Diagrams Game Store Project

I.ER + Table structure diagram
Gantt diagram

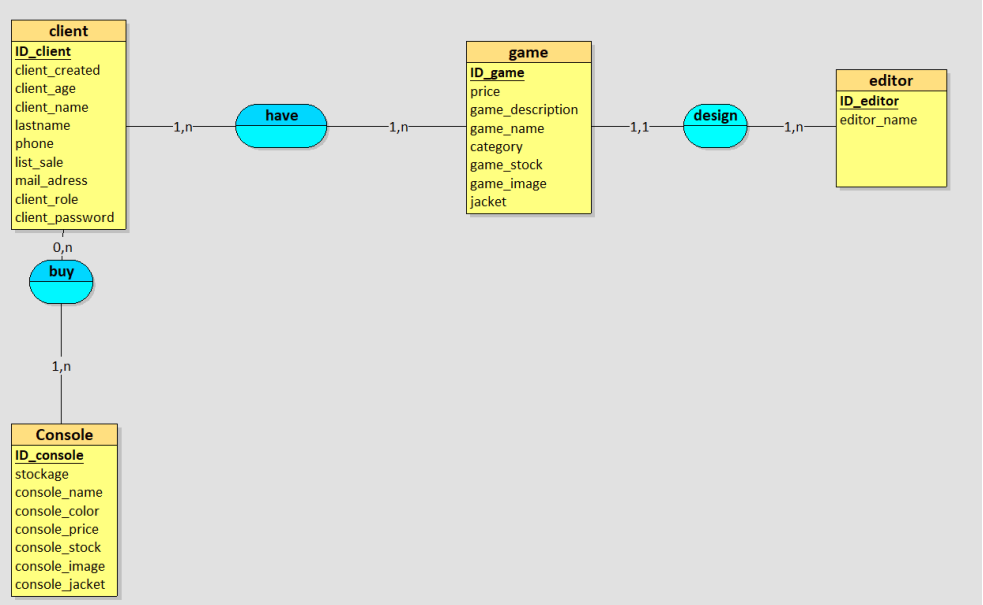
II.For each person of the group :
1) Use Case Diagram
2) Activity Diagram
3) Wireframe Diagram
4) Component Diagram
5) Sequence Diagram

III.Final diagrams to conclude the project
1) Gantt diagram
2) Class diagram

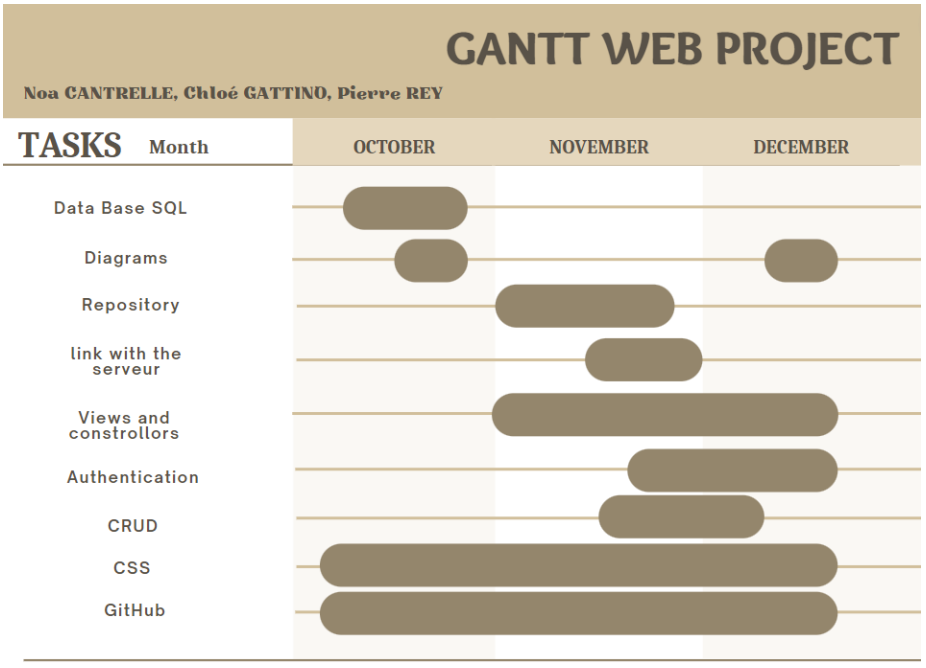
Noa CANTRELLE, Chloé GATTINO, Pierre REY

I - ER + Table structure diagram

Our project is to create an online shopping website with video games and consoles. We have different kinds of clients who can have access to it : a user, an admin and someone who hasn't an account yet. Each of them have a different role in the website. A payment is available only for users who want to buy a product.
Here is the following table structure diagram :

