

Web Statique & Dynamique

2023-2024

Théo CLERE, Maxime BROSSARD, Sandro SPINA, Ceif-Edine
MAROUANI, Julien BUC
CPE LYON

Table des matières

I.	Diagrammes de conception	2
A.	Création d'un utilisateur :.....	2
B.	Connexion d'un utilisateur	3
C.	Diagramme de classe User-> Card	4
D.	Diagramme des instances réel	5
E.	Diagramme de l'application 3 Tiers.....	6
F.	Table des illustrations	7

I. Diagrammes de conception

Les différents diagrammes ont été réalisés dans un but de compréhension et/ou faisant suite à une demande précise de la part du corp enseignant.

A. Création d'un utilisateur :

Lorsqu'un nouvelle utilisateur arrive sur la page de l'application web, celui-ci va créer un compte. La page de connexion/création de compte est afficher grâce au service rendu par l'authentification manager qui va vérifier le cache du navigateur et chercher si une session active existe (via l'existence d'un TOKEN). Le service appelé pour la création d'un compte est dans le UserManager, celui-ci va créer un utilisateur en base de données (via l'ORM) et passer à l'étape suivante, qui est l'attribution aléatoire de 5 carte, donc une boucle de 5 tour est exécuter sur CardManager. Une fois les cartes créer, pour assurer la persistance, celle-ci sont ajouter en base de données et l'utilisateur (qui viens d'être créé) finis par être retourner à notre client.

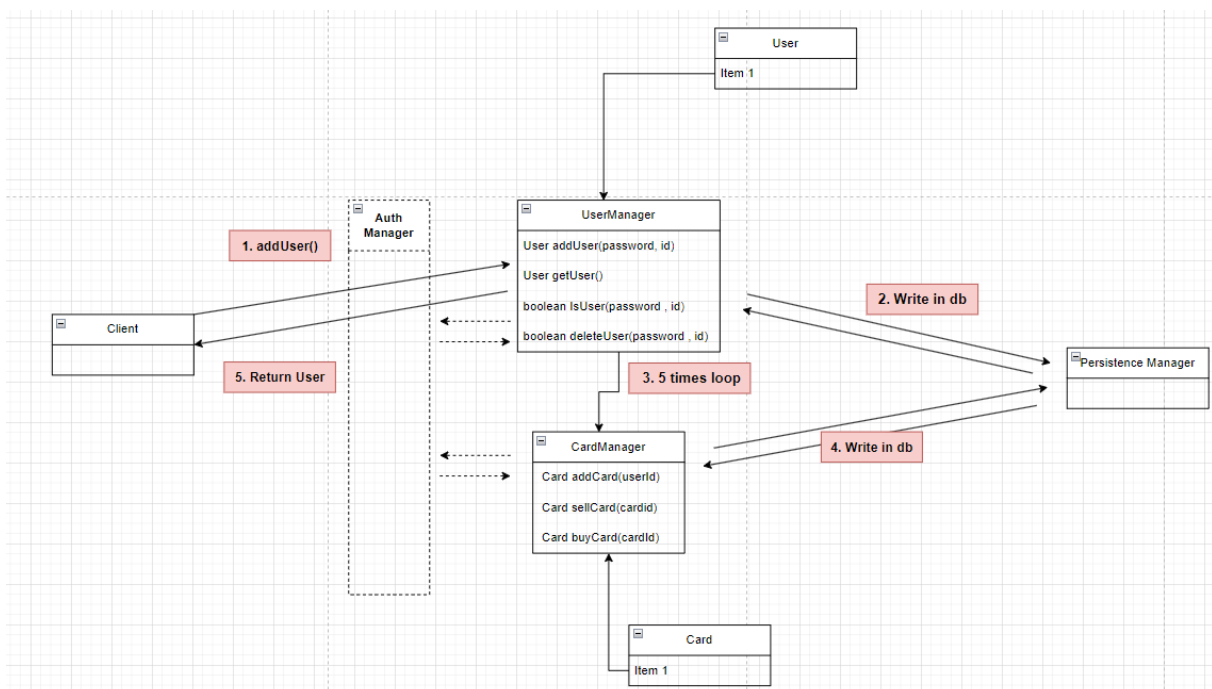


Figure 1 - conception du workflow de création d'un utilisateur

Le diagramme précédent s'accompagne d'un diagramme de séquence précisant mieux l'intervention et l'ordre des différents tiers de l'application.

