Architecture

The main purpose of this project is to create a Massive Multiplayer Online game inspired by Pokemon.

Technical and Conceptual Problems :

To implement the graphical user interface, our game will has a problem of compatibility with multiple devices (Smartphone , PC , Tablet).
We also have to choose between many types of maps (Matrix, 2D maps, 3D maps, Google maps).
The second problem is to choose the server architecture, because we have to choose a different architecture depending on how many players are expected.
Assuming a smaller scale, we can use a simpler design for the backend architecture, without a need for load balancing between multiple backend servers.

Concepts :

Throughout this project, we decided to use a MVC “Model View Controller” design, this solution will help us to separate the backend logic from the frontend that represent the program. In fact, this is an important design pattern which will allow us to separate between the database, the display, and the game logic.

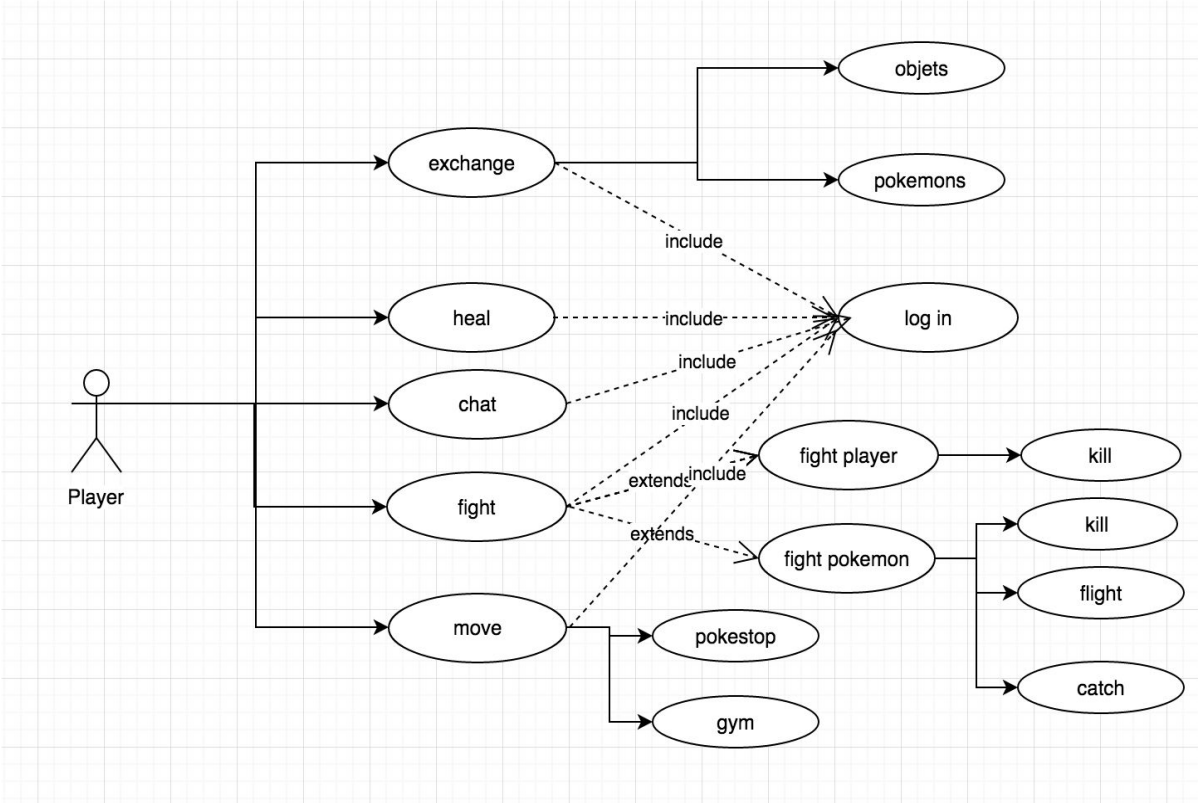
Possible extensions :

Currently, the only considered extension is the use of Google Maps as a map provider at the end of our project. In the specifications we talked about the exchange of objects and pokémons between players will also be a possibly extension.

