

Empires of Avatharia

Group 22

Christopher Raschke

Johan Wessman

Jashar Ghavampour

Johan Granberg

1. Functional Requirements

1.1 – Registering

When the user first comes in contact “with Empires of Avatharia” he/she will probably need to register him- or herself as a new user on the “Empires of Avatharia” webpage;

The registering form will take less than 5 minutes to fill in. The registering form requires the following input from the user;

- His or her full name
- Username
- A unique email-address
- Desired password
- In-game choices;
 - In-game faction
 - In-game start location

The password must be at least 5 characters long and has to consist of at least one number. It can consist of any of the characters A-Z and/or any of the numbers 0-9. It can be no longer than 20 characters. The password has to be filled in twice to make sure that the user doesn't mistype his password.

The email-address must be valid. It is validated through an email that is sent to the user after the form has been filled in.

The user can choose to side with any of the 3 in-game factions. Depending on his or her choice he will then select an in-game starting location. The browser will display an in-game-map so that this choice could be done easily by clicking on the desired location of the map. The location is represented both graphically and by coordinates X and Y. Because of the rules of the game some starting locations will be invalid (depending on the user's previous choice of faction etc). If the starting location that the user selected is invalid then he or she will be given alternatives that are located nearby his desired starting location. The user can then choose one of these alternatives or try a new spot on the map.

If the user should provide invalid input in any of the required fields then he or she is alerted through a message in his or her webbrowser. The message should contain information of what field/s that contained the invalid input.

After the form has been filled in the user is sent back to the root of the “Empires of Avatharia” webpage.

1.2 Logging in

In order to be able to play the game or access any other of the “Empires of Avatharia” material, the user has to identify him- or herself by “logging in”;

The “login”-screen can be found in the root of the “Empires of Avatharia” webpage. You are required to fill in two fields in order to log in; username and password. You will also have the alternative to recover your password. The login should take less than 10 seconds in 90 % of the cases to process. The password and username are compared with the (hopefully) corresponding fields in the database. If the username and password are correct then the user is sent forward to the in-game environment.

1.2.1 Recover password

If the user clicks on the “recover password” link located near the “login screen” then the system will require the user to input his username. The system will then check if the password has been recovered earlier this day, in which case the request is discarded. This precaution is taken to prevent spam/abuse. If it hasn’t been recovered this day then an email is sent, to the appropriate user, containing the password. The user is then returned to the login-screen.

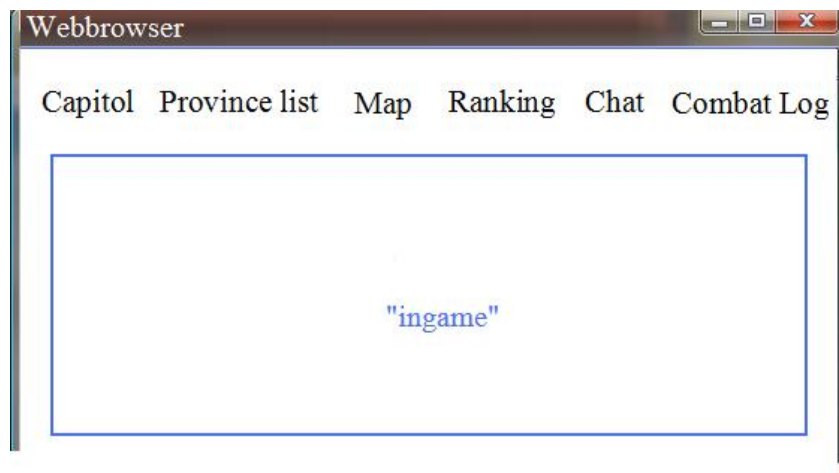
1.3 The ingame-environment

The ingame environment is centred around a main window that is centred in the webbrowser. Everything that has to do with the actual game takes place in this window.

The user may choose to change his survey of the game by clicking on one of the many tabs. When a tab is selected, the main window changes to a window corresponding to the chosen tab. The tabs include;

- Capitol
- Province list
- Map
- Ranking
- Chat
- Combat Log

(These tabs are explained in detail later on). This window is by default set to the capitol. Here is a rough example of how the windows and tabs may be orientated:



1.3.1 Capitol

When the user has processed the login he is then sent to an ingame environment. This environment is set to show the users “Capitol” by default. Much of the game takes place on this screen and much of the strategic decision making is done here;

A capitol has some important properties:

- They cannot be “captured” by other players (see Fighting).

