Ben Gurion University

The program in Software Engineering

Collectomania

ARD Document

Authors: Yuri Gabaev

Maria Kertsburd

Olga Sibiryov

Amir Eluk

Revision: 1.0

Date: 30 December, 2011

Contents

1. Introduction

1.1. The Problem Domain

Nowadays it’s a very common approach that different commercial companies try to get customers’ attention and of course to have more customers and more users of their problems. Unfortunately there is a lack of new ideas or new instruments for this purpose.

So, here the collectomania enters. This project attends to give an answer for this purpose’ as described at the vision of the project.

1.2. Vision

Collectomania is a web application, which allows different users to collect cards in different albums, which are managed entirely by other high level users (so called managers).

Each manager can start an album, which follows some rules, determined by the manager at the creation of the album. These rules can be divided into different categories and allow different options on managing the album. After the album is created, simple users can assign themselves to the album and start collecting cards in this album. These cards can be obtained by getting pin-code, by exchanging cards or by playing on them with other users. Of course, there is a competition between users on who finishes the collecting of cards in the album first.

In general, this application can be very useful for large commercial companies’, that look for an opportunity to attract more customers and they can use this application for promotional purposes.

1.3. Stake Holders

Collectomania is a system designed for allowing different users to collect and share different items between them. These items can cover different aspects. It also allows other users to start new albums covering any selected subject.

The users: Collectomania enables web users to collect different items and share them between with each other. We will call for these pieces of information for name cards. Users can also exchange cards according to different rules. In general, this application is for everyone, who is the user of different companies’ products. The goal of each user is to complete the album first. To complete album mean to collect all cards of it and to paste them to the right place in album.

The customer: Commercial companies which wish to get to their customers, so they user this application in the promotional purpose. The customer is going to be the manager of his album. Each album defines a competition between users. Customer may set rule with number of winners. It means that first X users, who collected all album cards and paste them in album, take the award.

1.4. Software Context

There are 2 major parts in the application.

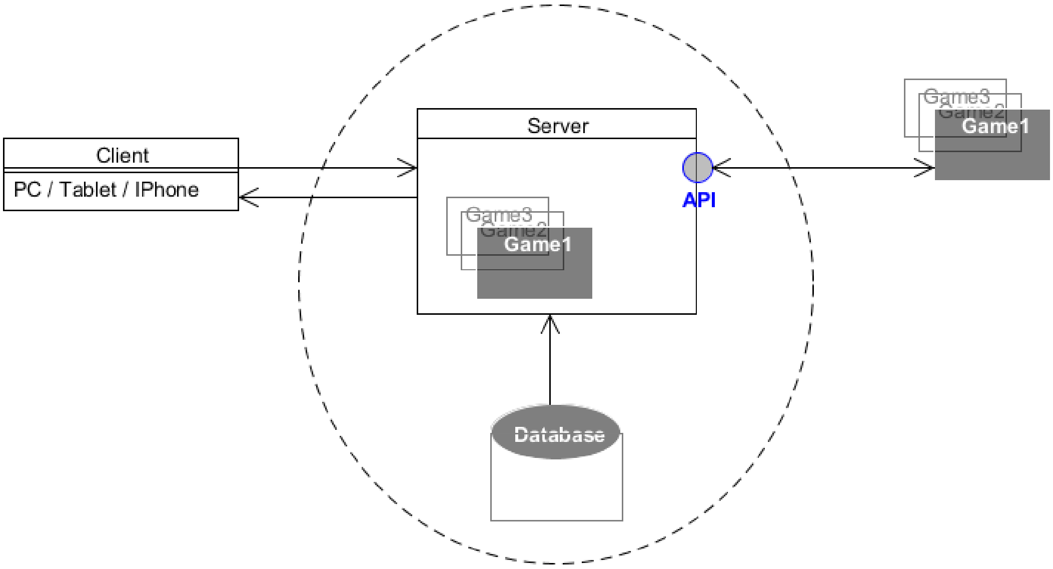
Client side – component which runs in on client's PC / tablet / Smartphone etc. In short every device that supports Flex technology of Adobe. Both simple user and manager interact with a system in a client side. All actions that are not changing the state of the system will be implemented here.

Server side – software component, that process all the events and queries.

Database - stores information about all users, managers, their albums and cards.

Internal mini-games – games for exchanging cards. These games are built in the system.

External mini-games – games for exchanging cards. These games communicate with server through API.



1.5. System Interfaces

First of all, Collectomania is a web application, thus it should be able to cooperate with different web interfaces. Furthermore, it is basically a user-driven application, therefore it has to be able to take care of different user-driven events.

1.5.1 Hardware Interfaces

As it has been said before the basic idea behind the collectomania is to be used in commercial and promotional purposes of different companies, so these companies may like to allow users to gain pin-codes in different ways. So, if we would like to expand our application to cover these ways in the future use, we would want to connect our system with other possible hardware interfaces.

1.5.2 Software Interfaces

Collectomania will use HTTP services in order to perform a communication through the web application between different users.

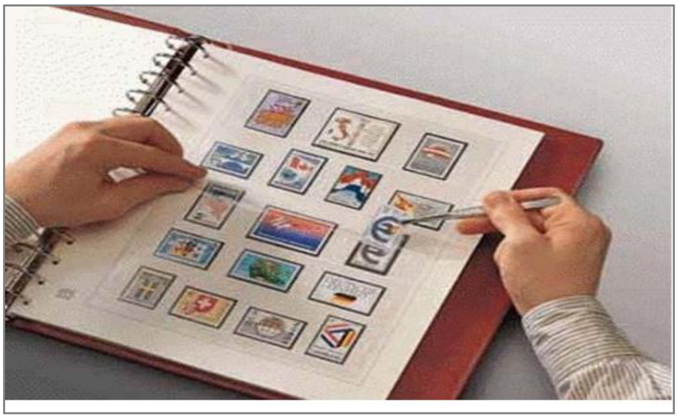
This application will also provide an API which allows adding additional mini-games to the application.

1.5.3 Events

There are going to be two different kinds of events: internal and external.

Internal events come from the timer, distributes to users sets of cards once in period. Both length of period and set of cards are specified by album manager.

External events are from the kind of entering a new pin-code by the user, so there is a need to answer to this event and it can also affect the state of users’ album.



2. Functional Requirements

2.1. General and administration requirements: (system)

#	Requirement	Description	Comments	Priority
1	**Support multiple albums**	The user will be able to obtain several albums and to watch all of them		1
2	**Support games**	There will be an option to add games for exchange cards. For more info see mini-game appendix.	In our implementation we will implement Rock-Paper-Scissors game.	2
3	**Enter pin code/** **Receive new cards**	User will insert a pin code, which she received by sending SMS.	The game manger decides how new codes can be obtained, and how codes are translated to actual cards	2
4	**Allow admin to Add/Remove/Update Player**	The information about players is stored in the database: - username - password - Information about his albums and pictures		3
5	**Decision about album winner**	* The system must be able to detect the winners of an album and notify to other owners of this album. * The creator of an album should be able to determine the number of possible winners. * The creator of an album should be able to define the winning criteria: - Completing the whole album - Completing a page in the album	If number of winners reached the maximum, then all other users are notified that the competition on the album ends.	2
6	**Saving current state**	The system will automatically save the state of all users' albums and cards after each transaction.		2
7	**Card multimedia**	Card can be linked to accompanying information such as text, and multimedia link		1
8	**Code types for receiving cards**	For more info see appendix code-types	Each code type will have different meaning	3
9	**Site Help**	The system should have help screens explaining the concept and goal. The least should cover what the site is about, how to join, how to play ant etc. The help should be accessible from different pages on the web site.		2

2.2. User requirements:

#	Requirement	Description	Comments	Priority
1	**Registration**	The user will be able to register himself in the system: - Creating new unique username. - Creating a new password, as secure as possible.	The system is going to enforce some basic rules for creating a password.	2
2	**Login**	The user will be able to login to the system. The system should also support auto-login by remembering the user credentials. The system should present a checkbox at login time with an automatic login option. When logging-in, the system must present a welcome screen, a news banner, and a list of available albums.		1
	Log-out	A user should be able to logout of the system. This will cancel the auto-login until the next login.		1
3	**Un-registration**	The user will have an option to un-register, in which case all his albums and cards will be terminated and his name will be removed from hall of fame		3
4	**Modify profile details**	User will be able to change his profile details		2-3
5	**Watch album**	The user will be able to see his albums and get different information about the album: - How many pictures are in the album? - List of locked cards. - List of free cards. - How many cards she still has to paste in order to complete the album. - Is user's album still participating in competition?	The list of pasted pictures and not pasted pictures are actually presented by the colored and shaded slots in the album.	1
6	**Watch card**	The user will be able to inspect any cards in the album and its characteristics: - Value of the card - Quantity - To which album it belongs - ID - Additional information		1
7	**Receive code**	User will have an option to receive codes for new cards by several ways	User can get some new card/cards by entering the secret code. Codes can be obtained by any means defined by the game creator\manager. (Facebook "Like", Google "+1", SMS, printed on Coca Cola screws, etc…)	3

