

SOFTWARE DESIGN DESCRIPTION FOR DISPOSOSOFIA

18.01.2013

Design For Version 1.1.0

Prepared by çiftçiler

TABLE OF CONTENTS

RECORDS OF CHANGES	4
1. OVERVIEW	5
1.1 PURPOSE.....	5
1.2 SCOPE.....	5
1.3 INTENDED AUDIENCE.....	5
1.4 REFERENCES.....	5
2. DEFINITIONS	5
3 DESIGN VIEWPOINTS	6
3.1 INTRODUCTION.....	6
3.2 CONTEXT VIEWPOINT.....	6
3.2.1 <i>DESIGN CONCERNS</i>	6
3.2.2 <i>DESIGN ELEMENTS</i>	7
3.2.2.1 Create Services.....	8
3.2.2.2 Insert Services.....	8
3.2.2.3 Update Services.....	8
3.2.2.4 Delete Services.....	8
3.2.2.5 Search Services	9
3.3 LOGICAL VIEWPOINT.....	9
3.3.1 <i>DESIGN CONCERNS</i>	9
3.3.2 <i>DESIGN ELEMENTS</i>	9
3.3.2.1 <i>User Module</i>	9
3.3.2.2 MusicCollection Module	10
3.3.2.3 BookCollection Module	11
3.3.2.4 FilmCollection Module	12
3.3.2.5 StampCollection Module.....	13
3.3.2.6 InsectCollection Module.....	14
3.3.2.7 GenericCollection Module.....	15
3.4 TRACEABILITY MATRIX.....	16
3.5 DEPENDENCY VIEWPOINT.....	17
3.5.1 <i>DESIGN CONCERNS</i>	17
3.5.2 <i>DESIGN ELEMENT</i>	17
3.5.2.1 <i>UserCollection Module Dependency Description</i>	17
3.5.2.2 MusicCollection Module Dependency Description	17
3.5.2.3 BookCollection Module Dependency Description	17
3.5.2.4 FilmCollection Module Dependency Description	17
3.5.2.5 StampCollection Module Dependency Description.....	18
3.5.2.6 InsectCollection Module Dependency Description.....	18
3.5.2.7 GenericCollection Module Dependency Description	18
3.5.3 <i>EXAMPLE LANGUAGE</i>	19
3.6 USER INTERFACE VIEWPOINT.....	19
3.6.1 <i>DESIGN CONCERNS</i>	19
3.6.2 <i>DESIGN ELEMENTS</i>	20

3.6.2.1 UserCollection Module Interface Description.....	20
3.6.2.2 MusicCollection Module Interface Description.....	21
3.6.2.3 BookCollection Module Interface Description.....	22
3.6.2.4 FilmCollection Module Interface Description.....	23
3.6.2.5 StampCollection Module Interface Description	24
3.6.2.6 InsectCollection Module Interface Description.....	25
3.6.2.7 GenericCollection Module Interface Description.....	26
3.6.3 <i>EXAMPLE LANGUAGE</i>	26
3.7 INTERACTION VIEWPOINT.....	40
3.7.1 <i>DESIGN CONCERNS</i>	40
3.7.2 <i>DESIGN ELEMENTS</i>	41
3.8 STATE DYNAMICS OVERVIEW	41
3.8.1 <i>DESIGN CONCERNS</i>	41
3.8.2 <i>DESIGN ELEMENTS</i>	42

RECORDS OF CHANGES

VERSION NUMBER	DATE	NUMBER OF PAGE, FIGURE, TABLE OR SECTION	A* M D	TITLE OR BRIEF DESCRIPTION
1.0.0	3.12.2012			Creation of the Document
1.1.0	17.01.2013		A	Traceability Matrix
1.1.0	17.01.2013	Pg. 8 / Figure 1	A	Class Diagram for Web Service Files
1.1.0	18.01.2013	Pg. 8 / 3.2.2.1 – 3.2.2.5	A	Addition to 'Design Elements' Section
1.1.0	18.01.2013		M	Renaming Names of Figures

*A: Added M: Modified D: Deleted

1. OVERVIEW

This document describes the design for hobby collection system named Dispososofia which will operate in mobile platform. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it contains the information required to write code. In order to construct the system modularly, we will investigate the system in context viewpoint, composition viewpoint, logical viewpoint, dependency viewpoint, interface viewpoint, structure viewpoint, interaction viewpoint and state dynamics viewpoint. Moreover we will provide some diagrams in order to increase the understandability.

1.1 PURPOSE

This document is prepared for the purpose of constructing design model with structural, hierarchical, interface, module level representations. Design model that is used in this software design document is improved before the code is generated for the purpose of quality. For the aim of constructing Dispososofia software, graphical representations and verbal explanations were added.

1.2 SCOPE

Architectural design, dependencies and the detailed design for the Dispososofia software will be provided in this document. The purpose, identification, function and subordinates, dependencies, interfaces, resources, states and the data attributes of the each component will be given.

1.3 INTENDED AUDIENCE

Developers, programmers and people who are interested in collections and Android applications can be intended audience of Dispososofia.

1.4 REFERENCES

IEEE standard 1016-2009 recommended practice for software design descriptions.

UML: The Unified Modeling Language User Guide by G. Brooch, J. Rumbaugh and I. Jacobson, Addison-Wesley Pub Co; ISBN:0-201-57168-4.

Android Training Tutorials, <http://www.lynda.com/Android-training-tutorials/947-0.html>

Android Application Development for Dummies by D. Felker

Android UI Fundamentals Develop and Design by J. Ostrander

2. DEFINITIONS

IEEE – Institute of Electrical and Electronics Engineers, the “world’s largest professional association for the advancement of technology”.

PHP - Hypertext Preprocessor, server-side HTML embedded scripting language.

Java - a programming language and computing platform first released by Sun Microsystems in 1995 and it is the underlying technology that powers state-of-the-art programs including utilities, games, and business applications.

Class Diagram – A UML format that describes classes graphically. Specifically, it describes their instance variables, method headers, and relationships to other classes.

Sequence Diagram – A UML document format that specifies how object methods interact with one another.

UML – Unified Modeling Language, a standard set of document formats for designing software graphically.

Dispososofia – A hobby collection application for mobile phones using Android operating system.

3 DESIGN VIEWPOINTS

3.1 INTRODUCTION

Several design viewpoints in terms of design concerns for use in SDD will be defined in following subsections. UML shall be used as a design language.

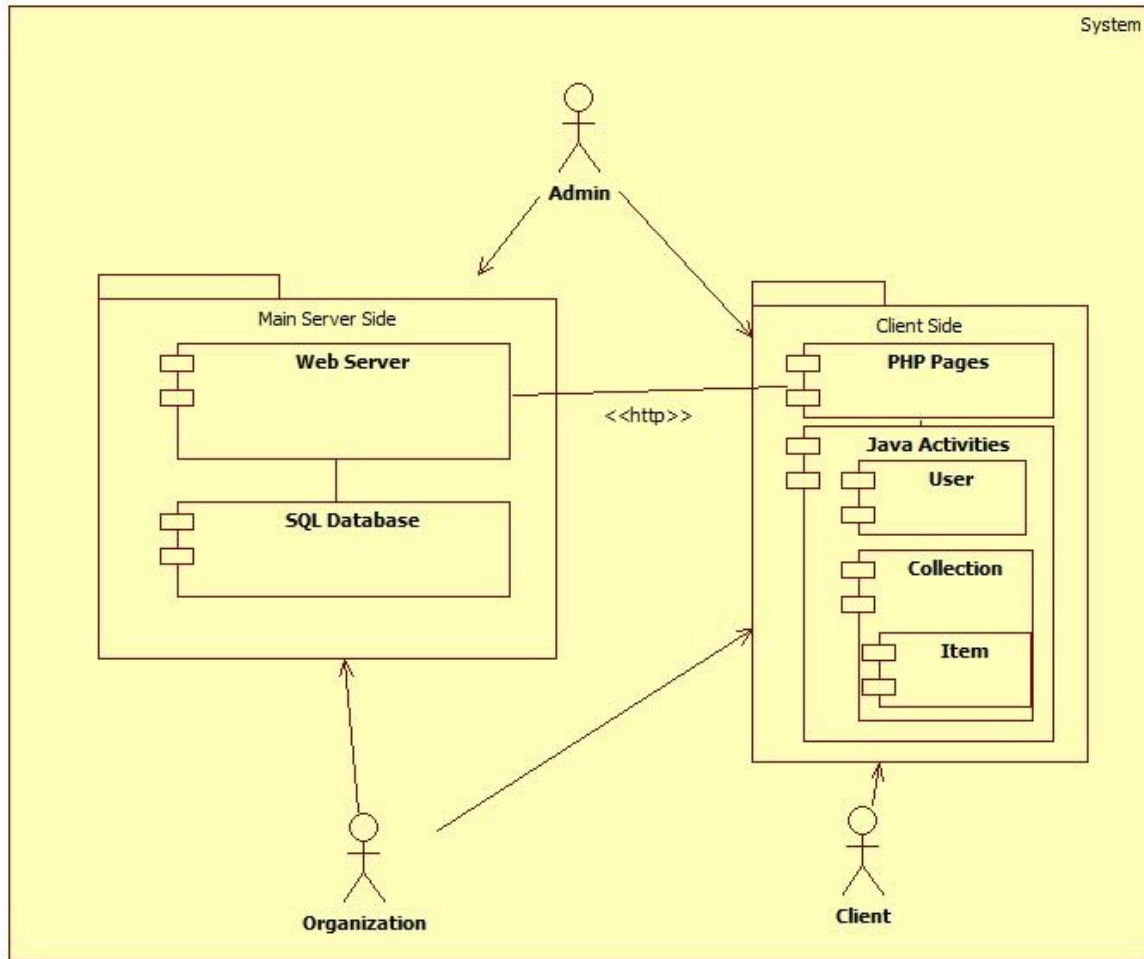
3.2 CONTEXT VIEWPOINT

Services provided by a design subject is characterized by the context viewpoint with reference to an understandable context. That context is defined by reference to actors that include users and other stakeholders, which interact with the design subject in its environment. Services represent fundamentally functional aspect or predicted cases of use of the design subject.

