SOFTWARE REQUIREMENTS SPECIFICATION FOR DISPOSOSOFIA

Requirements for Version 1.1.0

Prepared by çiftçiler

18.01.2013

TABLE OF CONTENTS

RECORDS OF CHANGES

*A: Added M: Modified D: Deleted

VERSION NUMBER	DATE	NUMBER OF PAGE, FIGURE, TABLE OR SECTION	A* M D	TITLE OR BRIEF DESCRIPTION
1.0.0	11.11.2012			Creation of the Document
1.1.0	17.01.2013	Pg. 8 / Section 1.4	M	Adding Page Titles to References
1.1.0	17.01.2013	Pg. 9 / Section 1.5	M	Editing Overview Section
1.1.0	17.01.2013	Pg. 9 / Section 2.1	M	Editing Product Perspective Section
1.1.0	17.01.2013		A	Conclusion
1.1.0	17.01.2013	Feature 3.2.1.6	M	Editing the feature of version 1.0.0
1.1.0	17.01.2013	Feature 3.2.1.7	M	Changing the scope of the feature
1.1.0	17.01.2013	Feature 3.2.2.2	M	Changing the name of the feature
1.1.0	17.01.2013	Feature 3.2.2.1	D	Deleting ‘Designing Template’ feature
1.1.0	17.01.2013	Feature 3.2.1.8	M	Editing the feature of version 1.0.0
1.1.0	17.01.2013	Diagram 3.2.2.1.2.1	M	Modifying the diagram
1.1.0	17.01.2013	Page 12 / Figure 2	M	Modifying the figure

1. Introduction

1.1 Purpose

This document aims to present a description of the behavior and the requirements of the mobile collection application Disposofia. This document is intended to decrease the effort needed for development, provide statements for validation and verification. The statements in this document will set up a basis for functionality, performance external interfaces, attributes and design of the system.

1.2 Scope

This document details the software requirements for the hobby collector mobile application project named as Disposofia v1.1.0. It will later be used as a base for the extension of the existing software itself. This document follows the IEEE standard for software requirements specification documents.

The hobby of collecting items is in interest of many collector people. They start to collect stamps, books, films, music CDs/DVDs, insects, coins, etc. but when the number of collected items increases, it gets more difficult to keep track of the collection repository. To hold the information about owned items and to be able to reach the documentation (features, details or pictures) of a wished item takes space, time and effort. The list of the collected items is held primitively in text files, or agendas. Disposofia gives convenience and comfort of reaching the searched item in the crowded collection. Moreover Disposofia will essentially perform functions such as adding, deleting, modifying, tracking place of items of collection, borrowing and lending, tracking of borrowed items, management of wanted items, and backup. Also it gives chance to users to design their own template beside using default templates.

The purpose of this project is to provide an easy and entertaining way to hold the collected item information on a mobile application and so provide efficiency in time and effort.

1.3 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers
JRE: Java Runtime Environment
JVM: Java Virtual Machine
DPI: Dots Per Inch
HTTP(Hyper Text Transfer Protocol): It is a transaction oriented client/ server protocol between a web browser and a web server.
XML(Extensible Markup Language): It is a markup language that was designed to transport and store data.

1.4 References

[1] Griffith Media Collection Manager, http://griffith.cc
[2] Flinsoft: Flin4Cell, http://www.flinsoft.com/cell.htm
[3] Movie Collection & Inventory, http://play.google.com/store/apps/details?id=com.metosphere.moviefree&hl=en
[4] Book Inventory Manager, http://play.google.com/store/apps/details?id=com.smartware.mobile.android.bim
[5] TinEye, http://www.tineye.com/
[6] A Picture is Worth a Thousand Keywords, Image-Based Object Search on a Mobile Platform, http://www.cs.utexas.edu/~ai-lab/pubs/yeh_et_al_chi2005.pdf
[7] Comparison of Android devices, http://en.wikipedia.org/wiki/Comparison_of_Android_devices#Tablet_computers
[8] Android Dalvik, http://code.google.com/p/android-dalvik-vm-on-java/downloads/list

[9] GNU General Public License, http://www.gnu.org/copyleft/gpl.html
[10] Backup, http://en.wikipedia.org/wiki/Backup

1.5 Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.
The third chapter, System Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.
Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. Overall Description

In this section, we will give background information about specific requirements of the system. General issues that involve in the production and outline of the functional requirements will be noted, too.

2.1 Product Perspective

Dispososofia is a multifunctional tool for holding collection data. It is free and open source with a GNU General Public License (GPL). Although there are many programs for holding collection data, there are not so many programs to hold a collection with so many features, or to hold more than one collection in a single program as Dispososofia does. Users will have the chance to create new collection types if the built-in collectible types would not be enough or sufficient for the need of users.

2.2 Product Features

The planned version provides:
- Adding, deleting and modifying collection items
- Tracking and searching of items in the database
- Following other user's collection, if they were given permission to others to see, and commenting on other people's hobby items
- Using default or user-designed template
- Borrowing or lending appropriate products if available
- Enable regular backups to protect the collection in case of system failures.
Shortly, it provides an environment for both organizations and users to reserve and share their hobby items

2.2 Product Functions

The entrance to the system and connection to the remote database will be provided through a mobile interface. The following diagram shows the major components of the system and external interactions among them.