8	**Receive card by winning game**	The winner of the game takes his opponents' pictures and keeps his own.	The user will be able to look for specific opponents or cards	1
9	**Receive card(s) by exchange chosen of set of cards with another player**	The user will be able to choose set of cards and exchange it with another player.	The user will be able to look for specific opponents or cards	1
10	**Start new album**	The system must allow adding a new album to a user's account, based on the user's request.	The album must be defined first by a manager	2
11	**Wish & Trade list**	User has a list of cards he wants to receive and a list of cards he is willing to trade or to play on.	The system will match appropriate opponents to play with	2
12	**Notifications**	The user will be able to read notifications posted by the manager on the main screen of the application.	Important notifications like a user completing an album & competition timeout will pop-up & disappear after a few seconds on all screens See Manager requirements #3	1-2
13	**Invite a friend**	A user can send an email to a friend with a link to a specific album (this can be done from a page or from an album). The receiving user will get a link where he can join the site. If he is already a member and does not have this album he will be asked whether to add this album.		2

2.3. Manager requirements:

#	Requirement	Description	Comments	Priority
1	**Login**	The manager will be able to login to the system.		2
2	**Create new album**	The manager will be able to create a new album and determine a set of rules for this album.	The determined rules wouldn't be changed during the life of the album.	2
3	**Watch album**	Manager will have an option to review his albums		3
4	**Watch card**	Manager will have an option to review the cards of his albums		3
5	**Publish notifications**	The manager will be able to post personal and/or public notifications about different topics related to the albums and pictures.		2
6	**Give codes for cards to player/s**	Manager can send codes to a group of players. The manager decides the type of code and can query the group of players to receive the bonus		2-3
7	**Determine/change the value of the picture**	The manager will be able to determine/change the value of the picture according to the given formula.		3
8	**Turn inactive cards to active (according to the set of rules)**	The admin will be able to activate\deactivate cards.		2
9	**Reset Pin codes**	A manager can reset the pin codes in case of a request by third party.		3

2.4. Rules requirements:

#	Requirement	Description	Comments	Priority
1	**Rules**	Each album will have a set of rules affecting its gameplay	See Appendix "Rules" for more information	1
2	**Set Rules**	Rules are created at album creation, or at game creation		1
3	**Modify Rules**	Some rules can be altered dynamically at any time		2
4	**Delete rules**	Some rules will be removed		3
5	**Game rules**	The system will have game level rules		1
6	**Album rules**	The system will have album level rules		1
7	**Card rules**	The system will have card level rules		2

2.5. Game requirements:

#	Requirement	Description	Comments	Priority
1	**Play on specific card**	The user will be able to select specific card to play on.	The system will find other online users that want to play on that card	1
2	**Play with specific user**	The user will be able to select specific user to play against		1
3	**Play using wish list**	The user will be able to publish his wish and trade lists and receive relevant opponents list	The system will find other online users that want to play on the cards	1

3. Non-Functional Requirements

3.1. Performance Constraints

3.1.1. Scalability

a. The system will support up to 100 registered users. All users could be on-line at the same time.
b. For future scalability all system resources will be built to hold 10 to 100 times this number when using data structures and run-time prediction.

3.1.2. Availability

a. The system should be available for users at least 95% of the time. While scheduled down-time will be known from advance and reduced to minimum.
b. The system is supported while on-line, there is no need to put the site down in order to update every-day game rules etc…

3.1.3. Portability

a. Users could log on with their username from anywhere in the world.
b. The web-site will be platform independent and should work on IE9 Firefox and chrome
c. Additional software installation may be required upon first use (Flash & Flex browser plug in).

3.1.4. Security

a. User identification and authentication is achieved by using a unique username and a secret password for every user.
b. A hash value of the password is saved on the server.
c. HTTP protocol will be used for unencrypted transmission of data.
d. Pin-codes should be virtually impossible to fake.

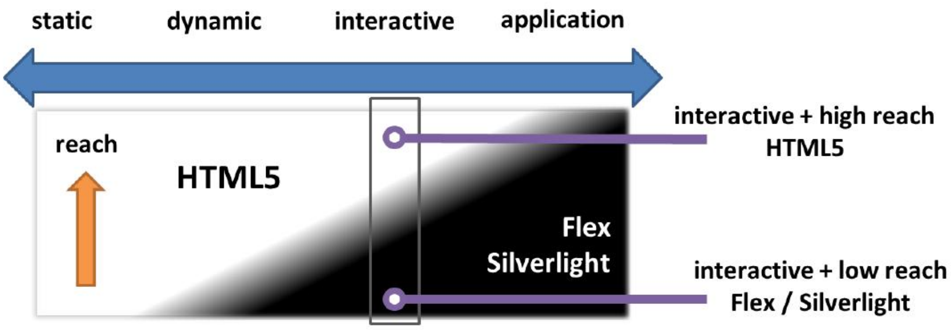
3.1.5. Stability

a. The system should be immune to crashed due to incorrect input from users.
b. In case of internal server error any data not stored in DB is lost. Critical data like transactions and card ownership is written straight to the DB and considered "done" only when this write is accomplished. Data of lesser importance like whish\trade list can be saved on server memory and written to DB once every minute.
c. In any case the data should remain coherent (no duplication or permanent lost of cards) and the user data should be synchronized with the server data.

3.2. Platform constraints

As it was written before, client side will be developed in Flex4 technology of Adobe. Choice was made according to article which compares HTML5, Flex and Silverlight and called "Flex, Silverlight or HTML5? Time to decide…". The diagram (page 16 in the article) shows the relation between application requirements, 'reach' level of application and software technologies for convenient implementation. When 'reach' determined by combining application features with browser version adoption. (page 11)

"In order to differentiate between different web sites and applications we will start by looking at a single metric: complexity. The most basic, least complex web sites are comprised of largely static content, they may have complex server-side logic generating and maintaining this content, but once it reaches the browser it does not change, resembling the pages of a book. As the complexity increases, web sites become more dynamic, with user interactions resulting in small changes to the page currently being displayed. Further increases in complexity result in truly interactive or immersive web sites where there is no longer the concept of a page. Finally, at the most complex end of the spectrum we have applications, which perform complex and useful business functions and often feel somewhat disconnected from the web." (page13)



Because we are developing interactive web-application, we are standing in a dark side of rectangle. In addition, Silverlight technology is still not so wide spread as Flex. Thus, our choice was **Flex4**. So, using the application requires Adobe Flash Player installation in a web browser.

The server side we will implement with **J2EE** and **Apache Tomcat** – web server and servlet container. We will also use **MySQL** for our database management.

We still do not know which layer it is convenient to use for messaging implementation between server and client. The options are **BlazeDS** or **SOAP**. Hopefully, prototype will help us to make a right decision.

4. User Scenarios

4.1. Actors

There are 4 main actors in the system

Unregistered user – guest in the system. He may only to get familiar with application by view available albums and games for exchanging cards. Of course he may register into the system.

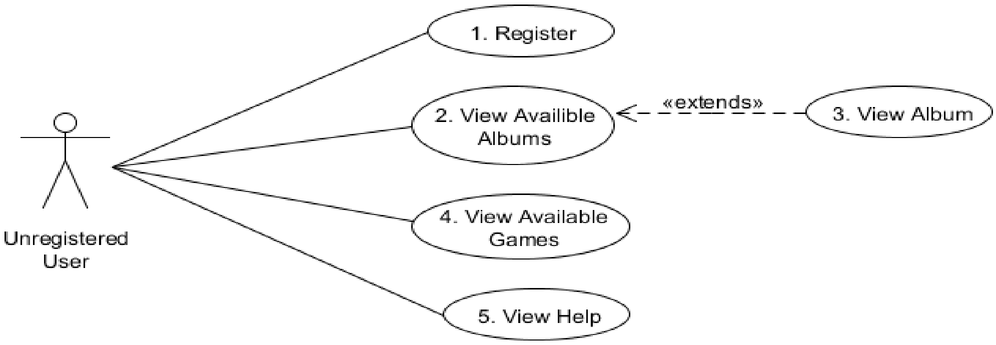
Registered user – this user can watch the albums, he participates in. He can collect cards in this album, participate in mini game on some of his cards, or exchange cards with other users. He can also get a new pin-code and enter it into the system in order to get new cards.