3.2.1 DESIGN CONCERNS

The purpose of the context viewpoint is to classify offered services of design subject, its actors which are users and other interacting stakeholders, to establish the system boundary and to effectively indicate the scope of use of design subject and operation.

3.2.2 DESIGN ELEMENTS



Services takes queries from java activities on device via http post method and send them to the MySQL. Results of queries are read to a stream and edited before presenting to user on device screen. Following class diagram describes the structure of PHP files for web services.

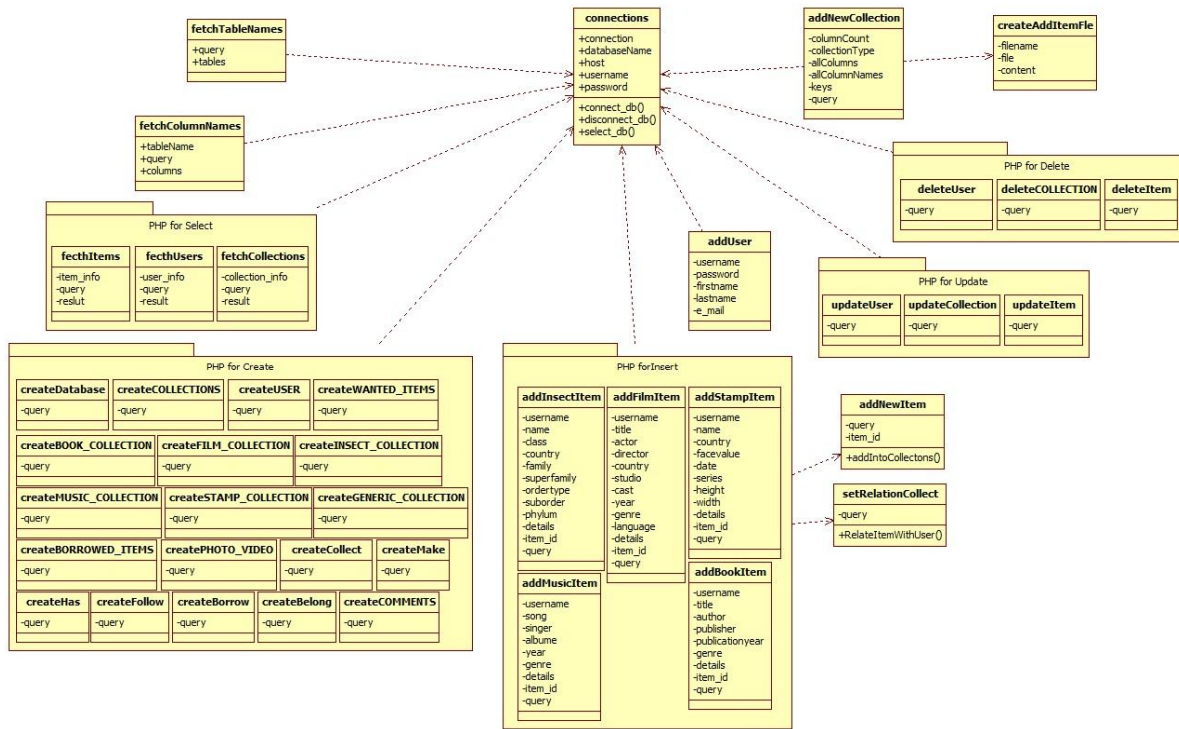


Figure 1 - Class Diagram for Web Service Files

We can divide them into five general classes.

3.2.2.1 Create Services

These set of services are programmed in order to create a service, a database, tables of collections (for both predefined collection types as book, stamp, music, film, insect and user-defined generic type), users, wanted items, comments and relations among them.

Every generated generic type of collection feels the need for services that can add, remove, modify and search on them. These requirements are also considered and PHP files responsible for these new collection types are also supported.

3.2.2.2 Insert Services

Insert services are responsible for appending new row in a table. The table can be related with users, a predefined collection, a user-defined collection and dependency of any pair of tables.

3.2.2.3 Update Services

These services are triggers the database queries starts with "UPDATE ...". Any type of modification oriented PHP file can be considered in this class. They manage changes on attributes of an item and take into account of relational dependencies.

3.2.2.4 Delete Services

Delete operations are run on user table and collection tables. As other services, delete takes care of relations of table according to defined options.

3.2.2.5 Search Services

Service directs user input with a declared category (user, collection and item) to database to fetch convenient rows. If a user is sought, then service checks fields of user table to catch equivalence. If it is a collection, search is called for every collection names. Either if the search category is an item, all shared items are scanned. The results of queries are returned on http response.

3.3 LOGICAL VIEWPOINT

The aim of the logical viewpoint is to expand existing and designed types and their implementations as classes and interfaces with their structural static relationships. In this subsection the Dispososofia software system will be decomposed into modules. Each module will have a name and identification information which is unique to that module. These name and identification information will be used to identify the module. Purpose attributes will be the aim of the function. Other module that makes up the system will be referred at Relationship attribute. Also in order to provide a more understandable module description a class diagram for each module will be given.

3.3.1 DESIGN CONCERNS

The Logical viewpoint is used to address the development and reuse of adequate abstractions and their implementations.

3.3.2 DESIGN ELEMENTS

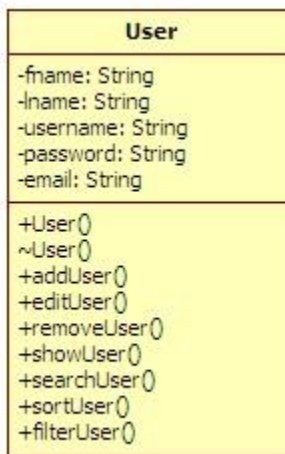
3.3.2.1 User Module

Background Information:

This module will be used for generating user profile. It has basic functions such that “addUser” that adds new user into system, ”editUser” that modifies information of existing user, ”removeUser” that will delete selected user from the system, ”searchUser” that returns the search result for the user, ”sortUser” that sorts the search result according to criteria of user who makes search, ”filterUser” that filters the search result according to criteria of user who makes search.

Description:

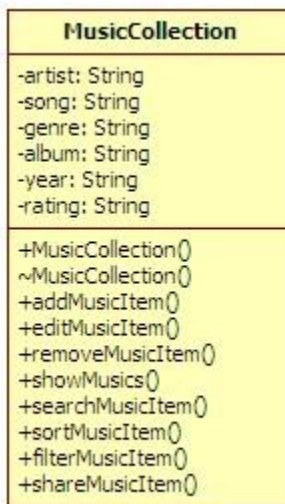
Identification:	User
Purpose:	Purpose of this module is to provide user profile for the Dispososofia software system.
Attributes:	“fname”, ”lname”, ”username”, ”password”, ”email” are needed attributes for User module.
Relationships:	This module directly associated with Generic Collection module.

Diagram:*3.3.2.2 MusicCollection Module***Background Information**

This module will be used to create a predefined music collection for the user by using items in the collections defined by organizations. It has some basic functions : ‘MusicCollection()’ creates an empty predefined music collection for the user, ‘addMusicItem’ allows user to add new songs into the collection; ‘editMusicItem’ and ‘removeMusicItem’ allow user to modify and remove music items which are already in the collection, respectively; ‘showMusics’ allows user to see the details of the selected song, like artist name, genre, year of the song; ‘searchMusicItem’, ‘sortMusicItem’ and ‘filterMusicItem’ allow searching, filtering with respect to some fields, and sorting the fields of the desired song from the collection list, respectively; and ”shareMusicItem” that publishes music item to other users if item has public permission.

Description:

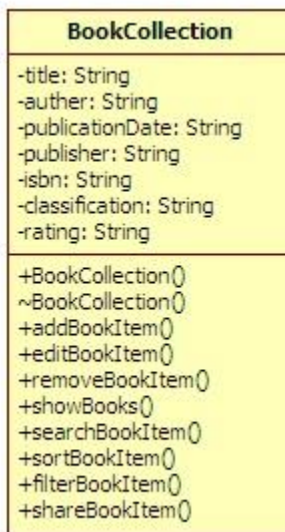
Identification:	MusicCollection
Purpose:	Purpose of this module is to provide pre-defined collection type for music collectors.
Attributes:	“singer”, ”song”, ”genre”, ”album”, ”year”, and “rating” are needed attributes for MusicCollection module.
Relationships:	This module inherited from Generic Collection module

Diagram:**3.3.2.3 BookCollection Module****Background Information:**

This module will be used for create pre-defined collection type for book collectors for the Dispososofia software system. It will contain the basic functions such that “addBookItem” that adds book item into collection, “editBookItem” that modifies book item in the collection, “removeBookItem” that deletes book item from collection, “showBooks” allows user to see the details of the selected book, like title, author name, publisher of the book, ”searchBookItem” that returns search result for the book item in the same type of pre-defined book collections, ”sortBookItem” that sorts the search result according to criteria of user in respect to fields of pre-defined book collection, ”filterBookItem” that filters the search result according to criteria of user in respect to fields of pre-defined book collection and ”shareBookItem” that publishes book item to other users if item has public permission.

