

League of Stones

Projet web

Alex JACQUOT

Aurélien THOMAS

Mamadou MENDY

Solen PENSEL

2018

Table des matières

I – Introduction	3
Langages utilisés :	3
Adresse de l’application :	3
II – Gestion de projet	4
III – Responsable technique	4
Choix des technologies :	4
Conception :	4
IV – Back-End	9
V – Front-End	9
VI – Conclusion	10
Bilan humain	10
Bilan Technique	Erreur ! Signet non défini.

I – Introduction

Dans ce projet, il est question de développer un *mashup* (mélange) de deux jeux vidéo. En utilisant le système de jeu de Hearthstone (HS) développé par Blizzard™ nous intégrerons les données ouvertes du jeu League Of Legends (LoL) développé par Riot Games™. Ce projet s'intitule "League of Stones" et il propose un jeu de cartes dont les cartes proviennent de LoL. Les deux joueurs s'affrontant possèdent un deck de 20 cartes et 150 points de vie. Chaque carte décrit un champion avec une statistique d'attaque et une de défense. Les joueurs jouent chacun leur tour dans l'objectif de réduire les points de vie de l'adversaire à 0 pour gagner la partie.

Langages utilisés :

REACT:

Bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

Material-UI:

Material Design est un ensemble de règles de design proposées par Google et qui s'appliquent à l'interface graphique des logiciels et applications.

Node.JS:

Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge.

Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS.

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js.

Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

MongoDB

Système de gestion de base de données orientée documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données.

Adresse de l'application :

<https://alex-jacquot.github.io/LeagueOfStones/>

II – Gestion de projet

Gestion des versions : Github : <https://github.com/alex-jacquot/LeagueOfStones>

Répartitions des tâches : Trello

Communication : Messenger, Slack

Documents techniques: Google Docs

III – Responsable technique

Choix des technologies :

Front-end : React

Raison : Framework front-end permettant la réalisation d'une application avec une page unique.

Back-end : node.js

Raison : Permet la gestion des requêtes AJAX mais également la gestion des paquets/dépendances de l'application et de react. Il permet également l'utilisation d'un serveur local pour tester l'application.

Conception :

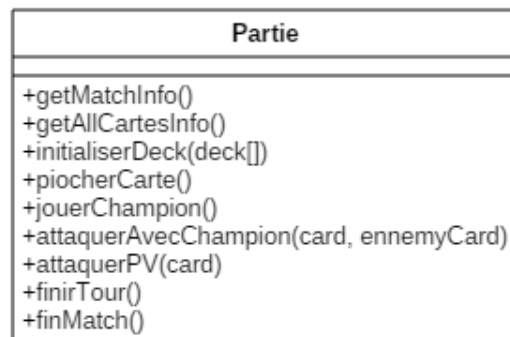
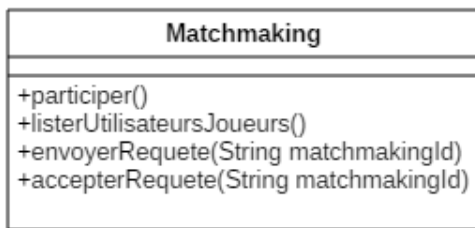
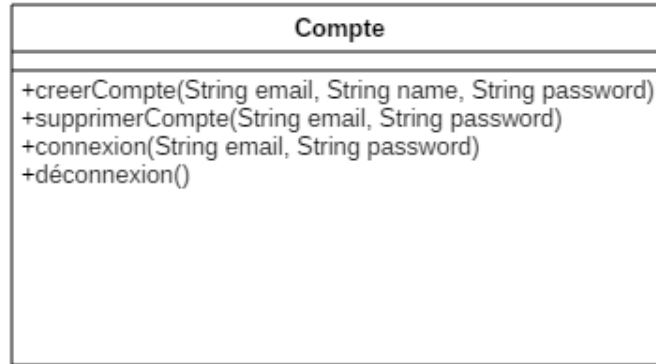
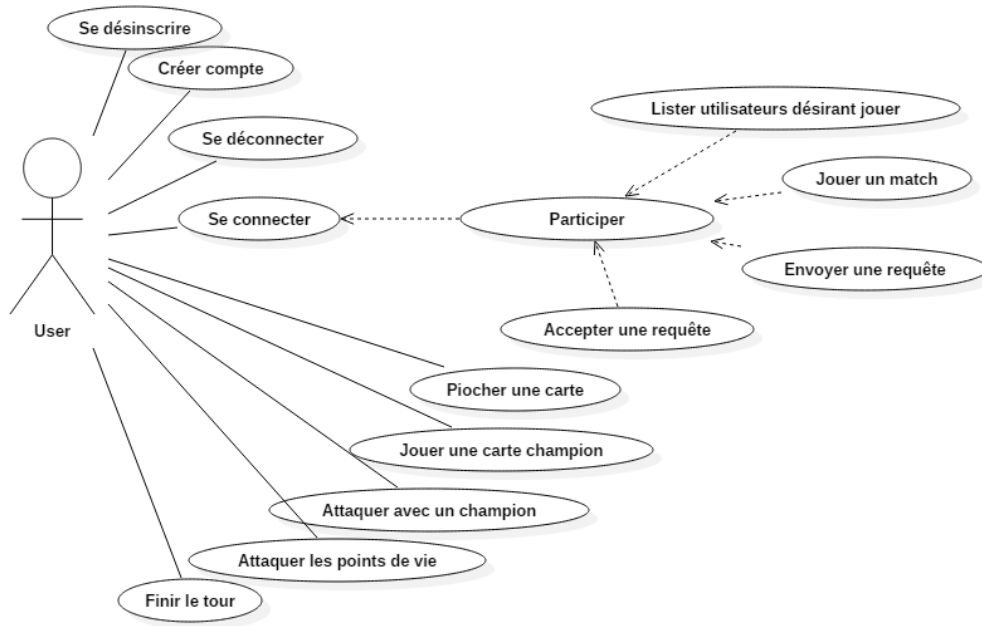


