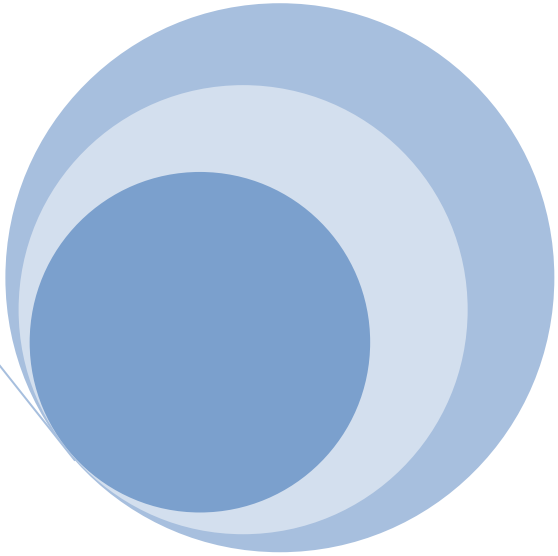

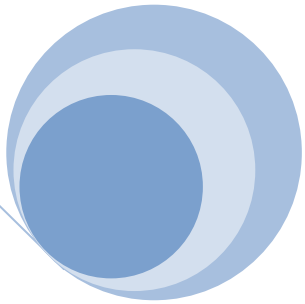

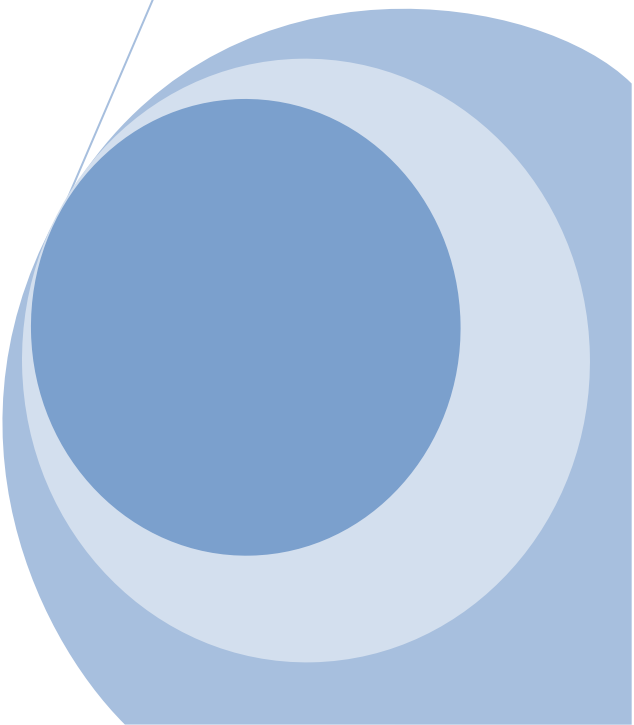

MOBCOLL
wonderland

Software Requirements Specification v1.1

Sinan Küçükköseler
Zeliha Şentürk
Cansu Tetik
Murat Türe
17.01.2013

Table of Contents

1. Introduction 4
   1.1. Purpose 4
   1.2. Scope of the Project 4
   1.3. Definitions and Abbreviations 4
      1.3.1. Definitions 4
      1.3.2. Abbreviations 4
   1.4. Overview 5
   1.5. References 5
2. Background 5
3. Overall Description 6
   3.1. Product Perspective 6
   3.2. Product Functions 6
   3.3. User Types, Constraints and Dependencies 7
   3.4. Hardware Requirements 7
4. Product Features 7
   4.1. Interface Requirements 7
   4.2. Functional Requirements 8
      4.2.1. Register 8
      4.2.2. Login 9
      4.2.3. Login / Register with Social Network Account 10
      4.2.4. Logout 10
      4.2.5. Create Collection 11
      4.2.6. Scan Image 11
      4.2.7. Search for Items 12
      4.2.8. View / Browse Collection 12
      4.2.9. Add New Item by Scanning 13
      4.2.10. Add New Item by Gesture 13
      4.2.11. Viewing Item Sets 14
      4.2.12. View Item Information 14
      4.2.13. Rate Other Collections 15
      4.2.14. Viewing Missing Items of Sets 15