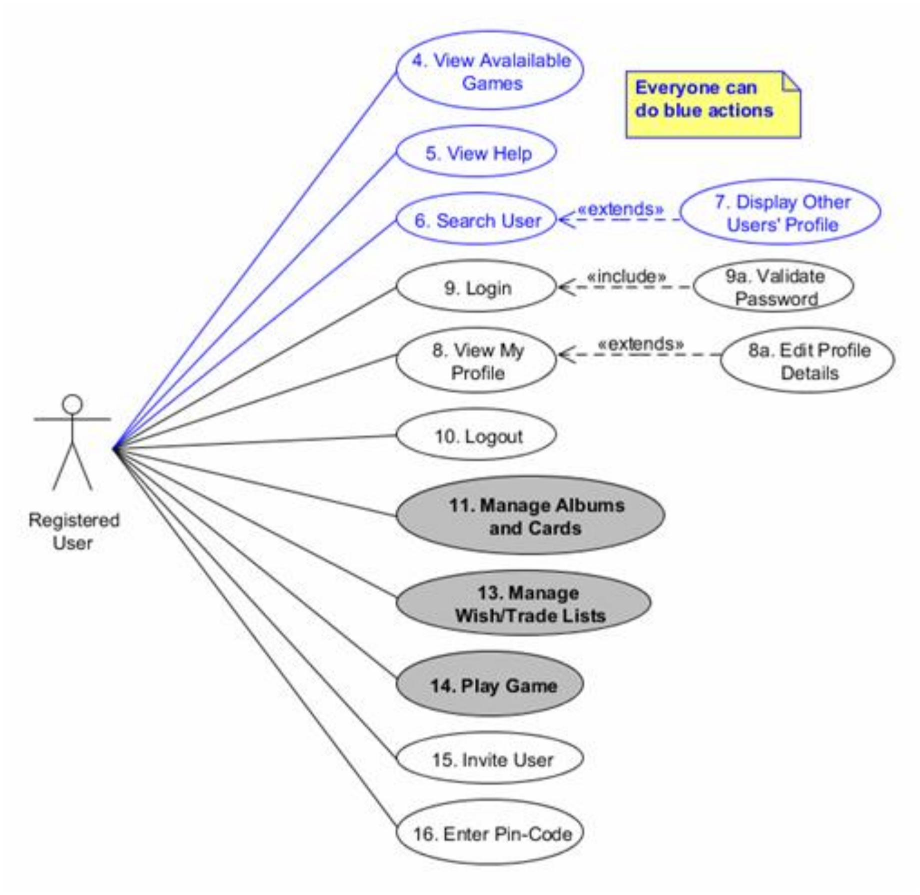
Manager – a high level user, who requests permission from system admin to uploads a new album and then manages it entirely by setting different rules.

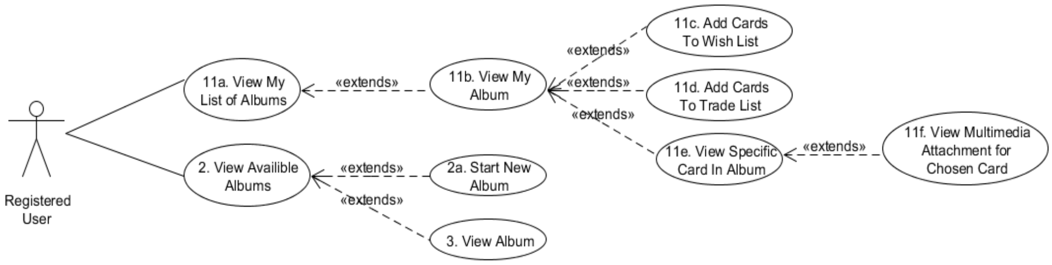
System administrator – grants permissions for different managers to open a new album, and so generates different pin-codes for the later use by the managers. This character is the one who manages the entire system.

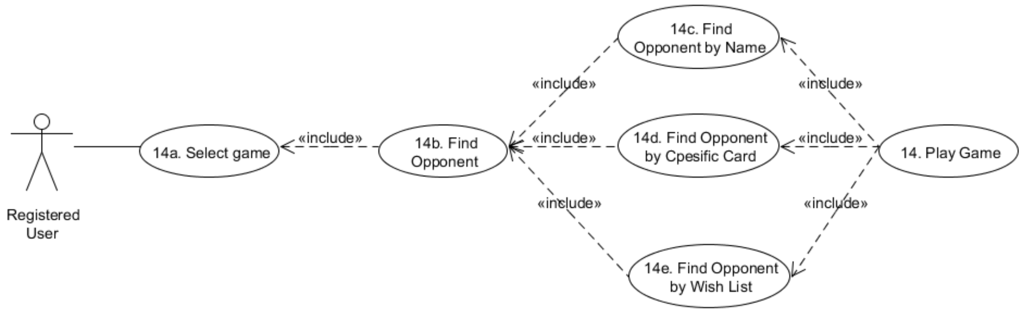
Timer – internal actor of the system. Timer distributes to users sets of cards once in period. Both length of period and set of cards are specified by album manager.

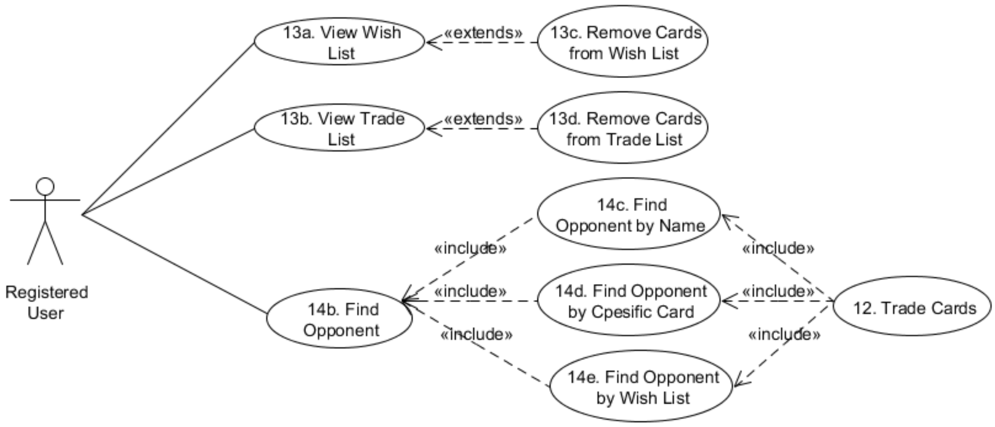
4.2. User scenarios diagrams

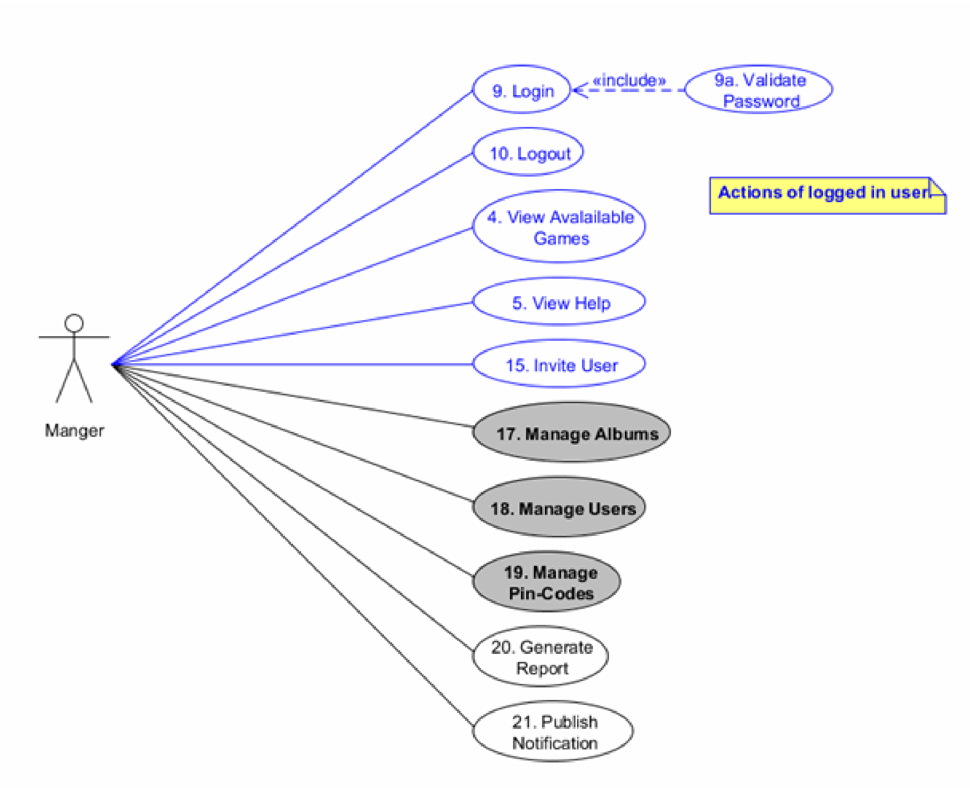


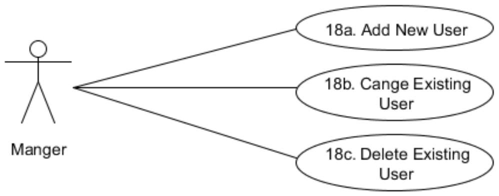

4. View Avalailable Games
Everyone can do blue actions
5. View Help
6. Search User
«extends»
7. Display Other Users' Profile
9. Login
«include»
9a. Validate Password
8. View My Profile
«extends»
8a. Edit Profile Details
10. Logout
Registered User
11. Manage Albums and Cards
13. Manage Wish/Trade Lists
14. Play Game
15. Invite User
16. Enter Pin-Code

