Electra
Group 14

Per Almquist
Peter Andersson
Marcus Bergenlid
Victor Mangs
Ali Mosavian

Contents

1 Preface

1.1 Version history

Version	Comment	Date
1.0	The first version.	28/12/2007

1.2 Intended readership

End users, developers, system administrators.

2 Introduction

2.1 Who are the users and what problem does the system solve for them?

We expect our main users to be men and women between 10 and 30 years of age, although it could attract people of other ages as well. What links our main users together is an interest in painting and competing against others online, and they are not willing to spend hours doing so. These users would probably want to use the system for no more than 20 minutes at a time.

The system is a game, which purpose is to entertain, so the problem in this case is the need for entertainment. We can, of course, never guarantee that we will find a solution for this problem. However, the specified main users will likely find value in our system, since the game is based on both painting and competing.

2.2 The main uses of the system.

This is basically a web based online game with an additional community based sidetrack. There will be two playing modes; Battle mode and Free sketch.

2.2.1 Battle mode

Every user has points, and to participate in a battle a user must be willing to put some of those points on the line. This is a 1-on-1 battle where the users play for a percentage of their own total points. The competition is started by one user (the challenger), by specifying 2 game options (explained later on) - time limit and how much points to bet (in percent). Provided that the challenged user accepts the options, the competition starts.

During gameplay, the users paint pictures of a given topic generated by the system. The time limit chosen will decide when the competition ends. After the competition stops, the pictures are put up on a voting page, where other users of the site will vote for the winner. When the voting validity time specified by the challenger runs out the outcome of the battle is decided, and points are distributed as specified.

2.2.2 Free sketch

In the free sketch mode the user can practice painting before challenging other users. The pictures will not be saved.

2.2.3 User narratives

High school student
The bell rings at Green's high school. Jim, who really likes to compete in almost every possible way, and his friend makes their way to the computer room to try out the new hyped Internet phenomenon. Jim finds a computer, sits down and opens the web browser. He enters the page, logs in and checks if he is still at the top of the ranking list. It turns out that Dan has surpassed him. Jim sees that Dan is online and decides to challenge Dan. He creates a game with 7% bet and a time limit of 30 seconds. He sends a challenge invitation to Dan. Dan accepts the challenge and the game starts. The random topic is "Clueless Dog". Jim starts painting a dog chasing his own house using the pencil with red color. When there is 5 seconds left he realizes that he will not be able to finish painting the house in time and quickly erases it using the eraser. The time runs out and Jim is somewhat discontent with what he created and gets nervous about what Dan has done. Both their pictures comes up on the screen and now Jim gets a heavy feeling in his chest - Dan drew a beautiful dog with an obviously confused face. Jim ask his friends to enter the page and vote for him. Jim then enter Dan's userpage and writes the comment "U're gonna looose, man :P !111!!1!" in Dan's guestbook. Afterwards he logs out and moves on to class.

Receptionist
Anne, a 23 years old woman with a great interest in painting, works in a reception at the Sköndik Hotel where she is just waiting for the time to pass. She reminds herself about the new internet game her friend introduced her to yesterday and decides to take a closer look. She logs in to her account and sees her personal page and a simple menu with a few links. She clicks on the voting link and discover that her friend just made a battle with someone, she clicks that link and two pictures appear. She want to vote for her friend but there are no names associated with the pictures so she votes for the one she finds more interesting. She gets inspired by her friend's battle and decides to practice a little before challenging someone. She navigates to the free sketch page and starts painting a cat. After a while she is satisfied with the picture and feels confident enough to challenge someone. She navigates to the Playing arena, but just as she is about to challenge someone in the list, a customer walks into the reception and she minimizes the web browser.

2.3 The context/environment in which the system is to be used.

The system is intended to be used at an internet café or on a home/school/work PC with a moderately fast internet connection (0.5 Mbit/s or more, both for uploading and downloading). The PC must have Firefox with the Flash plugin installed. The system may work on other web browsers as well, but we will consider it a bonus if it does. The average user is expected to use the system for short periods of time (5-20 minutes), several times a week.

2.4 The scope of the system.

Topic	In	Out
User need to login	x	
1-on-1 Battle mode	x	
Single player game mode	x	
Public forum		x
List of highest ranked users	x	
User profile[a]	x	
Downloading other peoples pictures		x
Sending private messages to users		x
Sending public messages to users	x	
Uploading pictures[b]		x
Private information about the user	x	
List of lowest ranked users		x
Tools used when painting[c]	x	
Buy extra painting tool for points		x
personal gallery	x	
Public gallery	x	
Several different multi player modes		x

[a] User presentation page showing status (online/offline), rank and guestbook.

[b] Except for the profile avatar which is the picture for the profile.

[c] Pencil, bucket, eraser and colors.

2.5 The main factors that need to be taken in to account when designing and building the system.

- Graphical interface.
- How to make sure that competitors start and finish at the same time
- How to vote.
- How to calculate the rankings.
- Tutorials for new users.
- How to save pictures (user created, within a competition, gallery, owner of the pictures).
- Where to save data.
- How to handle pictures.
- How to handle chat.
- Whether or not the player B should be able to see player A's painting in real time during the battle.

- Database design.
- The login mechanism.

2.6 Technologies and risks

All of the technologies mentioned below are mature and well tested as well as widely used. They are de facto standard today and we do not believe that there will be any problem in integrating them together. What might be an issue for some of the group members is the fact that they are not familiar with these technologies or web application development for that matter. But since the knowledge is present within the group their questions can be answered.

2.6.1 Server side technologies

The server side application will be written in the Java programming language using the Java Platform, Enterprise Edition (JEE) API. For the actual web server which will act as the container and provider for the JEE API, Tomcat will be used. Both of these are widely used and accepted and provide good development support as well as good performance. We do not expect to experience any severe problems on this part, apart from the one time setup that is required. As for the database we will be using Postgres 8.x. Postgres is a widely used and tested Database Management System (DBMS) which we do not expect to experience any problems with.

2.6.2 Client side technologies

The painting feature will be written in Flash. Flash provides very good facilities for graphical development and has a good debugger as well as been widely used, hence we do not expect to experience any major problems. Some of the dynamic widgets on the actual web page will be written in JavaScript and interpreted in the browser. JavaScript is usually somewhat troublesome to work with as it is weakly typed and variables are created on the fly. But as these snippets of code will be few and small, we do not expect to experience any major delays because of them.

2.6.3 Development tools

The entire group will be using the NetBeans IDE for the server side development and the code base will be kept synchronised through an SVN server and the SVN plugin for NetBeans. For the client side code either NetBeans IDE or Komodo IDE will be used. Viewing the result and debugging will be done using Firefox with the FireBug plugin. Flash will be used for development of the painting application.