4.2.15. Setting Item Information 16
4.2.16. Viewing Other Public Collections 16
4.2.17. Comparing Collections 17
4.2.18. Lend and Borrow 17
4.2.19. Creating a Switching Deal 18
4.2.20. Viewing Switching Deals by Collector 18
4.2.21. Viewing Switching Deals by Missing Item 19
4.2.22. Real Time Item Information in Camera Mode 19
4.3. Non Functional Requirements 20
4.3.1. Performance Requirements 20
4.3.2. Reliability 20
4.3.3. Usability 20
4.3.4. Security 20
5. CONCLUSION 20

1. Introduction

1.1.Purpose

In this document detailed information regarding the MOBCOLL is given; namely the purpose, features and interfaces of the system and how the system works under specific constraints. Not only the internal structure of the system, but also its reactions to the external structure are also covered in this document.

1.2. Scope of the Project

Keeping track of collection repositories has been a tough problem for collectors. This project comes up with a solution to this problem by allowing the users to have control on their collections, arrange and filter their repositories, view other collectors' items, and check the availability of missing items. MOBCOLL is an application simply to share the collections with other collectors, get detailed information about items, and search for the wanted or missing items. Users can register to the database after downloading the application, and start using the features mentioned above. The application recognizes the image via mobile device's camera, searches the database for that item and shows the basic information regarding the item such that title, set, release date, and owners

1.3.Definitions and Abbreviations

1.3.1. Definitions

Term	Definition
Image Processing	Any form of signal processing applied to images (with the aim of recognition in this project)
Search Engine	A computer program which finds information on the Internet by looking for words that are typed in

1.3.2. Abbreviations

Term	Definition
SRS	Software Requirements Specification
OS	Operating System

1.4. Overview

The next chapter provides information regarding the already existing systems and applications. The third chapter explains the overall description about the project by describing the product perspective, functions and user types constraints and dependencies. In the last chapter of the document, product features and functions are presented to the audience.

1.5. References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Background

Existing media collection systems allow users to select a supported source and try to fetch all the related information online. They provide not only limitless number of catalogues with immense capacity of items, but also a well-developed search engine to filter collection repositories. Moreover, users can publish their collections and keep in touch with other collectors by visiting their profiles via some collection manager applications. These systems have a wide range of use, such that cooking recipes, CD, coins system collection, banknotes collection, stamps catalogue, action figures system collection, DVD movie, postcards collection, books system collection, phone cards, audio - video file. Media collection systems have the following features:

- Integrates with a database
- Enables users to sort and filter libraries
- Search
- Publish
- Register, login, logout
- View collections of other collectors
- Exports to CSV, PDF, XML, HTML and Excel files
- Has built-in multi-language support

MOBCOLL will include the first six functionalities mentioned above. Even though it is not a prior concern, multi-language support is one of the considered features to be added to the application.

Moreover, there are many card collection management systems. One of them is Stickeroid, with this application the user can keep track of their cards and stickers. However, it lacks recognition of items via Android based device's camera and collection sharing is not allowed. Another popular mobile collection application is Pokémon Trading Card Manager. As its name implies, it is developed only for Pokémon collection cards. It offers some good

facilities to users, on the other hand it does not support card recognition via scanning and its interface is not user-friendly.

3. Overall Description

3.1.Product Perspective

MOBCOLL is a mobile application that will be running standalone in an Android OS device; which will run along and communicate with a database server and web-services running between. It will run as a regular mobile application which may use the functionalities the device and user lets. It is not projected to integrate with other applications on the device. It will have functions as data managing, image recognition, web based searching and web based operations specific to item collecting. It will use existing protocols to communicate with the database server, send and receive requests and data, recognize images and make use of internet within the device's tools.