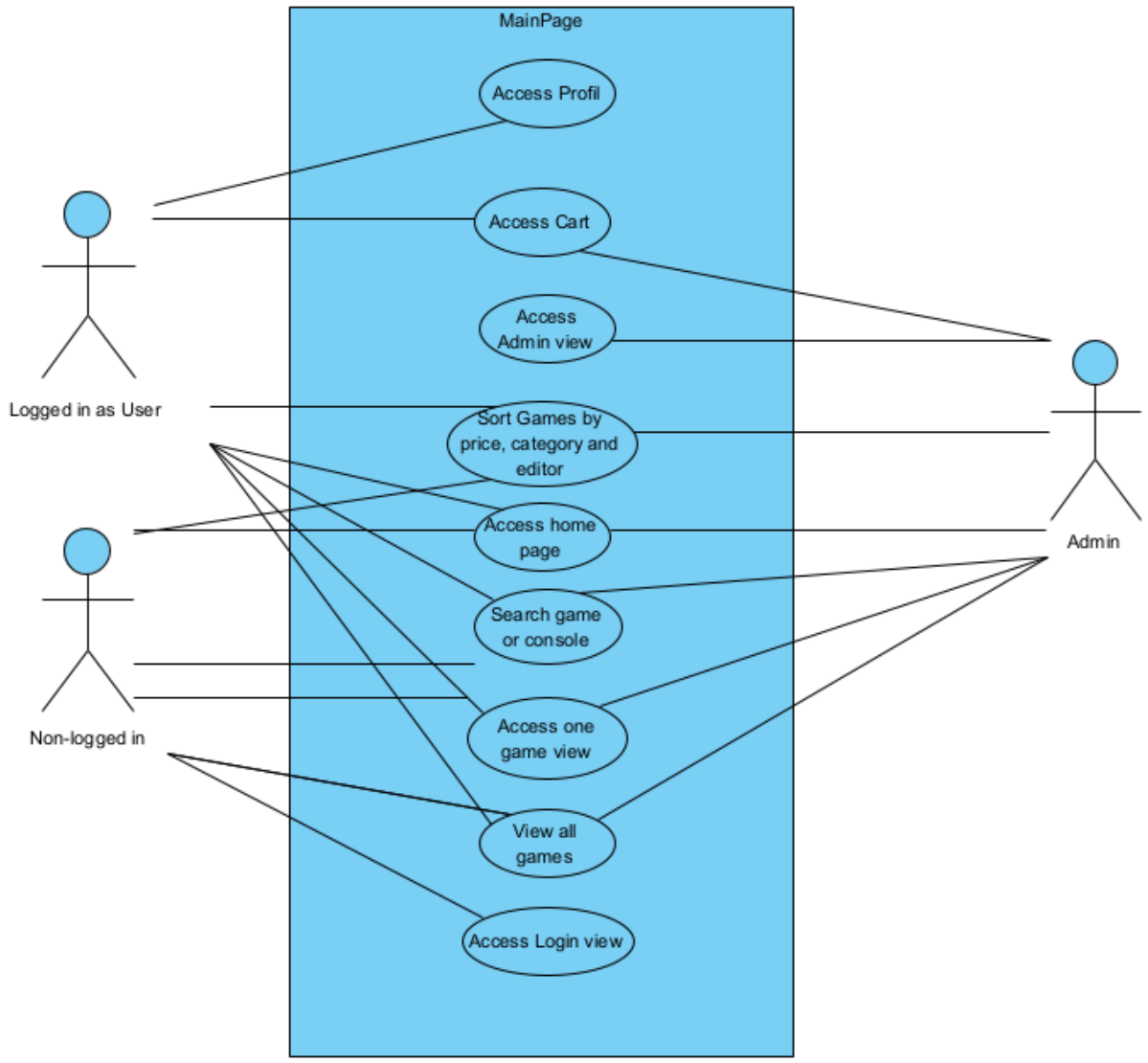


This is our GANTT DIAGRAM, a schedule with all tasks we have done during the semester :

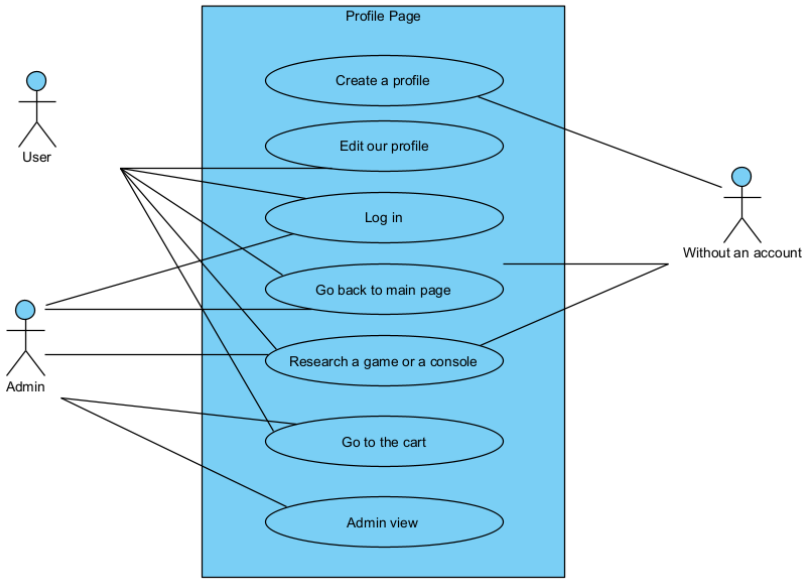


II- 1) Use case diagram

Pierre's Use Case Diagram : Case where someone access our website, shows the difference of permissions between logged in (USER and ADMIN) and non-logged in users