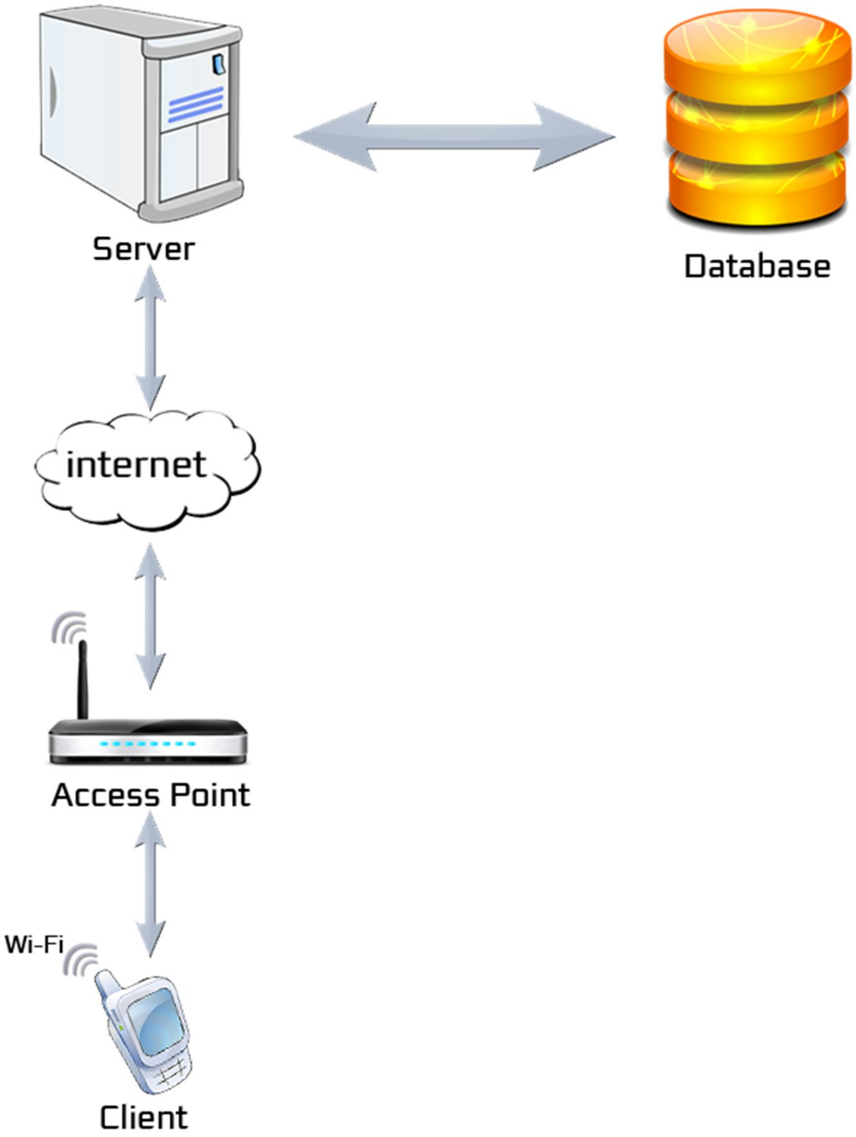


Figure 1 - Simple Network Diagram for Disposososofia

System will provide set of functions which will be applied during the use of it. The set of these functions can be collected under those titles:

- Connecting to the database
- Searching of a collected item on database

This includes the name based searching; feature (content) based searching

- Saving the modifications that have been done on the database

This includes the insertion, deletion, modification of a collected data item

- Backup-restoring of the database

This includes the backup of database in determined intervals for the reason of rescuing and preserving the database when a collapse occurs.

2.3 User Characteristics

This Software Requirements document is intended for:
− Developers who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (design and code the application – it sets the guidelines for future development).
− Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.
− End users of this application who wish to read about what this project can do.

2.4 Constraints

a) Regulatory Policies:

Developers of this system should take into account that the program must be an Android application requiring the Internet connection. Functionality that the program provides should consistent with existing platform.

b) Safety and Security Considerations:

Since the system database connection requires password and user id, developers should take into consideration the safety issues. Also, it satisfies Internet connection, which also requires safety and security issues.

3. System Requirements

3.1 External Interface Requirements

User interface and software interface are prominent point of interaction among elements of the system. For the reason that the application runs on a mobile device, it shall adopt with different screen sizes and resolutions. Any disregarded configuration setting shatters the synergy between the user and Disposofia. Because the product is written in Java programming language, certainly the operating system on mobile device needs a virtual machine to interpret the byte-code of the software. User Interfaces and Software Interfaces is given in more detailed within proceeded sections. Any processor architecture that is supported by Android which is a Unix like OS, as hardware interfaces; and under the name of memory constraints, dynamic random access memory which can take java byte code in, shall be proper for the Disposofia.

3.1.1 User interfaces

Android powered mobile devices differ from each other with respect to screen sizes; for the mobile phones it varies from 2.55 inches to 5.5 inches; in the case of tablets dimension can be more than 10 inches. Therefore user interface shall be qualified for each display dimensions of all these mobiles and supports all screen formats so that it is convenient for customers as many user as possible. Distinct size scales (hdpi, mdpi, ldpi, where h stands for high, m for medium and l for low) requires distinct page layouts for all actuators stands and waits user's preference on display.
Small icons shall be provided for action caller on the system facilitates the users' recognitions and raises usability nonfunctional requirement.