3.2. Product Functions

MOBCOLL consists of these main tasks:

- Authorization
- Authentication
- Data representing and data managing
- Image recognition
- Web based search

To start with authorization, every user can and may see the collection data which he/she is authenticated to. Users may manage and edit data only about their own collection and they will decide upon their own collections that if they can be viewed by other users. Authentication is the mechanism that every user may use the application with their personal username and password that will let the application know their identity. So the application will run its operations particularly for that user.

Data representing and data managing is the process that the application requests the collection data from the database server through the web services and the user can view their collection, its details. All the permanent data will be stored in the server database. Also users may operate and changes on their own collections or search through other users' public collection data. Image recognition is the mechanism that lets the user scan the collection items with the device camera and let the application recognize it and do necessary operations. Web based search is the process that user may search for any collection information through web via the existing internet connection of the device such as Wi-Fi.

3.3. User Types, Constraints and Dependencies

Users of this application are any Android device user that loads this application to their devices. All of the users are in the same class, only one type of user exists. Operating environment is, as just mentioned above, is an Android OS mobile device. An android device that can support basic dependencies of the application is expected for proper user experience. On the other hand, our database server and services can operate on any OS like Windows or Ubuntu that can supply the database server's fundamental dependencies and needs. One important constraint is privacy and security. Users should be accessing only the authenticated data, vice versa. Another important constraint is the reliability and the operating time of the image recognition process. The mechanism has to work within an acceptable reliability and the process should not take more time to effect the user experience. There are existing libraries that helps through image recognition such as OpenCV, etc. Project depends on these.

3.4. Hardware Requirements

A camera higher than 1.3 mega-pixel is needed for scanning images and decent image processing. The mobile application will be implemented for android higher than version 2.3. So a phone with android gingerbread version is needed. For the web service part, internet is needed to access the web services.

4. Product Features

4.1. Interface Requirements

There will be six menus in the user interface. At first, the user should register and login to get to the main menu tab. Then via that menu, user can easily get to the other menus like search, browse or camera tabs.