Description:

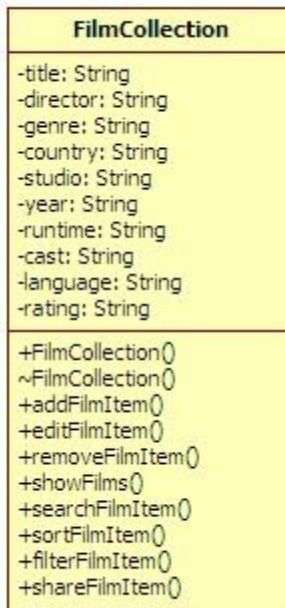
Identification:	BookCollection
Purpose:	Purpose of this module is to provide pre-defined collection type for book collectors.
Attributes:	“title”, ”author”, ”publicationDate”, ”publisher”, ”isbn”, ”classification” and “rating” are needed attributes for BookCollection module.
Relationships:	This module inherited from Generic Collection module

Diagram:**3.3.2.4 FilmCollection Module****Background Information:**

This module will be used for create pre-defined collection type for film collectors for the Dispososofia software system. It will contain the basic functions such that “addFilmItem” that adds film item into collection, “editFilmItem” that modifies film item in the collection, “removeFilmItem” that deletes film item from collection, ‘showFilms’ allows user to see the details of the selected film, like film name, actors, director of the film, ”searchFilmItem” that returns search result for the film item in the same type of user-defined collections, ”sortFilmItem” that sorts the search result according to criteria of user in respect to fields of pre-defined film collection, ”filterFilmItem” that filters the search result according to criteria of user in respect to fields of pre-defined film collection and ”shareFilmItem” that publishes film item to other users if item has public permission.

Description:

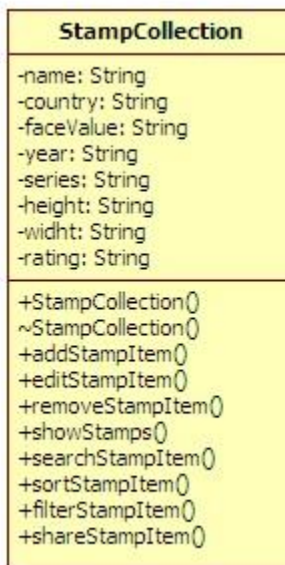
Identification:	FilmCollection
Purpose:	Purpose of this module is to provide pre-defined collection type for film collectors.
Attributes:	“title”, “actor”, ”director”, ”publicationDate”, ”genre”, ”country”, ”studio”, “year”, ”runtime”, ”cast”, ”language” and “rating” are needed attributes for FilmCollection module.
Relationships:	This module inherited from Generic Collection module

Diagram:**3.3.2.5 StampCollection Module****Background Information:**

This module will be used for create pre-defined collection type for stamp collectors for the Dispososofia software system. It will contain the basic functions such that “addStampItem” that adds stamp item into collection, “editStampItem” that modifies stamp item in the collection, “removeStampItem” that deletes stamp item from collection, ‘showStamps’ allows user to see the details of the selected stamp, like stamp name, country, year of the stamp, ”searchStampItem” that returns search result for the stamp item in the same type of user-defined collections, ”sortStampItem” that sorts the search result according to criteria of user in respect to fields of pre-defined stamp collection, ”filterStampItem” that filters the search result according to criteria of user in respect to fields of pre-defined stamp collection and ”shareStampItem” that publishes stamp item to other users if item has public permission.

Description:

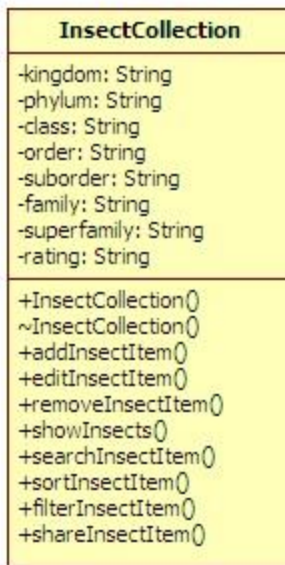
Identification:	StampCollection
Purpose:	Purpose of this module is to provide pre-defined collection type for stamp collectors.
Attributes:	“name”, ”country”, ”faceValue”, ”year”, ”series”, ”height”, “width” and “rating” are needed attributes for StampCollection module.
Relationships:	This module inherited from Generic Collection module

Diagram:*3.3.2.6 InsectCollection Module***Background Information:**

This module will be used to create a predefined insect collection for the user from the items of collections defined by organizations. It has some basic functions : ‘InsectCollection()’ creates an empty Insect collection for the user, ‘addInsectItem’ allows user to add new items into the collection; ‘editInsectItem’ and ‘removeInsectItem’ allow user to modify and remove items which are already in the collection, respectively; ‘showInsects’ allows user to see the details of the selected item; ‘searchInsectItem’, ‘sortInsectItem’ and ‘filterInsectItem’ allow searching, filtering with respect to some aspects, and sorting the fields of the desired insect from the collection list, respectively; and ‘shareInsectItem’ that publishes insect item to other users if item has public permission.

Description:

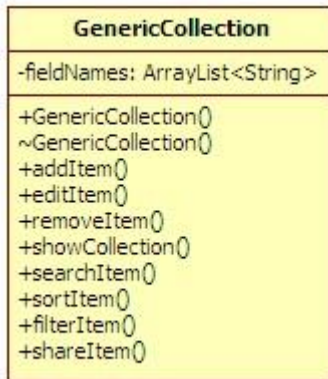
Identification:	InsectCollection
Purpose:	Purpose of this module is to provide pre-defined collection type for insect collectors.
Attributes:	“kingdom”, ”phylum”, ”class”, ”order”, ”suborder”, ”superfamily”, “family” and “rating” are needed attributes for InsectCollection module.
Relationships:	This module inherited from Generic Collection module

Diagram:**3.3.2.7 GenericCollection Module****Background Information:**

This module will be used for create user-defined collection type for the Dispososofia software system. It will contain the basic functions such that “addItem” that adds item into collection, “editItem” that modifies item in the collection, “removeItem” that deletes item from collection, ‘showItems’ allows user to see the details of the selected item, ”searchItem” that returns search result for the item in the same type of user-defined collections, ”sortItem” that sorts the search result according to criteria of user in respect to fields of user-defined collection, ”filterItem” that filters the search result according to criteria of user in respect to fields of user-defined collection and ”shareItem” that publishes item to other users if item has public permission.

Description:

Identification:	GenericCollection
Purpose:	Purpose of this module is to provide user-defined collection type in case of pre-defined collection types do not satisfy the needs of collectors.
Attributes:	“fieldNames” is needed attributes for GenericCollection module.
Relationships:	This module directly associated with User module

Diagram:**3.4 TRACEABILITY MATRIX**

In this sub section there is a match done between the designed modules and the use cases in the requirements report in order to keep avoid implementing additional functionality that the users did not want. Also all the functionalities that are specified in the software requirements specifications should be guaranteed to be implemented in the design. To do this we used a traceability matrix.

Use Cases (UCs)	User Module	Music Collection Module	Book Collection Module	Film Collection Module	Stamp Collection Module	Insect Collection Module	Generic Collection Module
UC1		X	X	X	X	X	X
UC2		X	X	X	X	X	X
UC3		X	X	X	X	X	X
UC4		X	X	X	X	X	X
UC5	X						
UC6	X						
UC7	X						
UC8	X						
UC9	X						
UC10	X						
UC11	X						
UC12	X						
UC13	X						
UC14	X						
UC15	X						
UC16	X						
UC17	X						

3.5 DEPENDENCY VIEWPOINT

3.5.1 DESIGN CONCERNS

The dependency viewpoint specifies the relationships of interconnection and access among entities. These relationships include shared information, order of execution, or parameterization of interfaces.

3.5.2 DESIGN ELEMENT

3.5.2.1 UserCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
MusicCollection	Association	Subsystem
BookCollection	Association	Subsystem
FilmCollection	Association	Subsystem
StampCollection	Association	Subsystem
InsectCollection	Association	Subsystem
GenericCollection	Association	Subsystem

3.5.2.2 MusicCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
UserCollection	Association	Subsystem
GenericCollection	Generalization	Subsystem

3.5.2.3 BookCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
UserCollection	Association	Subsystem
GenericCollection	Generalization	Subsystem

3.5.2.4 FilmCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
UserCollection	Association	Subsystem
GenericCollection	Generalization	Subsystem

3.5.2.5 StampCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
UserCollection	Association	Subsystem
GenericCollection	Generalization	Subsystem

3.5.2.6 InsectCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
UserCollection	Association	Subsystem
GenericCollection	Generalization	Subsystem

3.5.2.7 GenericCollection Module Dependency Description

DEPENDENCIES	TYPE	DESCRIPTION
UserCollection	Association	Subsystem
MusicCollection	Generalization	Subsystem
BookCollection	Generalization	Subsystem
FilmCollection	Generalization	Subsystem
StampCollection	Generalization	Subsystem
InsectCollection	Generalization	Subsystem