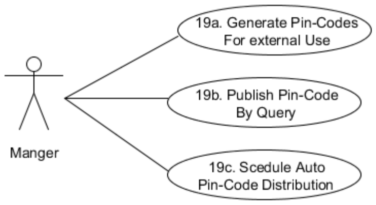

11c. Add Cards To Wish List
«extends»
11a. View My List of Albums
«extends»
11b. View My Album
«extends»
11d. Add Cards To Trade List
«extends»
Registered User
11e. View Specific Card In Album
«extends»
11f. View Multimedia Attachment for Chosen Card
2. View Availible Albums
«extends»
2a. Start New Album
«extends»
3. View Album

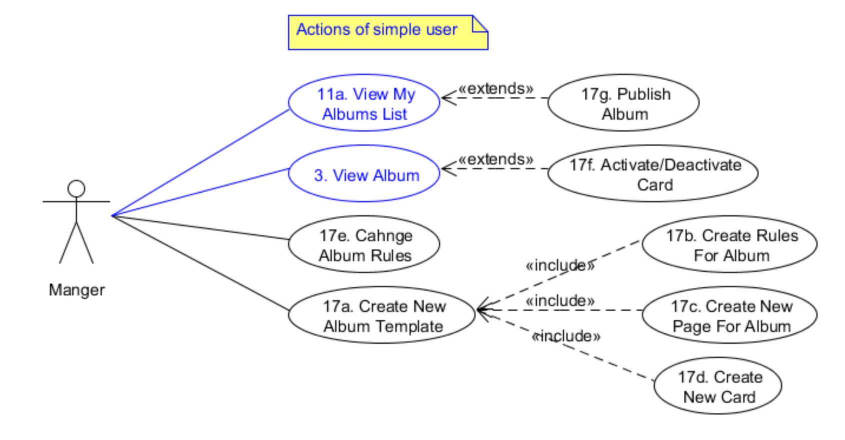

Registered User
14a. Select game
«include»
14b. Find Opponent
«include»
14c. Find Opponent by Name
14d. Find Opponent by Cpesific Card
14e. Find Opponent by Wish List
«include»
«include»
«include»
«include»
«include»
14. Play Game

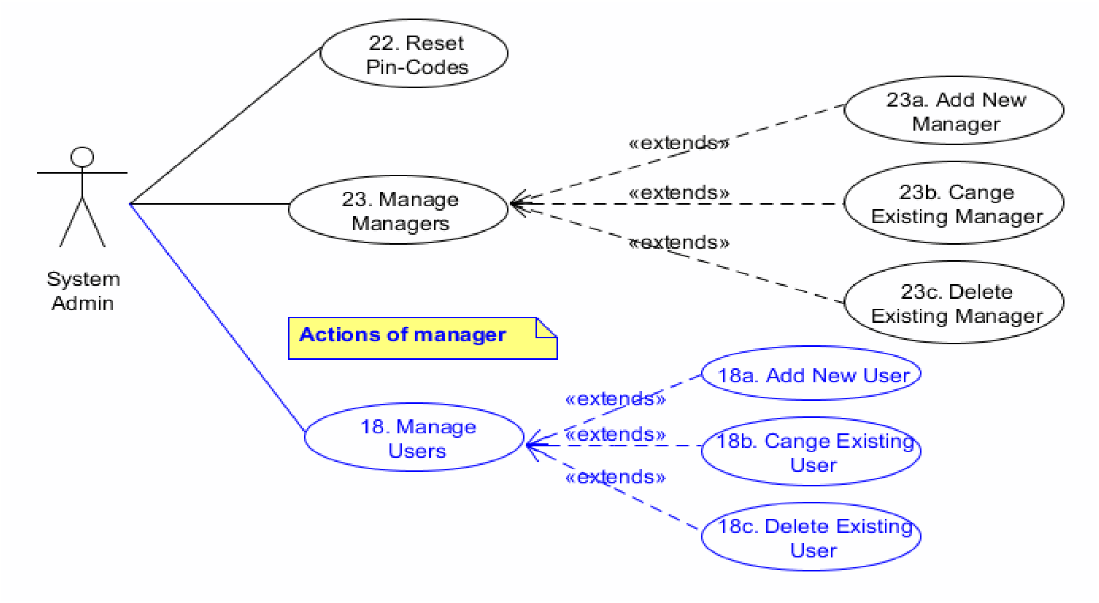

13a. View Wish List
«extends»
13c. Remove Cards from Wish List
13b. View Trade List
«extends»
13d. Remove Cards from Trade List
Registered User
14c. Find Opponent by Name
«include»
«include»
14b. Find Opponent
«include»
14d. Find Opponent by Cpesific Card
«include»
12. Trade Cards
«include»
«include»
14e. Find Opponent by Wish List

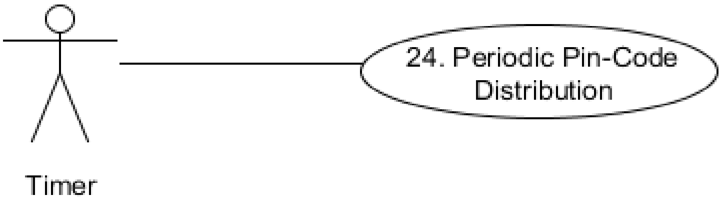

9. Login
«include»
9a. Validate Password
10. Logout
Actions of logged in user
4. View Avalailable Games
5. View Help
15. Invite User
Manger
17. Manage Albums
18. Manage Users
19. Manage Pin-Codes
20. Generate Report
21. Publish Notification


Manger
18a. Add New User
18b. Cange Existing User
18c. Delete Existing User


Manger
19a. Generate Pin-Codes For external Use
19b. Publish Pin-Code By Query
19c. Scedule Auto Pin-Code Distribution


Actions of simple user
Manger
11a. View My Albums List
«extends»
17g. Publish Album
3. View Album
«extends»
17f. Activate/Deactivate Card
17e. Cahnge Album Rules
17b. Create Rules For Album
«include»
17a. Create New Album Template
«include»
17c. Create New Page For Album
«include»
17d. Create New Card


System Admin
22. Reset Pin-Codes
23. Manage Managers
«extends»
23a. Add New Manager
«extends»
23b. Cange Existing Manager
«extends»
23c. Delete Existing Manager
Actions of manager
18. Manage Users
«extends»
18a. Add New User
«extends»
18b. Cange Existing User
«extends»
18c. Delete Existing User


Timer
24. Periodic Pin-Code Distribution

4.3. Unregistered user

UC01 – Register

Description: User wishes to register to become an active user

Actors: Unregistered User

Precondition: None

Post-condition: User has a unique username and password she can use to log in.

Basic Course of Action:

1. User choose to "register new user" via GUI element
2. User enters a unique username and password and chooses to continue
3. The system receives the users' data and validates its correctness
4. The system displays a "welcome %username " message for a few seconds
5. the system displays the "login" view

Extensions:

3a. Username is not legal (too shot, illegal characters or already taken). A message alerting the user is shown and the user enters a new name (go to step 2).

3b. Password is not legal (empty, not strong enough). A message alerting the user is shown and the user enters a new password (go to step 2).

UC02 – View Available Albums

Description: User wishes to browse through all playable albums

Actors: Any User

Precondition: None

Post-condition: List of albums is displayed to the user

Basic Course of Action:

1. The user chooses to "browse albums" via GUI
2. The system displays playable albums, the user can choose specific album to inspect.

UC03 – View Album

Description: User wishes to inspect a specific album with full detail

Actors: Any User

Precondition: User is logged in and user is viewing the album list (UC02)

Post-condition: Full album details displayed to the user

Basic Course of Action:

1. User chooses a single album from the album list
2. The system gets the selected album details and display the "album page" to the user
3. The system displays album rules
4. The system displays album pages

5. For each page, the system displays cards for this page that actor has

Extensions:

5a. If actor is Manager, the system will display all the cards for this album

UC04 – View Available Games

Description: Actor wants to see a list with available games.

Actors: Unregistered user, Registered user, Manager

Preconditions: None

Post-Conditions: List with available games displayed on the screen

Basic Course of Action**:**

1. Actor chooses to see a list of available games
2. The system searches all available games in DB
3. The system displays all games on the screen

Extensions: None.

UC05 – View Help

Description**:** User wishes to get help about the page he is currently viewing

Actors: Any User

Precondition: User is logged in and watching page# (i.e. main page, profile page, card page, etc…)

Post-condition: A help page describing page# is displayed.

Basic Course of Action**:**

1. The user chooses to get help about his current medical condition*.
2. The system retrieves the help page about the page that user is currently visiting
3. The system displays the help page to the user.
4. User decides to continue. The previous page is displayed.