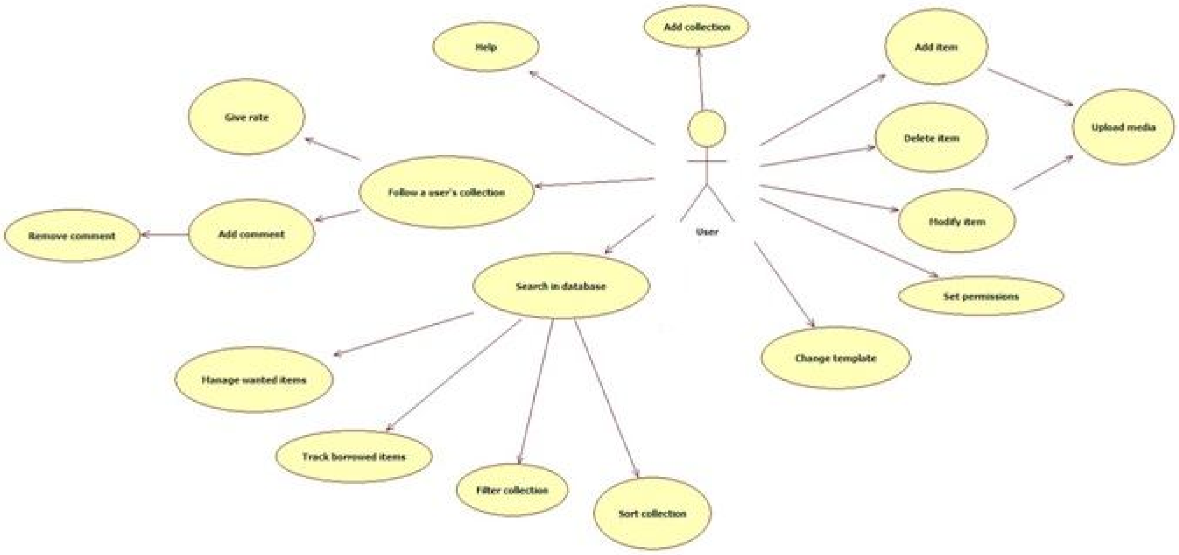


Figure 2 - User Interaction in the System

3.1.2 Software interfaces

Java compiled codes (i.e. byte codes) look for the JRE that also contains JVM to execute the app on mobile. Real time compiling makes Java language more portable. Android operating system has Dalvik virtual machine that based on register- based architecture. Dalvik can also be followed for latest updates and downloaded from [8].

3.1.3 Communications interfaces

For programming interface component of the communication, Java has SQL package that forms a basis for the calling SQL queries for data transaction among client side and database on server side. Generated data streams with the support of Java package flows through the Internet by TCP/IP network protocol.

3.2 Functional Requirements

3.2.1 Basic System Feature

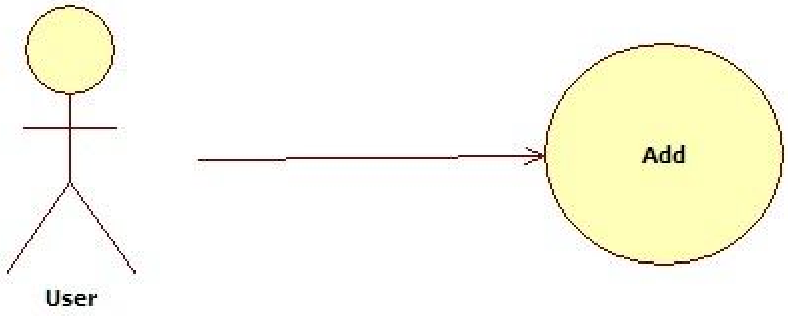
3.2.1.1 System Feature 1 - Adding an Item

3.2.1.1.1 Introduction/Purpose of feature

Add feature serves an empty form to the user by which s/he put a new item in her/his private collection. System warns user for the required fields of form to fill in, and also while user is putting the attribute on the line of field, written characters are checked to keep consistent relation in data type and data model. Irrelevant field generation of user corrupts the stability in database. User can add as many items as s/he wishes to its collection by using add function.

3.2.1.1.2 Stimulus/Response sequence

3.2.1.1.2.1 Diagram



3.2.1.1.2.2 Description

Primary actor:	User
Goal in context:	To add a new item in a collection
Trigger:	To tap on ‘Add New Item’ button in menu bar located on upside of the screen

3.2.1.1.3 Associated functional requirements

3.2.1.1.3.1 Functional requirement 1

System should provide an ‘add new item’ button which provides the user to add a new item information.

3.2.1.1.3.2 Functional requirement 2

System shall view a form according to the collection model that the user wants to add a new item in.

3.2.1.1.3.3 Functional requirement 3

For the sake of usability, system shall view the form in a manner that it can be filled easily.

3.2.1.1.3.4 Functional requirement 4

System shall give an option allowing the user to leave the process without complete.

3.2.1.1.3.5 Functional requirement 5

System should add the item in private collections in personal reserved data base by default.

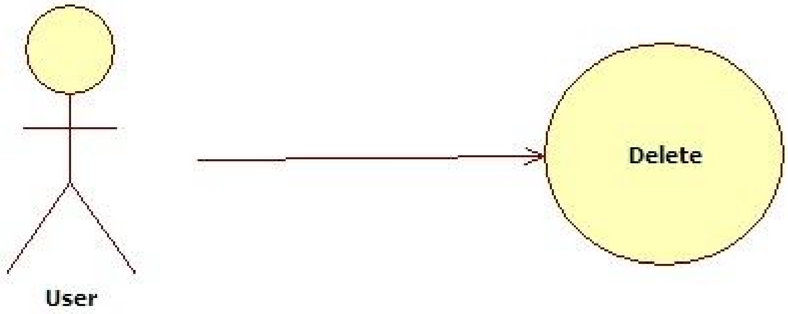
3.2.1.2 System Feature 2 - Deleting an Item

3.2.1.2.1 Introduction/Purpose of feature

Delete item feature is used for the discarded items from a collection. The collection can be both private and public. For this reason system has to be careful when the user attempts to remove an item. The process should respect the relational model among databases. Since deletion is an aspect with no return, system should ask user for the confirmation of the process. The volatilized data may not be restored. In overall, user can delete an old item by using this function. Moreover, user can also delete more than one item at a time.