3.5.3 EXAMPLE LANGUAGE

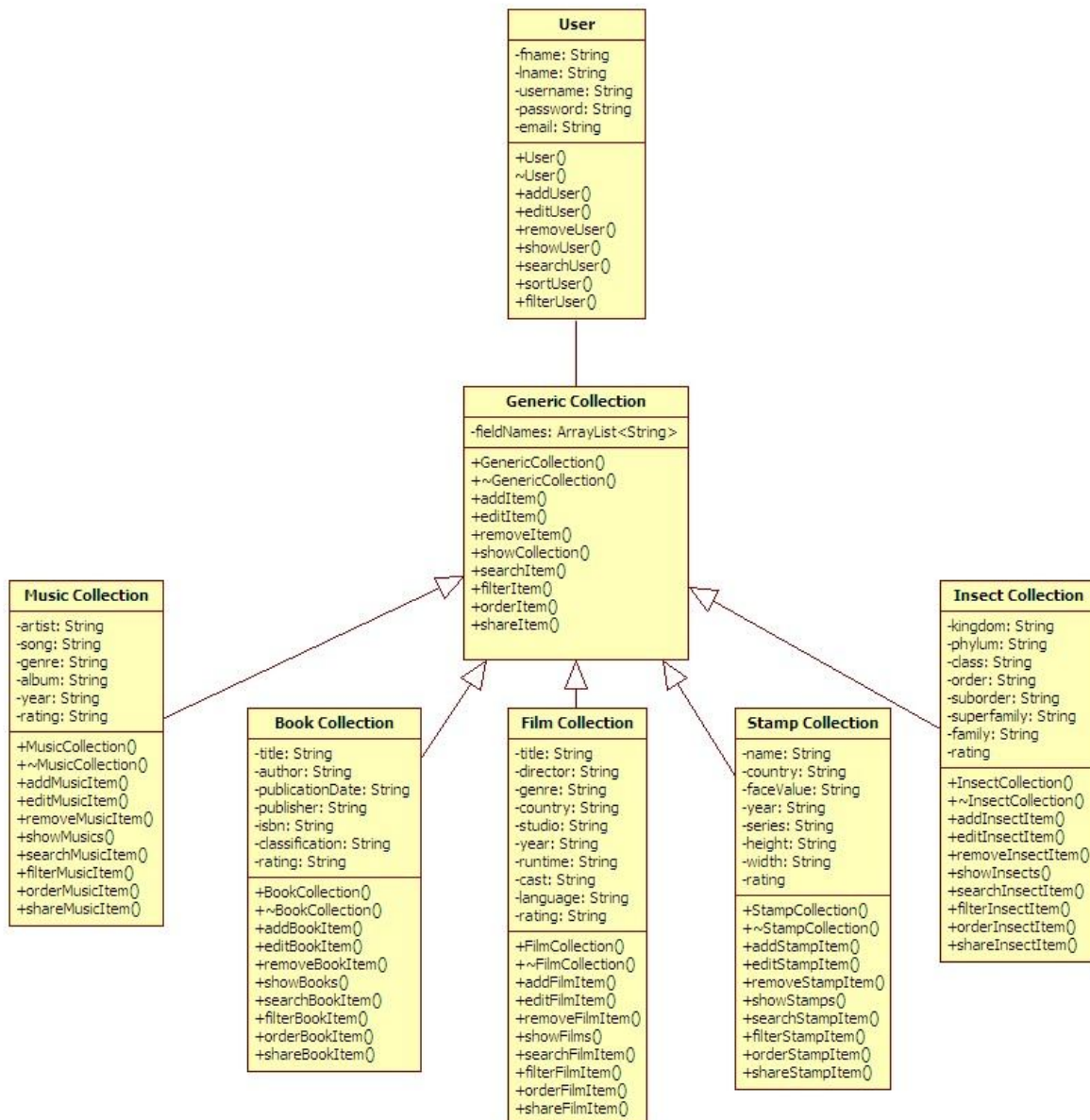


Figure 2 – Class Diagram of Java Activities

3.6 USER INTERFACE VIEWPOINT

The Interface viewpoint provides information designers, programmers, and testers the means to know how to correctly use the services provided by a design subject.

3.6.1 DESIGN CONCERNS

An interface view description serves as a binding contract among designers, programmers, customers, and testers. The interface description is used by technical writers to produce customer documentation or may be used directly by customers.

3.6.2 DESIGN ELEMENTS

3.6.2.1 *UserCollection Module Interface Description*

Name	Type	Accessibility	Type Signature
fname	attribute	private	String
lname	attribute	private	String
username	attribute	private	String
password	attribute	private	String
email	attribute	private	String
User	method	public	void(String)
~User	method	public	void(String)
addUser	method	public	void(String)
editUser	method	public	void(String)
showUser	method	public	void(String)
removeUser	method	public	void(String)
searchUser	method	public	void(String)
sortUser	method	public	void(String)
filterUser	method	public	void(String)

3.6.2.2 MusicCollection Module Interface Description

Name	Type	Accessibility	Type Signature
artist	attribute	private	String
song	attribute	private	String
genre	attribute	private	String
album	attribute	private	String
year	attribute	private	String
rating	attribute	private	String
MusicCollection	method	public	void(String)
~MusicCollection	method	public	void(String)
addMusicItem	method	public	void(String)
editMusicItem	method	public	void(String)
removeMusicItem	method	public	void(String)
showMusicItem	method	public	void(String)
searchMusicItem	method	public	void(String)
filterMusicItem	method	public	void(String)
sortMusicItem	method	public	void(String)
shareMusicItem	method	public	void(String)
giveRateMusicItem	method	public	void(String)

3.6.2.3 BookCollection Module Interface Description

Name	Type	Accessibility	Type Signature
title	attribute	private	String
auther	attribute	private	String
publicationDate	attribute	private	String
publisher	attribute	private	String
isbn	attribute	private	String
classification	attribute	private	String
rating	attribute	private	String
BookCollection	method	public	void(String)
~BookCollection	method	public	void(String)
addBookItem	method	public	void(String)
editBookItem	method	public	void(String)
removeBookItem	method	public	void(String)
showBookItem	method	public	void(String)
searchBookItem	method	public	void(String)
filterBookItem	method	public	void(String)
sortBookItem	method	public	void(String)
shareBookItem	method	public	void(String)
giveRateBookItem	method	public	void(String)

3.6.2.4 *FilmCollection Module Interface Description*

Name	Type	Accessibility	Type Signature
title	attribute	private	String
director	attribute	private	String
genre	attribute	private	String
country	attribute	private	String
studio	attribute	private	String
year	attribute	private	String
runtime	attribute	private	String
cast	attribute	private	String
language	attribute	private	String
rating	attribute	private	String
FilmCollection	method	public	void(String)
~FilmCollection	method	public	void(String)
addFilmItem	method	public	void(String)
editFilmItem	method	public	void(String)
removeFilmItem	method	public	void(String)
showFilmItem	method	public	void(String)
searchFilmItem	method	public	void(String)
filterFilmItem	method	public	void(String)
sortFilmItem	method	public	void(String)
shareFilmItem	method	public	void(String)
giveRateFilmItem	method	public	void(String)

3.6.2.5 StampCollection Module Interface Description

Name	Type	Accessibility	Type Signature
name	attribute	private	String
country	attribute	private	String
faceValue	attribute	private	String
year	attribute	private	String
series	attribute	private	String
height	attribute	private	String
width	attribute	private	String
rating	attribute	private	String
StampCollection	method	public	void(String)
~StampCollection	method	public	void(String)
addStampItem	method	public	void(String)
editStampItem	method	public	void(String)
removeStampItem	method	public	void(String)
showStampItem	method	public	void(String)
searchStampItem	method	public	void(String)
filterStampItem	method	public	void(String)
sortStampItem	method	public	void(String)
shareStampItem	method	public	void(String)
giveRateStampItem	method	public	void(String)

3.6.2.6 *InsectCollection Module Interface Description*

Name	Type	Accessibility	Type Signature
kingdom	attribute	private	String
phylum	attribute	private	String
class	attribute	private	String
order	attribute	private	String
suborder	attribute	private	String
family	attribute	private	String
superfamily	attribute	private	String
rating	attribute	private	String
InsectCollection	method	public	void(String)
~InsectCollection	method	public	void(String)
addInsectItem	method	public	void(String)
editInsectItem	method	public	void(String)
removeInsectItem	method	public	void(String)
showInsectItem	method	public	void(String)
searchInsectItem	method	public	void(String)
filterInsectItem	method	public	void(String)
sortInsectItem	method	public	void(String)
shareInsectItem	method	public	void(String)
giveRateInsectItem	method	public	void(String)

3.6.2.7 GenericCollection Module Interface Description

Name	Type	Accessibility	Type Signature
fieldNames	attribute	private	ArrayList<String>
GenericCollection	method	public	void(String)
~GenericCollection	method	public	void(String)
addItem	method	public	void(String)
editItem	method	public	void(String)
removeItem	method	public	void(String)
showCollection	method	public	void(String)
searchItem	method	public	void(String)
sortItem	method	public	void(String)
filterItem	method	public	void(String)
shareItem	method	public	void(String)
giveRateItem	method	public	void(String)

3.6.3 EXAMPLE LANGUAGE

After the installation of the software, on first sight system asks for a user account. In order to use the software, a user account on database is essential, for this reason a registration form is directed towards client. After filling all fields of form, database query to insert a new user is sent by a related PHP File