- The only province that you start.

A capitol is also a province which makes it necessary to explain what a province is.

1.3.1.1 Province

A province represents an online town. A province contains buildings and units (which will be explained later on). A province also contains (some) of the players resources that are measured in “Gold” and “Mana”.

The Gold and Mana attributes are basically two separate integers which will increase every hour depending on the player’s province setup. Players use resources to build buildings or to train units. The buildings and the units that are currently garrisoned in the town are represented with a graphical interface. The user can interact with the buildings and the units by clicking on them.

An essential part of each province is the buildings it contains which we will give a detailed explanation to:

1.3.1.2 Buildings

Buildings appear in provinces and have different properties and attributes that affect the game. (One of the common attributes is their “lvl”).

A building with a high “lvl” provides more benefits to a player than one with a low “lvl”. A building starts at “lvl 1” but can be upgraded using the towncenter to a maximum of “lvl 100”. These upgrades cost resources. These upgrade costs increase as the “lvl” of the building increases. An upgrade isn’t done instantly but requires time. It takes much more time to upgrade an already high “lvl” building than it takes to upgrade a low “lvl” one. A user may choose to speed up the required build time by spending more resources on the building. The buildings that you can build in a province include:

- 5 different buildings that enables the training of specific units.
- The “towncentre”
- The “magetower” which generates mana that increases with the “lvl” of the building.

A province (which includes the capitol) has one building with some very special properties called the “Towncentre”.

1.3.1.3 Towncentre

This building has some special properties which make it necessary to explain in detail. The properties include:

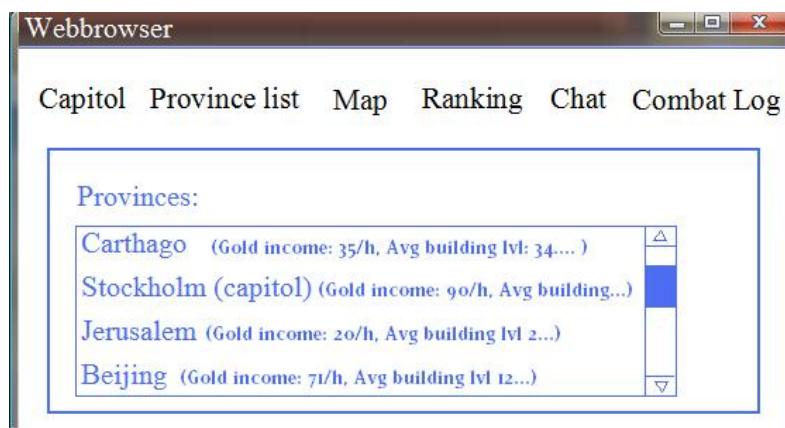
- You can build buildings here by selecting one from a menu and clicking on a “build” button. The build function can also “upgrade” a building. This is essentially overwriting it with a better building. When the user has selected the building he wants to build (or upgrade) and clicks on the “build button”

the system subtracts the required resources from the user's town's total resources.

- The only building that you don't have to build. You start with a "lvl 1" "towncentre" in each province.
- You can train units here by selecting the unit you want to train from a menu and clicking a "train unit" button. This option is not available if your town cannot "support" more units or if you don't have enough resources to train them. This subtracts the required resources from your town's total resources.
- You can change your "army setup" (which will be explained in details later)
- Your gold income is based on the "lvl" (short for "level") of this building (this is explained later)

1.3.2 Province list

When the user clicks on the "Province list"-tab, in his or her web browser, his main window is changed to a list, containing all his provinces. If the list is too long to fit in the window then scrolling is enabled. After each province name is a short line of information about that province. This information includes generated gold income and the average "lvl" of all the buildings in that province. Here is an example of how the layout may look when a user has clicked on the "Province list".