3.2.1.2.2 Stimulus/Response sequence

3.2.1.2.2.1 Diagram



3.2.1.2.2.2Description

Primary actor:	User
Goal in context:	To delete an existing item from the collection
Trigger:	To tap ‘Delete’ button located on the menu bar

3.2.1.2.3 Associated functional requirements

3.2.1.2.3.1 Functional requirement 1

User should be capable of deleting an item by tapping the ‘Delete’ button which exists on the menu bar.

3.2.1.2.3.2 Functional requirement 2

System should provide a dialog box window asking the user whether s/he is sure or not tapping the ‘Delete’ button.

3.2.1.2.3.3 Functional requirement 3

System shall take relational dependencies into consideration to avoid undesired data losses.

3.2.1.2.3.4 Functional requirement 4

User can select more than one item at a time to delete.

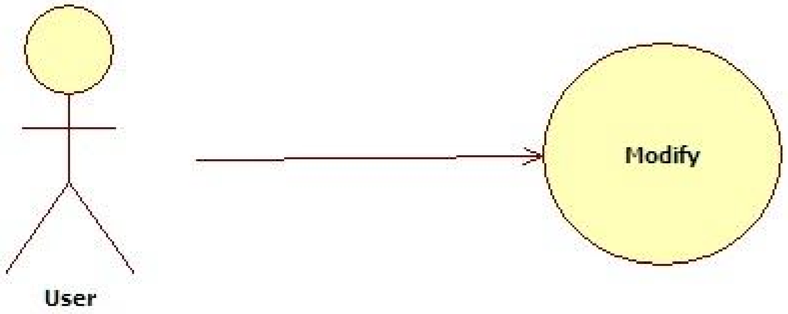
3.2.1.3 System Feature 3 - Modifying an Item

3.2.1.3.1 Introduction/Purpose of feature

Item in a collection may need revision and adjustment on some fields. For instance, a user can change the status of a nonpublic secret item to shared and accessible through the system and vice versa or user can change the name of field for serve different aim. User can make such modifications in the fields of their existing items.

3.2.1.3.2 Stimulus/Response sequence

3.2.1.3.2.1 Diagram



3.2.1.3.2.2 Description

Primary actor:	User
Goal in context:	To modify an existing item field from the collection
Trigger:	To tap 'Edit' button located on the menu

3.2.1.3.3 Associated functional requirements

3.2.1.3.3.1 Functional requirement 1

User should be capable of making modifications on existing item field information by tapping the 'Edit' button which appears after the item is selected.

3.2.1.3.3.2 Functional requirement 2

System should provide a dialog box asking the user whether s/he is sure or not tapping on the 'Edit' button.

3.2.1.3.3.3 Functional requirement 3

System shall reflect the changes on data bases that have relational dependencies.

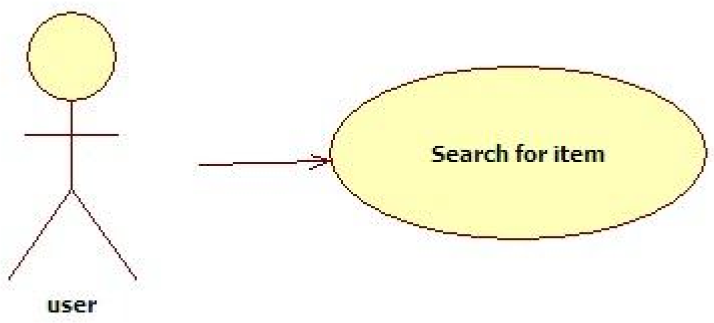
3.2.1.4 System Feature 4 - Searching an Item

3.2.1.4.1 Introduction/Purpose of feature

Especially for the crowded collections, user may suffer from looking an item. Searching feature reduces effort on retrieving a record. Collections may constitute of different fields, accordingly the feature is specialized for each collection to search based on fields of the collection. Since performance issue as a nonfunctional requirement is the key point for this feature, data structure for a collection consists of many record should be decided so that searching algorithm does not exceed order n.logn.

3.2.1.4.2 Stimulus/Response sequence

3.2.1.4.2.1 Diagram



3.2.1.4.2.2 Description

Primary actor:	User
Goal in context:	To search for of an item in a private or public collection
Trigger:	To tap on ‘Search an Item’ button in menu bar located on upside of the screen

3.2.1.4.3 Associated functional requirements

3.2.1.4.3.1 Functional requirement 1

System shall provide user with fields according to the collection that s/he searches for the item on. For instance, if user searches for a music CD, this function will not return the director which is not the attribute of a music CD.

3.2.1.4.3.2 Functional requirement 2

System shall do the search in a reasonable period so that the order of which less than or equal to n.logn.

3.2.1.4.3.3 Functional requirement 3

System shall serve the result of search in an order that takes account of relevance.

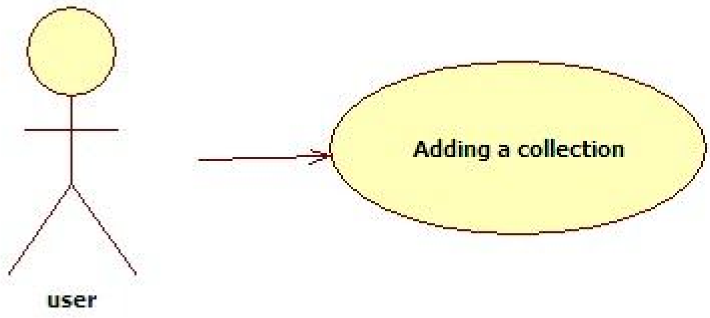
3.2.1.5 System Feature 5 - Adding a Collection

3.2.1.5.1 Introduction/Purpose of feature

For adding a new collection, user needs this feature. Adding a new collection can be done with a typical set of collection template, such as stamp, book, DVD. If user prefers a readymade collection type, system facilitates the process. In case of an individual collection creation, system presents type of fields that the user can append. Eventually a collection form of a combination of set of fields is created in personal database.