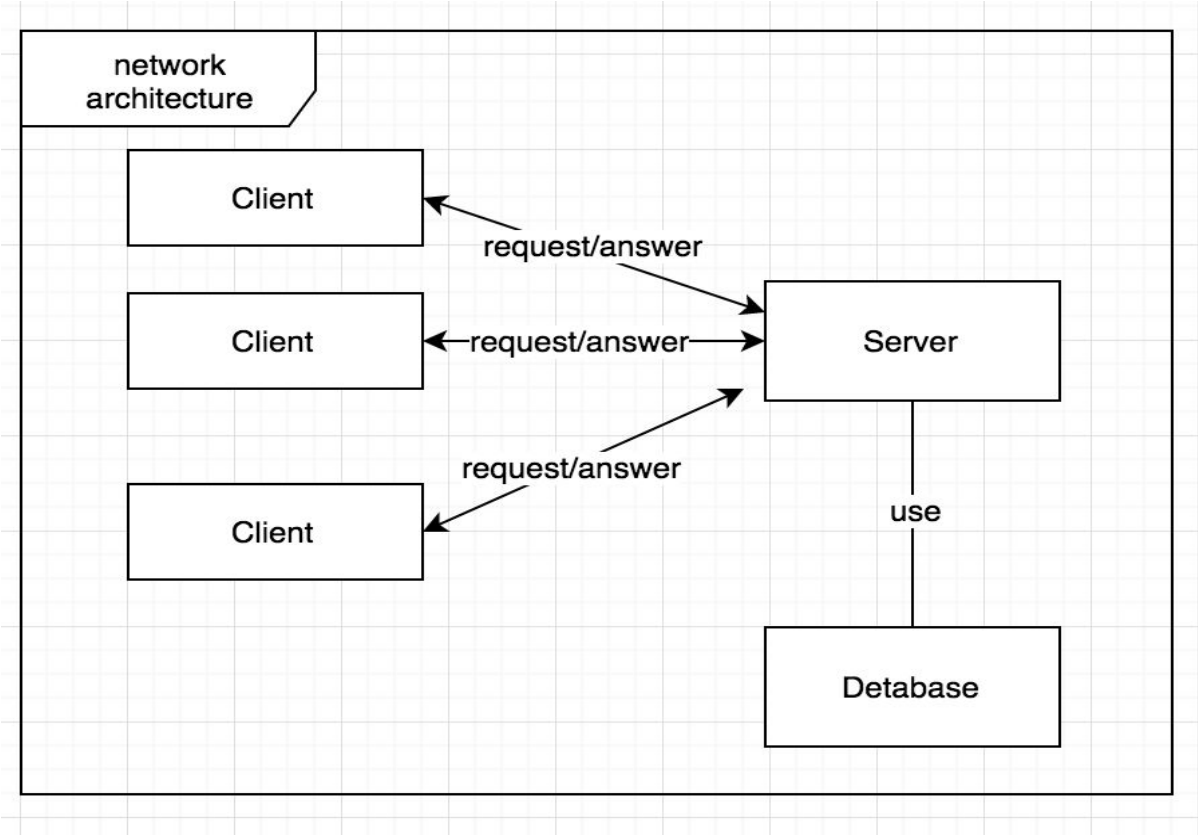
Diagrams :

Use case :

This diagram describes the interactions between the player and the whole game.