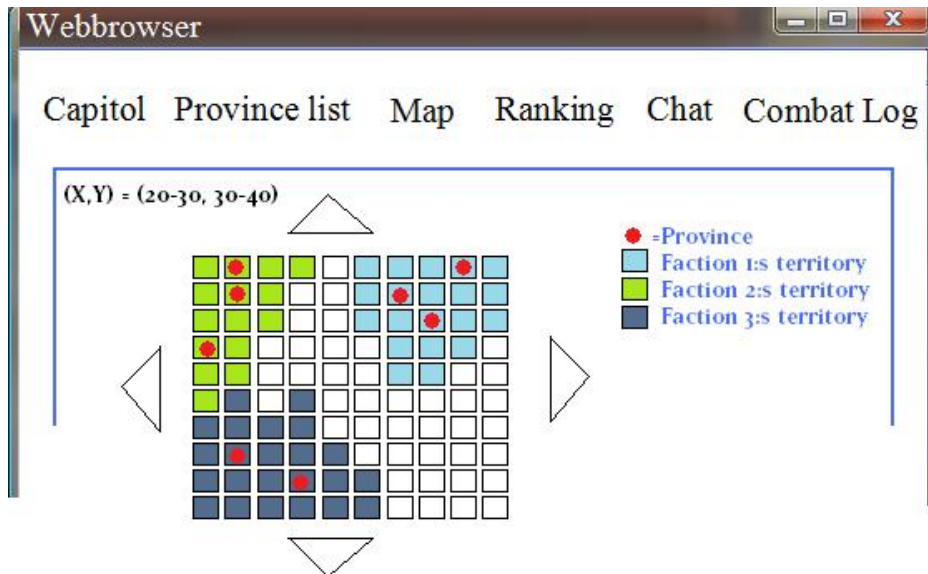


1.3.2 Map

The map is a very important part of the game since it represents the overall goal of the game: to capture territory for your own faction.

Each player is a member of exactly one faction. When the player has selected a faction (see registering) he cannot change it. The map contains of a 100*100 square grid. Each square has 2 properties; it can contain a town, and it belongs to a faction *or* no faction.

When the user selects the "Map" tab in his browser his view (main window) changes to a map. The map is a 10*10 subgrid of the real 100*100 map. This map may look something like this:



In this view the user can hold his or her mouse pointer over a spot on the map to get information about it. This information includes the owner of a province and the faction that it belongs to. The user can scroll by using the arrows (see the picture). This moves the current view by changing the coordinates of the top left square. The user can also view more details of a square containing a province by clicking on it. These details will include:

- The military strength of the province. Measured as Low, Medium or High. This is calculated relative the overall game standard.
- The average level of all the buildings in the province.
- The amount of resources in the province. . Measured as Low, Medium or High. This is calculated relative the overall game standard.

When the user clicks on a “square”, containing a province, you will also be given the option to send a “private message” to this player. Only the player which owns that province is able to see this message. That player may also choose to reply to that message. These messages can be viewed by clicking the Chat tab (more details of this later).

The map changes when a province is captured by an enemy faction player (details of this later) or when someone builds a new province (details later).

1.3.3 Ranking

The ranking is a way to represent how each player is doing compared to other players. Each player will be given a ranking according to his or her ingame-process. The ranking includes:

- **Total income from all provinces:** The sum of the income per hour from all the provinces that belongs to that player.
- **Average building lvl:** The average “lvl” of all the buildings that belongs to that player.
- **Military stats:** A measurement of how strong that player’s total army is. (see army for more details)
- **Total amount of provinces.**
- **Overall:** A combined score of all the previous points. Basically a weighted sum of all the above.

- **Ranking:** Shows which player that has the highest overall score by descending order. 1 is highest and 2 is second highest etc.
- **Name:** The name of the player

This score is represented by a scoreboard that is updated every day. The ranking scoreboard can be viewed by the user at any time by clicking on the “Ranking” tab in the ingame-environment.