*UI note: a help link (or button) should be available at any time. note2: the help page can be a popup or a different fame so the old page can be easily displayed again.

UC06 – Search User

Description**:** User wishes to find another user in the system

Actors: Registered user, Unregistered User, Manager

Precondition: None

Post-condition: List of users is shown on the screen

Basic Course of Action**:**

1. Actor chooses to find other users
2. The system displays the "search" page
3. The actor enters the name to find
4. The system searches for the user with the specified username

5. The system displays a list of users that their usernames matches the search criteria

Extensions:

5a. Null value entered or search yield no results. An error message is displayed asking the user to enter a new name (go to step2)

5b. Search result is too big to display in one page. Multiple pages are displayed with (←,→) arrows

At any step: user chooses to cancel operation. System main page is displayed. System dispose all search related objects.

UC07 – Display Other Users' Profile

Description**:** Actor wishes to get more details on another user

Actors: Registered User, Manager, Unregistered user

Precondition: Actor completed UC6a

Post-condition: Users' profile page is displayed.

Basic Course of Action**:**

1. Actor select one name from the list (UC06)
2. The system displays the selected user profile page

Extensions:

2a. if actor selects her own name, system displays own user home page (UC6c)

2b. if actor is logged out

The system displays required profile if and only if profile owner allows to all users to view his profile.

2c. if actor is not in friendship with profile owner

The system displays required profile if and only if profile owner allows to all logged in users to view his profile.

4.4. Logged-in User

UC02a – Start New Album

Description**:** User whishes to start collecting cards for a new album

Actors: Registered User

Precondition: User is logged in and he is viewing the album (UC03)

Post-condition: New album is added to the users' active album list

Basic Course of Action**:**

1. The user chooses to start the currently visited album
2. The system creates a new album instance and binds it to the user, adding it to the user list of active albums.

Extensions:

2a. Album is already active for the user, album or user max capacity reached.

- An error message is displayed
- The system displays the "all albums" list page

UC08 – View My Profile

Description**:** User wishes to see his own profile page

Actors: Registered User

Precondition: User is logged in, OPTIONAL: User have list of user names (as a result of UC06)

Post-condition: User is logged in and users' home profile page is displayed.

Basic Course of Action**:**

1. The user selects one name from the list, OR selects his profile page from the main system page.
2. The system displays users' profile page

Extensions: None.

UC08a – Edit Profile Details

Description**:** User wishes to change the his profile details

Actors: Registered User

Precondition: User is logged in and user is watching his profile page UC08.

Post-condition: Users' new details saved to the system

Basic Course of Action**:**

1. The user chooses to change his profile details
2. The system displays a list of details to change
3. The user changes his details
4. The system saves the new users data
5. The system displays the user home page(UC08)

Extensions: None.

UC09 – Login

Description: Actor wishes to login

Actors: Registered user, Manager

Precondition: None

Post-condition: Registered user sees list of his albums. Not registered user gets error message.

Basic Course of Action:

1. User opens the application.
2. System displays welcome screen
3. User enters username and password in suitable fields and clicks “Login” button
4. System validates username and password UC09a
5. System discovers “simple user” status
6. System updates user’s state in DB as “logged in”
7. System shows all user’s albums

Extensions:

4a. System cannot validate user’s details

4a.1. User gets an error message

4a.2. End of UC

5a. System discovers status if “manager”

5a.1 Step 6

5a.2 System shows all albums that manager created

UC09a - Validate Password

Description: One of the users wishes to validate his password.

Actors: All

Preconditions: The user has typed his username and password.

Post-Conditions: The main application window is presented.

Basic Course of Action:

1. The system finds the required username in the database.
2. The system compares the typed password ant the one that is written in the database.
3. If the passwords are the same, the main application window is opened for the user.

Extensions:

1a. the written username does not appear in the list:

1a1. the system presents an error message.

1a2. Go to UC01 (Register)

3a. the password are not the same:

3a1. the system presents an error message.

3a2. Go to UC09 (Login)

UC10 – Logout

Description: Actor wants to log out of the system

Actors: Manager, Registered user

Precondition: Actor is logged in

Post-condition: Actor sees welcome screen

Basic Course of Action:

1. Actor clicks “logout” button
2. System updates actors’ status in DB as “logged out”
3. System shows welcome screen

Extensions: None.

UC11 – Manage Albums and Cards

Description: User wishes to make some changes to his albums

Actors: Registered User

Precondition: User is logged in

Post-condition: None

Basic Course of Action:

1. User invokes UC02 or UC11a

Extensions: None

UC11a - View My List of Albums

Description: Actor wishes to view all his albums

Actors: User, Manager

Precondition: Actor is logged in and has at least one album

Post-condition: Actor sees all his albums and can choose one specific album

Basic Course of Action:

1. Actor chooses to view his album list (go to his homepage)
2. The system checks for all actors’ albums.
3. The system retrieves all albums from the DB
4. The system displays a list of all actors’ albums on screen

Extensions:

2a. Actor has no albums.

- The system will display relevant message.
- End of UC

UC11b - View My Album

Description**:** User wishes to view one of his albums

Actors: User

Precondition: User is logged in and has at least one album

Post-condition: User sees the album his wanted with all its data.

Basic Course of Action**:**

1. User chooses to view his album list (UC 11a)
2. The system displays a list of all users' albums on screen
3. User selects specific album from the list
4. The system retrieves all information about this album from DB
5. The system displays users' album with all cards. (User can manage cards that he already has)

Extensions:

2a. user has no albums. The system will display relevant message.

UC11c – Add cards to wish list

Description**:** User wishes to add cards to his wish list.

Actors: User

Precondition: User is logged in

Post-condition: Cards were added to wish list

Basic Course of Action**:**

1. User watches one of his albums (UC 11b)
2. User wishes to add a card to his wish list.
3. System shows all cards from this, when cards that are already in wish list marked. (So the user cannot add twice the same card)
4. User marks all cards that he wants to add to wish list
5. System adds user choice to DB
6. System refreshes the screen and shows all cards of current album when cards from wish list marked.

Extensions:

3a. User has no cards in chosen album, system will show appropriate
message to the user.

UC11d – Add cards to trade list

Description: User wishes to add cards to his trade list.

Actors: User

Precondition: User is logged in

Post-condition: Cards were added to wish list

Basic Course of Action:

1. User watches one of his albums (UC 11b)
2. User wishes to add a card to his trade list
3. System shows all cards from this album that user already has, when cards that are already in trade list marked. (So the user cannot add twice the same card)
4. User marks all cards to add to trade list
5. System adds user choice to DB
6. System refreshes the screen and shows all cards of current album when cards from trade list marked.

Extensions:

3a. User has no cards in chosen album, system will show appropriate message to the user

UC11e - View specific card in album

Description: User wishes to view one of the cards in one of his albums

Actors: User

Precondition: User is logged in and has at least one album

Post-condition: User sees the card in the album and the multimedia attachment.

Basic Course of Action:

1. User watches one of his albums (UC 11b)
2. The system displays users' album with all the cards and data
3. User selects a card by clicking on it.
4. The system retrieves card information from DB.
5. The system displays card on the screen
6. The system displays a link under the picture "View the multimedia content"
7. Users click on a link
8. UC11f

Extensions:

7a. user does not click on the link

- End of UC

UC11f - View multimedia attachment for chosen card

Description**:** User wishes to view multimedia attachment of card

Actors: User

Precondition: User is logged in and has selected a card from album UC11e

Post-condition: User the multimedia attachment for selected card.

Basic Course of Action**:**

1. User clicks on link “View multimedia content”
2. System checks if there is any multimedia content for this card in DB
3. System shows on the screen retrieved information

Extensions:

3a. there is no related multimedia content for selected card

- System shows appropriate message
- End of UC

UC12 – Trade Cards

Description**:** User wishes to trade cards

Actors: User

Precondition: User is registered and logged in.
Another user is registered, logged in and wants to trade

Post-condition: a trade is made

Basic Course of Action**:**

1. User select his cards for trade
2. User finds an opponent (UC14b)
3. User select opponent cards for trade
4. The systems sends the request to the opponent
5. The opponent accept the request
6. The system switches the cards owners

Extensions:

1a user has no cards to trade. User will get a relevant message
2a. no matches found. User will get a relevant message
3a. opponent has no cards to trade. User will get a relevant message
5a. opponent is offline. The request will hold for his response
5b. opponent declines. Step 6 will not happen. User will get a relevant message.

UC13 – Manage Wish List and Trade List

Description: User wishes to make some changes to his wish list or his trade list

Actors: Registered User

Precondition: User is logged in

Post-condition: None

Basic Course of Action:

1. User invokes UC13a or UC13b or UC12

Extensions: None

UC13a - View wish list

Description: User wishes to view cards from his wish list.

Actors: User

Precondition: User is logged in

Post-condition:

Basic Course of Action:

1. User chooses to view his wish list
2. The system looks in DB for all cards from user's wish list.
3. The system displays the list of cards on the screen

Extensions: None

UC13b - View trade list

Description: User wishes to view cards from his trade list.