3 Glossary

AP (Artistic points)
The measurement for comparing users that is used throughout the system. See Appendix A for more information.

API (Application Programming Interface)
An interface with prewritten programming code that helps the developers when creating the system.

Authenticate
Verify a user's identity with a password.

Battle
The competition phase, where two users compete against each other in the game of painting the best picture.

Bet
A wager, with AP, between two users.

Bucket
The painting tool, that is used to fill a bounded area on the canvas with a color.

Canvas
The area on which you paint.

Challenge
The competition phase, where one user engage another user to battle against him/her.

Chat room
A part of the Playing arena where a user can communicate with other users.

Client
The part of the system under discussion that requests and consumes the services provided by the system.

Client-server system
A system model that is based on communication between client(s) and server(s).

Community
The base for the social aspect of the system and it's users.

Competition
The three phases; Challenge, Battle and Vote.

Confidential
Something that is not intended to be known publicly.

Disconnect
An interruption in an existing connection to the system.

Download
A transfer of data from a remote computer to a local computer.

Encrypt
Conceal information by altering that information in a structured way.

Eraser
The painting tool that is used to clear parts of the canvas.
Executable
A file containing instructions that the computer can understand.
Free sketch
A part of the system where one can practice his/her painting skills.
Gallery
A part of the system that shows pictures that users has made in the past.
Golden vote
A special vote that decides the outcome of a tied vote.
Guestbook
A part of the system that enables users to post messages to other users.
HTML (HyperText Markup Language)
The standard language for formatting documents on the WWW.
HTTP (HyperText Transfer Protocol)
The standard protocol for transferring HTML documents on the Internet.
IDE (Integrated Development Environment)
An application that provides tools for developing computer applications.
Internet
A network connecting computers around the world.
Java
An object oriented programming language.
Login
The process of authenticating a user so that he/she can be able to use the system.
Minimal guarantee
A demand that is always fulfilled.
Online
The state that a user is in when he/she is connected to the system.
Password
A secret string used to authenticate a user.
Pencil
The painting tool, that is used to paint curves on the canvas.
Picture
The artifact made by a user after painting on the canvas in a battle.
Playing arena
The part of the system where users can compete against each other.
Postcondition
A condition that is to be satisfied after some action.
Precondition
A condition that is to be satisfied before some action.
Primary actor
In a use case, the primary actor is the one who strives to reach a goal by using the system.

Registration
The process where a user inputs information that is to be recorded by the system.
Requirement
A demand on the system that need to be met.
Scenario
A sequence of events.
Score
The number of AP gained by some action.
Server
The part of the system under discussion that response with and provides the services of the system.
Sign up
An action in the Playing arena that makes the user available to be challenged by other users.
Stakeholder
Someone that has interest in the system.
String
A sequence of characters.
SVN (Subversion)
A system that manages different versions of a file.
Time limit
The restricted period of time in which the battle or vote takes place.
Timer
A function of the system that measures the time.
Topic
A subject that sets the guide lines for a battle.
Trigger
An action that serves as a stimulus for a use case.
User
The abstraction of a person using the system.
Username
The alias that a user is known by when using the system.
Vote
The competition phase, where users can decide the outcome of a battle.
Web based
An application accessed via the WWW.
Web browser
A function, on the client side, of the system that views HTML documents.
WWW (World Wide Web)
The collection of documents residing on all Internet servers that use the HTTP protocol.

4 User requirements

4.1 Functional requirements

4.1.1 Registration and login

4.1.1.1 **People shall register to create a user of the system.**
The following information must be entered upon registration: username, password and e-mail address
Rationale: A user is needed in order to use the system.
References to system requirements: 5.1

4.1.1.2 **The system shall only be available to people that are logged in as users.**
Rationale: Users need to be identified and authenticated before they are able to use the system.
References to system requirements: 5.2

4.1.2 Profile

4.1.2.1 **Every user shall have artistic points (AP).**
When a user is created that user is given 100 AP. Users receive or loose AP when they win or loose a competition.
Rationale: The AP rank users. The more AP the higher rank.
References to system requirements: Appendix A

4.1.2.2 **Every user shall have a guestbook where other users can post messages.**
The guestbooks are publicly available but unique to every user.
Rationale: The guestbook enables users to communicate with each other.
References to system requirements: 5.4 5.5

4.1.2.3 **Every user shall have a personal gallery where all pictures made by that user are posted.**
Rationale: A personal gallery helps the user to show all his/her previous work.
References to system requirements: 5.6 5.7

4.1.2.4 **Statistics about users shall be publicly available.**
These user statistics are:

- the total number of competitions participated in.
- the number of won competitions.
- the AP.

Rationale: To show information about a user to other users.
References to system requirements: 5.8

4.1.3 Playing arena

4.1.3.1 **There shall be a Playing arena where users compete.**
Rationale: This will help the users that wants to compete to find each other.
References to system requirements: 5.9

4.1.3.2 **The Playing arena shall be viewable for all users.**
All users shall be able to follow what is going on in the Playing arena.
Rationale: Users might want to look at which users that are available for challenge but without being exposed for other users to challenge them.
References to system requirements: 5.9

4.1.3.3 **Users shall not be able to interact with the Playing arena unless they sign up in it.**
Rationale: To avoid cheating by giving tips, and to avoid disturbance during play.
References to system requirements: 5.9 5.11

4.1.3.4 **The Playing arena shall present the users that are willing to compete.**
Rationale: To let users know which users they can compete against.
References to system requirements: 5.9

4.1.3.5 **The Playing arena shall provide a chat room.**
In the chat room, users can post messages that are viewable to all other users.
Rationale: Users wants to communicate.
References to system requirements: 5.10

4.1.3.6 **Users should be able to view statistics about other users that are signed up in the Playing arena.**
Rationale: To enable information about the users available for challenges.
References to system requirements: 5.11

4.1.3.7 **A user shall be able to challenge other users in the Playing arena.**
The user specifies a few game options and then sends a challenge request to an opponent.
Rationale: Users wants to compete. The gaming options makes the game more versatile.
References to system requirements: 5.9 5.11

4.1.4 Paint

4.1.4.1 **There should be at least 60 and at most 80 colors available when painting pictures.**
Rationale: To create pictures with a variety of colors. There should not be too many colors available in order to keep the painting process simple.
References to system requirements: -

4.1.4.2 **There shall be a bucket, eraser and pencil available while painting pictures.**
The bucket is used to fill an area with a color. The eraser is used to clear an area. The pencil is used to paint curves.
Rationale: To create pictures there is a need of painting tools.
References to system requirements: -

4.1.5 Free sketch

4.1.5.1 **There shall be a free sketch mode available.**
On the free sketch users can practice their skills on their own. The pictures created in the free sketch will not be stored or published in any way.
Rationale: The free sketch enables users to learn how to paint pictures in the system.
References to system requirements: 5.19

4.1.6 Competition

4.1.6.1 General

4.1.6.1.1 **A competition shall consist of three phases: Challenge, battle and Vote.**

- Challenge: when a user asks another user to participate in a competition.
- Battle: the two users competes by painting pictures.
- Vote: when other users vote for one of the pictures created in the battle.