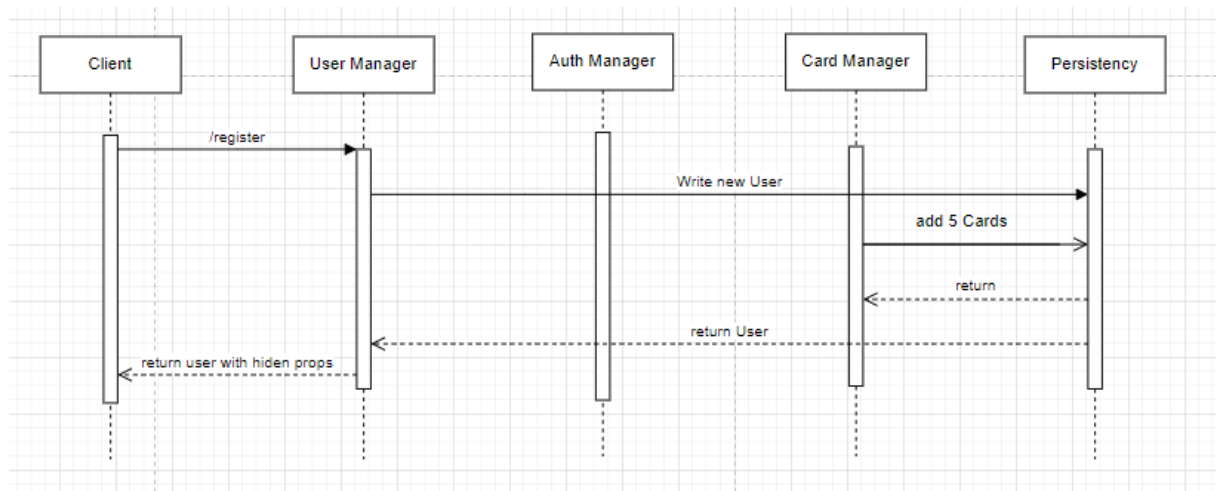


Figure 2 - Diagramme de séquence du workflow de création d'un utilisateur

B. Connexion d'un utilisateur

Ci-dessous le diagramme de séquence de connexion d'un utilisateur, comme précisé précédemment, un utilisateur a besoin d'un TOKEN afin de réaliser n'importe quel action sur l'application Web. Le diagramme suivant explique le processus d'obtention du TOKEN (stocké dans le navigateur du client final).

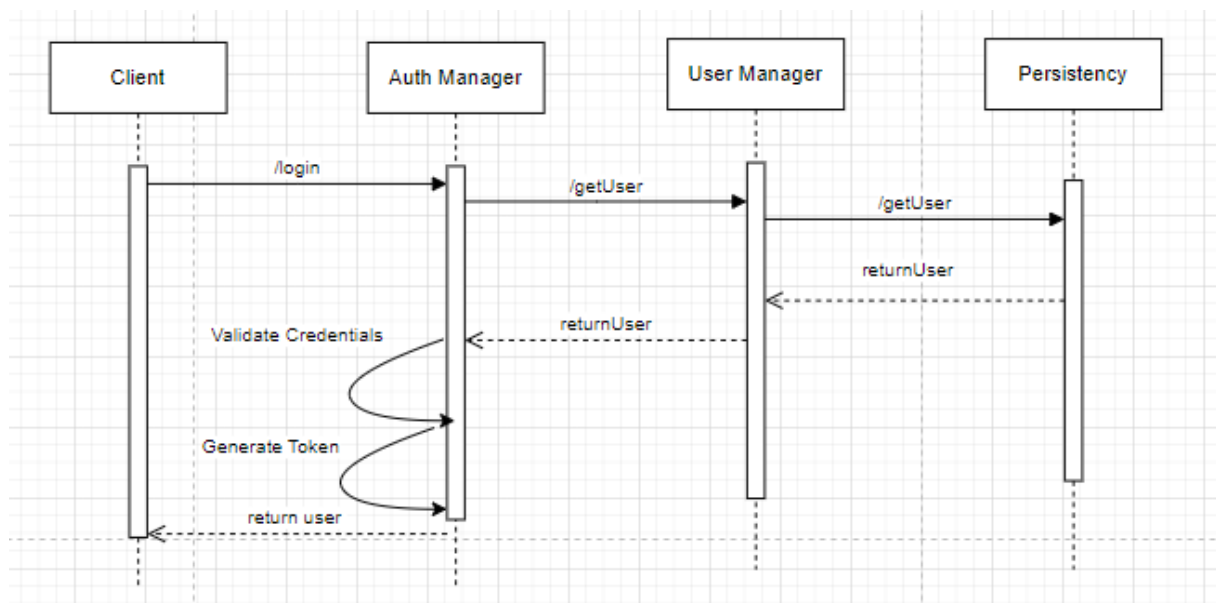


Figure 3 - Diagramme de séquence d'une connexion d'un utilisateur

C. Diagramme de classe User-> Card

Chaque utilisateur obtient des cartes lors de la création de son compte. Pour éviter les doublons et permettre à chacun de posséder des exemplaires d'une carte ou d'en avoir plusieurs sur le marché, nous avons mis en place des CardInstance, instance d'une carte (Card étant unique).