Registration

Dispososofia

First Name
Allen

Last Name
McMillan

User Name
User Name

Password
Password

Confirm Password
Password

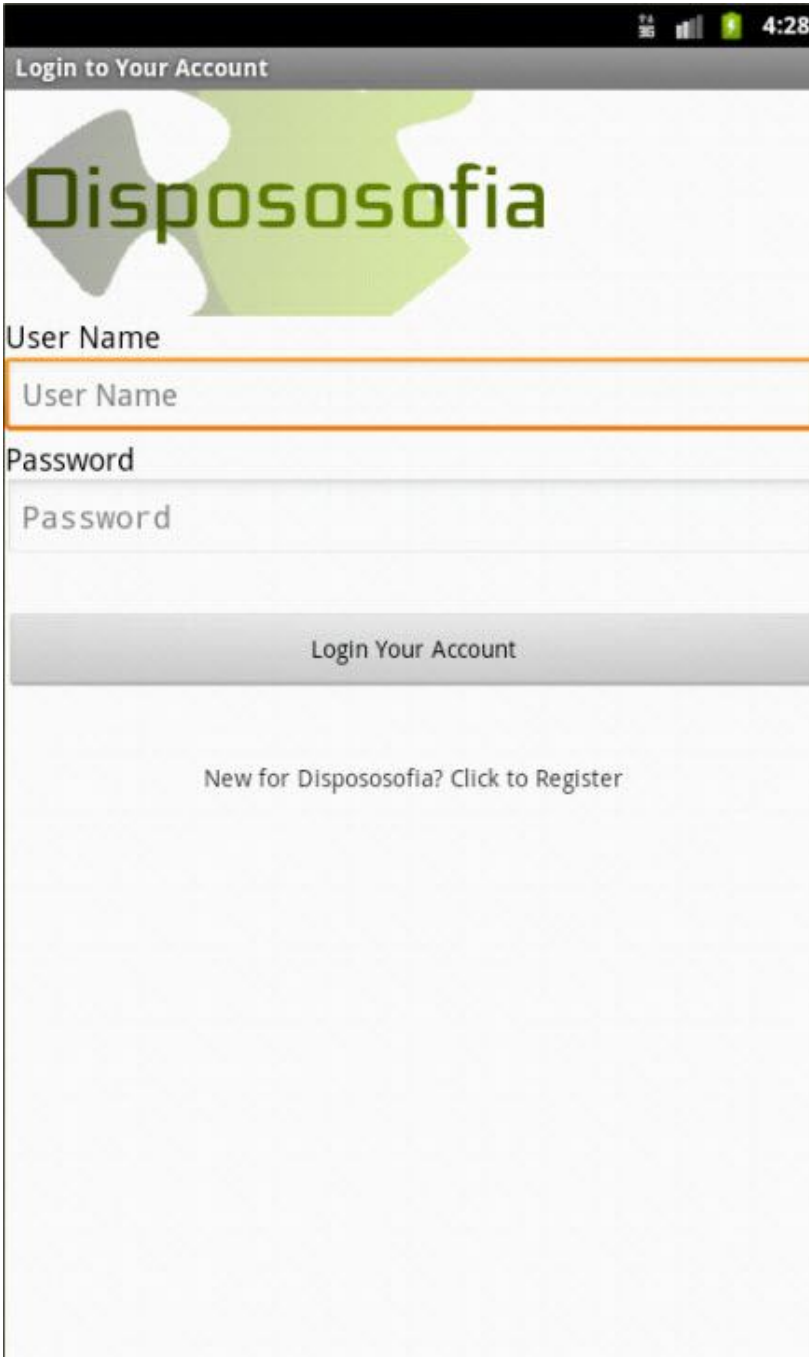
Email Address
Email Address

Register Account

Already Registered! Click to Login

User Interface 1: Registration screen welcomes new user

User logs in his/her account for subsequent accesses. Just username and password are needed to connect his/her database. For a new comer, a direct link is provided to see the registration form.



Login to Your Account

Dispososofia

User Name

User Name

Password

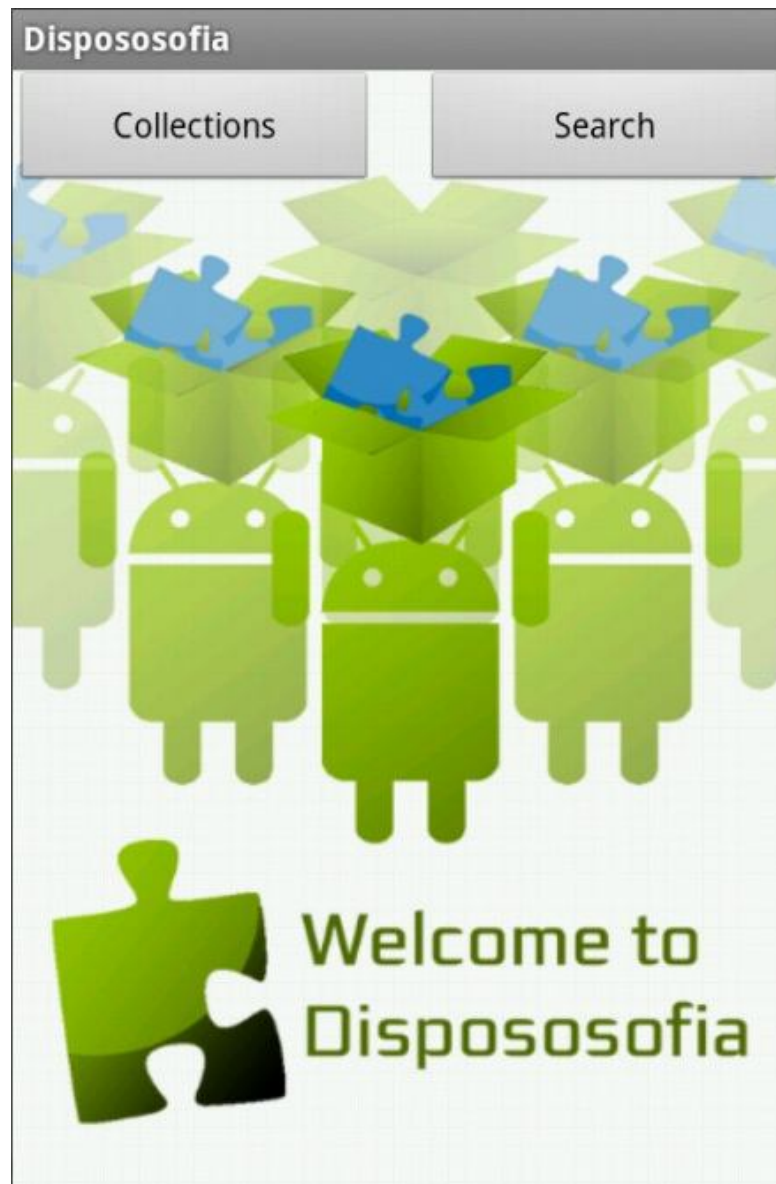
Password

Login Your Account

New for Dispososofia? Click to Register

User Interface 2: Log in screen shows form for registered user to log in

If valid user connects database, main screen listens client directives up for monitor his/her collections and search on users, collections, and items.



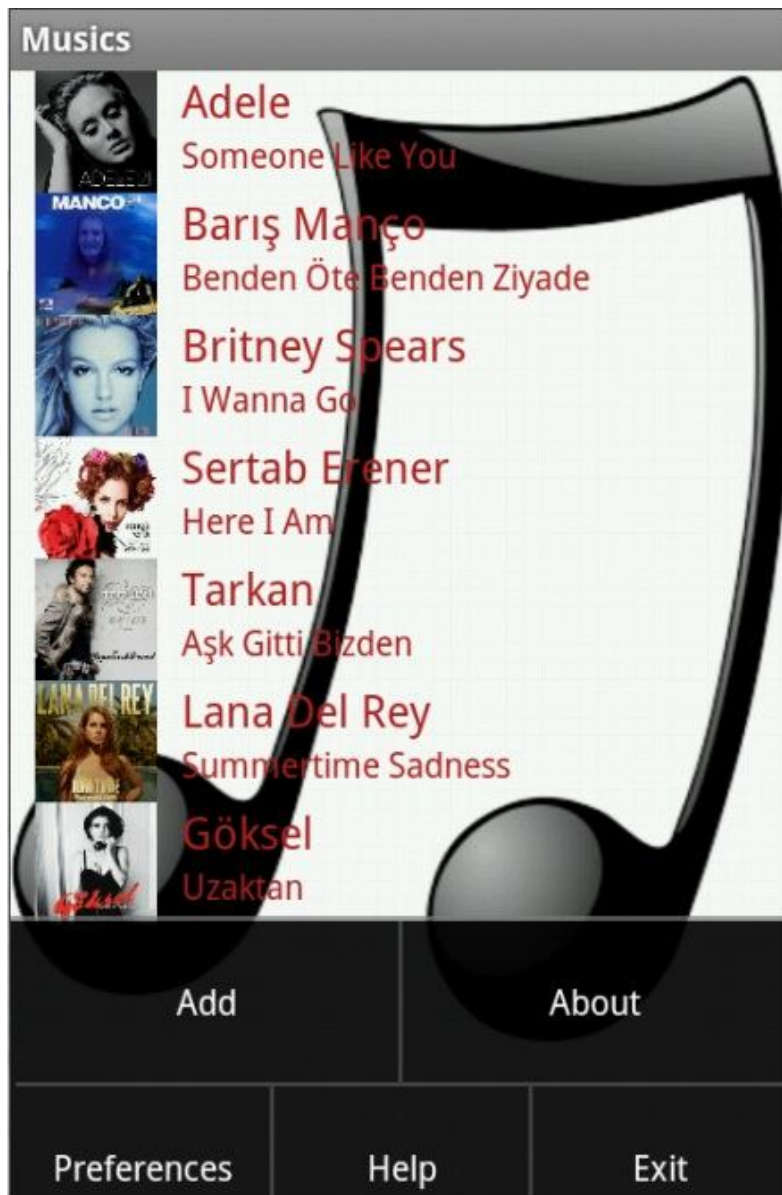
User Interface 3: Main screen with view collection and search options

When user comes collections screen by clicking on collection button from main screen, s/he can observe all crated collections. There can be both user defined collections and predefined basic collections, such as book, film, music, stamp, insect. On this list view user can see options as delete and edit by long tapping on a collection. Adding new collection can be done through menu button.