Diagramme de classe :

Diagramme des uses cases :



Maquettes:

League of Stones

Adresse mail

Mot de passe

[Connexion](#)

Utilisateur

Adresse mail

Mot de passe


[Inscrivez-vous](#)

Figure 1 Connexion Inscription

Rechercher un match

Figure 2 Lobby

League of Stones

Pseudo 

Adversaires disponibles

Joueur2484 veut vous défier

AccepterRefuser

Figure 3 Recherche de match

IV – Back-End

Bien que nous ayons fait tourner le serveur en local, nous avons jugé préférable de nous servir du serveur distant mis à disposition los.ling.fr.

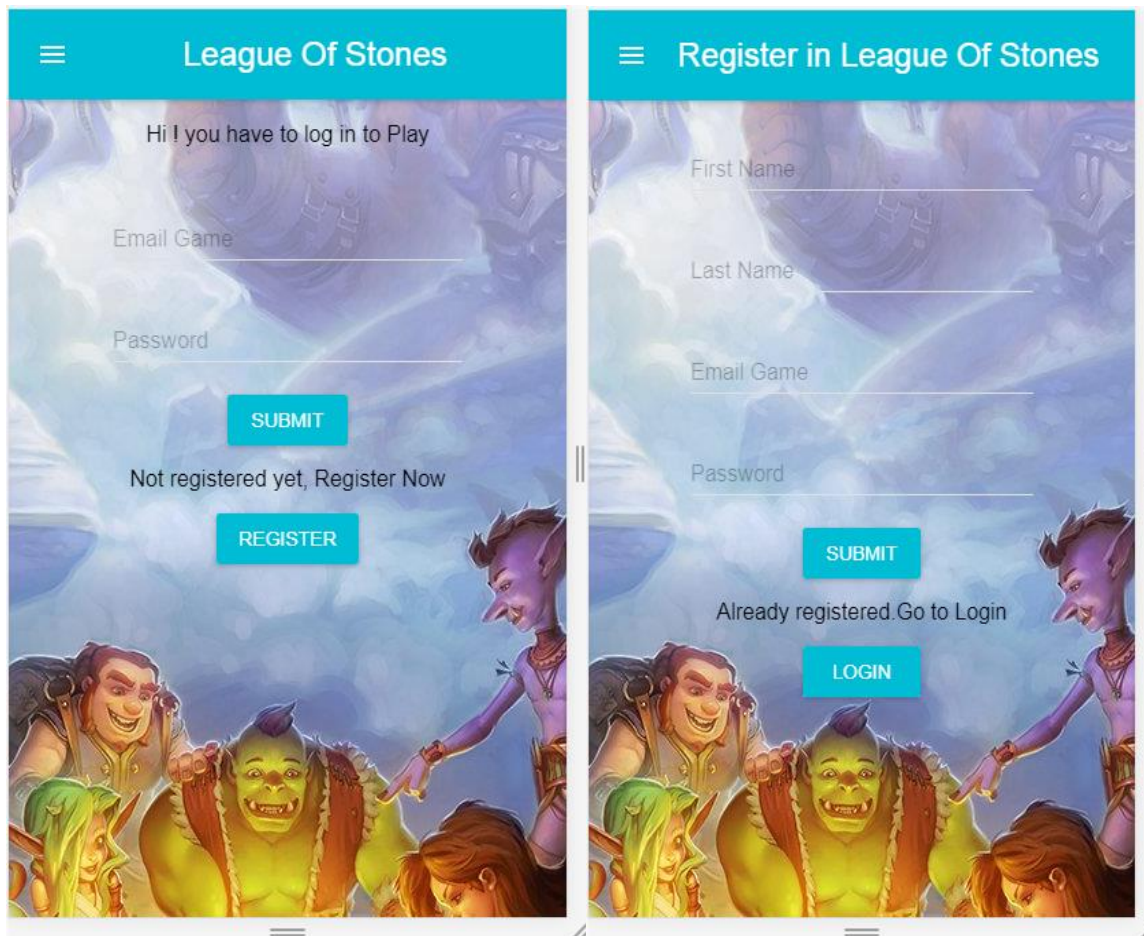
Nous avons pu implémenter les fonctions d'inscription et de connexion (`subscribe`, `unsubscribe`).

V – Front-End

Application des Technologies

ReactJS - Material-UI & Node : Interface de connexion et d'inscription

Ce système responsive fait avec Material-UI permet une simple utilisation de la plateforme. Material-UI propulse la plateforme sur un concept de “digital crafting”.



VI – Conclusion

Bilan humain

La marche de travail en général s’est révélé assez difficile, notamment en raison des circonstances extérieures de grève et de charge de travail supplémentaire qui en découle.

Mais nous avons réussi à nous arranger, dont entre autres prendre des réunions hebdomadaires.

Nous avons également choisi quelles fonctionnalités “sacrifier” afin de pouvoir rendre à temps un projet fonctionnel.

Bilan Technique:

Nous avons pu nous améliorer grandement dans les technologies React, Node, AJAX, et MongoDB.

Le front-end est géré par les technologies React, qui contient des appels à des fonctions back-end réalisant des requêtes AJAX sur le serveur distant constitué de Node et MongoDB .

Nous avons pu profiter de l’utilisation de l’outil Git, et notamment de la fonction d’hébergement de Github permettant d’avoir un serveur de test facilement accessible.