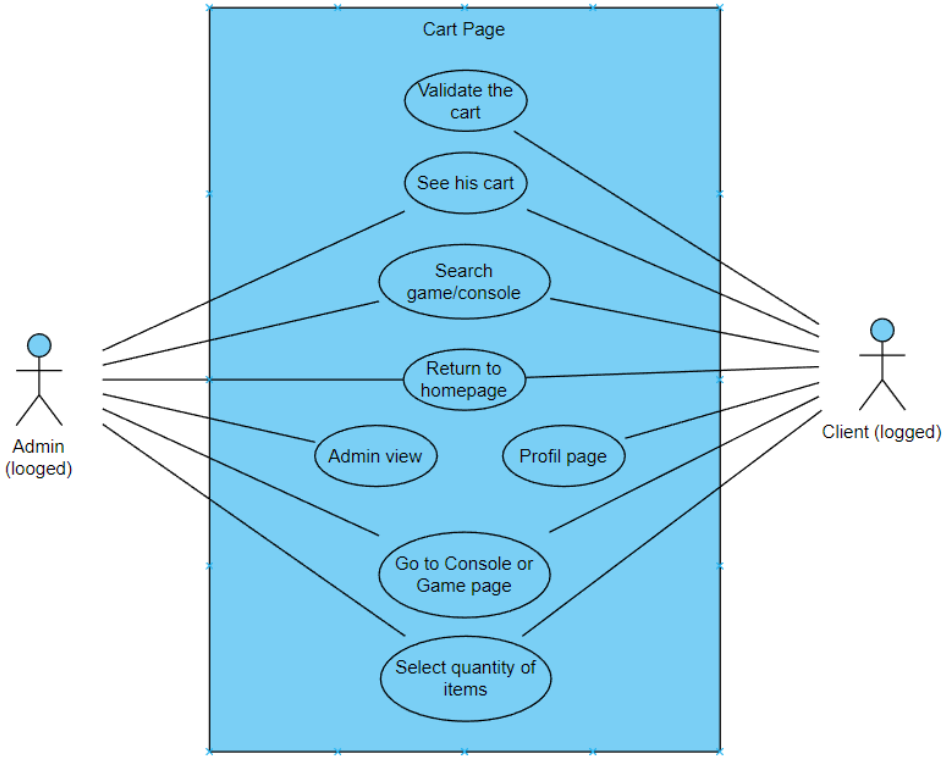


Chloé's : profile page with three types of user : the Admin, special user who can have access to the "admin view" to be able to edit, add or delete a game or console, the user who can edit his profile, go to the cart, and the one that hasn't any account but can see our products and create profile.



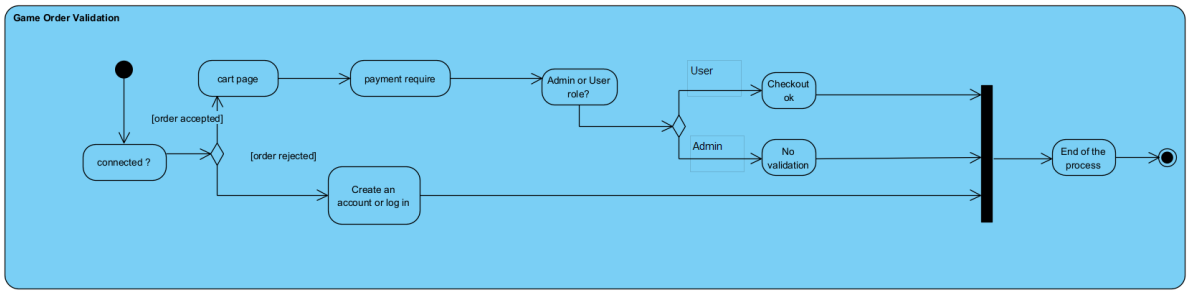
Noa : Cart page

In the shopping cart, whether you are a customer or an administrator, you have the same permissions. However, if you are not logged in, you cannot access the cart.

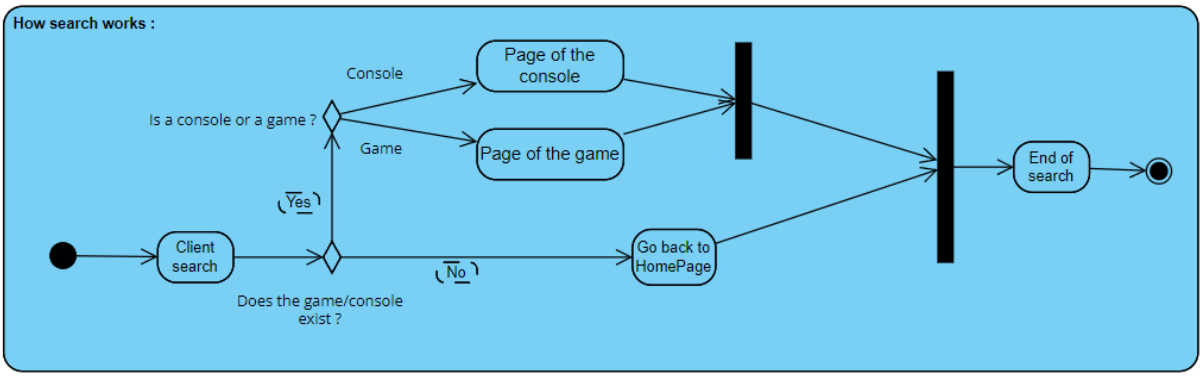


2) Activity Diagram :

