Software Requirements Specification

for

Collection$

Version 1.0 approved

Prepared by: Ronald Fairley

Jeremy Box

Lee Adlaf

Kuni Scissum

Team 2

October 8, 2013

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Project Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. System Features 3

3.1 Input and Edit Ojbect Information 3

3.2 Networking with Other Collectors 3

3.2 Object Identification 4

4. External Interface Requirements 5

4.1 User Interfaces 5

4.2 Hardware Interfaces 5

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

Appendix A: Glossary 6

Appendix B: Analysis Models 6

Appendix C: Issues List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Original | 10/8/2013 | Original Documentation | 1.0.0 |
|  |  |  |  |

# Introduction

## Purpose

Collection$ 1.0.0 is an Android-based application for collectors and hobbyists. The scope of this document is to introduce the application as a collection of databases that can be shared between users. This version will allow users to upload items in their collection to a database and provide information on the item. The information include name, year created, market value, and other properties. The application will also allow object identification assistance and networking with other users to a limited degree. The latter will be improved in following versions.

## Document Conventions

Any text that uses emphatic structure (bold, all caps, or italics) represents key information that requires special attention. Listed material represents further detail on the information presented immediately preceding the list in question.

## Intended Audience and Reading Suggestions

This document is for developers that will be working on the software interaction between the smartphone (Android), an online database, and overall social networking scheme. Also, testers are targeted in this document. Upon reading this document, testers will better understand the goal of the project, the methods used to achieve the goal, and the project’s functionality and features.

## Project Scope

The goal of this project is to produce a simple smartphone application that provides database management and sharing in a simple manner. For the future, Collection$ is expected to become a large community-oriented database tool that will allow the users to create and manage their own database as well as view and assist in building other collections.

For this project, the goal is to establish a database, a protocol for uploading data, ensure a secure method for accessing the database from an Android smartphone, while being able to share with other users simultaneously.

## References

This section will be updated as information is referenced in a relevant manner.

# Overall Description

## Product Perspective

Collection$ was conceptualized as a way of networking for hobbyists and collectors. Not only does it act as a social network, but it also helps to maintain a personal database, share information with other hobbyists, and find items that are based on others. This product will implement databases and Android OS programming.

## Product Feature

Collection$ has two main features: maintenance of personal collection(s) and assistance with object identification. Collection$ can be seen by others (if enabled), in order to collaborate as a community to maintain collections. For new objects in the collection, data can be inserted or edited manually. In viewing the collections, users can sort them according to type of items, names, year added, year created, and other item characteristics.

The second feature is networking. Collection$ will allow users to collaborate amongst each other in order to build up collections, share information, or to talk about their collections. Private messaging will be available. In the future, forums and bulletin boards may be available as well.

The third main feature is object identification. This feature is a long term goal that will allow for users to automate the object uploading process. For collection classifications that are common (coin collections, stamp collections, etc.) and have official identification, a 2D image can be taken and compared to the database of official items and the official title, year made, and other common information will be filled in automatically. Objects that exceed a 2D threshold may be identified using 3D object analysis in the future. The focus for version one is 2D object identification.