Actors: User

Precondition: User is logged in

Post-condition:

Basic Course of Action:

1. User chooses to view his trade list.
2. The system looks in DB for all cards from user's trade list.
3. The system displays the list of cards on the screen

Extensions: None

UC13c – Remove cards from wish list

Description: User wishes to remove cards from his wish list.

Actors: User

Precondition: User is logged in & user has cards in his wish list

Post-condition: Cards were removed from wish list

Basic Course of Action:

1. User watches his wish list (UC13a)
2. User wishes to remove a card from his wish list
3. System shows all cards of wish list

4. User marks all cards that he wants to remove
5. System removes marked cards from DB
6. System refreshes the screen and shows all cards that remained in the list.

Extensions:

3a.User has no cards in wish list, system will show appropriate message to the user

UC13d – Remove cards from trade list

Description**:** User wishes to remove cards from his trade list.

Actors: User

Precondition: User is logged in & user has cards in his trade list

Post-condition: Cards were removed from wish list

Basic Course of Action**:**

1. User watches his trade list (UC13b)
2. User wishes to remove a card from his trade list
3. User marks all cards that he wants to remove from his trade list
4. System removes marked cards from DB
5. System refreshes the screen and shows all cards that remained in the list.

Extensions:

2a.User has no cards in trade list, system will show appropriate message to the user

UC14 – Play Game

Description**:** User wishes to play game

Actors: User

Precondition: User is logged in.
Another user is logged in and wants to play

Post-condition: plays a game versus another user

Basic Course of Action**:**

1. User select a game from the games list (UC14a)
2. User finds an opponent (UC14b)
3. User challenge the opponent for a game
4. The systems sends the request to the opponent
5. The opponent accept the request
6. The systems starts the game
7. The game finished
8. The system decides who is the winner
9. The system added losers' bet cards to winners' bet cards

Extensions:

1a. no matches found. User will get a relevant message

3a. the opponent is offline. User will get a relevant message

4a. the opponent declines the request. User will get a relevant message

5a. the opponent declines the offer. Use case stops.

5b. the opponent changes the offer and sends the new offer to original user.

7a. one of the users logged off during the game. The system will consider him as the loser and proceed to step 9

UC14a – Select Game

Description**:** User wishes to select game

Actors: User

Precondition: User is logged in.

Post-condition: game request is prepared.

Basic Course of Action**:**

1. User wishes to play a game
2. The system retrieves list of available games on start up
3. The system displays the list of games
4. User click on a game to select it.

Extensions:

4a. user has no cards to play on. User will get a relevant message

UC14b - Find opponent (By name, by specific card, by wish list)

Description**:** User wishes to find an opponent

Actors: User

Precondition: User is logged in.
Game is selected

Post-condition: User sees the opponent and can challenge him

Basic Course of Action**:**

1. User chooses to find opponent by name button (UC14c)
2. User selects the opponent

Extensions:

1a. User chooses to find opponent by specific card (UC14d).

1b. User chooses to find opponent by wish list (UC14e)

UC14c - Find opponent by name

Description**:** User wishes to find an opponent by name

Actors: User

Precondition: User is registered and logged in

Post-condition: User sees the opponent and can challenge him

Basic Course of Action**:**

1. User enters the name of his opponent.
2. The system searches in the DB for all opponents that have the name entered as substring

3. The system displays a list of users on screen
4. User chooses from the list a specific user
5. The system retrieves the information of the specific user
6. The system displays the information on screen.

Extensions:

3a. no matches found. User will get a relevant message

UC14d - Find opponent by specific card

Description**:** User wishes to find an opponent by specific card

Actors: User

Precondition: User is registered and logged in

Post-condition: User sees the opponent and can challenge him

Basic Course of Action**:**

1. User opens a card (UC11e)
2. User wishes to find opponent for this card.
3. The system searches in the DB for all opponents that have the specific card.
4. The system displays a list of users on screen
5. User chooses from the list a specific user
6. The system retrieves the information of the specific user
7. The system displays the information on screen.

Extensions:

3a. no matches found. User will get a relevant message

UC14e - Find opponent by wish list

Description**:** User wishes to find an opponent by wish list

Actors: User

Precondition: User is registered and logged in

Post-condition: User sees the opponent and can challenge him

Basic Course of Action**:**

1. User wishes to find an opponent for wish list.
2. The system searches in the DB for all opponents that the at least one of the cards of users wish list.
3. The system displays a list of users on screen
4. User chooses from the list a specific user
5. The system retrieves the information of the specific user
6. The system displays the information on screen.

Extensions:

3a. no matches found. User will get a relevant message

UC15 – Invite friend

Description: Actor wants to invite a new user.

Actors: Registered user, Manager

Precondition: Actor is logged in

Post-condition: Appropriate message sent to user’s e-mail address

Basic Course of Action**:**

1. Actor wishes to invite a new user
2. System opens a screen with following fields: e-mail address, album name, free text.
3. Actor fills fields when e-mail address is necessary
4. Actor clicks “Invite” button
5. System checks fields content
6. System sends message to e-mail address mentioned in field above with link to the album and free text

Extensions:

- 5a. Email address is not possible or there is no album with entered name, system shows an error message and return to step 3
- 5b. Album name is empty, system sends general invention to e-mail address, End of UC
- 6a. Actor with entered e-mail has album with entered album name, system shows appropriate message, end of UC

UC16 – Enter pin code

Description: User enters a pin code

Actors: User

Precondition: User is logged in.
The user has legal pin code

Post-condition: The pin code received by the system and user receives cards

Basic Course of Action**:**

1. User enters the “Pin Code” section.
2. User enters the legal pin code he has
3. The system parses the pin code and marks it as “Used”
4. The system generates new instances of the cards according to pin code logic
5. The user receives the cards

Extensions:

2a. user has illegal pin code. User will get a relevant message

4.5. Manager

UC17 – Manage Albums

Description**:** Manager wishes to make some changes to his albums

Actors: Manager

Precondition: Manager is logged in

Post-condition: None

Basic Course of Action**:**

1. User invokes UC11a or UC17e or UC03 or UC17a

Extensions: None

UC17a – Create New Album Template

Description**:** Manager wants to create a new album template

Actors: Manager

Precondition: Manager is logged in

Post-condition: New album template is added to the managers' album list.
New album has "unpublished" state

Basic Course of Action**:**

1. The manager chooses to create a new album template
2. The system display "empty album template" prompting the manager to fill in basic album details (Name, Title, Cover Picture)
3. The manager fills the details and chooses to continue
4. The system prompt the user to create the rules for the album (UC17b)
5. The manager creates pages for the album (UC17c) until he chooses to continue
6. The system displays the album "preview" page.
7. The manager chooses to continue
8. The system saves the album in the manager list of album templates with "unpublished" state

Extensions:

5.a. The manager chooses to generate a default template album:

- Chooses the number of cards and the default number of cards per page
- Browse for a set of pictures for the cards on hard drive.
- The system automatically generates pages and fill them with default valued cards with the selected pictures
- OPTIONAL: The manages iterates over the cards changing the names and value of specific cards.

6.a. The manager wants to change some details

- Chooses which page or rules to change
- chooses to save the changes
- go to: step 6

UC17b – Create Rules For Album

Description**:** Manager wants to set rules for an album

Actors: Manager

Precondition: Manager is logged in, album creation is in progress

Post-condition: Album rules are saved for the album template

Basic Course of Action**:**

1. Manager wants to set rules for an album
2. The system display the set of mandatory rules
3. The manager sets the rule value (e.g. how many cards in the album) for each rule.
4. The manager chooses to add custom rules using "Custom Rule Templates" (to be described separately
5. The manager chooses to finish adding rules
6. The system displays a summary of the rules created
7. The manager chooses to continue
8. The system saves the rules set and link it to the album template

Extensions: None.

UC17c – Create New Page for Album

Description**:** Manager wants to add a new page to an album template

Actors: Manager

Precondition: Manager is logged in; Album template is being created (UC17a)

Post-condition: New page is created and linked to the album template

Basic Course of Action**:**

1. The manager chooses to add a new page to an album template
2. The manager chooses album background (via color picker or image uploader)
3. The manager chooses the number of cards the page will hold
4. The manager sets the layout for the cards In the page
5. The manager creates a new card for the page (UC17d) for each card **OR :**
6. The manager chooses an "unlinked" card from the unlinked cards pool.
7. The manager chooses to finish page creation

Extensions:

7a. Not all cards are set for the page: (there's a card slot but no card is registered with it)

- The page is saved as "unfinished"
- Album creation cannot complete while there are "unfinished" pages

UC17d – Create new card

Description: Manager wants to create a new card template for album

Actors: Manager

Precondition: Manager is logged in, page creation (UC17c) is in progress

Post-condition: A new card is created and linked to the page & album template

Basic Course of Action:

1. The manager chooses to create a new card
2. The manager fills card details (name, picture, value, special)
3. The manager chooses to finish card creation
4. The card is saved in the system
5. A link between the card and its parent page is created
6. A link between the card and its parent album template is created

Extensions:

5.a: Page creation is NOT in progress

- The card is saved as "unlinked" and can be linked in the future.