That scoreboard may look something like this:

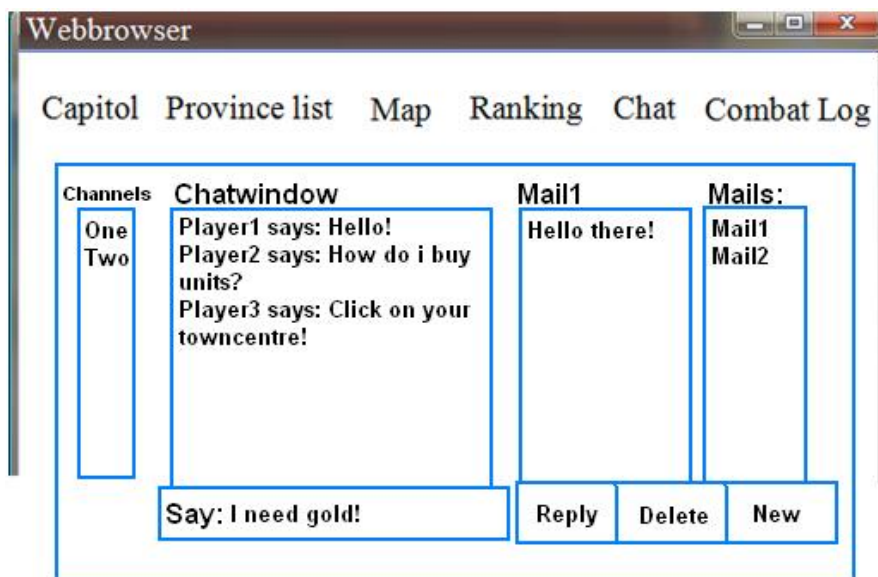
Ranking	Overall	Name	Military	Provinces	Build. lvl	Income
1	67890	Olle	3289	7	45	670/h
2	329	Per	789	2	4	20/h

1.3.4 Chat

When the user clicks on the Chat-tab in the ingame environment the main window changes to a new window. This window contains a mailbox and an instant chat window. In this window the user can:

- Send private messages (ingame mail) to another user.
- Read private messages.
- Manage his inbox by deleting or sorting his ingame mail.
- Reply to any ingame mail
- Chat with other players.

The chat contains different channels where users can communicate. A user can change the channel that he is using by selecting one from the “channels list”. The user chats by writing a string in the appropriate field (see picture) and pressing “ENTER”. When a message is sent using the chat all recipients receive the message within 5 seconds in 90% of the cases. When a mail (private message) is sent the receiving user should be notified and should receive the message within 10 minutes in 95% of the cases.

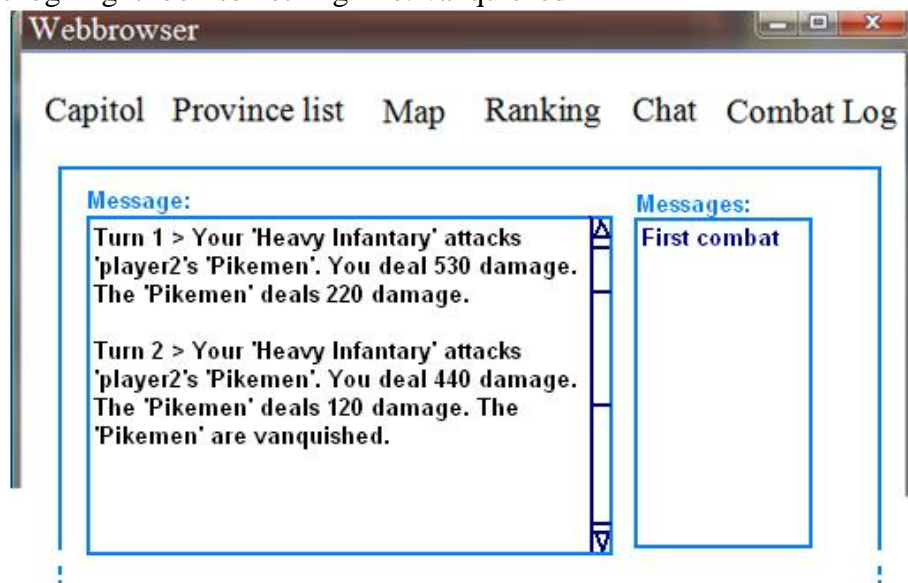


Scrolling is enabled if any of the field should be too small to contain all the information. A player can only chat in one channel at the same time.

You cannot type more than 1 message each other second. If a message is sent within 2 seconds of a previous it is discarded.