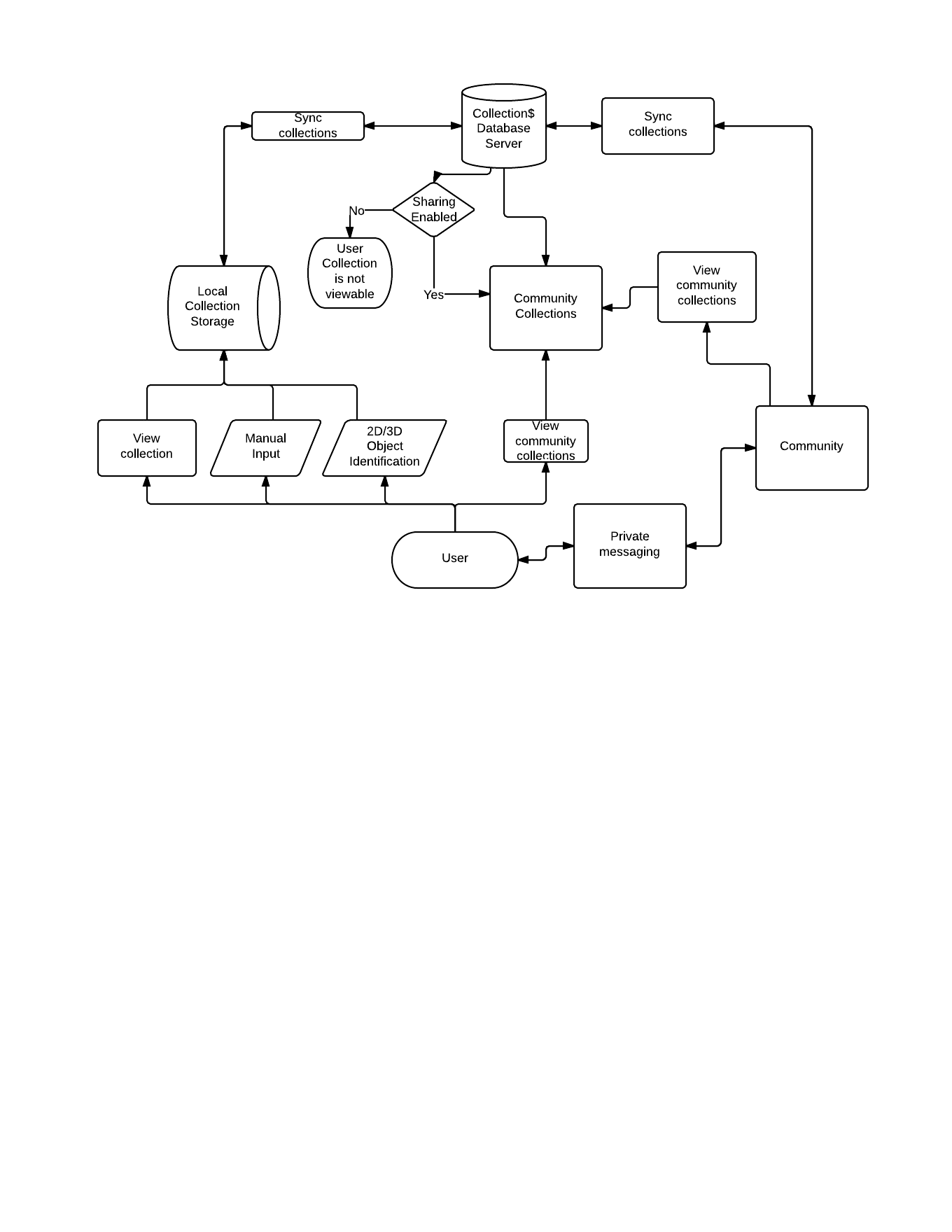


Figure 1. Collection$ Flow Chart

## User Classes and Characteristics

The targeted user classes include hobbyists and collectors. The users can either be prolific collectors or occasional hobbyists. The user should include those who have a desire to store their collection information in an organized manner or to share with others. The educational level should not be an issue; the age range for Collection$ should be wide, since there are younger users who can find use in this application as well.

## Operating Environment

For the version 1.0.0 the product will target Android operating systems. Devices includes tablets and smartphones. There are plans to implement a web application that can be accessed via computers. The Eclipse IDE will be used to develop this application.

## Design and Implementation Constraints

Preliminary design processes will be limited based on the technology available. In terms of hardware, several Android-based smartphones will be available to test the product. A database that can be accessed online must also be used to facilitate networking. Security measures will also have to be taken. Ideally, each user will have a username and password for user authentication. Collection$ will only be available in English.

## User Documentation

The support for this application will include tutorials and help information built-in to the application. As the product undergoes successive revisions, on-line help will become available. User documentation is currently unavailable.

## Assumptions and Dependencies

The development of the application relies on the following assumptions:

* Ability to maintain a cloud database
* Ability to find testers to assist with the development
* Ability to use Amazon Turk service to identify new additions to collection types and confirm authenticity (as allowable
* Ability to identify market value of objects by using websites such as Amazon and eBay

# System Features

## Input and Edit Object Information

3.1.1 Description and Priority

The core of the application is to allow users to add objects to their own collections and to add it to the community as a new object in the collection. This feature has a High priority as both the most basic feature to implement and as the main focus of the application.

There are several input parameters offered for this feature, including a title, description, value, research done for the object, status (up for sale, display only, etc.), condition, and photos of the object.

3.1.2 Stimulus/Response Sequences

Once a user is logged in, this option will be readily and clearly available to the user. The user only needs to choose this option to implement it. A “new-object” screen will appear with input parameters for the user.

3.1.3 Functional Requirements

REQ-1: Connection to cloud database.

When a new item is added, it should be included to the user’s online backup database, as well as a shared database amongst the community of users.

REQ-2: Secured account for individual users

Users will have their own accounts to maintain their own collections.

REQ-3: Verification system

To prevent the addition of objects that would not qualify as “relevant objects,” there should be a verification system for items that are added to the community database for collection types that are well established, such as quarter collections that have official mints.

## Networking with Other Collectors

3.2.1 Description and Priority

This feature is to create a community among users. This can act as both a verification process and a social networking experience such as Facebook, Instagram, and other such services. This feature has a high priority. While it is not integral to the creation of individual collections for each user and the overall functionality, it is a significant goal of this project.

3.2.2 Stimulus/Response Sequence

This feature will be available continuously. Because the database is backed up on a cloud service, it will be available at all times, given that the user allows for shared viewing of their collection. Users will also be allowed to communicate with each other via private messaging and online bulletin boards or forums. It is also possible to use a search engine to find specific characteristics that appear in other collections.