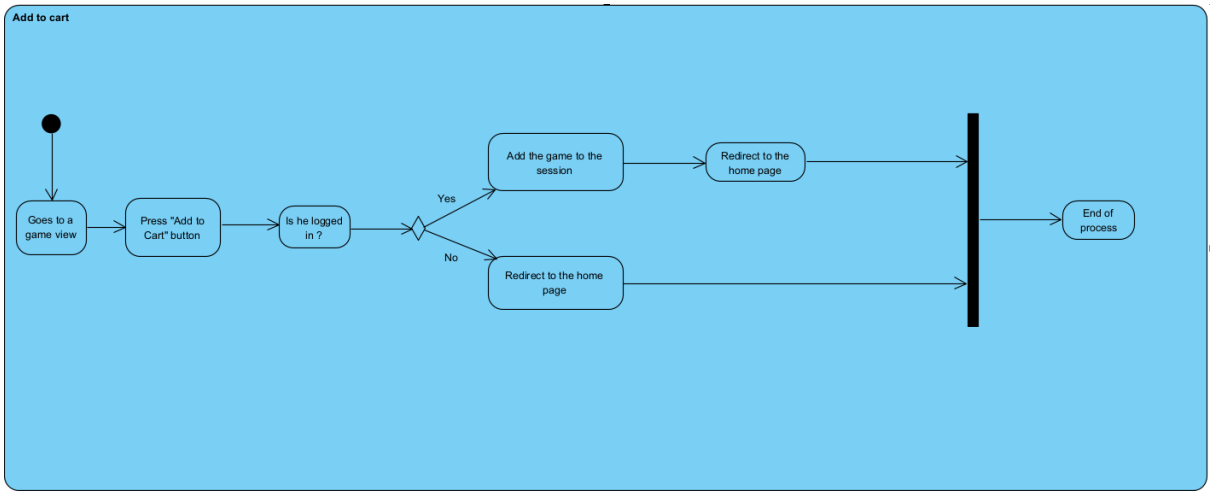
Chloé's Activity Diagram : when somebody go to cart page for the payment. Let's assume that it is an order for a game.



Noa's activity diagram about how the research works on the website

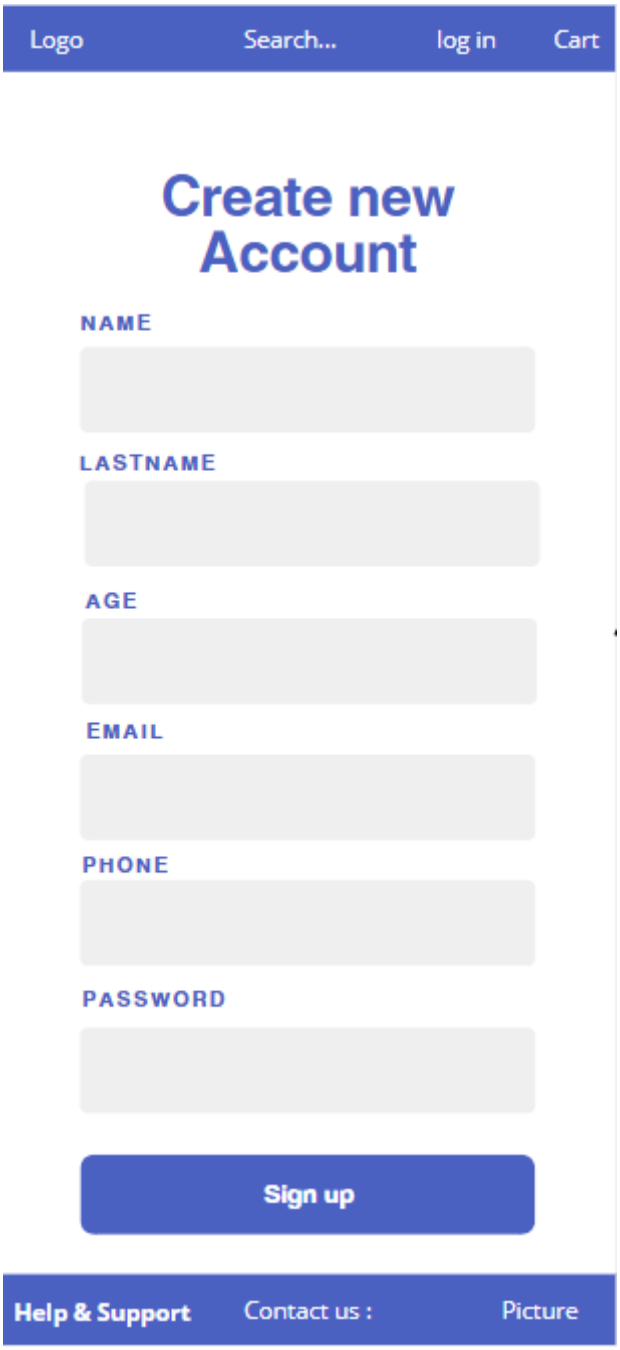


Pierre's activity diagram about how the "AddToCart" function works for a user. Basically the user goes to one game or console and presses the button to add the game to his cart. If he's logged in, he will have access to the game in his cart. If he's not, he won't be able to access the cart.