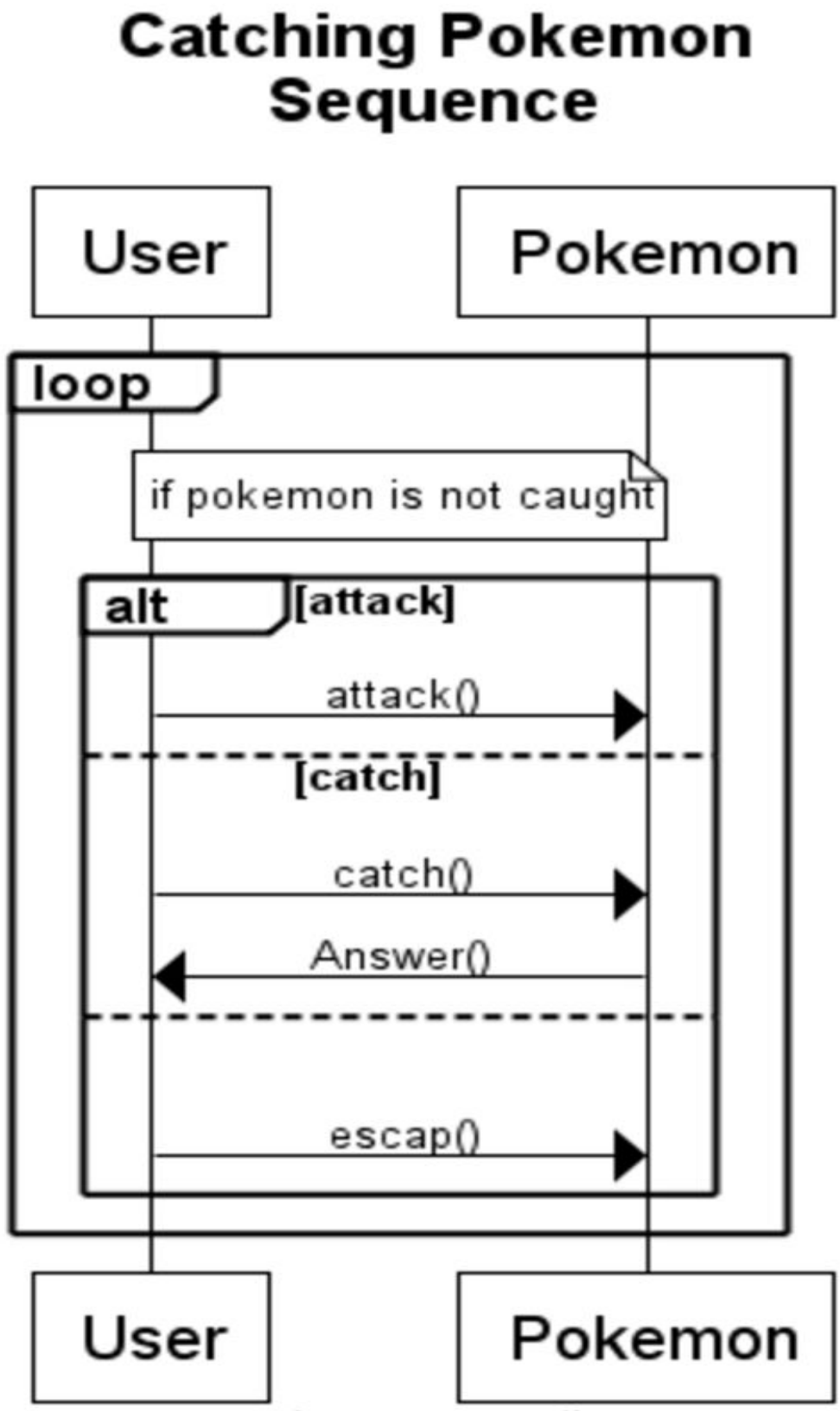


The network architecture :



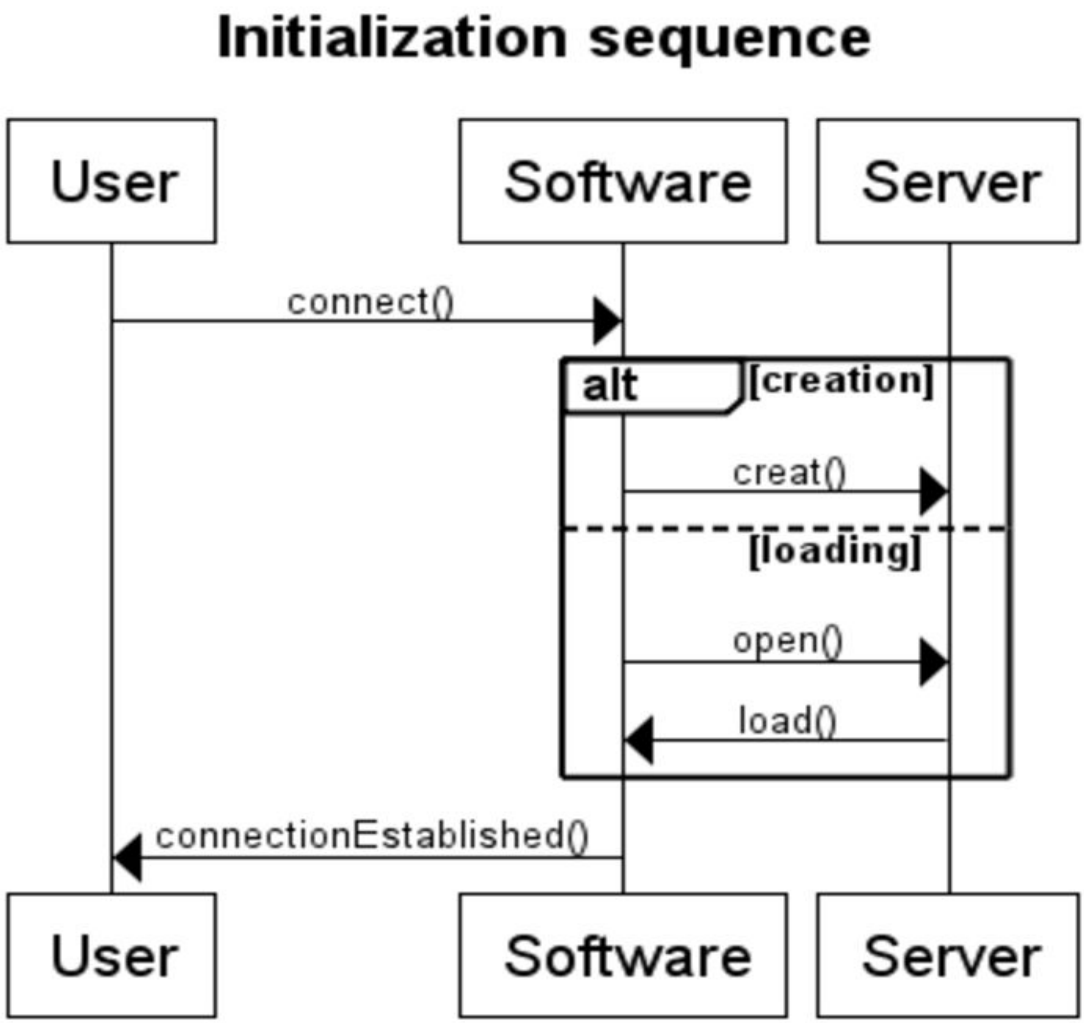
This one describe the network part of our game.