Rationale: You have got to tell someone that you want to battle them. The actual battle is a must. The voting decides who the winner is.
References to system requirements: 5.12 5.13

4.1.6.1.2 **Every started competition that makes it to the voting phase shall have a winner.**
If a tie occurs, one more vote (aka. Golden vote) is needed and that vote decides the outcome of the competition.
Rationale: To solve the problem of tied competitions.
References to system requirements: 5.15

4.1.6.1.3 **When a competition ends, both of the participants' AP shall be affected.**
The user who looses the competition looses his/her betted amount of AP and that amount of AP is received by the winner of the competition.
Rationale: A system often used in games. The users place bets, and the winner takes it all.
References to system requirements: 5.3

4.1.6.2 Challenge

4.1.6.2.1 **A user shall be able to specify gaming options when challenging another user.**
These options are:

- time limit: For how long the battle will last.
- Bet: The amount of AP to bet. Explained below.

Rationale: Different users may want to have different time limits. You can choose how big risk you want to take in a battle by being able to choose how large share of your AP you want to bet.
References to system requirements: 5.12

4.1.6.2.2 **When a user is challenged, the user shall be notified of that.**
The notification shall also provide a link to accept the challenge.
Rationale: This is how a user tells another user that he/she wants to battle.
References to system requirements: 5.12 5.13

4.1.6.2.3 **The bets shall relate to the amount of AP that each user has.**
A specified percentage of the challenger's AP and the opponent's AP at this moment.
Example: User A (100 AP) challenges user B (10 AP) and choose to bet 30%. That means, if B accepts the challenge, A has betted 30 AP and B has betted 3 AP.
Rationale: The betting system is constructed this way because if you challenge someone who has much fewer points than you, the amount you gain is less than if you challenge someone at your own skill level.
References to system requirements: 5.12, Appendix A

4.1.6.2.4 **While a user participates in a competition the betted amount of AP shall be reserved.**
That the betted amount of AP is reserved means that it can not be used in other bets, but that it is still part of the user's total amount of AP.
Rationale: AP are reserved to ensure that the user do not bet more AP than that user have.
References to system requirements: Appendix A

4.1.6.3 Battle

4.1.6.3.1 **The system shall randomly generate a topic for each battle that the users are intended to paint.**
Rationale: The topic is what the users are intended to paint.
References to system requirements: 5.19

4.1.6.4 Vote

4.1.6.4.1 **The voting phase for each competition shall last 24 hours.**
Rationale: The competition must have an end. If there are no time constraints competitions would be stuck in the voting phase forever.
References to system requirements: 5.14

4.1.6.4.2 **There shall be a voting page where users can vote on competitions that are in the voting phase.**
Rationale: To enable users to vote and determine who wins a competition.
References to system requirements: 5.16

4.1.6.4.3 **The weight of a vote shall be related to the total amount of AP that the voting user has.**
A vote from a user with lots of AP shall weigh more than a vote from a user with less AP.
Rationale: To let users with lots of AP be more influent in the outcomes of competitions.
References to system requirements: Appendix A

4.1.6.4.4 **Statistics showing the development of a vote should be available for presentation during a voting phase.**
These statistics are available for users that fulfill one of the following criteria:

- Has participated in that competition.
- Has voted on that competition.

Rationale: To enable users to follow the development of a vote as the vote phase of that competition continues. Also, people who vote will not be able to vote on the current leader (or the picture with lowest score) on purpose.
References to system requirements: 5.16

4.1.6.4.5 **Users shall receive AP by voting on competitions.**
Rationale: To encourage users to contribute to the community. Voting is essential for competitions.
References to system requirements: Appendix A

4.1.6.4.6 **A user shall only be able to vote in a competition once.**
Rationale: The outcome of the competition is likely to be unfair if a user is able to vote more than once on the competition. Without this constraint there are no limit in how much AP a user can receive by only voting.
References to system requirements: 5.15 5.16

4.1.6.4.7 **A user shall not be able to vote in a competition that he/she participates in.**
Rationale: Obviously, if you can vote for yourself, you lose the whole point of voting.
References to system requirements: 5.15 5.16

4.1.6.4.8 **The pictures in a competition shall be shown anonymously during the voting phase.**
Instead of showing which user that has created one of the two pictures.
Rationale: To avoid that pictures are judged based on other criteria than the actual pictures.
References to system requirements: 5.16

4.1.7 Other

4.1.7.1 **There shall be statistics that shows the ten users with the most AP.**
These statistics are available to all the users.
Rationale: To push users into competitive thinking. These statistics encourages this.
References to system requirements: -

4.1.7.2 **There shall be a global gallery where all pictures created in the system are shown.**
Rationale: The global gallery provides a way to find new pictures and shows what art has been created.
References to system requirements: 5.18

4.1.7.3 **A user shall be able to search for other users.**
They will be sought with respect to their username.
Rationale: To help users get in contact with each other.
References to system requirements: 5.17

4.2 Non-functional requirements

4.2.1 **The system shall be a client-server system.**
The web browser serves as the client.
Rationale: It is an online game and for the information to be accessed by other users, a common central storage is needed.
References to system requirements: -

4.2.2 **The client shall be web based.**
Rationale: Easy to use and no downloads of large executables needed.
References to system requirements: -

4.2.3 **The client shall work using the web browser Firefox version 2 with Flash 9 installed.**
Rationale: It is an online game so it makes sense that it should work on at least one web browser. Painting functions will be implemented with Flash.
References to system requirements: -

4.2.4 **The server shall work using Java Runtime Environment 5**
Rationale: The server will be written in the Java programming language, which is widely used and tested.-
References to system requirements:

4.2.5 **The pictures created in Battle mode shall be stored by the system.**
Rationale: Makes the use of the system much easier for the user.
References to system requirements: -

4.2.6 **The following information about a user shall be stored in the system: username, password, e-mail address.**
Rationale: A username to identify the user. A password to authenticate that user. An e-mail address to make it possible to contact the user outside the system.
References to system requirements: -

4.2.7 **The system shall not disclose the e-mail addresses to other users.**
Rationale: E-mail addresses are considered as personal information and shall be kept confidential.
References to system requirements: -

4.2.8 **Password and e-mail addresses shall be stored in an encrypted state.**
Rationale: To keep this sensitive personal data safe.
References to system requirements: -

4.2.9 **There shall be rules associated with the use of the system.**
These rules shall be agreed upon by every user while registering and followed thereafter.
Rationale: Users need to conform to certain behavior.
References to system requirements: -

5 System requirements

5.1 Create user

Function: Create a user (A).