3.2.1.5.2 Stimulus/Response sequence

3.2.1.5.2.1 Diagram



3.2.1.5.2.2 Description

Primary actor:	User
Goal in context:	To add a new collection in private database
Trigger:	

3.2.1.5.3 Associated functional requirements

3.2.1.5.3.1 Functional requirement 1

System shall provide a set of custom template collection, as banknote collection, stamp collection, book collection templates.

3.2.1.5.3.2 Functional requirement 2

For a user who wants to create his/her personal collection, wide set of field names and types should be given.

3.2.1.5.3.2 Functional requirement 3

To avoid from duplicate collections, system shall warn user having duplicates.

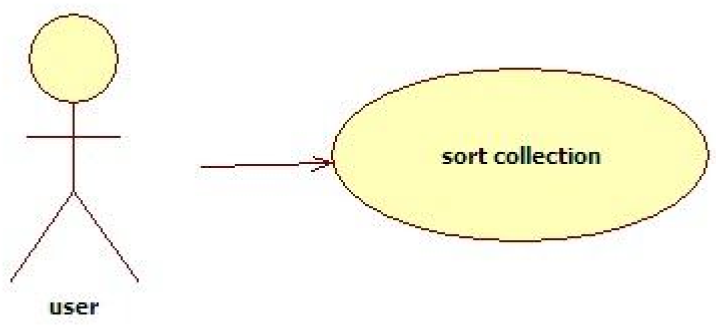
3.2.1.6 System Feature 6 - Sorting a Collection

3.2.1.6.1 Introduction/Purpose of feature

This feature is responsible for listing items in a collection by sorting the collection with respect to a specific order. The sorting type can be determined by alphabetic order or ratings of item in the collection. It is planned to the interaction between view of the collection and user.

3.2.1.6.2 Stimulus/Response sequence

3.2.1.6.2.1 Diagram



3.2.1.6.2.2 Description

Primary actor:	User
Goal in context:	To sort a collection with respect to a field
Trigger:	To tap the name of a field that the user wants to sort and change the order accordingly

3.2.1.6.3 Associated functional requirements

3.2.1.6.3.1 Functional requirement 1

Sorting should be done with regard to data formats.

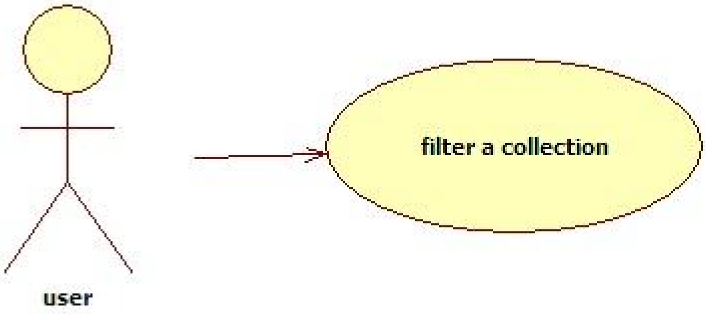
3.2.1.7 System Feature 7 - Filter a Collection

3.2.1.7.1 Introduction/Purpose of feature

This feature is responsible for returning items in a collection to according to specific key. This key may be one of the field name or combination of more than one field name. For example, user can get adventure films or Bon Jovi albums with this function as an output if s/he makes filtering according to film type or album artist respectively.

3.2.1.7.2 Stimulus/Response sequence

3.2.1.7.2.1 Diagram



3.2.1.7.2.2 Description

Primary actor:	User
Goal in context:	To show the items in a collection that can pass through the arranged filter
Trigger:	To tap on names of fields that form the filter

3.2.1.7.3 Associated functional requirements

3.2.1.7.3.1 Functional requirement 1

System should give the whole set of field names for a filter.

3.2.1.7.3.2 Functional requirement 2

The names of fields shall correspond to the collection that the user employs the filter on.

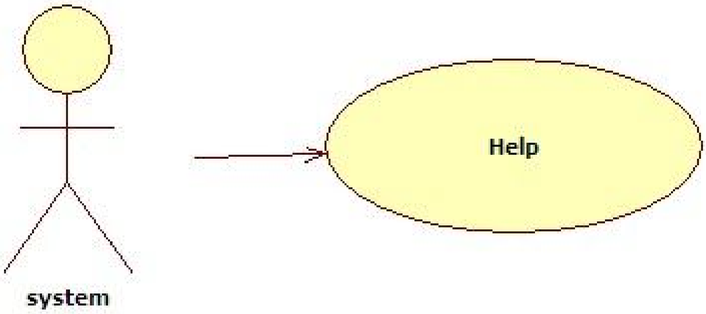
3.2.1.8 System Feature 8 - Helping

3.2.1.8.1 Introduction/Purpose of feature

This feature provides the user to be able to take any information about the system. In particular, it aims to give the user a prospectus of an system feature about "What is it for?" and "How should it be used?".

3.2.1.8.2 Stimulus/Response sequence

3.2.1.8.2.1 Diagram



3.2.1.8.2.2 Description

Primary actor:	User
Goal in context:	To give directions and tips to user about the functionality of the software
Trigger:	When the user tap "Help" tab

3.2.1.8.3 Associated functional requirements

3.2.1.8.3.1 Functional requirement 1

A 'Help' tab shall be provided on the top right corner of the view by the system. This tab shall include a layout of the system. Every title in the layout should be allowed to tap, and should show the information of that title after tapping.

