

# Strategy Game

Project Architecture

**CEBP**

## **ASSISTANT**

Luisa Grigorescu

## **TEAM**

Eliza-Maria Ciovică

Sarah Cîrpaci

Sebastian Petruc

Paula Ruști

# 1. SPECIFICATIONS

The Strategy Game will aim to be a new version of the well-known Catan Game.

It will be implemented as a distributed system, with multiple users, as players who are concurrently sending requests to a server.

The goal of this game is to reach 10 points, so the players will compete to gather resources to build the biggest settlement, as houses and cities are points. The resources are limited, consisting of brick, wood, sheep, grain and stone.

- One resource of brick plus one wood, one grain and one sheep make a house



- Two grains and three stones produce one city (2p) – but only if you have a house.



- One sheep, one grain and one stone produce a wanted resource to be used again.



- When is a players' turn, he can exchange resources with the other players, as needed.

The game takes place in a virtual common space, as it will be established a connection between the players and the server, through the console. Considering the command send by the player to the server and the resources the players possess, the server will parse the command and send back a response with some results or an error.

The server will communicate with a SQL database which will hold all the users and their number of resources (the game will begin with various users, an equal number of resources for each user, and some extra amount besides).

The users will be able to work with the following commands, the game having the following functional requirements:

- /login username

This will authenticate the current user in the system.

- `/list-all-resources extra_resources`

This will return the amount of extra resources available for all the players.

- `/list-player-resources resources_amount`

This will return the number of resources available for the current player.

- `/build-house player_wood player_brick player_grain player_sheep`

This will subtract one specified resource from the players number of resources and build him a house, gaining one point.

- `/build-city player_grain player_stone`

This will subtract two grains and three stones from the players number of resources and build him a city, gaining two points.

- `/get_new_resource player_sheep player_grain player_stone`

This will subtract one specified resource from the players number of resources and give him a new wanted resource.

- `/exchange-resource resource`

This will exchange one players resource with one of the other players.

- If the username is not valid or if the number of resources is less than required, or if the combination of resources is wrong, the server will send back an error.

For the event part, we will use Apache Kafka, so users could be able to send, save and request data whenever they need it. Through Kafka we're going to manage our data amount moving from one endpoint to another, ensuring all the data is delivered.

Non-functional requirements:

Due to the subject of this course, our game will consider the concurrency and capacity, so the service will be able to handle multiple computations executing simultaneously, for a various number of concurrent users, seen as threads.

The system will be reliable, while also having a good time performance, either just for the users' request or for the server' messages, varying from: 300ms to 600ms, depending if there is an error, the game having a 99.7% uptime.

Our SQL database will ensure the storage efficiency.

## 2. Architectural Design