3.2.3 Functional Requirements

REQ-1: Cloud database service

This will allow users to view other users’ collections while they are offline.

REQ-2: Security Features and Privacy Options

For users who would prefer to only maintain their collections for themselves and not for public sharing, the option should be available to control visibility of the collections.

## Object Identification

3.3.1 Description and Priority

This feature will help in “officially” identifying a new object. By validating an object’s identity, a standard for the types of collections can be established. This feature is especially relevant for items such as baseball cards, stamps, and coins.

There are two methods of object identification: 2D and 3D object identification.

This feature holds a medium priority. For the earlier versions of this product, the automated identification process is not essential to allow interactions between users or the maintenance of databases. However, as the product undergoes revisions the automated identification will become higher priority to provide a higher level of service to the users. 2D object identification will be prioritized and made available once complete.

3.3.2 Stimulus/Response Sequence

This option will be available as an alternative to manual input of objects. When a user attempts to input a new object using this method, a service will implemented that will analyze the object based on a product code, serial number, and/or picture identification service.

3.3.3 Functional Requirements

REQ-1: Product Identification Service

This is a service that will have to be created by the developers that can check the serial number or any form of product ID on a common type of object with an official database. For instance, if a user collects a specific brand of toys with a product ID, the information about the product can be automatically input for the user by picking the classification of item and inputting the product code.

REQ-2: Amazon Turk

For items that cannot be officially identified by another database, Amazon Turk can be used in order to identify a product that cannot be automatically identified.

# External Interface Requirements

## User Interfaces

The Android application will have a home page that redirects to each features as specific buttons are pressed. The user will have the option of viewing and editing their own collections, observing other collections, adding new objects, logging out/in to their account, and adjusting privacy settings.

As an Android application, the GUI will be designed for making a simple interface while making it as compact as possible.

## Hardware Interfaces

The hardware interface to the Collection$ application will include a large range of electronics that implement the Android OS. The primary hardware to focus on are smartphones.

## Software Interfaces

The software interfaces required by the Collection$ product include access to a database, a cloud service, and secure logins. The application will also be able to implement the Amazon Turk service in order to verify the identity of newly added objects to a collection genre that becomes recognized by a number of the users in the Collection$ community.

For server operations, Apache Tomcat 7.0 will be used. Apache Tomcat is an open source web server and servlet container developed by Apache Software Foundation. Tomcat provides a web server environment for Java code to run in. This method will be used to connect a MySQL database, which is an open source relational database management system. The MySQL database will be used to hold the items from each user’s collection.

## Communications Interfaces

Collection$ will require access to an online SQL-based database server for storing object information, which will be done as specified in the previous section, utilizing Apache Tomcat. The application will make queries through Tomcat to the MySQL database to view the user’s own collections and other users collections.

Email communications should be limited to notifications about the Collection$ application, including system updates, expected downtime, and server problems. Email will be necessary as a method of registering for an account, but SHALL NOT be shared with other users (users who wish to communicate with each other outside of the application will do so at their own discretion.) Private messaging will be enabled as a means of user-to-user communication.

JavaScript Object Notation (JSON) will be used to send data between the Android and the Java servlet. Because objects cannot be directly passed between machines, it is necessary to convert the information into a format that can be parsed and recreated once it reaches its destination. JSON will be sent to the servlet on the Apache Tomcat server via an Http Post.

# Other Nonfunctional Requirements

## Performance Requirements

Collection$ should be able to function both online and offline. If offline, the application will hold objects in the users personal database until able to “push” the new objects into the online database. This is to avoid a constraint on the users declaring that he/she must be connected for the application to be functional, no matter how minor this issue may be.

## Safety Requirements

Collection$’ server database will act as a backup for the user’s collection. In the event that the user’s collection is lost from their smartphone, an option will be available to download the collections. Syncing the collections will also be a feature to prevent data loss. Server maintenance and security will also be regulated to prevent harm to the user.

## Security Requirements

Each user will be required to register with an email and provide a secure password to login. Users will be urged to take caution in communications with other users and sharing of personal information.

## Software Quality Attributes

Collection$ will be built with future improvements in mind. The Collection$ application should also be eventually deployed to iOS, and should be deployed as web applications. Ease of use is a key attribute for this application. The users should spend more time maintaining their collections rather than trying to understand a wide variety of options and features that are not crucial to the process.

Appendix A: Glossary

IDE – integrated development environment

SQL – structured query language

Appendix B: Analysis Models

Figures and diagrams will be included in later revisions of this document.

Appendix C: Issues List

1. Method of object identification

A choice must be made on the minimum requirement for object identification (one picture from side, multiple pictures from different perspectives, etc.)

\*This list will be updated continuously over the course of the project.