1.3.5 Combat log

When the user presses the “Combat log”-tab his main window changes to a mailbox-like environment. This window will contain a list and a message field. When the user clicks on an item in the list it is displayed in the message field. The list contains all “combat logs” from all the combats that the players units have fought as well as other important ingame information. The combat log might look something like: vanquished



In order to explain the combat stage we first need to explain what a “unit” is:

1.3.6 Units

A unit represents a part of an army. A unit has some very important attributes:

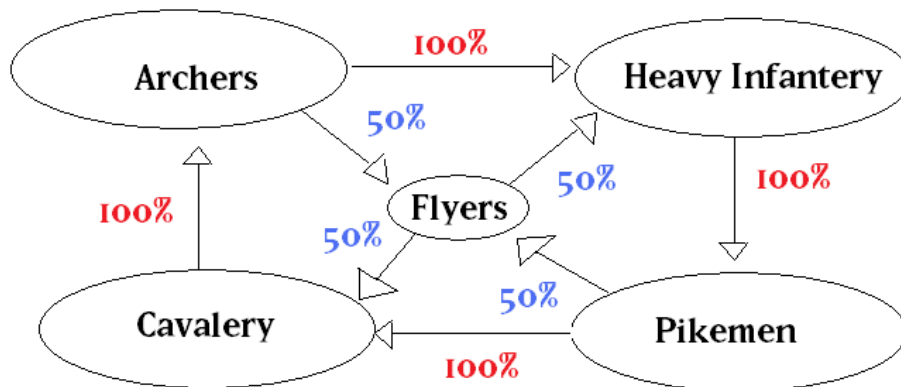
- **Attack:** Modifier when calculating damage during combat (see combat).
- **Defence:** Modifier when calculating damage dealt to your unit by your opponent’s unit.
- **HP:** Modifier for damage. If this modifier reaches zero the unit is considered vanquished (see combat).
- **Max HP:** The maximum amount of HP that the unit can achieve.
- **Speed:** How far a unit can travel every hour.
- **Level:** What level the current unit is at.
- **Experience:** How much experience the unit needs to reach the next level.

A unit starts at “Level 1” and increases his level by gaining experience. A unit can gain experience by damaging enemy units or by practicing in the “Towncenter” or “Magetower” building (see buildings). When a unit increases his level some of his attributes increase. These attributes are; Attack, Defence, Max HP, Speed and Level. The maximum level of a unit is

100. A unit can also have any of the following “traits” that increase each 5th level. These traits include:

- Critical Strike: Increases the maximum damage of the unit.
- Swift Blade: The unit deals damage to an enemy unit before it has a chance to deal damage to it.
- Twin Blades: The unit deals both “Swift Blade” damage and normal damage.
- Berserk: The damage that is done is calculated by the max-hp.
- Trample: If the unit vanquishes an enemy unit, the remaining damage is dealt to another unit in the enemy army.

There are also 5 “types” of units; Archers, Heavy Infantry, Pikemen and Cavalry. These units have specific bonuses against one another. These bonuses are represented in the picture below.



As you can see Cavalry has a 100% damage bonus against Archers, however Archers has a 100% damage bonus against Heavy Infantry and so on. Archers and Pikemen seem to have more bonuses than the other unit types, however these units move slower on the map than other units.

1.3.7 Combat

A unit can be sent into combat only if it is a part of an army. Each province can support 2 armies. Each province starts with two empty armies. Units can be moved between armies freely. A unit can be deleted from the army, in which case the user will be asked to confirm this action. If the user presses the “confirm” button then the unit is deleted from the army and also from the game. If the user clicks on the “abort” button the delete-request is discarded. One army is assigned to defend the province against enemy forces until defeated. When the defending army is defeated, or if there is no defending army, the “offensive army” will start defending the province. A defending army cannot be sent to attack another player's province. A player may send units to defend a province that belongs to a player of the same faction. An army has room for 9 units. When an army contains at least 1 unit and is positioned inside one of your provinces it can be sent to war against another player's province. Before sent in to war they must be given precisely one of the two missions:

1. **Pillage:** Steal some of the opponent's resources. The user specifies how many turns the army should fight before they try to retrieve with some resources.

2. **Conquer:** The province is taken from the defending player. This province is considered a new province for the attacking player but some of the buildings will remain at the same level.

1.3.7.1 War

A war takes place when an attacking army reaches an enemy province.