User Interface 4: Collection screen of a user (when the collection 'Music' is tapped)

Items belong to a collection can be listed by tapping on the collection. Each collection has different fields and different configuration settings on this view. For an example music collection is listed with album cover, artist and song. Adding a new item can be done again on menu button.



User Interface 5: The list of Items in a Collection(When the menu button is tapped)

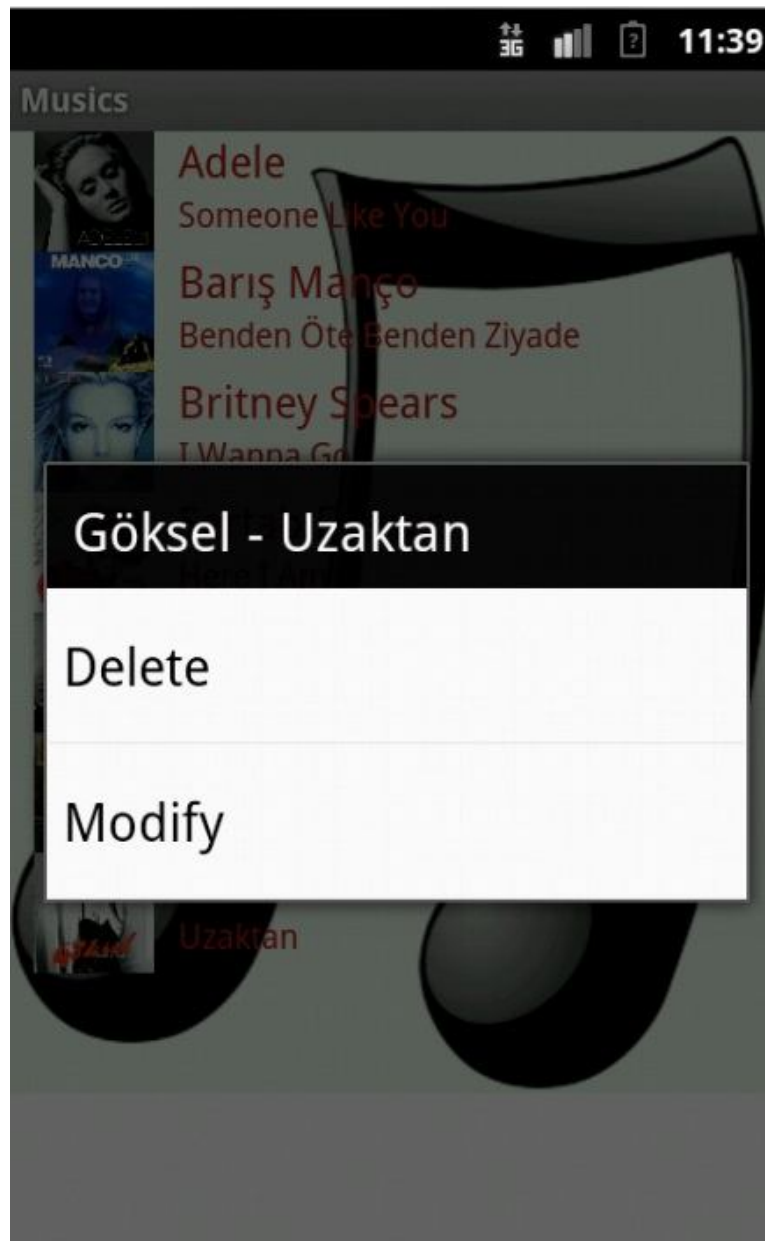
In the process of adding new item in a collection, after tapping the add button which can be seen on menu button a blank form consists of fields of the collection shown in first place. After user has sent the form on database, a new record is created in corresponding collection table.



The image shows a user interface for adding a new song. The title 'New Song' is at the top left. The background is a blue and white musical theme with notes and staves. The form consists of five text input fields, each with a label above it: 'Song Name', 'Singer', 'Album', 'Year', and 'Genre'. The 'Song Name' field is highlighted with an orange border. At the bottom center is a grey 'Add' button.

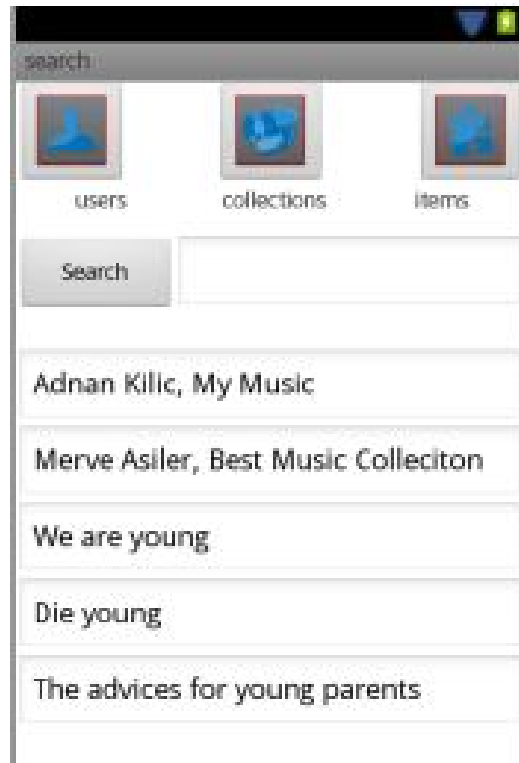
User Interface 6: Adding New Item To a Collection

When the user selects a collection and try to look at all its elements, s/he can tap long each item to delete or modify the content.



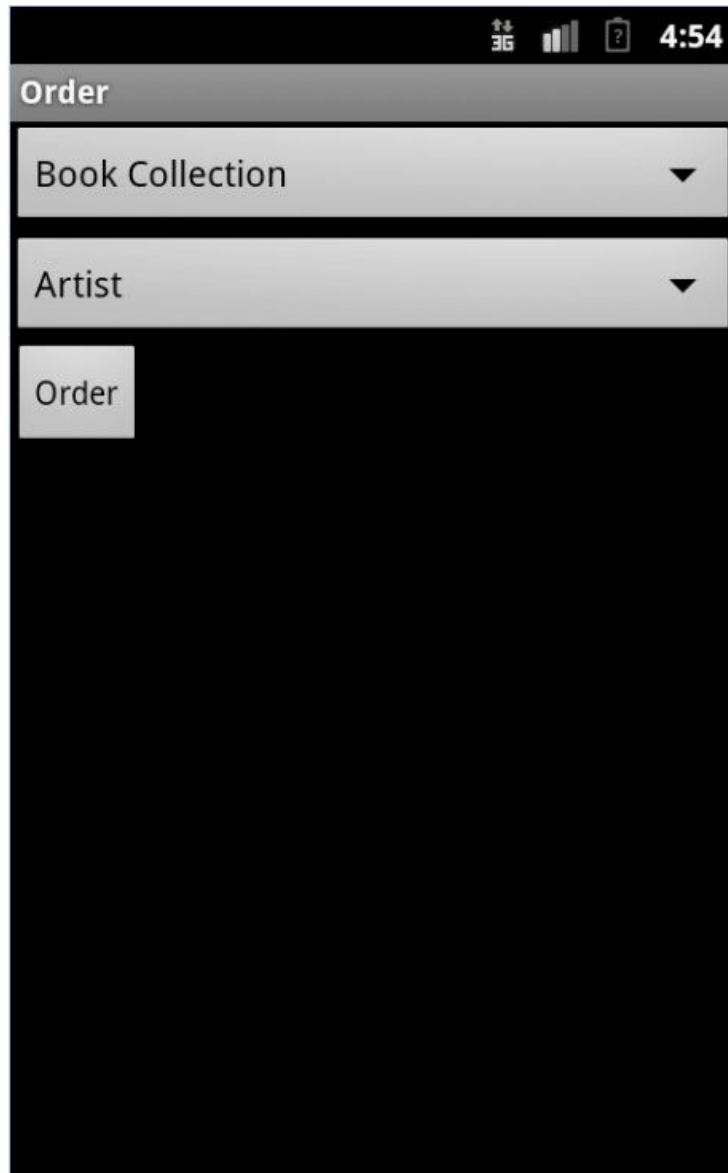
User Interface 7: Collection screen after long clicked by the user

Search option eases access on an item, a collection or a user. All related records in database are listed on screen. By filter, user can see the ones that satisfy certain conditions and by order, an organized view can be followed. Both filter and order options are nestled in menu button screen.