3.2.1.8.3.2 Functional requirement 2

Contact information of the system designers should be provided.

3.2.2 Additional System Feature

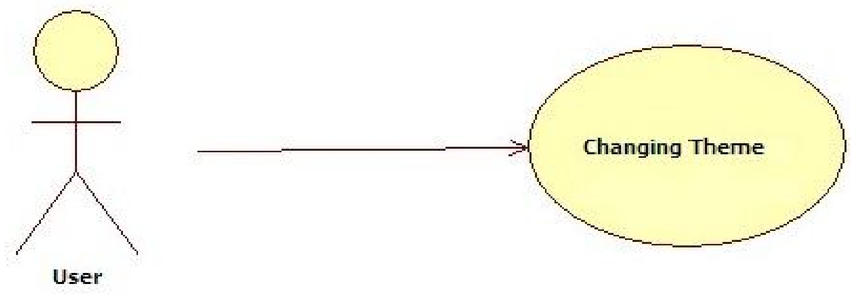
3.2.2.1 System Feature 1 - Changing Theme

3.2.2.1.1 Introduction/Purpose of feature

This feature provides the user to hold his/her collection in an available theme satisfied by the system. It aims bringing practicability and easiness to the user instead of preparing her/his own theme.

3.2.2.1.2 Stimulus/Response sequence

3.2.2.1.2.1 Diagram



3.2.2.1.2.2 Description

Primary actor:	User
Goal in context:	To change the theme
Trigger:	To tap "Change Theme" button

3.2.2.1.3 Associated functional requirements

3.2.2.1.3.1 Functional requirement 1

System should provide the user a 'Change Theme' button, and then present the available (prepared by system designers or user before) themes.

3.2.2.1.3.2 Functional requirement 2

The themes should have 'preview' and 'upload' buttons which satisfy the previewing (without saving) and uploading (with saving) that theme, respectively.

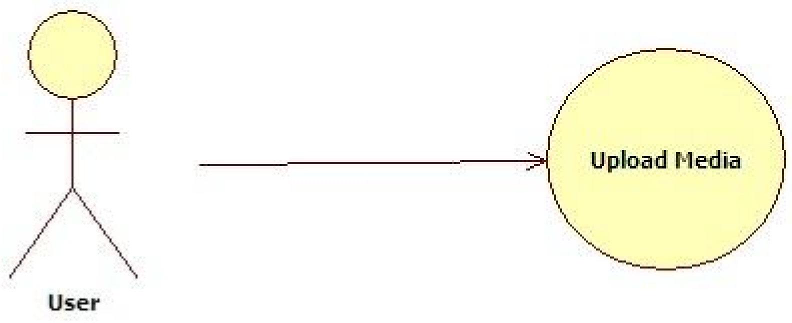
3.2.2.2 System Feature 2 - Uploading Pictures and Videos

3.2.2.2.1 Introduction/Purpose of feature

This feature provides the user to upload the pictures and videos related to that collection item.

3.2.2.2.2 Stimulus/Response sequence

3.2.2.2.2.1 Diagram



3.2.2.2.2.2 Description

Primary actor:	User
Goal in context:	To upload related videos and pictures for items
Trigger:	To tap 'Upload Picture' or 'Upload Video' buttons which are located next to the row of the item at which a video or picture will be uploaded

3.2.2.2.3 Associated functional requirements

3.2.2.2.3.1 Functional requirement 1

The user should be able to upload pictures or videos by tapping the 'Upload Picture' and 'Upload Video' buttons, respectively, which are available on the menu.

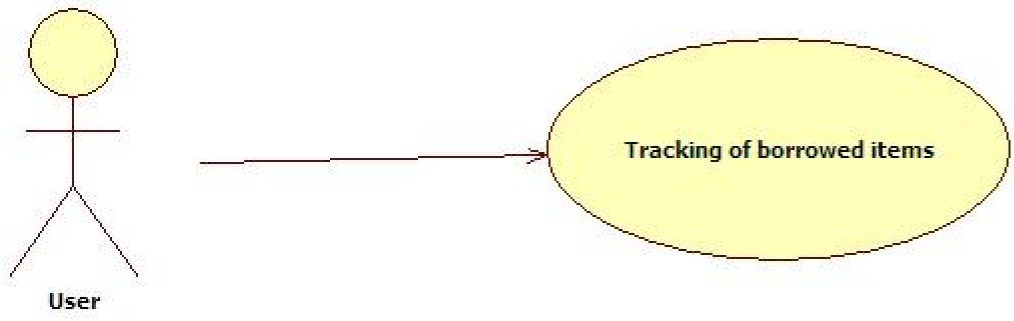
3.2.2.3 System Feature 3 - Tracking of borrowed items

3.2.2.3.1 Introduction/Purpose of feature

An item can be borrowed from someone else with an agreement on date of taking back. The feature saves the borrower from being ashamed towards lender by sending a message that alerts the day of giving back.

3.2.2.3.2 Stimulus/Response sequence

3.2.2.3.2.1 Diagram



3.2.2.3.2.2 Description

Primary actor:	System
Goal in context:	To follow the borrowed items
Trigger:	To tap "Track borrowed items" button

3.2.2.3.3 Associated functional requirements

3.2.2.3.3.1 Functional requirement 1

System will hold the name of user who borrowed the item.

3.2.2.3.3.2 Functional requirement 2

System will hold the information about the start time of borrow, end time of borrow, length of borrow time which will be defined by owner of the collection borrowed.

3.2.2.3.3.3 Functional requirement 3

System will give warning message to borrower three days before the end time of borrow.

3.2.2.3.3.4 Functional requirement 4

System will give warning message to collector who borrows the item.