Catching sequence :



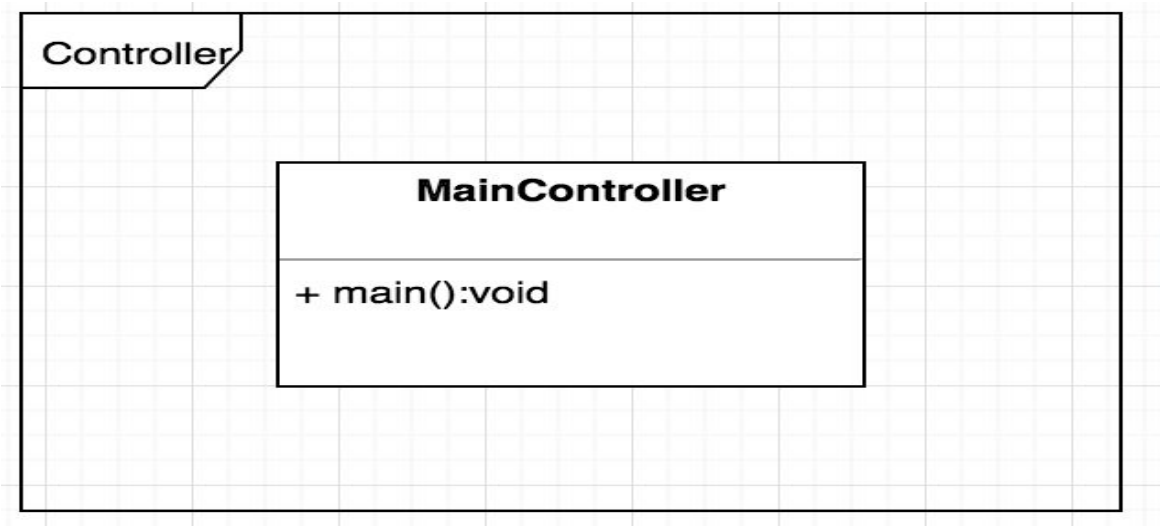
This diagram shows how to catch a pokemon, and as we can see the pokemon can escape after an attack.

The initialization sequence :