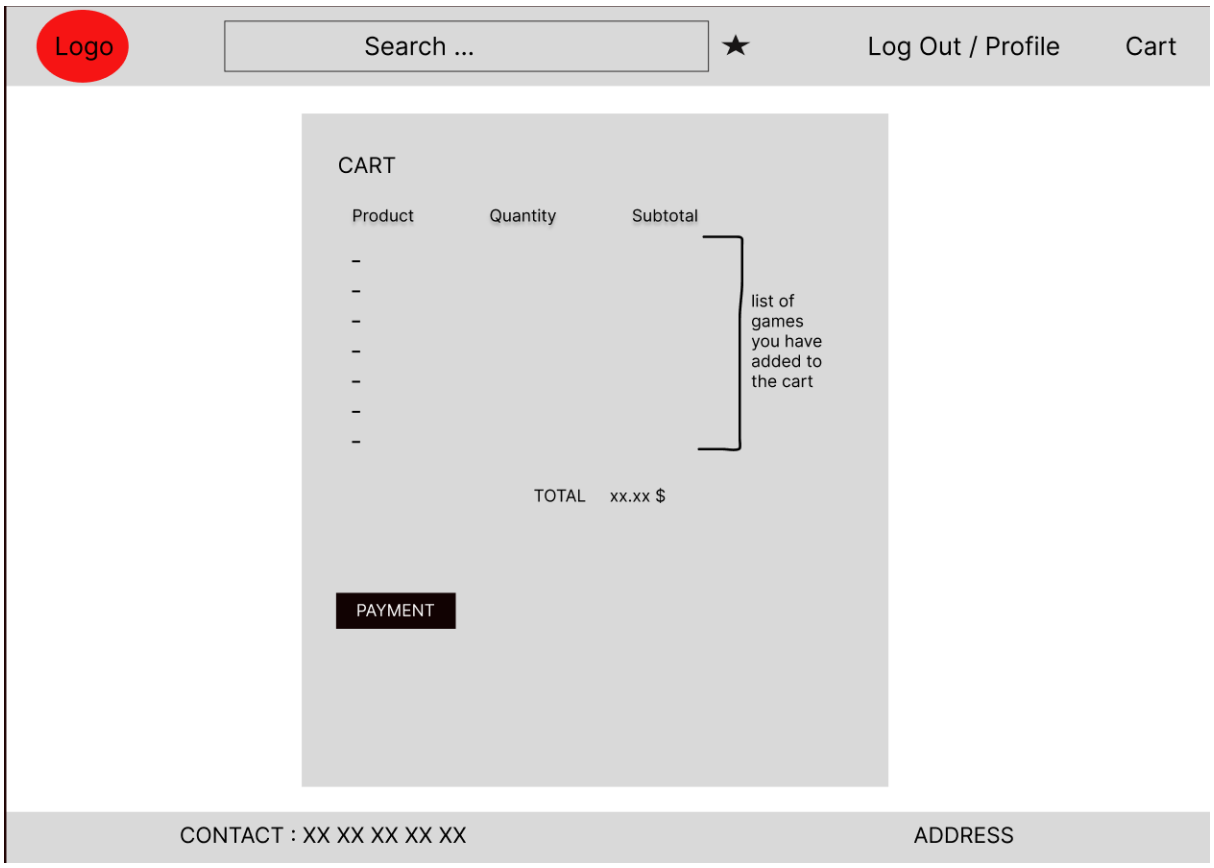


3) Wireframe Diagram

Chloé : the wireframe diagram shows how is representing the creation of a new account is our website:



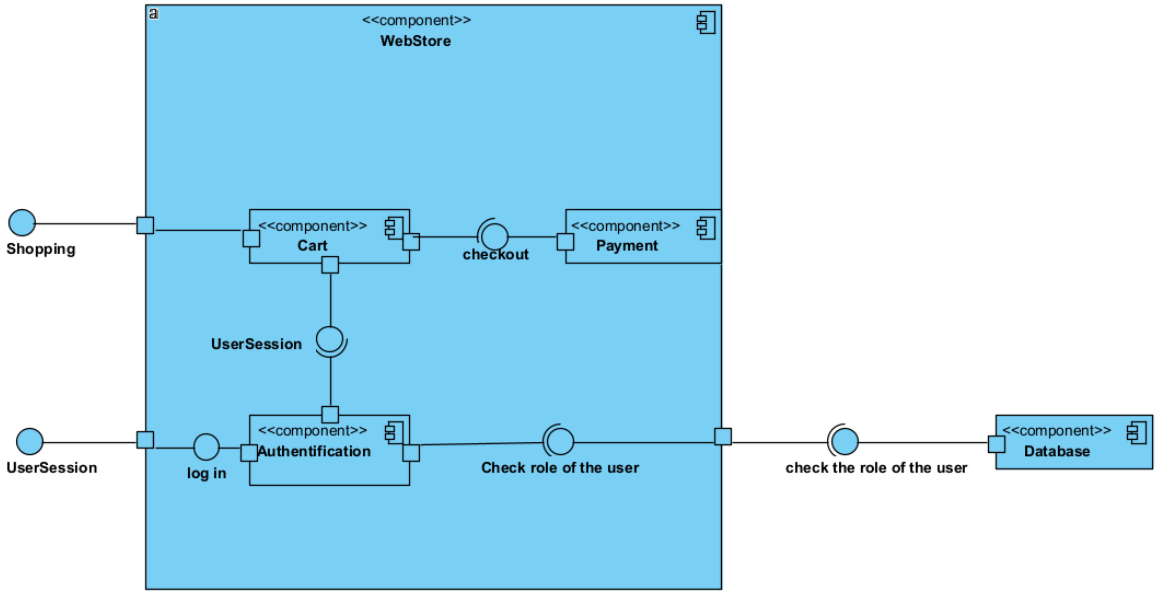
Noa’s Wireframe diagram about Cart page :



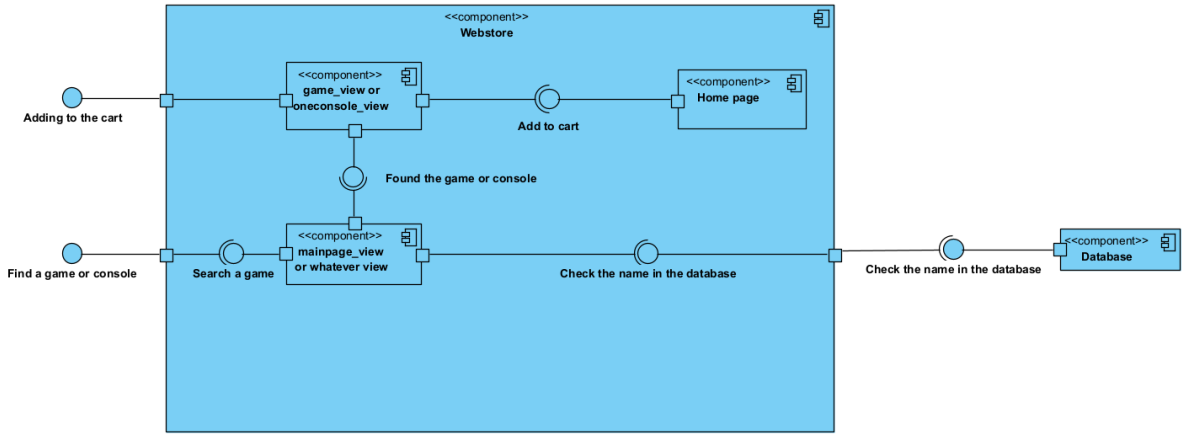
Pierre’s wireframe Diagram about Main Page