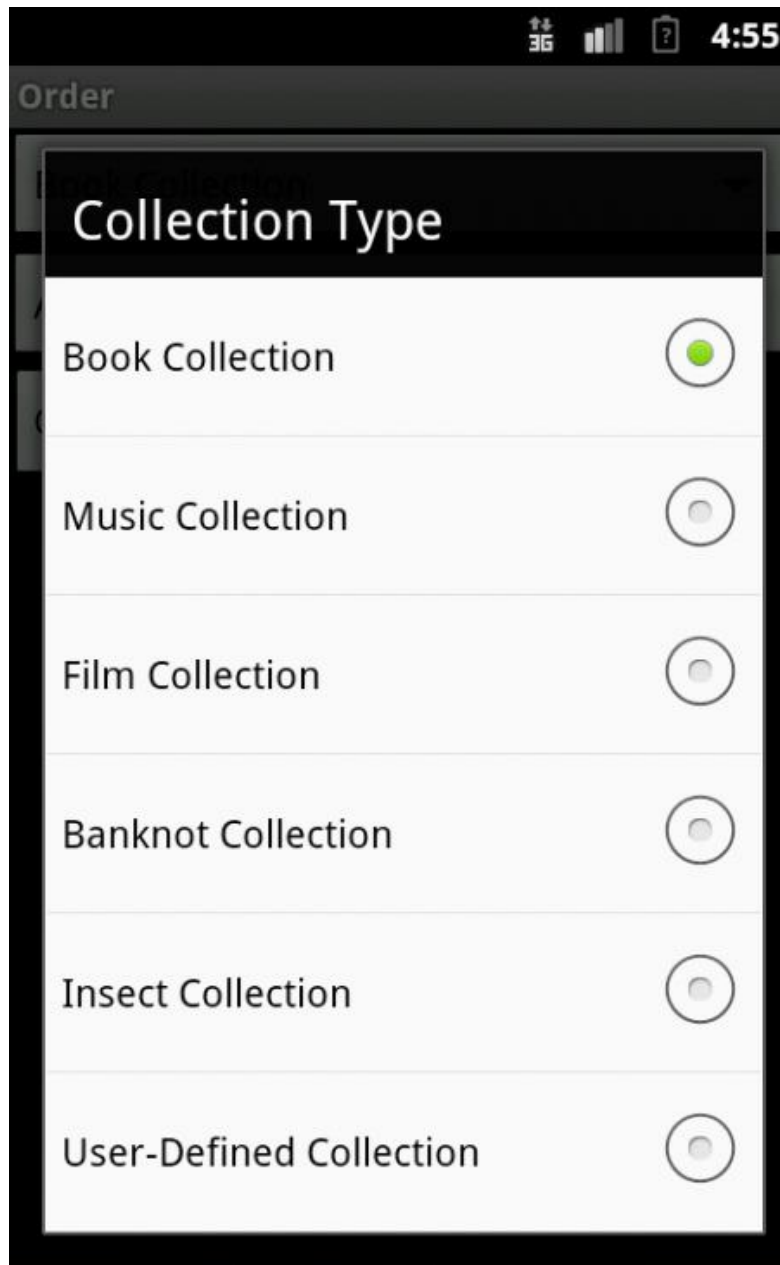
User Interface 8: Search screen serves user three options to search on: users, collections, items

Order Button presents the user an ordered list view of search results by providing the order with respect to wished field of that collection.



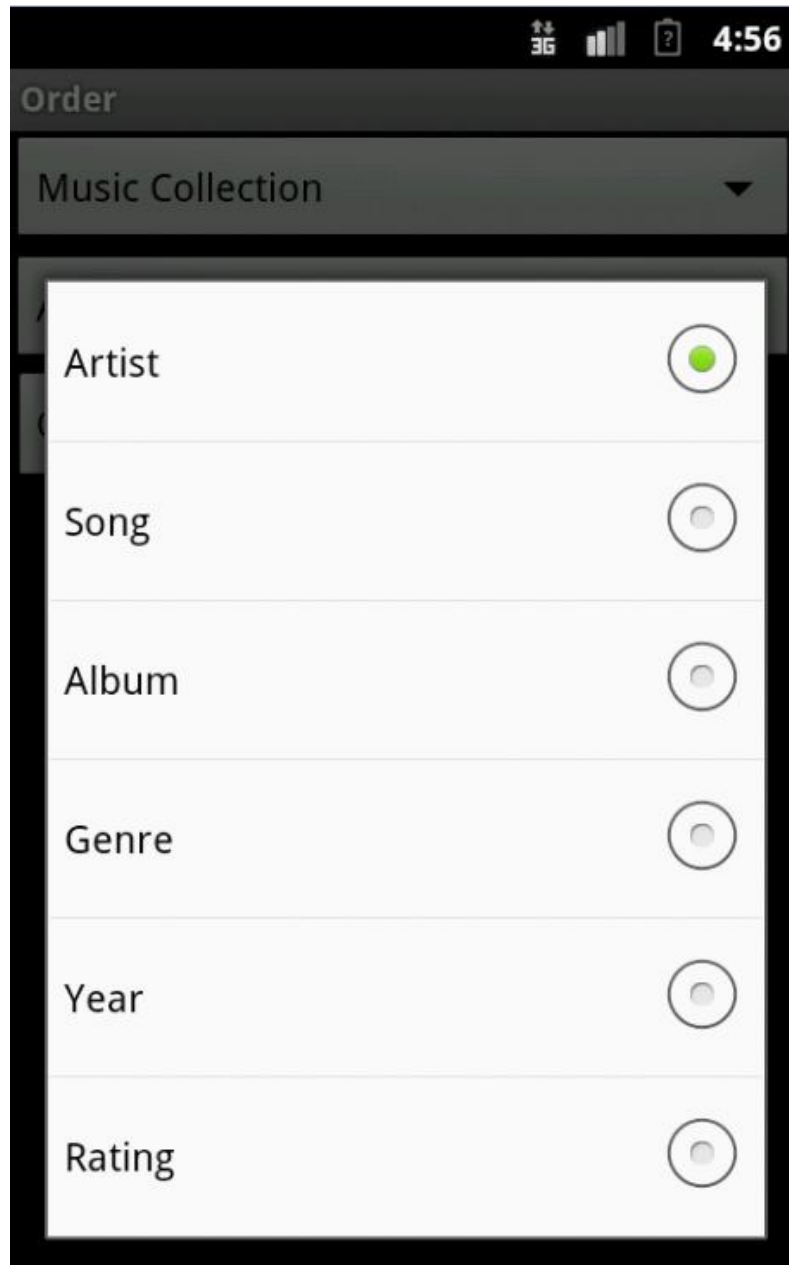
User Interface 8: Order screen asks user for a field according to which search results become ordered in a collection

To order the items of the wished collection, after tapping on the select menu of collection types, the existing pre-defined and user-defined collection types will be seen on the screen together with their radio buttons.



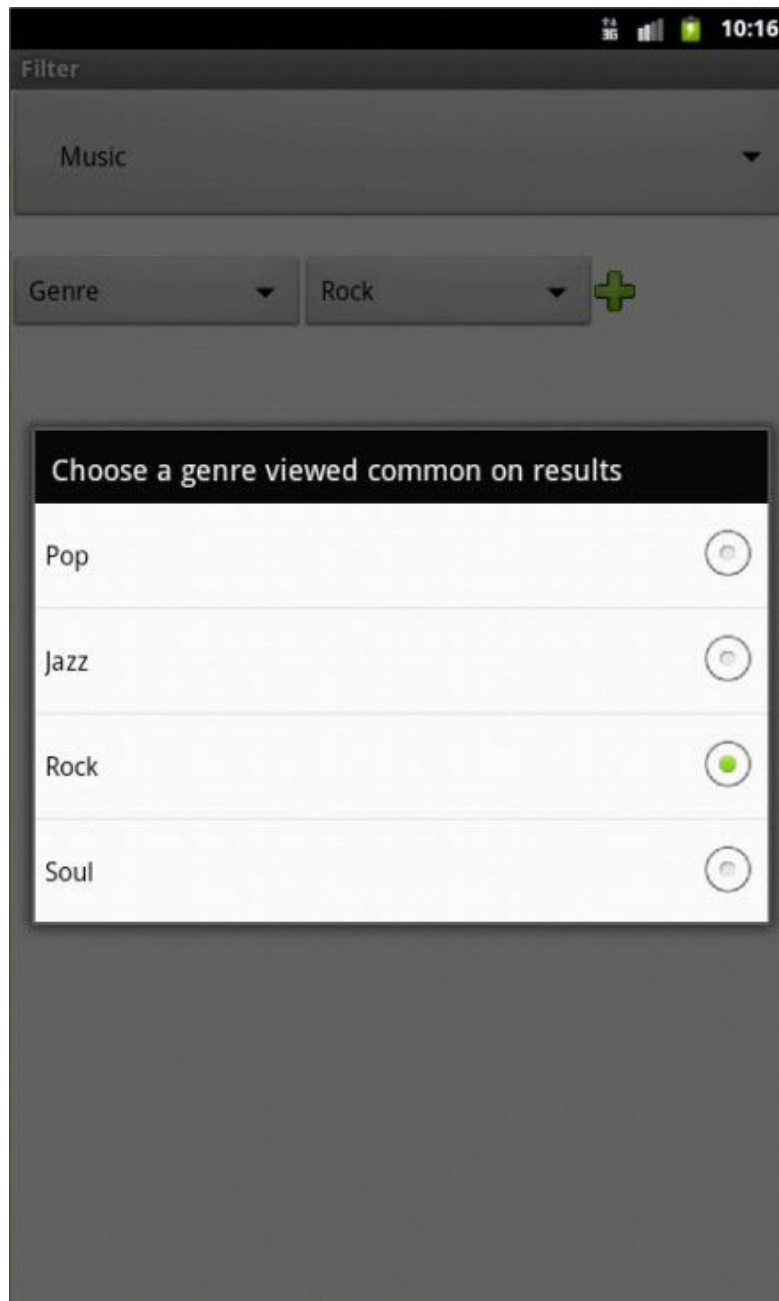
User Interface 9: List of Collections includes both predefined and user defined ones.

To order the items of the wished collection with respect to wished field of that collection, after tapping on the select menu of collection fields, the related fields of that collection will be seen on the screen together with their radio buttons.