UC17e – Change album rules (disable, change value)

Description: Manager wants to change some rules for album

Actors: Manager

Precondition: Manager is logged in. Manager is owner of an album

Post-condition: Rules in album were changed.

Basic Course of Action:

1. Manager view one of his albums
2. Manager wishes to change one of the rules.
3. System shows all dynamic rules for this album and its current value
4. Manager sets new options for album rules
5. System saves updated rules
6. System publishes notification about rule updates

Extensions:

3a. There are no dynamic rules in album, system shows appropriate message ,end of UC

UC17f – Activate/Deactivate card

Description: Manager wants to change card state from active to inactive or vice versa.
Inactive cards cannot be distributed.
Inactive albums cannot be started by users.

Actors: Manager

Precondition: Manager is logged in

Post-condition: Card.isActive = !(@pre card.isActive)

Basic Course of Action**:**

1. Manager views album UC03
2. Manager clicks on the button “Change cards state”
3. System displays the cards in the multiple choice list when not active cards are gray shaped.
4. Manager selects cards that he wishes to change their state
5. Manager clicks the button “change”.

Extensions: None.

**UC17g – **Publish Album

Description**:** Once manager completed album creation, he can publish his album to all users or he can wait for another desirable date to do this.

Actors: Manager

Precondition: Manager is logged in. Album creation completed.

Post-condition: Album.isActive = !(@pre album.isActive)

Basic Course of Action**:**

1. Manager views unpublished album UC03
2. Manager clicks button “Publish”
3. System adds the album to available album list

Extensions: None.

**UC18 – **Manage Users

Description**:** Actor wishes to manage owners of one of his albums

Actors: Manager, System Admin

Precondition: Actor is logged in

Post-condition: None

Basic Course of Action**:**

1. Actor invokes UC18a or UC18b or UC18c

Extensions: None

UC18a - Add New User

Description: Actor wishes to add new user to system.

Actors: Manager, System Admin.

Preconditions: Actor has submitted request for starting the album

Post-conditions: The data is registered in the system.

Basic Course of Action**:**

1. Actor receives a request from user to start his album.
2. Actor submits the request into the system.
3. The system presents a confirmation window for the actor.
4. Actor confirms.

5. The system stores data about the user for the specified album.
6. Actor receives a confirmation message from the system.

Extensions:

4a. Actor decides not to add the user for the album:

4a1. Finish use-case.

Failure flows:

5a. The system crashes during the attempt to store the data.

5a1. Go back to 2

UC18b - Change Existing User

Description: Actor wishes to change the saved information about existing user.

Actors: Manager, System Admin.

Pre-conditions: The user that has to be updated is registered in the system.

Post-conditions: The data is updated successfully.

Basic Course of Action**:**

1. Actor decides to change some stored data about the user.
2. Actor submits the request with the updated data into the system.
3. The system presents the confirmation window to the actor.
4. Actor confirms.
5. The system stores the updated data.

Extensions:

4a. Actor decides not to update the user's data:

4a1. Finish use-case.

4b. Actor decides that there are some more corrections that are needed:

4b1.Actor performs the corrections.

4b2. Go back to 2.

Failure flows:

5a. The system crashes during the attempt to store the data.

5a1. Go back to 2.

UC18c - Delete Existing User.

Description: Actor wishes to delete one of the existing users.

Actors: Manager, System Admin.

Pre-conditions: The user that is going to be deleted is registered in the system.

Post conditions: The user is deleted from system.

Basic Course of Action**:**

1. Actor decides to delete the user.
2. Actor submits his name to the system.
3. The system presents a confirmation window.
4. Actor confirms.

5. The data about the user is deleted.
6. The system presents a “done” message to the actor.

Extensions:

4a. The actor decides not to delete the user:

4a1. Finish use-case.

Failure flows: none.

UC19 – Manage Pin-Codes

Description**:** Manager wishes to make some actions with pin-codes for his albums

Actors: Manager

Precondition: Manager is logged in

Post-condition: None

Basic Course of Action**:**

1. User invokes UC19a or UC19b or UC19c

Extensions: None

UC19a – Generate pin-codes for external use

Description**:** Manager wants to generate some pin codes for his own reasons

Actors: Manager

Precondition: Manager is logged in. Manager is owner of an album

Post-condition: Manager gets specified number of pin-codes

Basic Course of Action**:**

1. Manager wishes to generate pin-codes
2. System shows screen with following fields: album name, code type and number of pin-codes to generate
3. Manager fills the fields
4. System generates wished number of pin-codes for specified album
5. System shows the list of generated pin-codes on the screen

Extensions:

2a. Album is closed for the competition, system shows an error message end of UC

UC19b – Publish pin-code by query

Description**:** Manager wants to generate publish new pin-codes to album users

Actors: Manager

Precondition: Manager is logged in. Manager is owner of an album

Post-condition: Each album user get new pin-code

Basic Course of Action**:**

1. Manager view one of his albums
2. Manager wishes to publish pin-codes

3. Manager chooses group of users from given options. (e.g. users online, active users est.)
4. Manager chooses pin-code type
5. System generate pin-code for each user
6. System sends generated pin code to user's e-mail (or publishes as private notification)

Extensions:

4a. Album is closed for the competition; system shows an error message, end of UC

UC19c - Schedule auto-code distribution

Description: The manager wishes to choose conditions for auto-code distributions.

Actors: Manager

Preconditions: There is a t least one pin-code existing for this album.

Post-conditions: The manager's choice is registered in the system.

Basic Course of Action**:**

1. The manager wishes to decide auto-code distribution.
2. The system presents a list of optional criterions for the distribution.
3. The manager chooses one of the criterions.
4. The system records his choice.

Extensions: none.

UC20 – Generate Report

Description: Manager wishes to get some statistics about one of his albums

Actors: Manager

Preconditions: Manager logged in

Post-Conditions: Report with statistics generated and displayed on the screen

Basic Course of Action**:**

1. Manager chooses to watch statistics
2. System displays list with all manager's albums
3. Manager chooses an album
4. System retrieves from DB suitable statistics
5. System displays retrieved statistics on the screen

Extensions: None.

UC21 – Publish notification

Description**:** Manager wants to publish notification about one of his albums

Actors: Manager

Precondition: Manager is logged in. Manager is owner of an album

Post-condition: All album users got notification

Basic Course of Action**:**

1. Manager wishes to publish a notification.
2. System shows new screen with following fields: album name, notification text
3. Manager fills the fields
4. System publishes new notification to all users of mentioned album

Extensions:

4a. Notification text is empty or too large, system shows appropriate message to manager, go to step 4.

4.6. System administrator

UC22 - Reset pin-code

Description: The system administrator wishes to reset existing pin code.

Actors: System Admin.

Preconditions: There is at least one existing pin-code.

Post conditions: The change to pin-code is registered successfully in the system.

Basic Course of Action**:**

1. The system admin decides to change the existing pin-code.
2. The system admin chooses one of the existing pin-codes.
3. The manager changes the pin-code according to the needs.
4. The system presents a confirmation window to the system admin.
5. The system admin confirms.
6. The change to the pin-code is saved.

Extensions:

3a. The manager decides to delete the pin-code entirely:

3a1. End of UC.

UC23 – Manage Managers

Description**:** Admin wishes to make some changes on managers

Actors: System Admin

Precondition: Actor is logged in

Post-condition: None

Basic Course of Action**:**

1. Actor invokes UC23a or UC23b or UC23c

Extensions: None

UC23a - Add New Manager

Description: The system admin wishes to add a new manager to the system.

Actors: System Admin.

Preconditions: The user has submitted request for being a manager.

Post-conditions: The data is registered in the system.

Basic Course of Action**:**

1. The system admin receives a request from user to be a manager.
2. The system admin submits the request into the system.
3. The system presents a confirmation window for the system admin.
4. The system admin confirms.
5. The system stores data about the new manager.

Extensions:

4a. The system admin decides not to add this manager.

4a1. Finish use-case.

Failure flows:

5a. The system crashes during the attempt to store the data.

5a1. Go back to 2.

UC23b - Change Existing Manager

Description: The system admin wishes to change the stored data about the existing manager in the system.

Actors: System Admin.

Preconditions: The manager that has to be updated is registered in the system.

Post-conditions: The data is updated successfully.

Basic Course of Action**:**

1. The system admin decides to change some stored data about the manager.
2. The system admin submits the request with the updated data into the system.
3. The system presents the confirmation window to the system admin.
4. The system admin confirms.
5. The system stores the updated data.

Extensions:

4a. The system admin decides not to update the user’s data:

4a1. Finish use-case.

4b. The manager decides that there are some more corrections that are needed:

4b1. The manager performs the corrections.

4b2. Go back to 2.

Failure flows:

5a. The system crashes during the attempt to store the data.