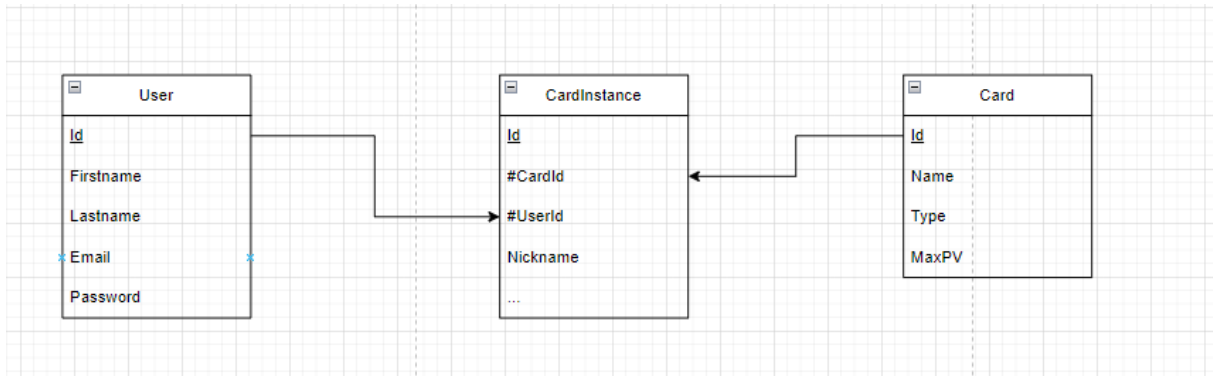


Figure 4 - Diagramme de classe Utilisateur -> Card

D. Diagramme des instances réel

Le diagramme suivant décrit et distingue les instances en jeu lors de l'exécution de l'application. On retrouve en premier lieu une porte qui va filtrer les interactions possibles de la part du client. Toute interaction doit être authentifiée et l'utilisateur n'interagit qu'avec le front. Le front fait appel à des routes gérées par le UserController et le CardController. En fonction de la demande, les différentes briques de l'application Spring interagissent afin de fournir la réponse adaptée à la demande du front. Une fois les données récupérées depuis la base de données, la logique est exécutée, une réponse est retournée et le front se met à jour afin de remplir les différents champs côté UI.