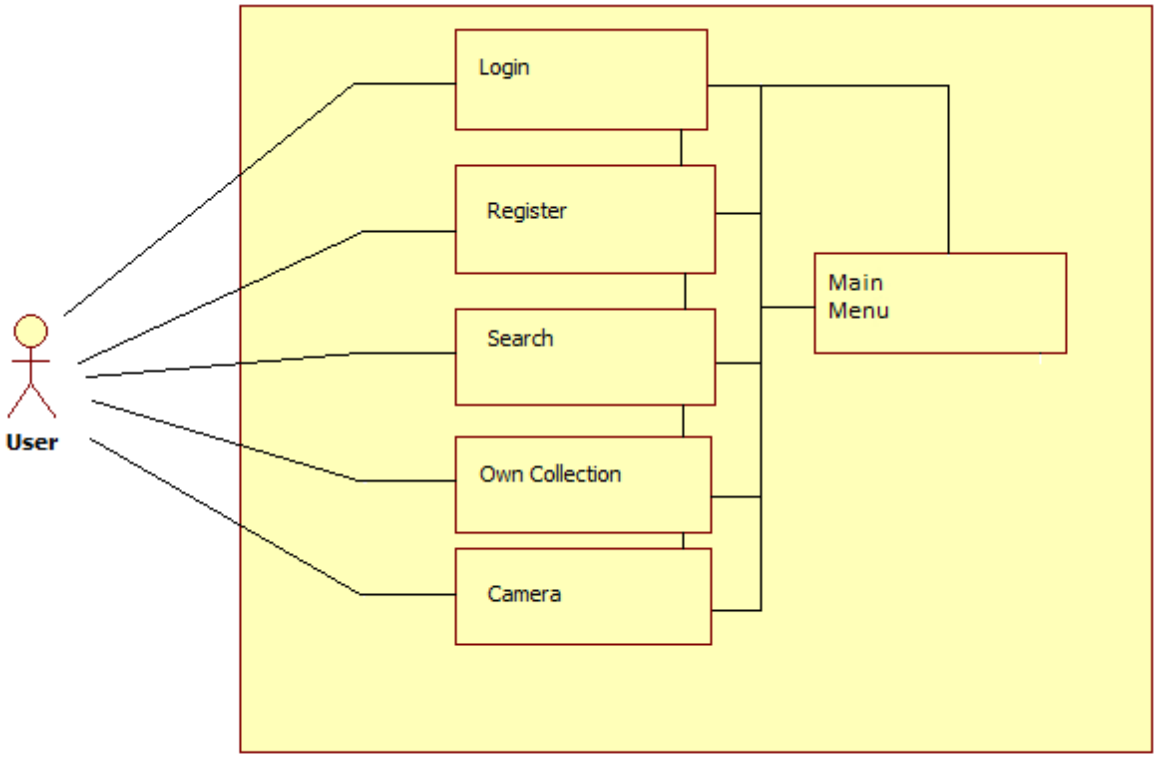

Figure 1 – Interface Requirements

4.2. Functional Requirements

In this section the use cases for the user are shown.

4.2.1. Register

User will register with their name, email and password. Application will record these and grant the user with the authentication when needed.

Severity : Must

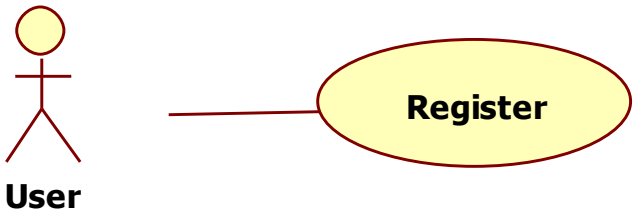

Figure 2 – Register Use Case

4.2.2. Login

Application asks the user for their username and password that have registered before in 4.2.1. User can login with their e-mail and password. Application will check these with the data on database server and authenticate the user if correct. If not, application asks for the correct login information again.

Severity : Must

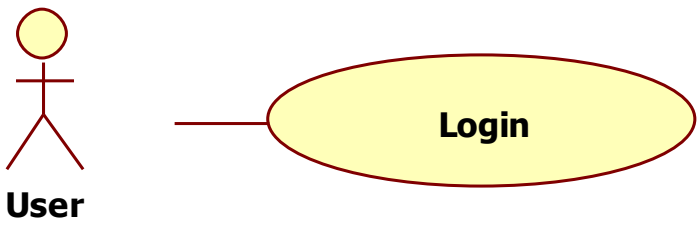

Figure 3- Login Use Case

4.2.3. Login / Register with Social Network Account

User can login and register to the application via Facebook and Twitter accounts.

Severity : Demanded

Figure 4- Login / Register with Social Network Account Use Case

4.2.4. Logout

User can logout from its current authentication. User may decide to do this operation during anytime while running the application. User must login again to return to previous state or screen.

Severity : Must

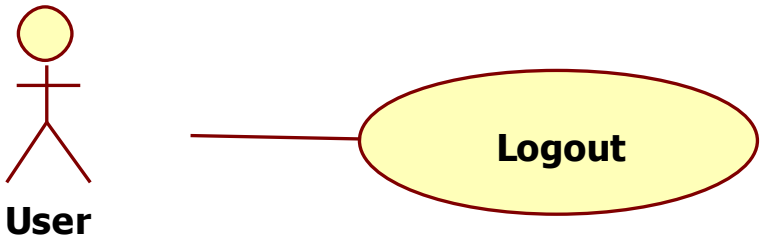

Figure 5 – Logout Use Case

4.2.5. Create Collection

User can create a new empty collection for a specific card collection.