5a1. Go back to 2.

UC23c - Delete Existing Manager

Description: The system administrator wishes to delete information about one of the existing managers in the system.

Actors: System Admin.

Preconditions: The manager, that is going to be deleted is registered in the system.

Post conditions: The manager is deleted successfully.

Basic Course of Action**:**

1. The system admin decides to delete the manager.
2. The system admin submits his name to the system.
3. The system presents a confirmation window.
4. The system admin confirms.
5. The data about the manager and his albums is deleted.
6. The system sends a notification to all users who participate in this manager’s albums.
7. The system presents a “done” message to the system admin.

Extensions:

4a. The system admin decides not to delete the manager:

4a1. Finish use-case.

4.7. Timer

UC24 – Periodic Pin-Code Distribution

Description: It’s time to distribute pin-codes for album

Actors: Timer

Preconditions: Time period was set by manager for album A.
There is still no winner for album A.

Post-Conditions: Each album’s owner gets pin-code (by e-mail or in notification area)

Basic Course of Action**:**

1. Time period since last distribution expired
2. Timer queries the system to generate one pin-code for each album A owner
3. The system sends pin-code to each user via e-mail or publishes it in the notifications

Extensions: None.

5. Risk Assessment & Plan for the Proof of Concept (The First Prototype Development)

There are several goals we want to achieve during prototype development. One of them is to understand which actions it is better to implement on the client side, and which actions is better to put on the server side.
Second goal is to improve the flow of described use cases.
Third one is do design user friendly graphical interface as much as possible.
We also believe that prototype developing will help us to understand which additional layers we need to put in order to easy communicate with server side (BlazeDS, SOAP technology or may be another one).

As it was described before, we would like to give our users a wide variety of capabilities for use and maintenance of the album. We want our user to be able to see his list of albums, enter to one of them, see his list of cards, manage his wish list and trade list etc...
So we would like to present some of these capabilities in our prototype, which would be written in flex technology and present some aspects of using the album by the simple user.
Our prototype would present the simple functionality of open the list of albums,
enter to one of them and perform some actions:

1. Scrolling the pages
2. Watch the list of available cards owned by user
3. Drag one of the cards to its place
4. Watch multimedia attachment for chosen card

We want to make sure that these actions can be performed easily by user and do not require any additional knowledge from user. Moreover, because it is relatively easy to develop graphical user interface in Flex, costumers and a simple user will see relatively a lot of functionality at early stage of development.

Risk assessment	Solution
Interactive and attractive UI development	As it was mentioned above, one of the goals is to provide user friendly interface and choose the most convenient platform for development. In order to make right decisions we checked some options in summer and read a lot about developing such interactive web applications as we want.
Learning new technologies (Flex 4)	We started to learn Flex in summer. Moreover, our technical advisors (Asaf and Eitan) help us and support us during development process.

Keep to the time scheduling	We won’t develop the entire system. There is implementation priority for each requirement and rule in description tables. When 1 stands for highest priority, 2 for middle and 3 for lowest.

Even if we would decide not to use the entire prototype in the end, some basic ideas of this implementation can be definitely used later in the final project. Of course, we can’t forget that designing and developing the prototype gives us an important experience in using flex and this experience will serve us as the project goes on.

6. Appendices

6.1. Code-Types

What is Pin-Code?

The code character string users can use to load their account with new cards.

To avoid code prediction by malicious user, each code is encoded with SHA1 hash function.

Each valid code is listed to no code can be used twice.

There are many possible ways to achieve new code, for example:

- Send/receive sms
- Buy staff in supermarket/store, pay bill for a service, etc. Code may be printed on the receipt.
- Album manager can send codes by e-mail.
- … and more …

Each card may be active or inactive. System releases to users only active cards. This feature allows to album manager to manage the order of releasing new cards.

Pin-code types

Type	Name	Description	Impl. priority
A	Fixed-Cards	The code describes exactly what cards are given	1
B	Fixed-Value	Random cards for a sum equal to the code value. Code creator can decide what albums to include	2
C	Fixed-Number	Fixed number of cards, actual cards and their value is random. Code creator can decide what albums to include	2
D	Fixed-Value from set	Same as fixed-value but the creator select a specific set of cards that can be given by the code	2
E	Fixed-Number from set	Same as fixed-number but the creator select a specific set of cards that can be given by the code (optional, the creator may decide the probability for each card to appear)	3
F	User Select from set	The user can select the card he desire from the chosen set	3

- Type code may be represented as first letter in pin-code string.

6.2. Rules

Game Rules

#	Name	Description	Dynamic	Legal values	Default value	Impl. priority
1	Max albums per user	The maximum number of albums any user can have active at the same time	+	Numeric ≥ 0	100	2
2	Max users per album	The maximum number of users who can participate in an active album (can be # of instances in case where a user have multiple albums of the same type)	+	Numeric ≥ 0	∞	2
3	Max on-line users	Self explanatory	+	Numeric ≥ 0	∞ (Dependent on server capacity)	3
4	Max users	Global maximum of users registered to the system	+	Numeric ≥ 0	∞	3

Album Rules

#	Name	Description	Dynamic	Legal values	Default value	Impl. priority
1	Album is active	Album must be active to be playable. Users can not modify, trade or get new cards for inactive albums. (Though they can still watch them)	+	Boolean	True	1
2	Album Size	The number of cards to complete this album	-	Numeric > 0	30	1
3	# Of cards per page	How many cards are shown in every page. This can later be modified per-page**	-	Numeric > 0	10	1
4	# of winners	How many players can win by completing this album	+	Numeric > 0	1	1
5	Reusable cards	Cards can be removed from album even after being locked.	-	Boolean	F	2
6	Cross album tradable cards*	If true- cards from this album can be traded in the free market with other cards. If false – cards can only be traded with card from the same album	+	Boolean	True	3
7	#Of albums per user	If a user can have multiple instances of the same album, this rule decides how many	+	Numeric ≥ 0	1	3

8	Card place	There are 3 options. 1. Card can be placed everywhere in album 2. Card can be placed only in defined page in album 3. Card can be placed only in defined place in defined square.	-	Page number in album and/or card cell in page	Card can be placed everywhere. Page number is ∞	2

Card Rules

#	Name	Description	Dynamic	Legal values	Default value	Impl. priority
1	Unique instance of card per user*	If true, users can have only one instance of the same card at the same time.	+	Boolean	True	2
2	Cards details visibility	What properties of cards the user can see when watching it (could be: value, #instances, cross-album, external link etc…) (May be implemented as checkbox)	+	Boolean per each detail		2
3	Inactive cards	Cards that cannot be given by auto-gen pin codes.	+	Set of numeric values. Each value represents a card.	None	3
4	Maximum number of cards for card	The chosen number represents the maximum number of cards each player can get for this card.	+	Numeric ≥1	1	3

Code Rules (per album\global)

#	Name	Description	Dynamic	Legal values	Default value	Impl. priority
1	Code value	the value of the code according to the code type	+	String		1
2	Auto code scheduling	When the system will auto send codes for users. The value represents the frequency in days. When zero, the code won't be sent.	+	Numeric 0≤n≤31	7	2
3	Auto code query	How to select the user to receive the free code. (The limited set of rules: by sms, at cash etc. Choice of moderator may be implemented as binary register.)	+	Moderator will see checkbox with free text	By sms only	2

Mini-game Rules

#	Name	Description	Dynamic	Legal values	Default value	Impl. priority
1	The number of cards each player can play on	The number represents the maximum number of cards that the player can play on.	+	Numeric ≥ 1	1	2

*Rules exist also for sub level, overridden if set.

6.3. About general mini-games

One way of exchanging cards in collectomania is to play on set of cards with another user. For this purpose exactly, we would like to create a mini-game or even to create an interface which would allow adding different mini-games.

Of course, each mini game has its own rules, but there are some general rules for all of them:

1. The user must play on at least one card. Otherwise it's not possible to participate in the mini-game.

   The exact maximal number of cards that the user is allowed to play on can be determined by one of the manager's rules.
2. At least two players have to participate in the mini-game.
3. The winner of the mini-game keeps his own cards and also gets his opponents' cards.

In this project we chose to develop a mini- game which is called Rock-Paper-Scissors.

6.4. Rock-Paper-Scissors

This game will be between two players only and shall consist of three rounds. The winner of at least two of them keeps his set of cards and also takes his opponent's set of cards.

On the screen of the mini-game each player will see three opened images of his options: stone, paper, scissors; and three closed images of his opponent (representing the same options as his). When the two players make their choice (by clicking on the chosen image), two chosen images will be opened at the center of the screen and the message with the winning image will appear in front of two players. The winner of every round is determined by the following rules:

1. Rock wins scissors.
2. Scissors win paper.
3. Paper wins rock.

If there is a tie, one round is added automatically. Generally, while there is a tie, one round is added.

When the three rounds are over, a message with the winner's name will appear in front of two players, and as it is mentioned above: The winner takes it all!