Description: Register a new account in order to create a user that will have access to the system.

Inputs: Username, password and e-mail address.

Source: Input from the person trying to register.

Outputs: One of the following:

- Ok, if A was created.
- Not ok, if the registration failed and A therefore was not created.

Destination: A's client.

Action: Create a new entry in the database with the information provided by the input. This record represents a user of the system.

Requires: Username to be unique in the system.
The same password to be input twice.

Precondition: The person has accepted the rules of the community.

Postcondition: A is able to login to the system.

Side-effects: -

References to user requirements: 4.1.1.1

5.2 Login

Function: Login.

Description: Identify and authenticate a user.

Inputs: Username and password.

Source: Input by user (A).

Outputs: -

Destination: -

Action: Verify that the username-password pair exists in the database. If username-password pair does not exists the system will notify A that login failed. Otherwise A will get access to the system.

Requires: -

Precondition: -

Postcondition: A is logged in and has access to the system or at least A gets a message that login failed.

Side-effects: -

References to user requirements: 4.1.1.2

5.3 AP transfer

Function: Transfer AP.

Description: Interchange of AP between users.

Inputs: Amount of AP (x) to be transferred, From a user (A), To a user (B).

Source: Information from the outcome of a competition.

Outputs: -

Destination: -

Action: Decrease A's total amount of AP by x and increase B's total amount of AP by x.

Requires: -

Precondition: A and B has previously ended a competition where B was the winner.

Postcondition: A has reduced his/her total AP by x and B has increased his/her total AP by x.

Side-effects: -

References to user requirements: 4.1.6.1.3

5.4 Guestbook post

Function: Guestbook post.

Description: A user posts a message in a guestbook.

Inputs: Message (m), user (B).

Source: User (A)'s client.

Outputs: -

Destination: -

Action: Find B's guestbook and add m to it.

Requires: -

Precondition: A has written m.

Postcondition: m is saved in B's guestbook.

Side-effects: -

References to user requirements: 4.1.2.2

5.5 View guestbook

Function: Guestbook view.

Description: View messages in a user's guestbook.

Inputs: User (A), message id ($m1$), message id ($m2$).

Source: User (B)'s client.

Outputs: All messages in the range $[m1, m2]$.

Destination: B's client.

Action: System finds all messages in A's guestbook with an id (i) that is in the range $m1 <= i <= m2$. If there are no messages with an id in that range then nothing is returned.

Requires: -

Precondition: -

Postcondition: -

Side-effects: -

References to user requirements: 4.1.2.2

5.6 View personal gallery

Function: Personal gallery view.

Description: View pictures in a user's personal gallery.

Inputs: User (A), picture id ($p1$), picture id ($p2$).

Source: User (B)'s client.

Outputs: All pictures in the range $[p1, p2]$.

Destination: B's client.

Action: System finds all pictures in A's personal gallery with an id (i) that is in the range $p1 <= i <= p2$. If there are no messages with an id in that range then nothing is returned.

Requires: -

Precondition: -

Postcondition: -

Side-effects: -

References to user requirements: 4.1.2.3

5.7 Personal gallery entry

Function: Personal gallery entry.

Description: Add a picture to a user's personal gallery.

Inputs: Picture (p), user (A).

Source: A competition that A has participated in.

Outputs: -

Destination: -

Action: Finds A's personal gallery and adds p to it.

Requires: -

Precondition: A has painted p in a competition.

Postcondition: p is added to A's personal gallery.

Side-effects: -

References to user requirements: 4.1.2.3

5.8 View statistics

Function: Show statistics.

Description: System shows community statistics for a user.

Inputs: User (A).

Source: User (B)'s client.

Outputs: Number of competitions (c), number of won competitions (w), AP.

Destination: B's client.

Action: Finds c, w, and AP that belongs to A in the database and returns it to B's client.

Requires: A exists in the system.

Precondition: -

Postcondition: -

Side-effects: -

References to user requirements: 4.1.2.4

5.9 Playing arena

Function: Playing arena (PA).

Description: The PA shall be the place where users compete with each other. Users shall be able to challenge other users as well as receive challenges from other users. Once two parts have agreed to play, the actual battle shall start. Users willing to play shall be presented in such a way that a user looking for a competitor will find an opponent quickly.

When a user(U) first enters the PA, U shall not be able to interact with the PA. U should only be able to see what other users are signed up and see messages posted in the chat room.

References to user requirements: 4.1.3.1 4.1.3.2 4.1.3.3 4.1.3.4 4.1.3.7

5.10 Chat room

Function: Chat room in Playing arena (PA).

Description: Once a user enters the PA, that user should start receiving messages posted in the chat room. It should not be necessary to sign up in order to see the messages. However, to post messages a user should have to sign up in the PA.

References to user requirements: 4.1.3.5

5.11 Sign up in Playing arena

Function: Sign up in Playing arena (PA).

Description: In order to interact with the PA, a user must sign up in the PA. This way, all users willing to play will be gathered at one place, instead of having users aimlessly search for competitors among members that might not even be online at the same time.

Inputs: A user of the community (A).

Source: The set of users of the community looking for competitors that still has not signed up in the PA.

Outputs: -

Destination: -

Action: System marks A as signed up in the PA and provides A with the following. A should be able to view some statistics about other members in the PA. These statistics should be at least: Total AP, Wins and Losses.
A shall also be able to post messages in the chat room that shall be provided by the PA. Furthermore, A shall now be able to specify game options used when challenging other users, challenge other users, receive challenges from other users, accept/decline challenges and most of all, actually participate in battles.

Requires: -

Precondition: A is not already signed up in the PA.

Postcondition: A is signed up in the PA and A is able to interact with the PA.

Side-effects: -

References to user requirements: 4.1.3.3 4.1.3.6 4.1.3.7

5.12 Send a challenge invitation

Function: Send a challenge invitation.

Description: Sends a challenge invitation from one user to another user.

Inputs: time limit (t), AP to bet in percent (b), challenging user (A), opponent (B).

Source: Challenging user.

Outputs: A challenge invitation.

Destination: The opponent's client.

Action: The system calculates the challengers bet, that is b % of A's total AP. The system then calculates B's bet, that is b % of the opponents total AP. The system then creates a challenge invitation that consists of A's bet, B's bet, b and t. The invitation is then sent to B's client.

Requires: The two users A and B has signed up in the Playing arena.

Precondition: For each user U the following must be true: The difference between U's total AP and U's reserved AP is greater than U's bet.

Postcondition: B recieves a challenge invitation.