Whenever an army reaches a war he gets a timestamp. This timestamp represents his queue in the line to succeed in missions. During a war there will always be two sides: The defending side and the attacking side. The attacking side contains of all attacking armies (can contain of units from different players) and the defending of all defending (can contain units from different players) and also a guardian. The guardian is considered a unit during the war. It has all the attributes that a unit has. It can be killed during a war but it will (unless the province is taken) be resurrected after each war. This way the defending province will always have a defending unit. The guardians strength depends on which mission the attacking player has. The guardian also grows in strength if the defending player has a high "lvl" "Towncentre". If the attacking player has the intention to "pillage" then a relatively weak guardian is summoned. If the mission was to "conquer" the province then a relatively strong guardian is summoned. *(These guardians are made to prevent abuse and harassing of new players)*

The combat will be turnbased. A pillage mission ends after an amount of turns specified by the attacking player. A conquer mission will continue until the attacking army/ies is defeated or until every defending army is defeated including it's guardian.

The army with the lowest timestamp will (if he had chosen the conquer mission and is not vanquished to the last unit) conquer the province.

If many armies have chosen to pillage (and have won the war and have units remaining in the army) the army with the lowest timestamp will pillage first then the one with the second lowest timestamp and so on.

When all the missions have been carried out the units return to the provinces that they belong to unless they managed to conquer the province in which case they remain in the province as a defending army.

2. Non-functional Requirements

There are several different types of non-functional requirements that our system must meet, though to different degrees. To make it clear we will for each type describe how and to what degree our system must meet the requirement.

2.1 Performance Requirements.

To begin with we will talk about the *performance requirement*. Our system will not make performance an issue at the user end, and therefore there is no real requirement on how much the system will use the user's computers resources.

On the other hand our system must meet a high performance requirement when handling the communication between users and server. The system must be efficient when sending information between user and server so there is no apparent delay while playing.

Having a cap on the maximum number of users that can be connected at a given time will also help the system be more efficient.

2.2 Space Requirements.

Another type of requirement that is interesting is the *space requirement*. Since the game is played through a web browser the user doesn't have to install anything and there is no real requirement on the space that the system must meet for the user.

The system will run on a server computer where the game and information database will be stored. This is no real requirement either though.

We are confident that any modern computer has the space required to store the game and database.

2.3 Reliability Requirements.

The third requirement type is *reliability*. Our system must have high reliability; it is highly undesirable to experience server crashes. The server computer needs to be reliable and our program must be written in a good way, and also tested so that crashes are avoided.

If a crash does occur, the system must stay consistent and information about what the users have done saved. This will most probably be solved by the system using a database with logs and backup as the only means to saving information, no files that must be rewritten etc.

2.4 Portability Requirements.

The next requirement is *portability* and where the system is to be used. The requirement in our case is that the game must be able to be played on most web browsers, but only on computers.

The game is not supported on other platforms than computers, even though it may work on a mobile phone with a web browser and the like.

2.5 Usability Requirements.

Usability is another type of requirement that we want the game to meet to some extent.

The game must be easy-to-use and almost like navigating any usual web page. This means that there will be no advanced interfaces and the like, which will make the game easy to play, even for people with internet-usage only and casual gamers.

2.6 Privacy Requirements.

Privacy is also something we want to include in the system. The privacy requirement is, for our system, that a user's information is not shown to the public. This includes information like the user's real name and e-mail address.

2.7 Standards Requirements.

The last requirement which our project involves is a *standards requirement* throughout the development of the system. All of the documentation and written work must be saved as .doc , or standard word format so to speak. We have chosen this format to make sure that everyone in the group can open and read/change documents sent around before handed in as pdf-versions.

3. Use Cases

All use cases are written using the model presented by Alistair Cockburn¹. These are all on the following form:

Primary Actor

Goal

Eventflow

Extensions (*optional*)

Xn. Problem

Xn1. Solution

Variations (*optional*)

3.1. Registering

Primary actor: The user

Goal: Make a registration to the game.

1. The user enters the game website.
2. The user proceeds by going to the registration page.
3. User enters the account information that he/she wants.
4. The system sends a confirmation mail to the specified mail address.
5. The user then receives a confirmation mail that the registration was successful.

Extensions:

3a. One of the submitted fields have incorrect values entered.

3a1. The user receives a notification about what field was wrongfully entered.

3a2. The user supplies the information again but correct.

3b. The username and/or e-mail address already exists in the database.

3b1. The user is informed that the username and/or e-mail address is already linked to another account.