3) Component diagram :

Chloé's diagram about buying a product from the webstore with the checking of the authentication and the payment at the end. We have to check if the client is an admin or a user or not, otherwise he can't buy a game.

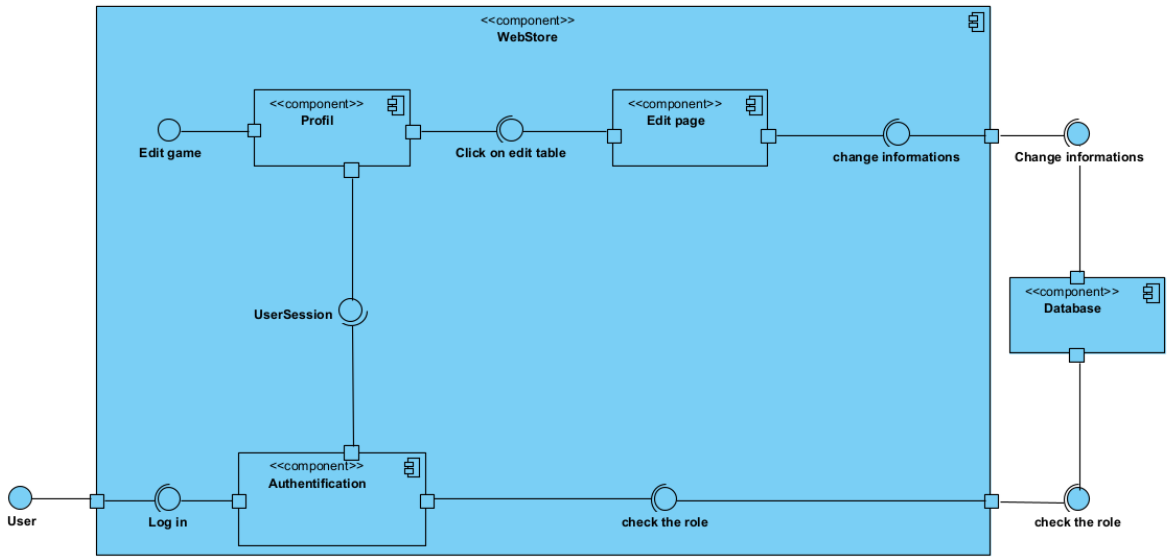


Pierre's component diagram is about searching a game from any view on our website. We have to access the database to know if the game exists or not and then it renders the view of this game. It also shows how we had a game to the cart, using the Add To Cart function that redirects the user to the Home page after adding the game to the session of the user.



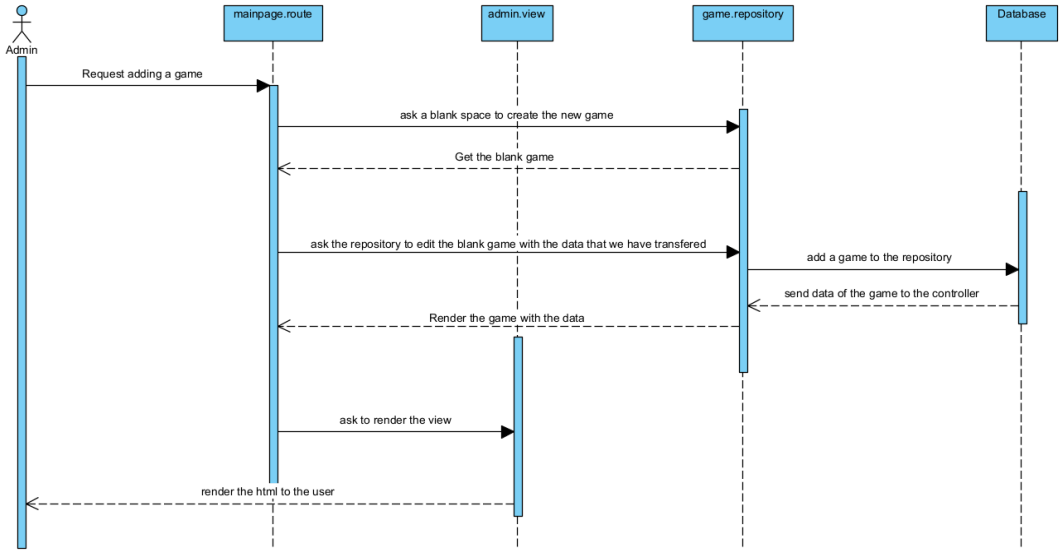
Noa's component diagram:

In the case of your are an admin, you can edit a table. The first thing is that you have to be an admin, so there is a check when you log on to the site if you are an admin. In this case, you can go to your profile and access the edit table. You have to click on the edit table and change what you want.