Severity: Must

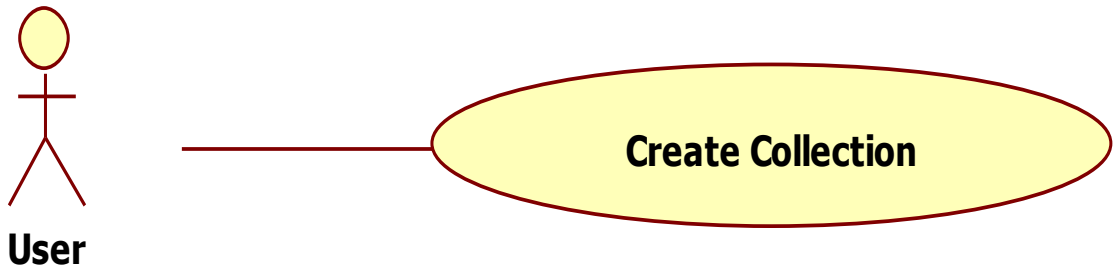

Figure 6 – Create Collection Use Case

4.2.6. Scan Image

User can use the device camera to scan images of items to synchronize them with their collections. Image will be recognized and the recognized item will be included in the collection of the current user. Or the user will be shown the item(s)' details.

Severity : Must

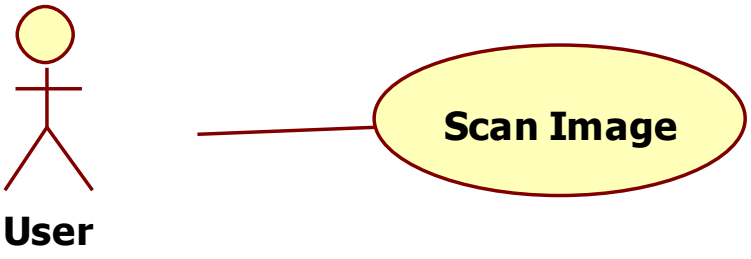

Figure 7 – Scan Image Use Case

4.2.7. Search for Items

User can search for items by scanning the card or using search box and typing the name of the item.

Severity: Must

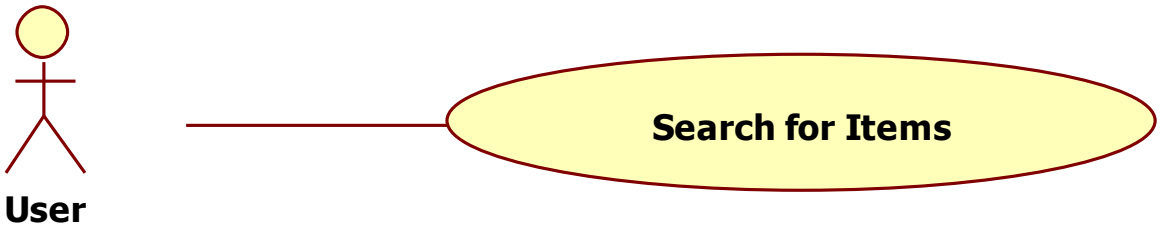

Figure 8 – Search for Items Use Case

4.2.8. View / Browse Collection

User can view and browse the items in its collection. User will also be shown the details of the collection. User may group their collections into item sets.

Severity : Must

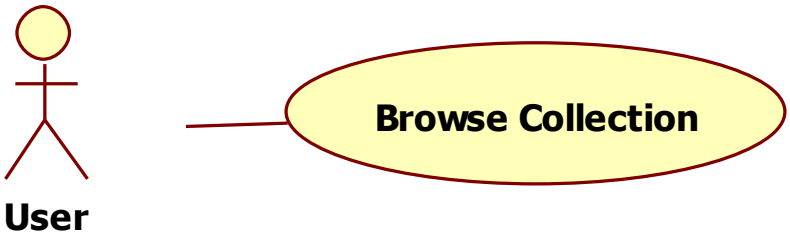

Figure 9 – Browse Collection Use Case

4.2.9. Add New Item by Scanning

User can add new items to their collection by scanning them with camera as seen on

Severity : Must

Figure 10 – Add New Item by Scanning Use Case

4.2.10. Add New Item by Gesture

User can add new items to their collection by searching for a specific item and applying a defined gesture.

Severity: Must

Figure 11 – Add New Item by Gesture Use Case

4.2.11. Viewing Item Sets

There are known complete sets of a specific collection item such as : a complete set of some collection cards. User can view complete item sets like in this example.