User Interface 10: List of fields for corresponding collection

From the filter screen, the user will be able to be obtaining items which have a specific content by tapping that content. After entering the content information, the related items which include that content will be gathered from the database and shown on the screen.



User Interface 11: Filter screen purifies the search results

After tapping on the 'Submit' button, the items including the specified content will be shown together with its collection type and the part displaying how it passes inside the field.



User Interface 12: Filter can be done by different attributes of a collection

If user does not want to use one of the basic collection layouts (music, book, stamp, insect, film), a user defined collection type can be created on Dispososofia. User decides the attributes of the collection type and then just add click button at the bottom. Screen return search results with a filtered view.

The screenshot shows a mobile application interface titled "Add New Collection Layout". At the top, there is a status bar with signal strength, battery, and time (2:24). Below the title bar, there is a list of fields for a collection layout. The fields are: "Banknote Collection", "Currency Name", "Country", "Denomination", "Serial No", "Year", "Material", and "New Field Name". To the right of each field is a red "X" icon, indicating that the field is required or has a validation error. The "New Field Name" field has a green "+" icon, indicating it is optional or can be added. At the bottom of the screen, there is a button labeled "Add Layout".

User Interface 13: User can add own layout on Add New Collection Layout Screen

3.7 INTERACTION VIEWPOINT

3.7.1 DESIGN CONCERNS

For designers, this includes evaluating allocation of responsibilities in collaborations, especially when adapting and applying design patterns; discovery or description of interactions in terms of messages among affected objects in fulfilling required actions; and state transition logic and concurrency for reactive, interactive, distributed, real-time, and similar systems.

3.7.2 DESIGN ELEMENTS

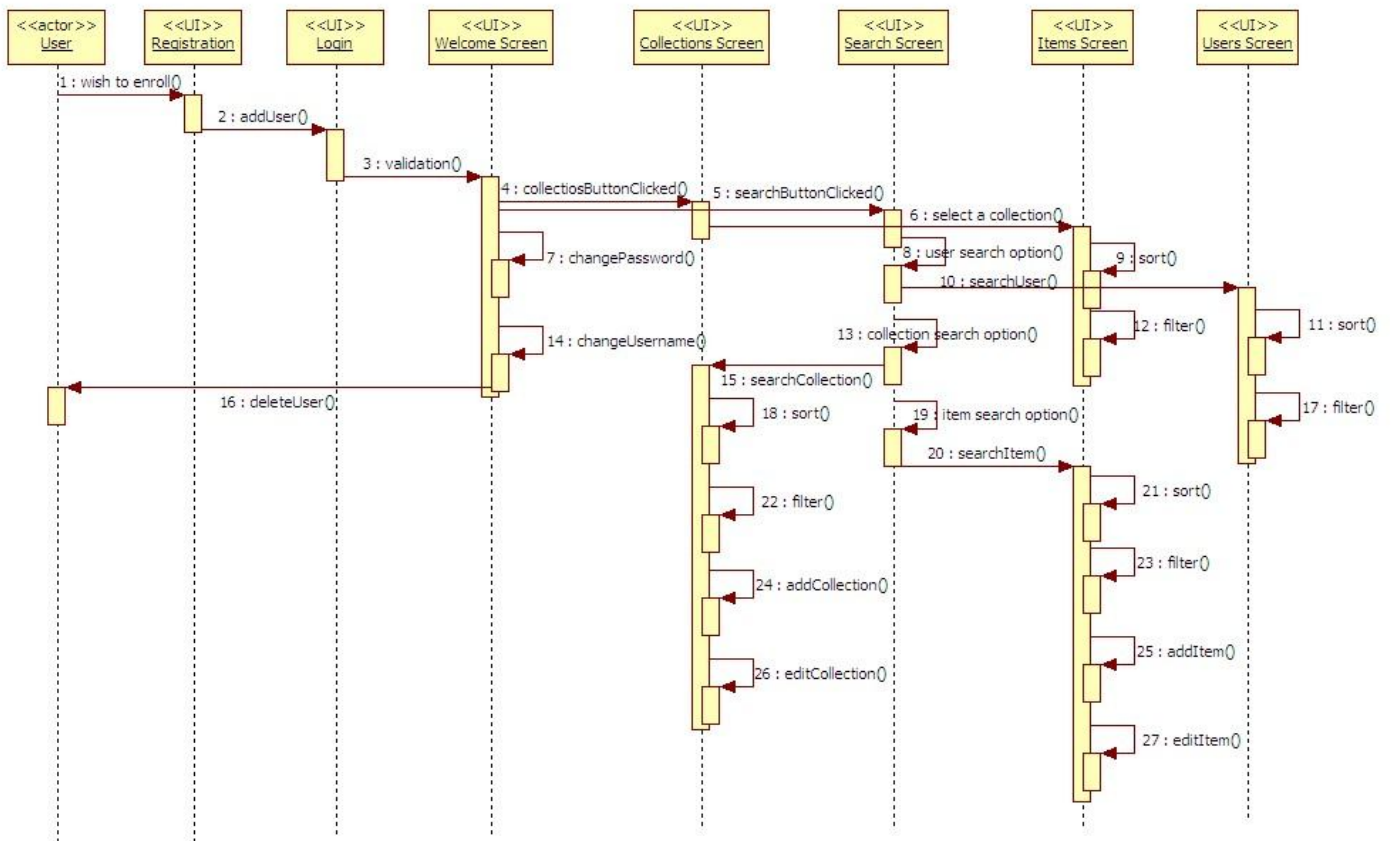


Figure 3 – Sequence Diagram

3.8 STATE DYNAMICS OVERVIEW

Reactive systems and systems whose internal behavior is of interest use this viewpoint.

3.8.1 DESIGN CONCERNS

System dynamics includes modes, states, transitions, and reactions to events.

3.8.2 DESIGN ELEMENTS

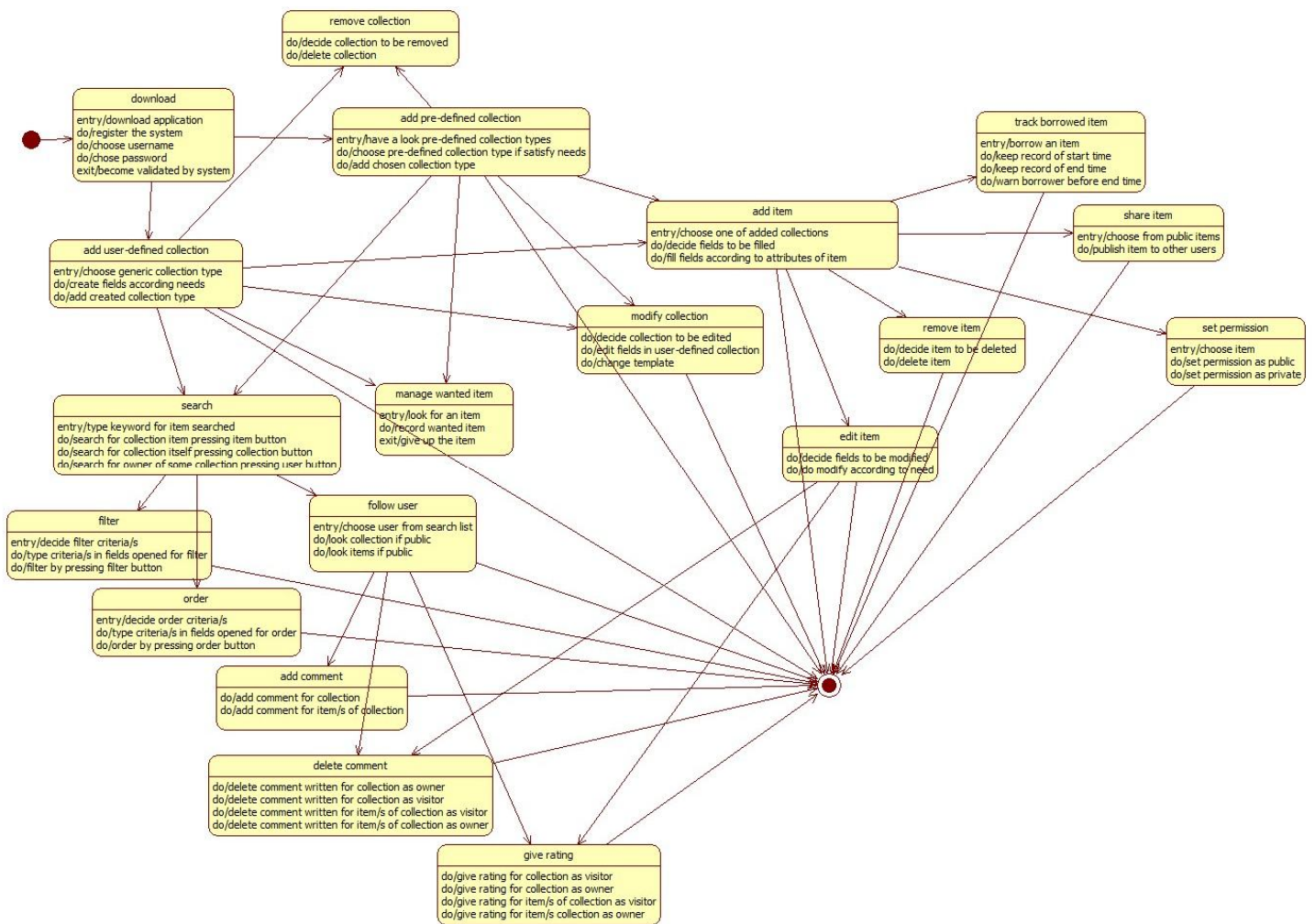


Figure 4 – State Diagram