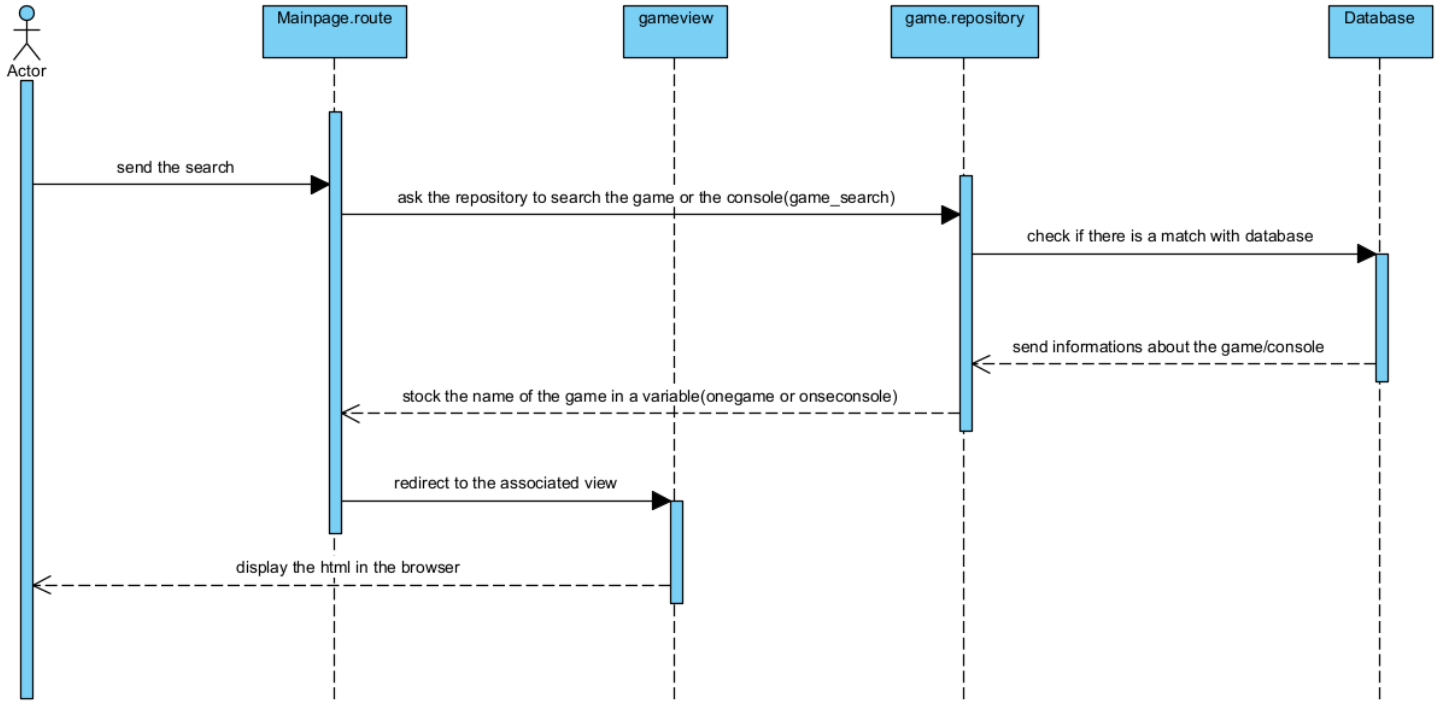
4) Sequence Diagram

Chloé : an admin who adds a new game in the database through the view where the interface is. Let's assume that it is an admin, because it is the only one who is able to add a game.

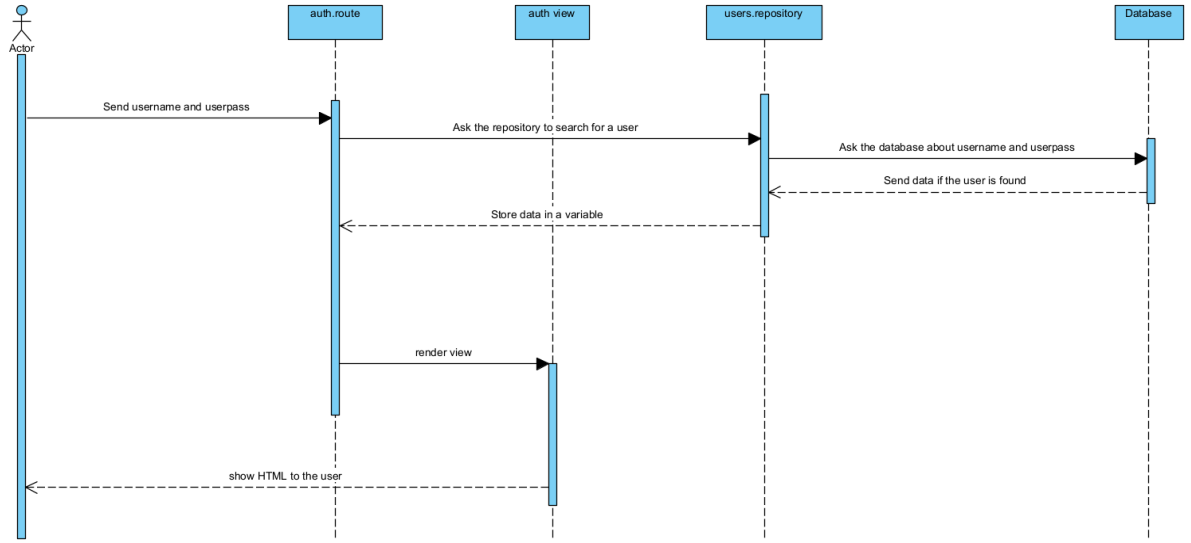


Noa : How to find a game or a console on our website.

