1 Conceptual Modelling

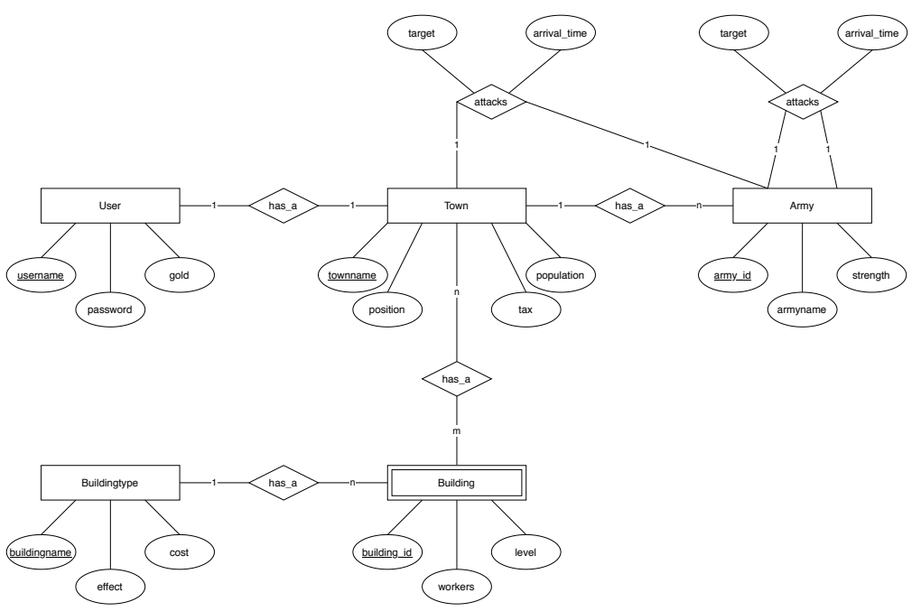
1.1 Description

Savagery is a multiplayer online game, which can be played on any modern web browser. It is a very simple strategy game in which the player has to manage a small town. He can construct new buildings, upgrade existing buildings, distribute his workforce across buildings and recruit armies to attack other players towns.

Buildings and soldiers are payed with gold and every town has a fixed population. The main objective of the player is to find a cost efficient mix between constructing new buildings and upgrading existing buildings to gain various bonuses.

The player can also recruit armies to attack other towns. These armies have a name and a strength, which determines their manpower. Certain buildings in the players town also influence the offensive and defensive values of the players armies. To attack other towns, the army must march a certain time, dependent on the position of the start and target town. During the march, the army is a valid target for other armies and can be intercepted before it reaches its destination. If the attacking army reaches its target, the result of the battle is calculated and, in case of victory, spoils of war, in form of gold, are taken from the attacked town. After that both armies return to their towns of origin.

1.2 ER-Diagram



2 Use-Case Design

2.1 Registration

2.1.1 Description

Objective:

Create an account for a new user.

Description:

A user can fill in a username and password and hit the ”Register Account”-button to save his login data to the database.

Preconditions:

-User not logged in

-User in ”Login”-screen

Expected Execution:

-User fills in data

-Data is validated

-Data is saved to database

-User is logged in and redirected

Postconditions on Success:

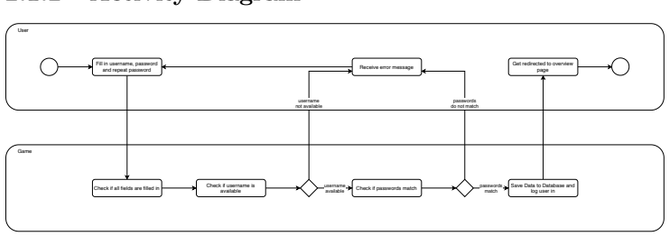
-Userdata saved to database

-User logged in

Postconditions on Error:

-Errormessage displayed

2.1.2 Activity Diagram



2.2 Manage Buildings (Main Use-Case)

2.2.1 Description

Objective:

Build or upgrade buildings or redistribute workers.

Description:

A user can manage his town by adding or leveling up buildings for a gold cost and optimize his workforce by changing their workplaces as needed.