Side-effects: -

References to user requirements: 4.1.6.1.1 4.1.6.2.1 4.1.6.2.2 4.1.6.2.3

5.13 Answer a challenge invitation

Function: Answer a challenge invitation.

Description: A user (A) receives a challenge invitation. He/She can then either accept or decline the invitation.

Inputs: Challenge invitation, accept/decline message, text message (optional).

Source: The A's client.

Outputs: Accept/Decline message, optional text message.

Destination: The challengers client.

Action: The system redirects the answer to the challenger's client, if the answer is decline the system will include the text message if there is one.
If the answer is accept then the system will commence the game. The system will also mark the users' betted AP as reserved.

Requires: -

Precondition: The challenged user has received a challenge invitation from the challenger.

Postcondition: The challenger has received an answer.

Side-effects: -

References to user requirements: 4.1.6.1.1 4.1.6.2.2

5.14 Voting time exceeded

Function: Voting time exceeded.

Description: Check to see if there is a winner in the competition.

Inputs: Two competitors (A,B), voting points for each competitor (vp_A, vp_B), Betted amount of AP for each competitor (b_A,b_B).

Source: System.

Outputs: -

Destination: -

Action: If $vp_A > vp_A$ then A is declared as a winner and B's bet is transferred from $B \rightarrow A$. If $vp_B > vp_A$ then B is declared as a winner and A's bet is transferred from $A \rightarrow B$. If one of the conditions above are true then b_A is subtracted from A's reserved AP and b_B is subtracted from B's reserved AP. If $vp_A = vp_B$ then the competition is marked as a Golden vote competition.

Requires: -

Precondition: There is a competition between A and B that entered the voting phase at least 24h ago.

Postcondition: Either a winner is declared or the competition is marked as a Golden vote competition.

Side-effects: -

References to user requirements: 4.1.6.4.1

5.15 Voting

Function: Voting.

Description: A user puts a vote on one of the pictures in a competition.

Inputs: Picture (p), competition (c), user (A)

Source: The voting A's client.

Outputs: -

Destination: -

Action: The system calculates the points to be given to the picture on behalf of A's total AP. Adds the points previously calculated to p. The system then add a mark on c to know that A has voted in this competition. Finally the system adds 5 AP to A's total AP.

Requires: There is a competition in the voting phase.

Precondition: The voting user has not participated in this competition nor has he/she voted in the competition before.

Postcondition: p has increased it's score in the competition.

Side-effects: -

References to user requirements: 4.1.6.4.6 4.1.6.4.7 4.1.6.1.2 Appendix A

5.16 The voting page

Function: Voting page.

Description: The voting page is the place where users can see all competitions that are in the voting phase. Users can then vote for one of the pictures in a competition provided that the user has not voted in that competition before and that the user has not participated in the competition. When voting in a competition the voting user does not know which user created what picture. If the user has already voted in the competition or if the user participated in the competition then he/she is able to see the current standings in the competition.

References to user requirements: 4.1.6.4.2 4.1.6.4.4 4.1.6.4.6 4.1.6.4.7 4.1.6.4.8

5.17 Search user

Function: Search user.

Description: Finds other users by their user name.

Inputs: String of characters (s).

Source: User (A)'s client.

Outputs: List of usernames.

Destination: A's client.

Action: The system finds all users with a username that match s. That is, all usernames containing a substring anywhere that is equal to s. The search will not be case sensitive.

Requires: -

Precondition: -

Postcondition: A retrieves the search result.

Side-effects: -

References to user requirements: 4.1.7.3

5.18 Global gallery

Function: Global gallery.

Description: The global gallery is the place where all pictures ever made in a competition are posted. The system posts the pictures in the global gallery as soon as the competition where the pictures were made has finished. In the global gallery a user can see information about the pictures such as:

- Who created the picture.
- When the picture was created.
- What topic the creator were trying to illustrate.
- If the picture won the competition or not.

References to user requirements: 4.1.7.2

5.19 Paint

Function: Paint.

Description: The painting function will provide the user with three tools bucket, pencil, and eraser along with about 70 different colors. The user can then combine the bucket and pencil with any color to paint what they like. This function is used both in free sketch mode and in Battle mode. However the function will differ a little depending on which of the two modes it is used in. The differences are that in free sketch there are no associated time limit and no topic.

References to user requirements: 4.1.5.1

6 Use Cases

6.1 Register an account

Primary actor: User (*A*).

Stakeholders: *A* - Want to register.

Preconditions: The user has entered the system web page.

Postconditions (same as Success guarantee): A user account is created.

Minimal guarantee: System notifies *A* if the account was created or if any input need alteration.

Trigger: *A* starts to register.

Main scenario:

1 *A* enters his/her desired username, password and e-mail.

2 System validates the information.

3 System creates an account.

4 System notifies *A* that the account is created.

Alternative flow:

2 a. The username already exists.

1 *A* is notified of that.

2 Return to 1.

2 b. The password is invalid according to the system password policy.

1 *A* is notified of that.

2 Return to 1.

2 c. The e-mail is invalid.

1 *A* is notified of that.

2 Return to 1.

6.2 Login

Primary actor: User (*A*).

Stakeholders: *A* - Want to login.

Preconditions: *A* has previously created an account.

Postconditions: *A* gets access to the system.

Minimal guarantee: No user can login without a valid username and a corresponding password.

Trigger: *A* starts to login.

Main scenario:

1 *A* enters his/her username and password.

2 System validates the information.

3 System makes the functions that require authentication available to *A*.

Alternative flow:

2 a. There are no user with that username in the system.

1 System returns a message that the username/password combination is incorrect.

2 Return to item 1.

2 b. The password does not belong to the given user.

1 System returns a message that the username/password combination is incorrect.

2 Return to item 1.

6.3 Challenge another user

Primary actor: User (A).

Stakeholders:

A - Wants to challenge User (B).

B - Wants to be challenged.

Preconditions: A has logged in and B has signed up in the Playing arena.

Postconditions: B receives a challenge-request.

Minimal guarantee: -

Trigger: A enters the Playing arena.

Main scenario:

1 System presents the Playing arena and battle options.

2 System adds A to the playing list.

3 A sets up his/her desired battle options (see use case 6.11).

4 A tries to challenge B.

5 System sends the challenge request to B.

Alternative flow:

4 a. A views information about B.

1 System presents how much of B's AP that would be put on the line.

2 Return to 4.

5 a. B is already challenged by another user.

1 System notifies A that B is being challenged by another user.

2 Exit scenario.

5 b. B does not have enough AP that are not reserved.

1 System notifies A that B are unable to receive the request.

2 Return to 3.

6.4 Battle another user

Primary actor: User (*B*).

Stakeholders:

B - Wants to paint as good as possible.

User (*A*) - Wants to paint as good as possible.