3b2. The user enters a new username and/or e-mail address.

3.2. Login

Primary actor: The user

Goal: To log in into the game.

1. The user enters the game website.
2. The user proceeds by pressing the login button on the main page.
3. The user enters his/hers username and password and presses login.
4. The system verifies that the password entered is the real one corresponding to the username specified.
5. The user is logged in and can play.

¹ http://alistair.cockburn.us/index.php/Structuring_use_cases_with_goals

Extensions:

- 3a. The username includes incorrect characters.
 - 3a1. The website notifies the user about the username including an illegal character.
 - 3a2. The user enters the username and password again, hopefully correct this time.
 - 3a3. The system verifies that the password entered is the real one corresponding to the username specified.
 - 3a4. The user is logged in and can play.
- 3b. The username entered is not associated with any account.
 - 3b1. The system notifies the user that an account with the specified username doesn't exist.
 - 3b2. The user tries to re-enter the username.
- 4a. The password entered was not correct.
 - 4a1. The user receives a notification that the password was incorrect and is prompted to enter the correct one.
 - 4a2. The user enters the password again and presses login.

3.3. Build a building

Primary actor: Any registered player.

Goal: To start building a building.

1. The use case starts when the player has logged in. (*See the "login" use case for more information*)
2. The system will display the ingame-environment.
3. The user will select the "Province list"-tab.
4. The system will change the "main view" to the "Province list"-tab.
5. The user will select a province from the list.
6. The system will change the "main view" to the corresponding "province".
7. The user will click on the "towncentre" building.
8. The system will change the "main view" to the "towncentre" and present a list with all the buildings that the user is able to build in this province.
9. The user will select one of these buildings and then click on the "buy"-button.
10. The system will process the request for buying this building.
11. The use case ends.

Extensions:

- 3-9a. The user might select another tab.
 - 3-9a1. The use case continues at 4 whenever the user selects the "province list"-tab or at 6 if the user selects the "Capitol"-tab.
- 7a. The user may click on any other building.
 - 7a1. The use case continues at 8 whenever the user selects the "towncentre".
- 9a. The user might have insufficient resources to build the selected building.
 - 9a1. The building is not built.

Variations:

- 3. The user may choose to click on the "Capitol"-tab in which case the use case continues at 6.
- 5. The province can be any of the provinces in the game that the player owns.
- 8. The buildings can also be "upgrades".

9. Possible buildings are:
- a) Magetower
 - b) Any building that enables units (5 total).
 - c) Upgrades on the towncentre.

3.4. Build a unit

Primary actor: Any registered player.

Goal: To start building a Unit

1. The use case starts when the player has logged in. (*See the “login” use case for more information*)
2. The system will display the ingame-environment.
3. The user will select the “Province list”-tab.
4. The system will change the “main view” to the “Province list”-tab.
5. The user will select a province from the list.
6. The system will change the “main view” to the corresponding “province”.
7. The user will click on the “towncentre” building.
8. The system will change the “main view” to the “towncentre” and present a list with all the units that the user is able to build in this province.
9. The user will select one of these units and then click on the “buy”-button.
10. The system will process the request for buying the unit.
11. The use case ends.

Extensions:

3-9a. The user might select another tab.

3-9a1. The use case continues at 4 whenever the user selects the “province list”-tab or at 6 if the user selects the “Capitol”-tab.

7a. The user may click on any other building.

7a1. The use case continues at 8 whenever the user selects the “towncentre”.

9a. The user might have insufficient resources to build the selected unit.

9a1. The unit is not built.

Variations:

3. The user may choose to click on the “Capitol”-tab in which case the use case continues at 6.

5. The province can be any of the provinces in the game that the player owns.

9. Possible units are special cases of:

- a) Archers
- b) Cavalery
- c) Pikemen
- d) Heavy infantry
- e) Flying

3.5. War

Primary actor: Any registered player.

Goal: To send an army to war.

1. The use case starts when the player has logged in. (*See the “login” use case for more information*)
2. The system will display the ingame-environment.
3. The system will change the “main view” to the “Province list”-tab.
4. The user will select a province from the list.
5. The system will change the “main view” to the corresponding “province”.
6. The user will select his “offensive army” by clicking on it.
7. The user will then click on the “send to war button”.
8. The system will change the “main view” to the map.
9. The user will select an enemy province.
10. The system will display a dialogue box with 3 options; Pillage, Attack, Abort.
11. The user selects one of these options.
12. The system will send the selected army to war.
13. The use case ends.