Preconditions:

-User logged in

-User in "Building"-screen

Expected Execution:

-User selects whether to build something, upgrade something or manage the workforce

-Resources (gold or workforce) are checked

-The desired change is written to the database

-The page is refreshed for the user to see the change

Postconditions on Success:

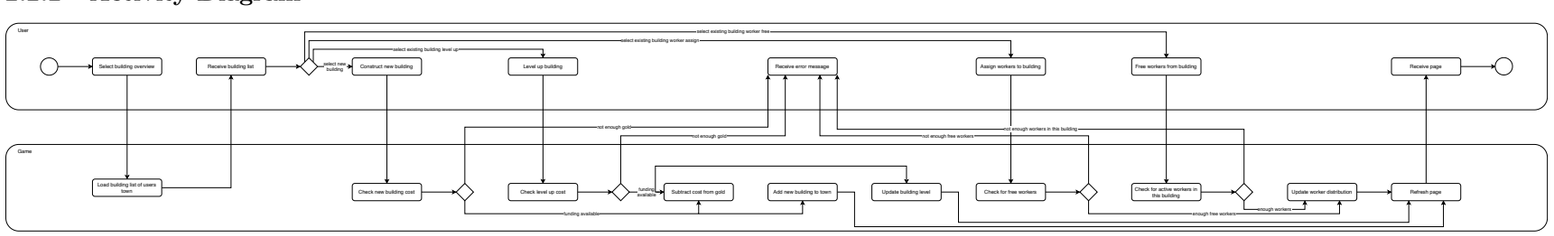
-Changes saved to database

-Page refreshed

Postconditions on Error:

-Errormessage displayed

2.2.2 Activity Diagram



2.3 Manage Armies

2.3.1 Description

Objective:

Create new armies or split or merge existing armies.

Description:

A user can manage his armies by adding new armies with a specified strength for a gold cost or change the counts and strengths of his existing armies by splitting or merging them.

Preconditions:

-User logged in

-User in ”Army”-screen

Expected Execution:

-User selects whether to create a new army or split or merge an existing army

-Resources (gold or army strength) are checked

-The desired change is written to the database

-The page is refreshed for the user to see the change

Postconditions on Success:

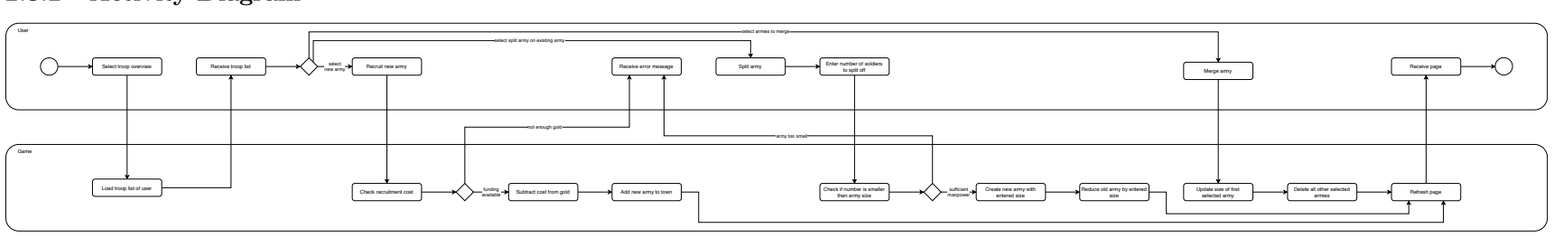
-Changes saved to database

-Page refreshed

Postconditions on Error:

-Errormessage displayed

2.3.2 Activity Diagram



2.4 Plan Attack

2.4.1 Description

Objective:

Attack another players town or marching army.

Description:

A user can attack targets like other towns or marching armies in his vicinity with one of his armies.

Preconditions:

-User logged in

-User in "Attack"-screen

Expected Execution:

-User selects a target from the list

-User selects one of his armies to carry out the order

-The arrival time is calculated

-Target and arrival time are written to database

-The page is refreshed for the user to see the change

Postconditions on Success:

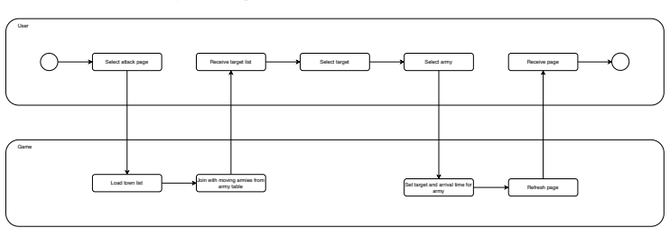
-Target and arrival time for corresponding army is saved in database

-Page refreshed

Postconditions on Error:

-Errormessage displayed

2.4.2 Activity Diagram



2.5 Attack Results (Elaborate Use-Case)

2.5.1 Description

Objective:

Display the results of a fight.

Description:

After a fight occurred, the results are calculated and both the attacking and defending players get notified.

Preconditions:

-User logged in

-Time of arrival of attacking army passed

Expected Execution:

-Fight is triggered after time of arrival passed

-Necessary data is loaded from database

-Casualties and spoils are calculated

-Army strengths are updated

-Army targets and time of arrivals are reset

-Report is displayed to the user

Postconditions on Success:

-Target and arrival time for corresponding armies are unset

-Army strengths are updated

-Spoils of war go to the winner

-Report is generated and shown to the player

Postconditions on Error:

-Death and Decay (probably written to system log, since no user input is involved)

2.5.2 Activity Diagram