Preconditions: *A* and *B* has signed up in the Playing arena.

Postconditions: The competition has entered the voting phase.

Minimal guarantee: -

Trigger: *B* receives a challenge-request from *A*.

Main scenario:

1 *B* accepts the challenge request from *A*.

2 System presents a painting page for *A* and *B* along with a topic.

3 System starts the timer on the specified time.

4 *A* and *B* paints until timer reaches zero.

5 System receives both the pictures.

6 System lets the users see each others pictures.

7 System moves the competition into the voting phase.

Alternative flow:

1 a. *B* does not accept *A*'s challenge.

1 *B* types an explanation about why he/she did not accept the challenge.

2 The system notifies *A* that *B* has rejected the challenge-request along with *B*'s explanation.

3 Exit scenario.

1 b. *A* cancel the challenge-request before *B* accepts it.

1 The system notifies *B* that *A* has canceled the request.

2 Exit scenario.

4 a. One of the users disconnect while painting.

1 The system declares the other part as winner.

2 Exit scenario.

4 a. Both of the users are disconnected while painting.

1 The competition ends without a winner.

2 Exit scenario.

6.5 Vote for a competition

Primary actor: User (*A*).

Stakeholders:

Participants - Wants to win the competition.

Other users - Want to vote to gain AP or because of interest.

Preconditions: There are at least one open competition in the voting phase.

Postconditions: One vote been added to one of the participants of the competition.

Minimal guarantee: The sum of votes are in a consistent state.

Trigger: *A* starts to vote.

Main scenario:

1 System presents a list of all current active competitions.

2 *A* selects one of the competitions.

3 System displays the two pictures created in the competition's battle phase.

4 *A* chooses to vote for the left picture.

5 System calculates the points awarded to the picture based on *A*'s total AP.

6 System adds the result in 6 to the left picture's score.

7 System adds five AP to *A* for voting.

8 *A* sees statistics of the current standing in the competition.

Alternative flow:

4 a. *A* decides not to vote for one of these pictures.

1 *A* selects another competition.

2 Return to 4.

4 b. The voting time went out while watching the pictures.

1 System notifies *A* that the voting is closed.

2 Return to 2.

5 a. The competition is marked as a Golden vote.

1 System declares the creator of the selected picture as winner and presents that user.

2 System adds five AP to A for voting.

3 Return to 2.

6.6 End competition

Primary actor: System.

Stakeholders: Users (A, B) (the participants) - Both want to win competion.

Preconditions: There currently is a competition going on.

Postconditions: One of the participants is declared as winner, the winner's AP has increased, the looser AP has decreased.

Minimal guarantee: The users AP are left in a consistent state.

Trigger: The time limit of a voting phase has exceeded.

Main scenario:

1 System compares the total voting points for the two pictures and declares a winner (A) and a looser (B).

2 System marks the competition as finished.

3 System transfer the appropriate amount of AP from B to A.

4 System adds one to A's number of won competitions and adds one to B's number of lost competitions.

Alternative flow:

1 a. The pictures voting points are equal.

1 System marks the competition as Golden vote.

2 Exit scenario.

6.7 End a Golden vote competition

Primary actor: System

Stakeholders: Users (A, B) (the participating) - Both want to win the competition.

Preconditions: A competition voting phase has ended and there is a tie.

Postconditions: The competition is marked as finished. One of the users is declared winner.

Minimal guarantee: One of the users is declared winner.

Trigger: A user votes on a Golden vote competition.

Main scenario:

1 System declares the creator of the voted picture as a winner (user A) and the other as a looser (user B).

2 System marks the competition as finished.

3 System transfer the appropriate amount of AP from B to A.

4 System adds one to A's number of won competitions and adds one to B's number of lost competitions.

6.8 Transfer AP

Primary actor: System

Stakeholders: Users (A, B) - Want to gain AP.

Preconditions:

There has been a competition and A won.

The bet was 30%. For A, this was 20 AP and for B, it was 15 AP.

Postconditions: 15 AP has been transferred from B to A. All AP reserved in this competition are freed.

Minimal guarantee: Users AP are left in a consistent state.

Trigger: The competition has ended and A is the winner.

Main scenario:

1 The system removes 15 AP from B.

2 The system reduces B's reserved AP by 15.

3 The system adds 15 AP to A's total AP.

4 The system reduces A's reserved AP by 20.

6.9 Post message in guestbook

Primary actor: User (A)

Stakeholders:

A - Wants to contact User (B) (owner of guestbook).

B - Wants other users to post messages in his/her guestbook.

Preconditions: A is logged in the system.

Postconditions: B's guestbook is augmented with a new message.

Minimal guarantee: B's guestbook is left in a consistent state.

Trigger: A enters B's profile

Main scenario:

1 System present B's profile to A.

2 A signs B's guestbook.

3 System posts the message which will be presented in B's guestbook.

6.10 Sketch in free sketch mode

Primary actor: User (A).

Stakeholders: A - Wants to simply sketch without entering a competition.

Preconditions: A is logged in.

Postconditions: A has tested some of the functionalities of the game.

Minimal guarantee: -

Trigger: free sketch mode is entered.

Main scenario:

1 Systems presents a canvas with some painting tools to A.

2 A tries out different tools to see what they do.

3 A leaves the page when he or she is done.

6.11 Sets up battle options

Primary actor: User (A).

Stakeholders: A - Wants to create a competition.

Preconditions: A has signed up in the Playing arena.

Postconditions: A is ready to challenge someone with the specified battle options.

Minimal guarantee: Battle options for a competition are set.

Trigger: A starts to create a competition.

Main scenario:

1 A selects the amount to bet in percent of total AP.

2 System translates the percentage amount into actual AP and presents it to A.

3 A selects the desired battle time limit.

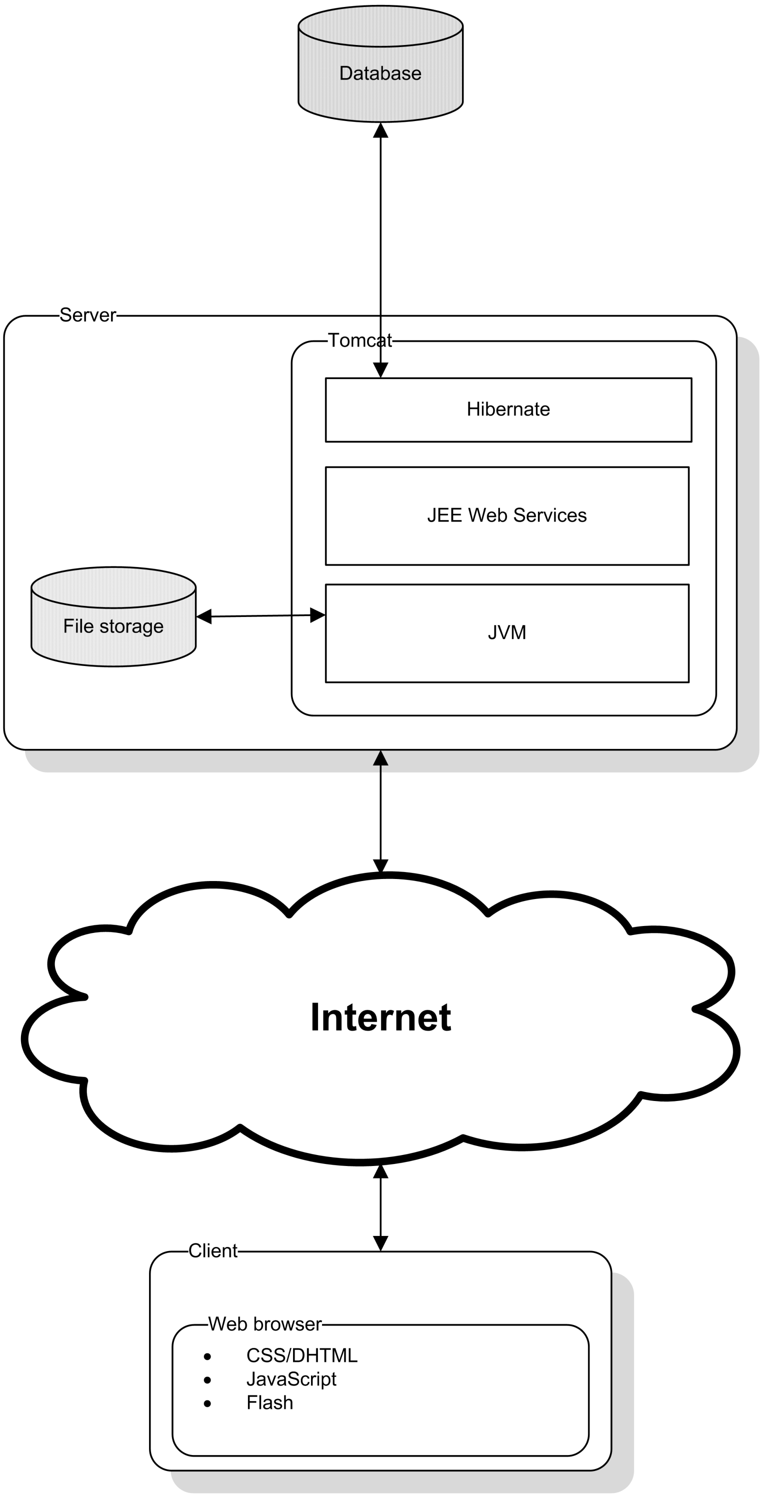
4 A moves on to the arena page with the selected options.

Alternative flow:

1 a. A wants to change the bet.

1 Return to step 1.

7 System architecture



7.1 Tomcat

Tomcat is a a web container that implements the JavaServlets and JavaServer Pages specifications from Sun Microsystems. Tomcat also includes an internal HTTP server. For more information visit http://tomcat.apache.org/.

7.1.1 Hibernate

Hibernate is a library for the Java programming language, providing a framework for mapping Java classes to a traditional relational database. We will use hibernate as an interface to the database.

7.1.2 JEE Web Services

JEE Web Services is an API for developing web applications in Java.

7.1.3 Java Virtual Machine (JVM)

JVM is an environment that understand Java code, a JVM is needed within the server to run the system.

7.2 Client

7.2.1 CSS/DHTML

Dynamic HTML is a collection of technologies used to create interactive and animated web sites. Cascading Syle sheets (CSS) is a language for specifying the presentation of a document written in a markup language such as HTML.

7.2.2 JavaScript

JavaScript is a scripting language that is most often used on dynamic web sites. The client downloads the script and executes it. We will among other things use JavaScripts to process and present data within the client.

7.2.3 Flash

Flash is mostly used for creating animations that can be viewed over the Internet. We will use Flash to develop the painting part of the system.

8 System evolution

8.1 Fundamental assumptions on which the system is built

The system is an online game and the reason for building one is based on the assumption that small online games where several users play against each other are quite popular these days. They serve as a break in people's everyday job, and are not played for long.

In order to have people come back and play our game more than once, we introduce a small community based side-track. Reasons for this have their roots in the fact that online communities are very popular these days, and people visits them regularly.

8.2 Anticipated changes due to hardware evolution

Of course, computers are always changing and if we find some new supreme technology to store large amounts of data in a convenient way we might want to upgrade our system to work with this technology. This will all be done on the server side of the system and should not affect the way in which users use the system.

8.3 Anticipated changes due to user needs

Since the system is an online game with a community based side-track, it is very probable that the system will change with time, as the user needs change very rapidly. Therefore, we will consider each user requirement in turn when anticipating system evolution.

Abbreviation: User requirement = UR.

Regarding UR 4.1.1.1
Not likely to change.

Regarding UR 4.1.1.2
This requirement means that a user needs to login every time that user wishes to use the system in any way. So, if a user just wishes to see who currently are at the top of the rankings, this could get annoying. In an evolutionary viewpoint we find this likely to change in a way to let a user actually use some of the system functionality without having to login, such as view rankings.

Regarding UR 4.1.2.1
That users have AP is not very likely to change. However, as we have specified things at the moment, a new user will be given 100 AP to begin with. This might change if, for example, we discover that users find it easier to register a new account when low on AP instead of battleing their way up.

Regarding UR 4.1.2.2
If the system has been up and running for a long time, there might be a lot of old guestbook messages that are being stored and takes a lot of space. This is very likely to happen. We might have to come up with a better way to store messages, simply remove them when they become "too old" or perhaps set a limit on how many messages will be saved by the system.

Regarding UR 4.1.2.3
Pretty much the same as with UR 8.3, except it is about pictures now.

Regarding UR 4.1.2.4
This requirement is mainly concerned with statistics shown on the users personal pages. More statistics could be added if users express interest in it but otherwise this is not likely to change.

Regarding UR 4.1.3.1 - 4.1.3.4
We do not expect these to change.

Regarding UR 4.1.3.5
Well we do not really expect the main chat room to change, however, we might include the possibility to start private conversations if the user base grows large. If there is a lot of users posting messages in the main chat it will be quite cumbersome to see which messages are directed to a specific user.

Regarding UR 4.1.3.6
At the time of writing we plan to have statistics about wins, losses and total amount of AP. It is possible that we include more statistics or exclude some of the current ones, but we will probably keep the amount of statistics at around 3 items.

Regarding UR 4.1.3.7
Not likely to change.