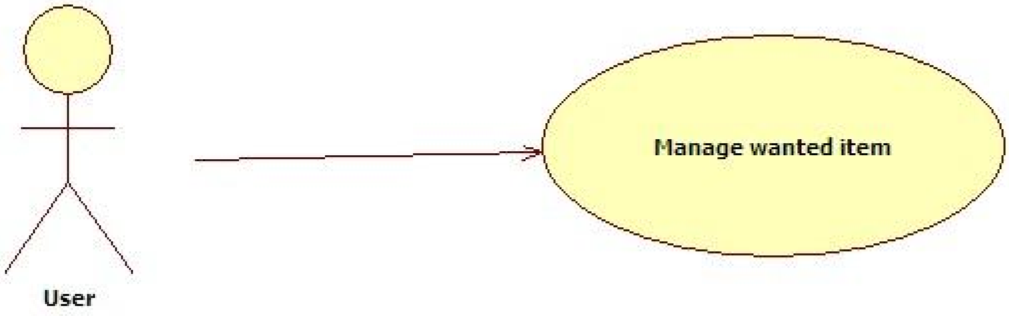
3.2.2.4 System Feature 4 - Wanted item management

3.2.2.4.1 Introduction/Purpose of feature

This feature consists of a table which shall show the wished but missing collection items. It shall be visible to followers so that a follower who has that item can get into contact with the user. It shall be able to be in the place of a reminder of wished items to the user.

3.2.2.4.2 Stimulus/Response sequence

3.2.2.4.2.1 Diagram



3.2.2.4.2.2 Description

Primary actor:	User
Goal in context:	To keep the system informed about the items that looked for by the user
Trigger:	To tap “Want an item” button

3.2.2.4.3 Associated functional requirements

3.2.2.4.3.1 Functional requirement 1

A table that shall hold the wished items notifications should be provided by the system.

3.2.2.4.3.2 Functional requirement 2

The table shall be able to be marked by a follower in the case that s/he wants to share that item information with the user.

3.2.2.4.3.3 Functional requirement 3

An alert system should be built for the user in the case of a wished item notification comes from a follower.

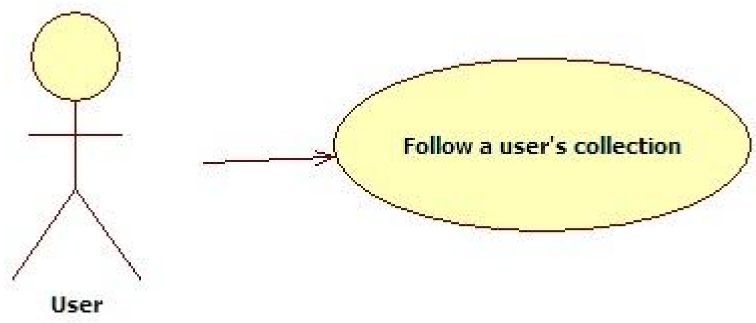
3.2.2.5 System Feature 5 - Following the User’s Collection

3.2.2.5.1 Introduction/Purpose of feature

This feature provides the other people who are interested in collections to follow any user who use this program through the own network of the application. With this feature, it is purposed that the users can recognize each other and so get interaction with each other, and get fun by observing each others’ public collection.

3.2.2.5.2 Stimulus/Response sequence

3.2.2.5.2.1 Diagram



3.2.2.5.2.2 Description

Primary actor:	User
Goal in context:	To follow other user’s collections
Trigger:	To tap ‘Follow this collector’ button

3.2.2.5.3 Associated functional requirements

3.2.2.5.3.1 Functional requirement 1

A ‘Follow this collector’ button shall be submitted by the system in the top right corner of the user’s page.

3.2.2.5.3.2 Functional requirement 2

System will hold the number of person who follows that collection.

3.2.2.5.3.3 Functional requirement 3

System will update number of person who follow that collection when the number of follower is changed.

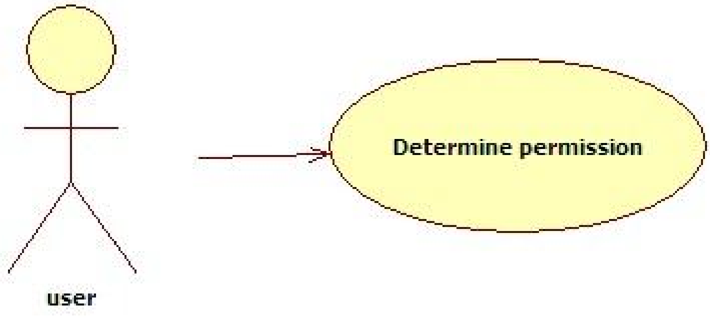
3.2.1.6 System Feature 6 - Setting Permissions of Items

3.2.1.6.1 Introduction/Purpose of feature

This feature provides the user to determine a permission setting on visibility of each collection item in the network that is created by the application. It aims that the user gets satisfaction by sharing that item to followers, or makes it invisible to the others in the case that s/he wants to hold it as secret. There will be 2 privacy levels; one of them is making everything secret as default, the other one is giving permission only to followers.

3.2.1.6.2 Stimulus/Response sequence

3.2.1.6.2.1 Diagram



3.2.1.6.2.2 Description

Primary actor:	User
Goal in context:	To set permission of items about which part of collection can be seen by others
Trigger:	To modify the state of a field in an item from private to public.

3.2.1.6.3 Associated functional requirements

3.2.1.6.3.1 Functional requirement 1

Each item shall have a permission settings menu next to it.

3.2.1.6.3.2 Functional requirement 2

Scope of being public for each item should be determined by a state of a field shows that it is sharable or not for the user.