Extensions:

- 3-9a. The user might select another tab.
- 3-9a1. The use case continues at 4 whenever the user selects the “province list”-tab or at 6 if the user selects the “Capitol”-tab.
- 6a. The offensive army might be empty.
- 6a1. The army cannot be sent to war.
- 9a. The user might select an allies province or click on an empty square.
- 9a1. The operation is aborted.
- 10a. The user may choose to abort.
- 10a1. The use case resets to 5.
- 10b. The province that the player has chosen in 9 might be an enemy capitol.
- 10b1. The Attack option is not available.

Variations:

- 3. The user may choose to click on the “Capitol”-tab in which case the use case continues at 6.
- 5. The province can be any of the provinces in the game that the player owns.
- 9. The province can be any other province that belongs to a player of another faction.

3.6. Private Message

Primary Actor: Any registered player.

Goal: Write a private message.

1. The use case starts when the player has logged in. (*See the “login” use case for more information*)
2. The user presses the “Chat” tab.
3. The program displays the Chat window.
4. The user presses the “new message” button.
5. The program shows an empty mail.
6. The user inputs receiver name and a text message to the receiver.
7. The user presses send.

8. The system sends the private message to the receiver.
9. Usecase ends

Extensions:

1. See login.
- 2-7a. The user might select another tab.
- 2-7a1. The use case continues at 3 whenever the user selects the “Chat”-tab
- 7a. No receiver name.
- 7a1. Nothing happens.
- 7b. No text message.
- 7b1. Nothing happens.

Variations:

6. Receiver is
 - a. a player in your faction.
 - b. a player in an enemy faction.
 - c. a server admin.

3.7. Chat message

Primary Actor: Any registered player.

Goal: Write a chat message.

1. The use case starts when the player has logged in. *(See the “login” use case for more information)*
2. The user presses the “Chat” tab.
3. The program displays the Chat window.
4. The user chooses a channel by selecting a channelname from the a list.
5. The program displays the channels chat window.
6. The user types a message.
7. The user sends the message by pressing “ENTER”.
8. The program sends the message to the people in the channel.

Extensions:

1. See login.
- 2-7a. The user might select another tab.
- 2-7a1. The use case continues at 3 whenever the user selects the “Chat”-tab
- 7b. The user has already sent a message within the last two seconds.
- 7b1. The send is ignore. The user must wait an additional second before sending a new message.

Variations:

4. Channelname is
 - a String.
 - a Number.

3.8. Check “Ranking”

Primary actor: Any registered player

Goal: To view the total ranking

1. The use case starts when the player has logged in. (*See the “login” use case for more information*).
2. The system will display the in-game-environment.
3. The user will select the “Ranking” tab.
4. The system will change the “main view” to the “ranking view”.
5. The user will see a table with player aliases along with some numbers that indicate the player’s current scores in various categories, with a set maximum number of rows per page.
6. The user will click on the “next page” button and gets a list of lower ranked players, all displayed in a descending order.
7. The use case ends.

Variations:

6. The user might go to a certain ranking page without clicking the “next page” by clicking on a number in a list of the “nearest” page numbers.

3.9. View Map

Primary actor: Any registered player

Goal: To view the game map

1. The use case starts when the player has logged in. (*See the “login” use case for more information*).
2. The system will display the in-game-environment.
3. The user will select the “Map” tab.
4. The system will change the “Main view” to the “Map” tab.
5. A grid of 10x10 colored squares will be displayed, where each square represents a section of territory that can be possessed by a faction. The user clicks on one of the four arrows that enable scrolling in the selected direction to another adjacent set of squares (all representing land).
6. The map will shift in the selected direction, and new squares will be displayed.
7. The user hovers the mouse cursor over a square and information about the contents of the square are displayed in a side panel near the grid.
8. The user clicks one of the squares and more details are displayed in the side panel.
9. The use case ends.

Exceptions

- 5a The user may scroll to the “end” of the grid (as it is a total of 100x100 squares)
- 5a1. Further scrolling in the selected direction is disabled.