Severity : Must

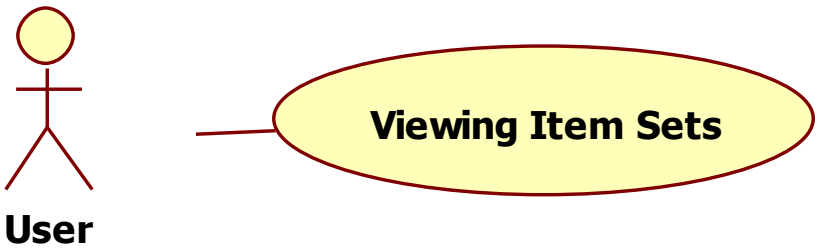

Figure 12- Viewing Item Sets Use Case

4.2.12. View Item Information

User will be able to see the item information by selecting the card on any application page.

Severity : Must

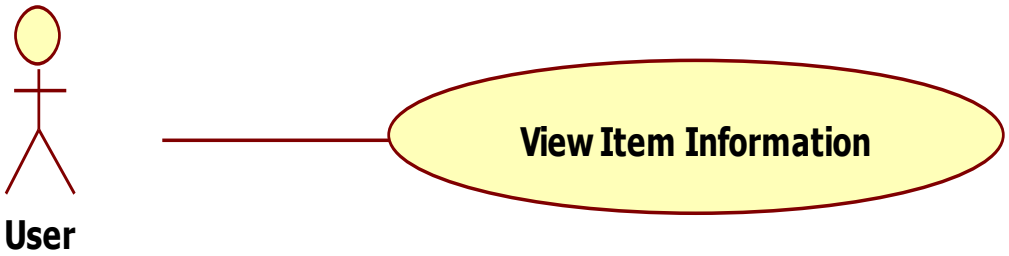

Figure 13 – View Item Information Use Case

4.2.13. Rate Other Collections

User can rate other collections of other users. Total rate of some particular collection can be seen.

Severity : Nice to have

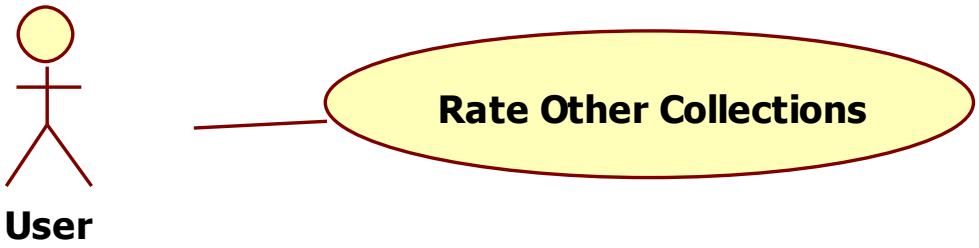

Figure 14 – Rate Other Collections Use Case

4.2.14. Viewing Missing Items of Sets

User can view only the missing items of a particular item set.

Severity : Nice to have

Figure 15 – Viewing Missing Items of Set Use Case

4.2.15. Setting Item Information

It is mentioned that user can view item and collection details. User can set and change the details of their own collections. One descriptive example is setting location information of the items.

Severity : Nice to have

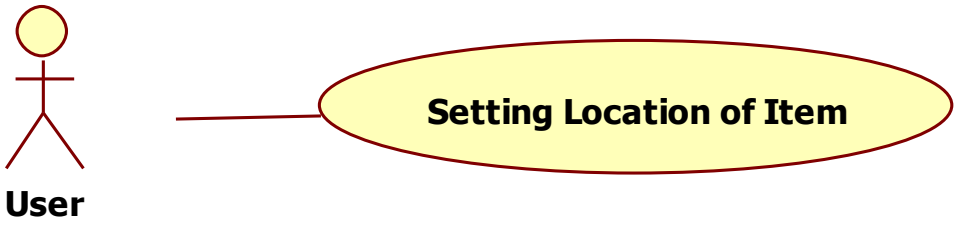

Figure 16 – Setting Location of Item Use Case

4.2.16. Viewing Other Public Collections

User can view other collections which are made public by their owners. To make it more clear, user can only view collections that are set as public by the owner of the collections.