Regarding UR 4.1.4.1 - 4.1.4.2
Additional colors and tools could definitively be added or removed as time moves on, users would probably be a bit bored if the actual gameplay never changed.

Regarding UR 4.1.5.1
If we change what tools are provided in Battle mode, these changes will be reflected here - this is, after all, the practice mode.

Regarding UR 4.1.6.1.1
If this were to change we would probably have to redo most of the system, so this is not likely to change.

Regarding UR 4.1.6.1.2 - 4.1.6.1.3
Not likely to change.

Regarding UR 4.1.6.2.1
To keep things simple when creating a game we have only 2 options. Unless there is a change in UR 4.1.6.1.1 this is not very likely to change.

Regarding UR 4.1.6.2.2
Not likely to change.

Regarding UR 4.1.6.2.3
If a lot of users express their anger over how the betting system works we might change it into a more "classic" form where a user can bet how many AP he/she likes.

Regarding UR 4.1.6.2.4
Not likely to change.

Regarding UR 4.1.6.3.1
We could have let the users choose topic as a battle option, and if a lot of users demand this, there is a strong possibility we change it.

Regarding UR 4.1.6.4.1
This is very likely to change. When we start the system up, there will be very few users and we need longer times in the voting phase to gather enough votes so we can get a somewhat fair result. If there are a lot of users, a competition will probably gather more votes in a shorter time. When that happens we will probably shorten the voting time or at least present the competitors with an option of how long the voting phase will be.

Regarding UR 4.1.6.4.2
Not likely to change.

Regarding UR 4.1.6.4.3
The reason for this UR is that a user with more AP shall have more authority. If a lot of users find this unfair or if we feel that this is not as good as we intended, this might change.

Regarding UR 4.1.6.4.4
Not likely to change.

Regarding UR 4.1.6.4.5
One obvious question is how much AP a user receive when voting on a competition. This will probably change as the user base grows. If there are very few users there will be few competitions and since this is the only way for a user who has lost a lot of his/her AP to gain more AP, the member should get more AP for each vote in order to get on his/her feet.
If there is a lot of competitions on the other hand, a member should get less AP so that the member do not get a massive advantage by just voting.

Regarding UR 4.1.6.4.6 - 4.1.7.1
Not likely to change.

Regarding UR 4.1.7.2
The global gallery is absolutely subject to change since the number of pictures in the system will grow rapidly, so new ways of presenting the images will be needed.

Regarding UR 4.1.7.3 - 4.2.2
Not likely to change.

Regarding UR 4.2.3
Technology is always moving forward and it is very probable that users update their web browsers to the latest versions. The system should of course join the evolution and conform to the advances in web browser technology. Also, users use different web browsers than just Firefox, and for the system to hold, we should make sure it work on different web browsers.

Regarding UR 4.2.4
This is not really something that will affect the users of the system directly.

Regarding UR 4.2.5
The amount of pictures will rise rapidly so what is likely to happen is that we run out of storage. When this happens we will have to increase the storage capacity, change how we store the pictures into a more compact form or start removing pictures.

Regarding UR 4.2.6
Eventually, users will (hopefully) want to change password. The system should provide facilities for doing so in the future.

Regarding UR 4.2.7 - 4.2.8
Not likely to change.

Regarding UR 4.2.9
Users might find loopholes in the rules of the system and therefore the rules are subject to change into more constrained rules.

A The system of Artistic Points

This appendix describes how the system of Artistic Points (AP) and the related voting system shall work. AP are the currency in the system, the more of those the better. It is the intention that every user will strive to receive as much AP as possible by winning competitions and by voting on competitions that the user is not participating in.

A.1 Starting AP

When a user account is created that user will initially receive 100 AP. Users can then win and loose AP by competing and voting. A user's total AP can never go below zero. A user with zero AP is not able to bet any AP and therefore he/she ca not loose any.

A.2 Competition

When a user participates in a competition the user jeopardize some of his/her AP. The system will reserve the amount of AP that a user bet during a competition. This means that the reserved amount of AP can not be used in future competitions unless it is released, but it will still be counted in the users total AP. The rest of the user's AP will be referred to as available AP. This ensures that a user is able to pay up in case of loosing a competition, the user will at least have that amount of AP. When a competition ends, the amount of AP that were reserved for the competition will become available to new competitions.

A.3 Betting

A user can challenge another user. When doing so, the user defines the rules of the game. One of these rules is the total AP that the competitors are going to bet. The challenging user will set an integer value between 1 and 100 that represents the percentage of total AP that the user will bet (Note: total AP, not available AP). The opponent will then bet the same percentage of his/her total AP which means that the actual amount of AP that each of the two players bet are different (If their AP are not equal). A user's bet will always be an integer value, if it turns out not to be an exact integer value it will be rounded to the closest integer. This procedure is illustrated in the example below.

Example: User A (with a total of 100 AP) challenge User B (with a total of 202 AP) and chooses to bet 20%. This corresponds to A's bet being 20 AP and B's bet being 40 AP (20% of 100 = 20 and 20% of 202 = 40.4 respectively).

This means that users with more AP takes a higher risk than users with less AP. This is motivated by the assumption that a user with a lot of AP probably are quite good at this game and any other user that manage to beat him/her should therefore gain a little extra.

A user can not bet more than he/she can afford i.e. the bet exceeds the user's available amount of AP. This follows from the discussion of reserved AP above. This rule does (of course) also apply to the opponent, the opponent can not accept a bet that exceeds his/her available AP.

Example: User A (with a total of 100 AP of which 20 are reserved) challenge User B (with a total of 200 AP of which 125 are reserved) and chooses to bet 50%. The challenge is then rejected because B is not able to accept it.

A.4 Voting

A user will receive AP by voting on other competitions. If a user's total AP is down to zero this is the only way to receive some AP. Every user that votes for a competition will receive 5 AP. This means that users will be willing to contribute to the system by voting because it will pay off. But at the same time he/she will not be able to build a fortune on just voting because of the restriction that a user only can vote for the same competition once.

When a user votes for a picture in a competition that picture will get votes depending on the voting user's total AP. The entire set of users in the system are partitioned into four distinct subsets on behalf of their total AP. How the users are partitioned into subset is described next.

Find the user with the highest total AP and let M be that user's total AP, then make three limits L_1, L_2, L_3 that is 25% of M, 50% of M and 75% of M respectively. Let A denote the total amount of AP for an arbitrary user. The groups are then

G_1 - Users with between 0 and L_1 (limits included)

G_2 - Users with A between $L_1 + 1$ and L_2 (limits included)

G_3 - Users with A between $L_2 + 1$ and L_3 (limits included)

G_4 - Users with A between $L_3 + 1$ and M (limits included)

A vote from a user in group G_1, G_2, G_3, G_4 counts as 1,2,3 and 4 votes respectively.

Index