First the controller is called, it communicates with the repository which compares the entered data with our database. If there's no match between the database and the game_search, we are redirected to the home page. However, if there's a match, the database sends back the information about the game or the console to the repository. The controller stocks informations in a variable (oneconsole or onegame). The controller redirects to the view which corresponds and the view displays to the user the html in the browser.

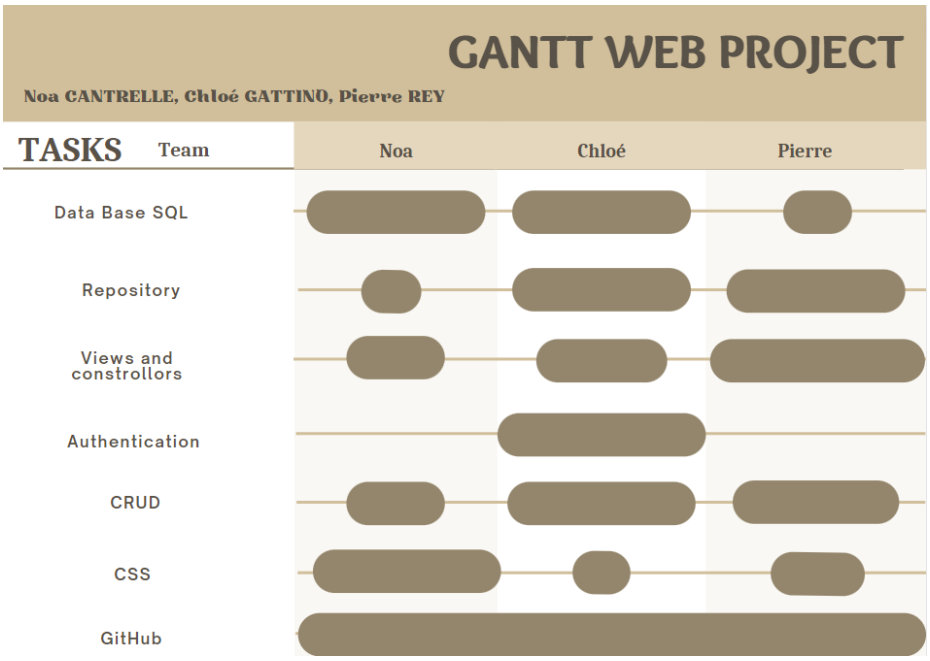


Pierre’s sequence diagram about logging into our website. The user will writes his username and userpass that will be passed to the auth.route (controller) which will send these info to the repository so it can ask the database if the username and userpass are valid. If it does, then there is data in the variable in the controller. In this case, it will render a view that will render the HTML page to the user.

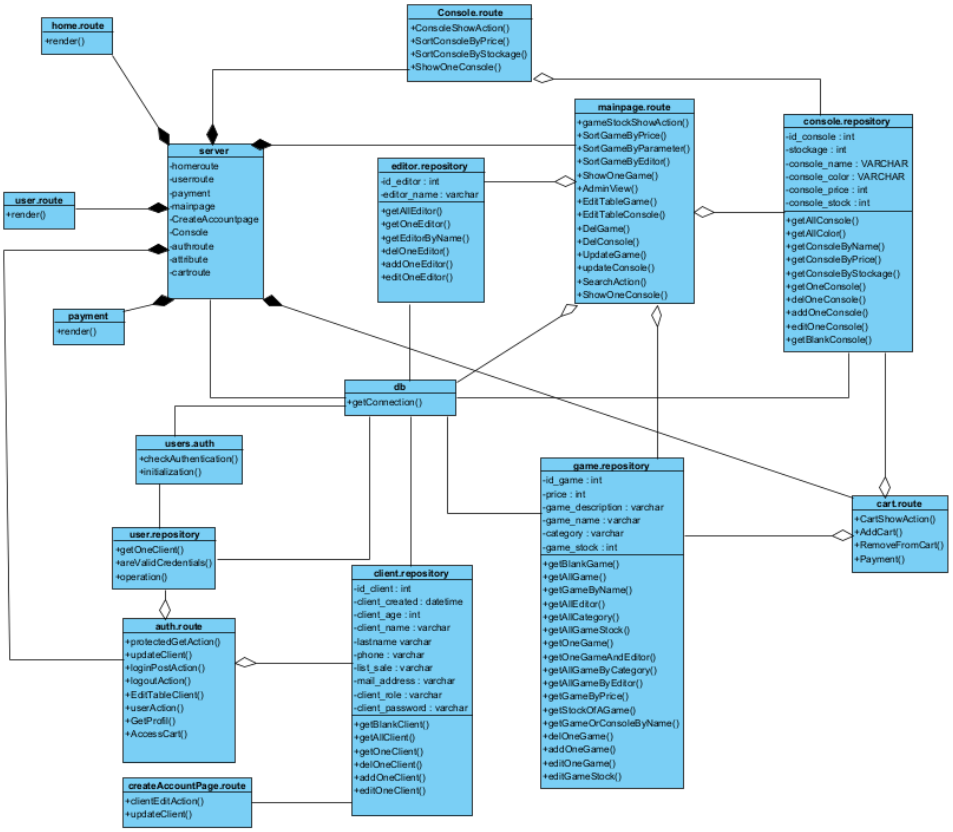


III- Final diagrams

Gantt diagram : Here is how we worked and who did what



Class diagram : let's see how our classes are linked between each others



Sources :

payment page : https://www.w3schools.com/howto/howto_css_checkout_form.asp