Severity : Must

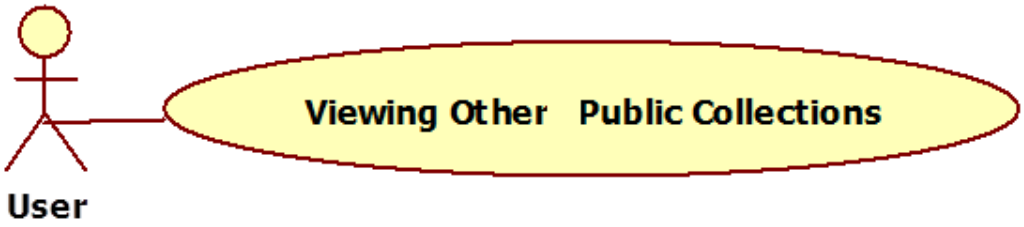

Figure 17 – Viewing Other Public Collections

4.2.17. Comparing Collections

User can compare their own collections or any other public collections to see the missing and surplus items.

Severity : Nice to have

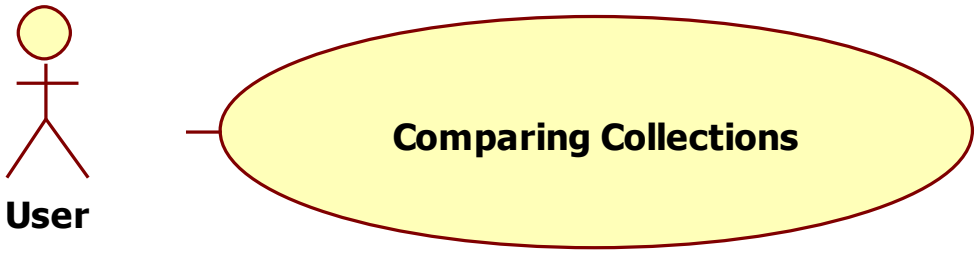

Figure 18 – Comparing Collections Use Case

4.2.18. Lend and Borrow

User will be able to send borrow requests to other collectors, and they can lend the items by accepting these requests.

Severity : Must

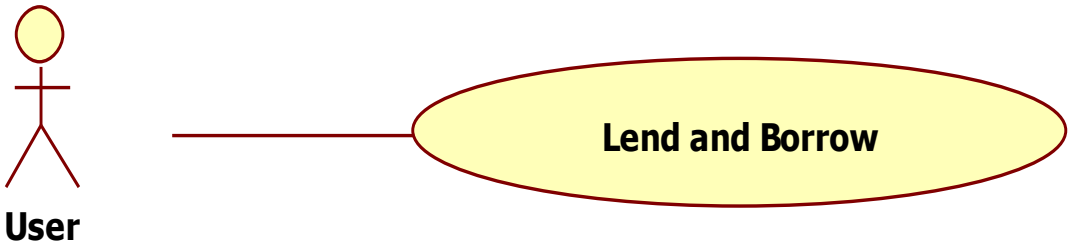

Figure 19 – Lend and Borrow Use Case

4.2.19. Creating a Switching Deal

User can create switching deals. Deals should be created according to the application's deal format like selecting offered item(s), etc. And the deal will be seen by the user that the deal is offered.

Severity : Nice to have

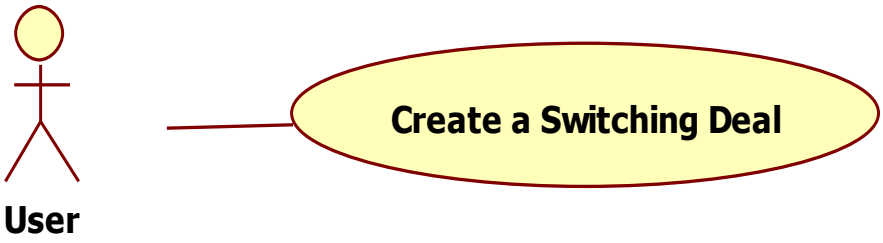

Figure 20 – Create A Switching Deal Use Case

4.2.20. Viewing Switching Deals by Collector

User can search for switching deals with respect to their creators.