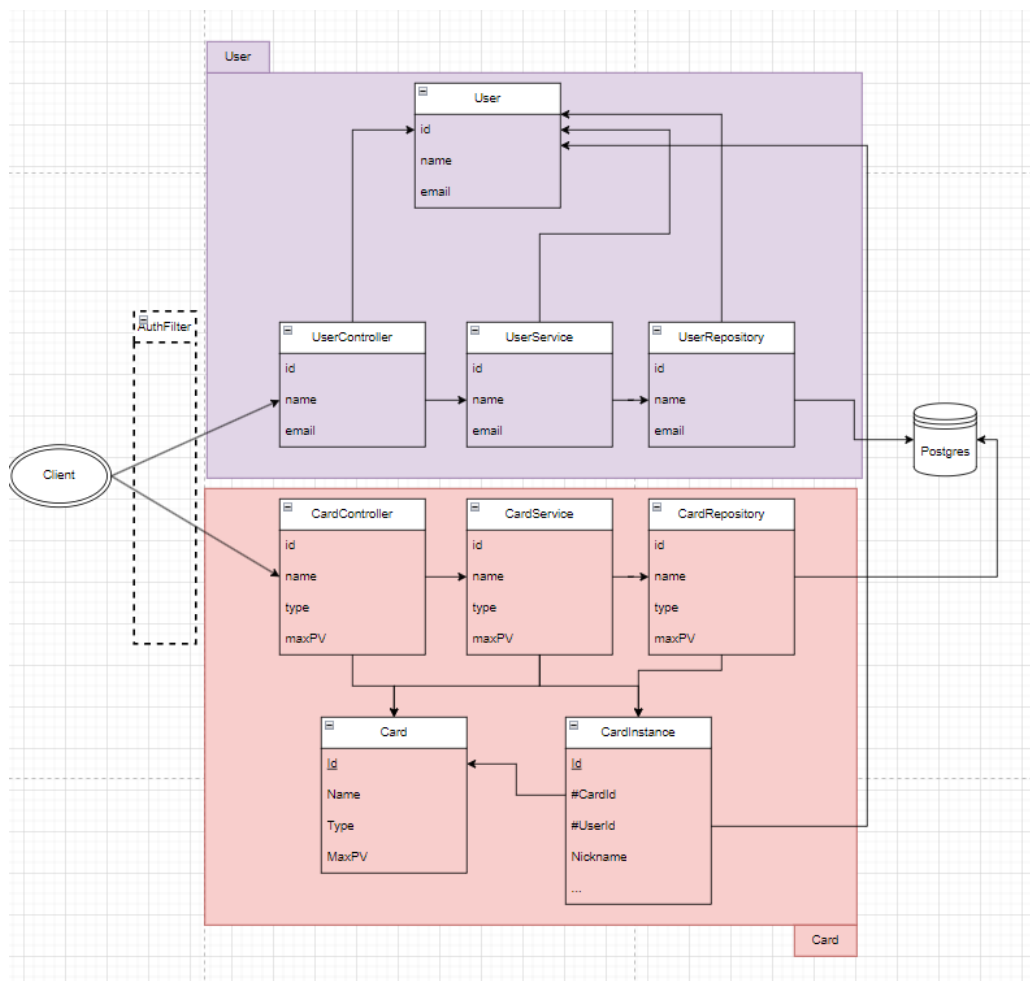


Figure 5 - Diagramme des briques en interactions dans l'application web

E. Diagramme de l'application 3 Tiers

Diagramme décrivant l'architecture 3 Tiers adoptée par le projet dans le cadre de l'utilisation d'une base de données PostgreSQL dans le cloud, un serveur Spring Boot et un front-end React.

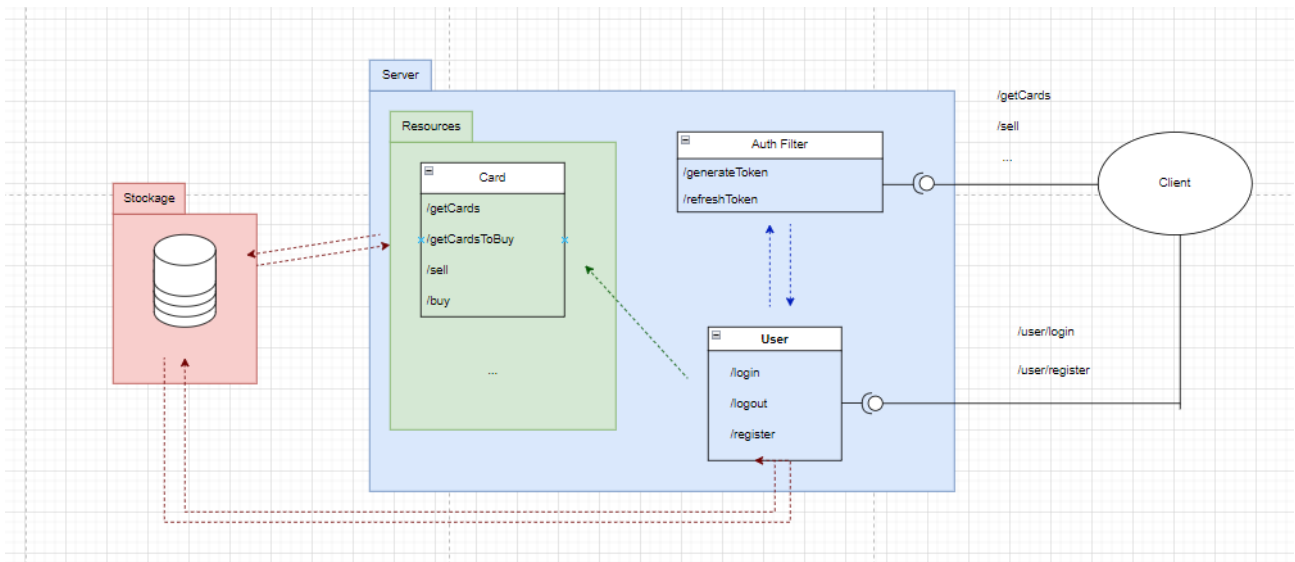


Figure 6 - Schema de l'application 3 Tiers

F. Table des illustrations

Figure 1 - conception du workflow de création d'un utilisateur	2
Figure 2 - Diagramme de séquence du workflow de création d'un utilisateur.....	3
Figure 3 - Diagramme de séquence d'une connexion d'un utilisateur	3
Figure 4 - Diagramme de classe Utilisateur -> Card.....	4
Figure 5 - Diagramme des briques en interactions dans l'application web	5
Figure 6 - Schema de l'application 3 Tiers	6