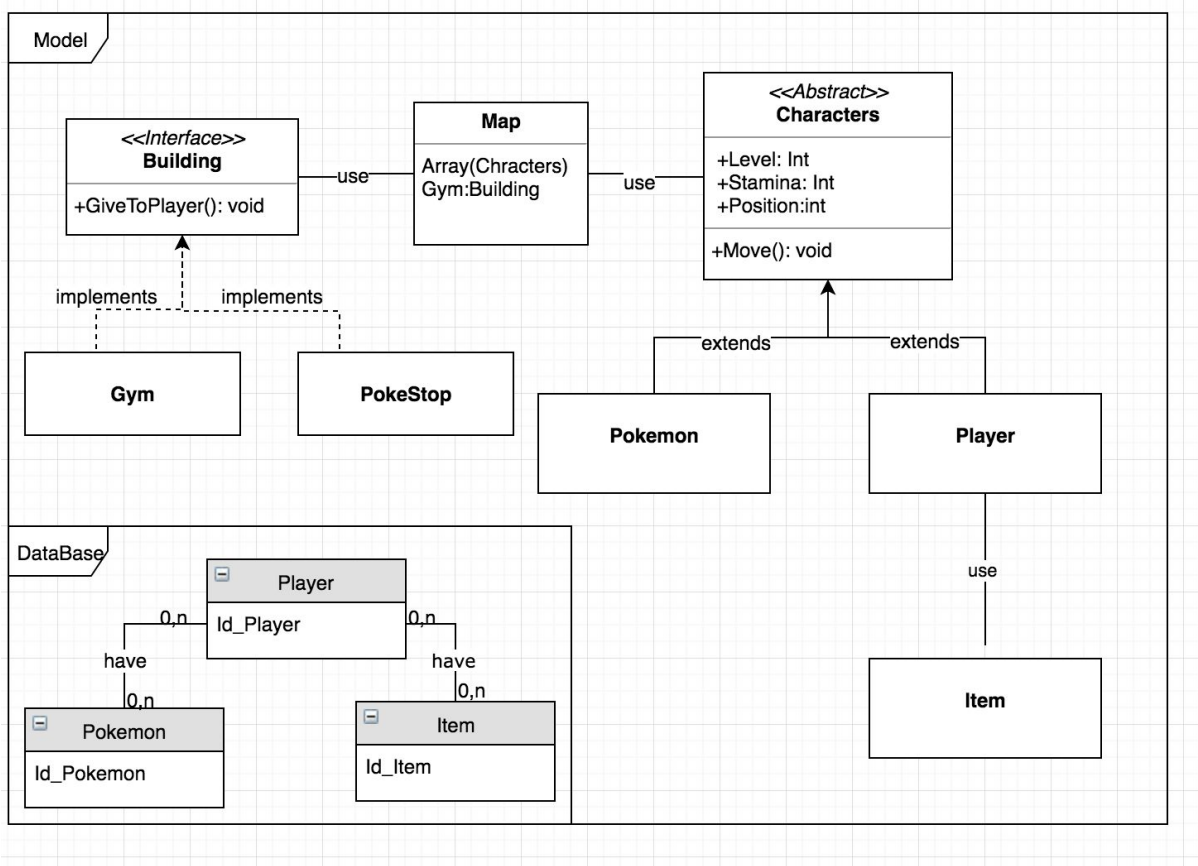


The purpose of this one is the creation of an account in the first connection, and then the connection to our account every time we wanted to play.

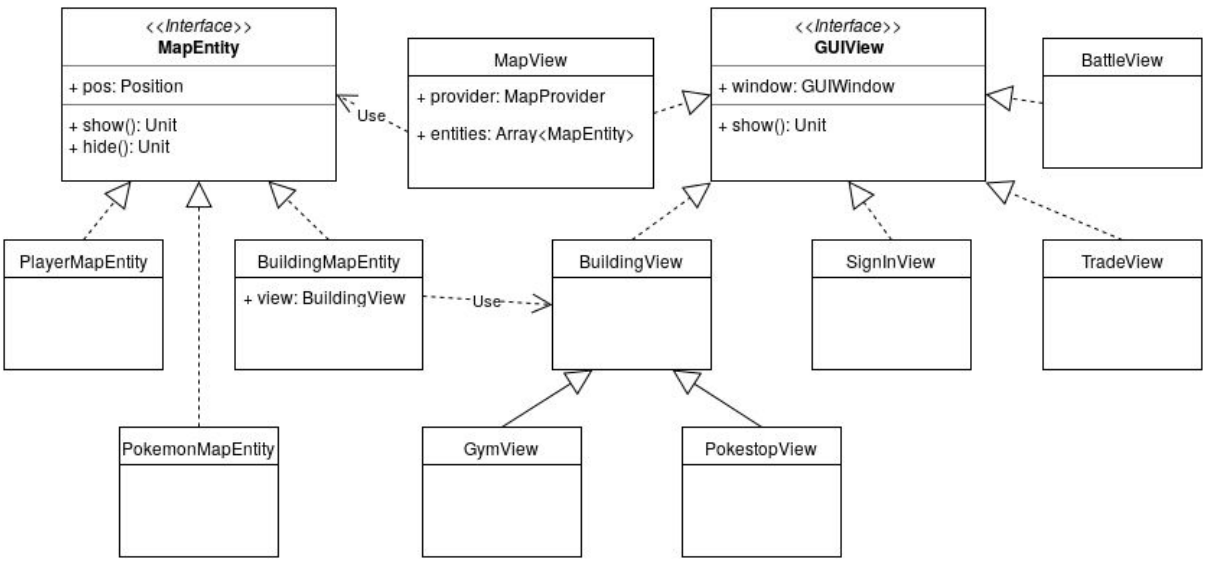
The controller diagram :



The model diagram :



The view diagram :



The general view components of our game will be represented as shown previously.