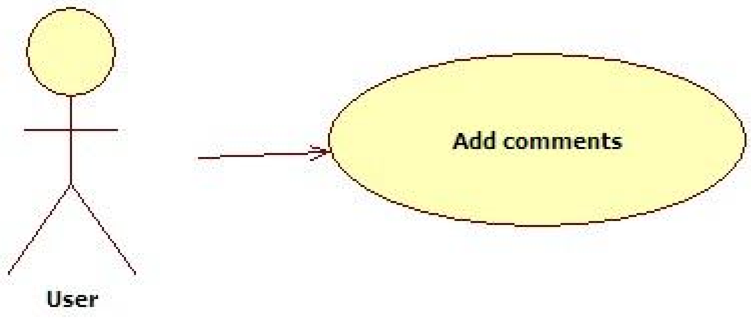
3.2.2.7 System Feature 7 - Adding Comments

3.2.2.7.1 Introduction/Purpose of feature

This feature provides the user and her/his followers make comments for a collection item. The comments shall be provided to be made under the pictures or videos of that item.

3.2.2.7.2 Stimulus/Response sequence

3.2.2.7.2.1 Diagram



3.2.2.7.2.2 Description

Primary actor:	User
Goal in context:	To add comments about a collection item
Trigger:	To tap 'Add Comment' button

3.2.2.7.3 Associated functional requirements

3.2.2.7.3.1 Functional requirement 1

An 'Add Comment' form shall be provided under the each photo and video. The comments shall be shown together with its commentator name, and the date it is made.

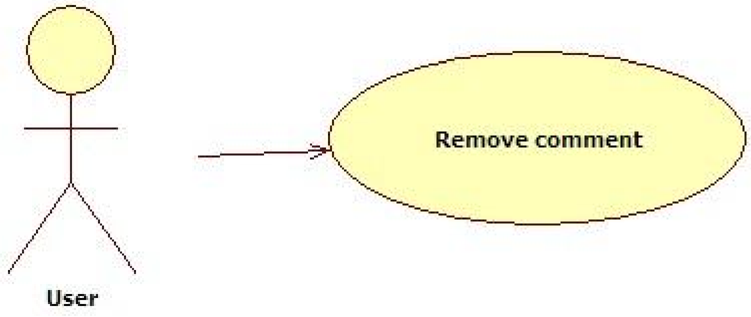
3.2.2.8 System Feature 8 - Removing Comment

3.2.2.8.1 Introduction/Purpose of feature

This feature provides the user and her/his followers remove the comments which were made by them for a collection item. The user, as the administrator of her/his own system, can remove the comments of her/his followers, also.

3.2.2.8.2 Stimulus/Response sequence

3.2.2.8.2.1 Diagram



3.2.2.8.2.2 Description

Primary actor:	User
Goal in context:	To delete comments about a collection item
Trigger:	To tap 'Remove Comment' button

3.2.2.8.3 Associated functional requirements

3.2.2.8.3.1 Functional requirement 1

A delete comment button next to the each comment should be provided by the system.

3.2.2.8.3.2 Functional requirement 2

A delete action can be performed by the user who made the comment.

3.2.2.8.3.3 Functional requirement 3

A delete action can be performed by the user whose item was commented.

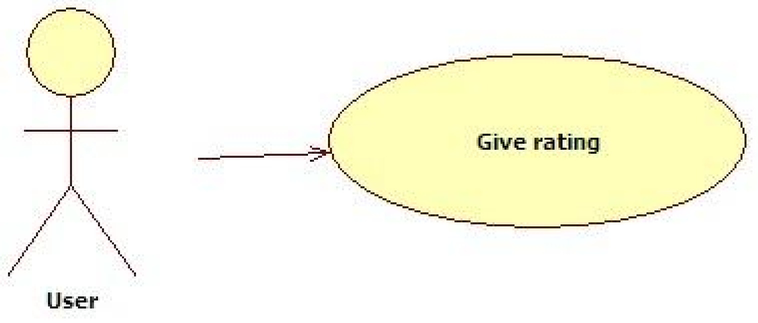
3.2.2.9 System Feature 9 - Giving Ratings

3.2.2.9.1 Introduction/Purpose of feature

This feature provides the user and her/his followers give rating for a liked item. Rating information of the items is going to be used in ordering after the item search.

3.2.2.9.2 Stimulus/Response sequence

3.2.2.9.2.1 Diagram



3.2.2.9.2.2 Description

Primary actor:	User
Goal in context:	To rate or give point to a collection item
Trigger:	To tab ‘Rating Indicator’

3.2.2.9.3 Associated functional requirements

3.2.2.9.3.1 Functional requirement 1

A rating indicator should be provided next to the each collection item by the system.

3.2.2.9.3.2 Functional requirement 2

Rating indicators should be allowed to tap. A user or follower shall be able to give point to that item from the pointing options chart, which will be shown by the system after tapping the rating indicator.

4. Nonfunctional Requirements

4.1 Backup

Since all users would like to use Dispososofia to keep information about their items, preventing any data loss becomes an important issue for the application. For this reason, system will provide backup of datum of collection for all users in case of system corruption or data loss. System will perform periodically based backing up automatically and change the content of old backup. The period will depend on decision of the user.

4.2 GNU GPL

This projected will be released under the GNU General Public License. It is a free software license which allows everyone use or make some changes to redistribute it without paying anyone.

- Anyone can run the program with any purpose
- Source code of the application should be accessed to let users see how the program works or use it if necessary
- Source code of the application can be changed or improved for the use of everyone again

5. Conclusion

In conclusion, this document is prepared to give details about the requirements of the mobile collection application Disposofia. In the process of preparation of this report, needs of the collectors and lack of similar applications in the market are taken into consideration.