Severity : Nice to have

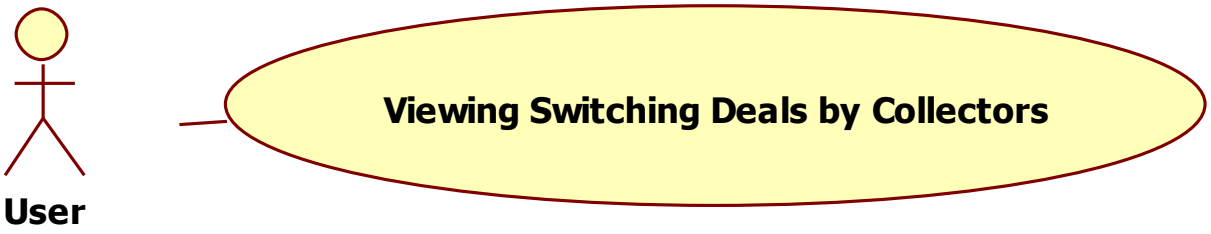

Figure 21 – Viewing Switching Deals by Collectors Use Case

4.2.21. Viewing Switching Deals by Missing Item

User can search for switching deals with respect to their missing items. With the help of this, users may complete their own collections.

Severity : Nice to have

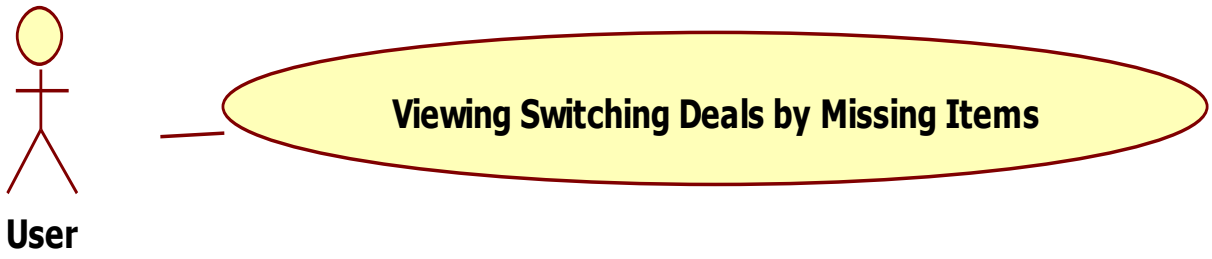

Figure 22 – Viewing Switching Deals by Missing Items Use Case

4.2.22. Real Time Item Information in Camera Mode

User can get real-time augmented information feed about the items that are being scanned. User then also may add the items to their collections as mentioned in 4.2.4.

Severity : Demanded

Figure 23 – Real Time Card Info in Camera Mode Use Case

4.3.Non Functional Requirements

4.3.1. Performance Requirements

To increase responsiveness, the mobile application's graphical interface will work at a rate of minimum 25 fps.

4.3.2. Reliability

To ensure reliability for the web services, whenever possible database information about the collector will be kept at only the users phone, until the user choose to make its collection public. This will reduce the database query overhead of the server, thus making the web service more reliable and responsive.

4.3.3. Usability

One of the main concerns is the usability of the mobile application. The menus can be reached from a main menu which makes it easier to reach to other tabs. Registering new items to the collection through scanning them with the camera , makes the job a lot easier and exciting, thus increasing the usability.

4.3.4. Security

One of the most standard security , which is input validation, will be used to defend against sql injection attacks. Encryption of the server - client communication will be used against network spoolers.

5. Conclusion

MOBCOLL will be a user friendly mobile application which will help collectors keep track of their collections and keep in touch with the other collectors to share, view and trade the items in their collections. With the provided functionalities users can register new items to their collection repositories by just scanning them with their phone's camera. Communicating with the server through web services will enable the users to browse other collectors' items as well as complete sets of their particular items.