest priority. The item description provides an in depth detail of each item, containing: item image,serial number, model number, price, quantity, and date of purchase.

4.1.2 Stimulus/Response Sequences

The user will have to go into the item and manually click and add all the necessary information listed earlier. To add an image the user would click the load image button and be taken to the image gallery, where they can pick the image to add. Next the user will click on the other field boxes, listed earlier, where the keyboard will appear for them to add the necessary information.

4.1.3 Functional Requirements

When selecting an image it will take the user to the image gallery. From there they can choose images of all types(png, jpg,ect...) For other fields it can only accept exspecting types. For example the price field can only expect numerical values. From there everything besides the image is put into object then into an array and then serialized.

4.2 Use of multiple Directories

4.2.1 Description and Priority

Having multiple Directories was also an essential part of this app. Allowing for both personal and business gives the benefit of storing both sets of items on one device.

4.2.2 Stimulus/Response

The user will be promoted at the start of the app with the option to choose from either business or personal. After choosing one they will then be taken to the item view for that directory.

4.2.3 Functional Requirement

The business and personal are both array names to the array that contain their respective item lists. The array list are then serialized into strings and saved.

5. Other Nonfunctional Requirements

5.1 Safety Requirements

Everything besides photos are stored locally on the app. If the app is deleted all information regarding the items in the app will be lost. Please refrain from deleting the app to keep all information.

5.2 Security Requirements

Since this is a product that stores information about personal belongings, it is highly recommended that the user have a passcode on their phone. The Trackaholic does not password protect accounts, so anyone who has access to the phone has access to the app.

5.3 Software Quality Attributes

Due to the simplicity of Trackaholic it will be easy to maintain and if desired implemented into other apps. Porting to other types of hardware can be done with ease due to the app being written in java. The database may need to be modified for the specified hardware.

5.4 Business Rules

Some of the limitations of the user include not being able to alter certain properties of items for legitimacy reasons. For example, the purchase date of an item can cause complications if it's tempered with.