

ChessJS

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	3
2	Analyse / Conception.....	4
2.1	Diagramme de classe	4
2.2	Maquettes.....	4
2.3	Concepts	9
2.4	Méthode de gestion de projet	10
2.5	Plannification finale.....	10
2.6	Stratégie de test.....	12
2.7	Risques techniques	12
2.8	Base de données.....	12
2.8.1	MCD	12
2.8.2	MLD	13
2.9	Dossier de conception	13
3	Réalisation.....	13
3.1	Dossier de réalisation	13
3.2	Description des tests effectués.....	14
3.3	Erreurs restantes	14
3.4	Liste des documents fournis	14
4	Conclusions.....	14
5	Annexes.....	16
5.1	Résumé du rapport du TPI / version succincte de la documentation	16
5.2	Sources – Bibliographie.....	16
5.3	Journal de travail	16
5.4	Manuel d'Installation	16
5.5	Manuel d'Utilisation.....	16
5.6	Archives du projet.....	16

1 Analyse préliminaire

1.1 Introduction

Le projet consiste en un jeu d'échec en réseau. Les joueurs pourront créer de multiples parties avec différents adversaires. Il sera possible d'arrêter la partie en cours puis de la continuer ultérieurement

1.2 Objectifs

Le premier objectif est d'avoir un jeu d'échec fonctionnel. Le déplacement des pièces, la détection des « échecs » & « échecs et mat » doivent se faire selon les règles. Il y a certaines règles que je n'implémenterais pas, notamment le pat, le petit et grand roque ainsi que le changement de grade du pion lorsqu'il atteint le bord de l'adversaire. Plutôt que d'implémenter ces petites règles, j'ai préféré me concentrer sur le deuxième objectif décrit ci-dessous.

Le second objectif est la gestion en temps réel du déplacement des pièces. Je profite de ce projet pour apprendre la manipulation de base des « WebSocket » avec NodeJS. Le serveur doit gérer l'affichage en temps réel en fonction de la couleur du joueur. Les 2 joueurs doivent avoir leurs « camps » de leurs coté et donc les calculs de déplacements doivent être adaptés également. Lorsqu'un joueur déplace une pièce, le déplacement se fait en temps réel sur l'écran de tous les joueurs connectés à la partie.

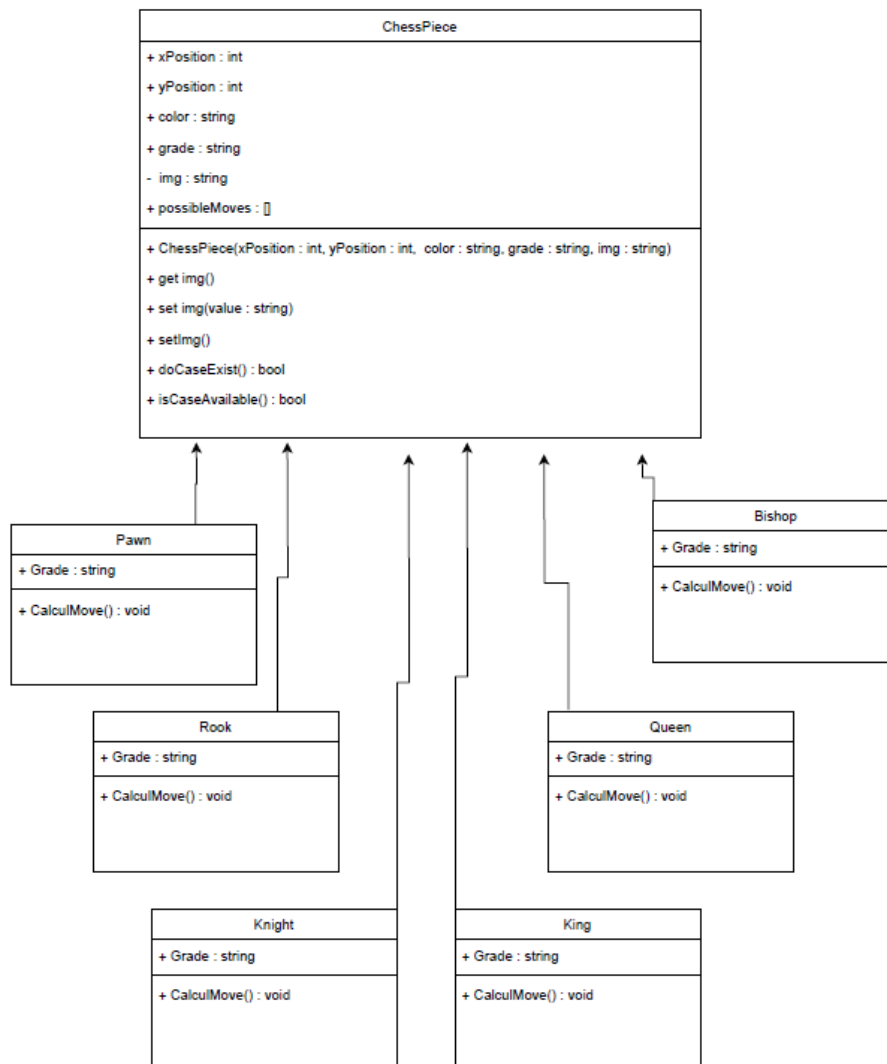
Un troisième objectif est de créer une gestion efficace des utilisateurs lors du push en production. Lors de mon dernier projet, j'ai eu des problèmes d'authentification des utilisateurs car le navigateur bloquait les requêtes POST. A priori l'implémentation des tokens « JWT » devrait résoudre le problème.

Un objectif serait du push en production sur le serveur de swisscenter. Normalement le serveur est dans tous les cas en « production » sur le réseau local.

2 Analyse / Conception

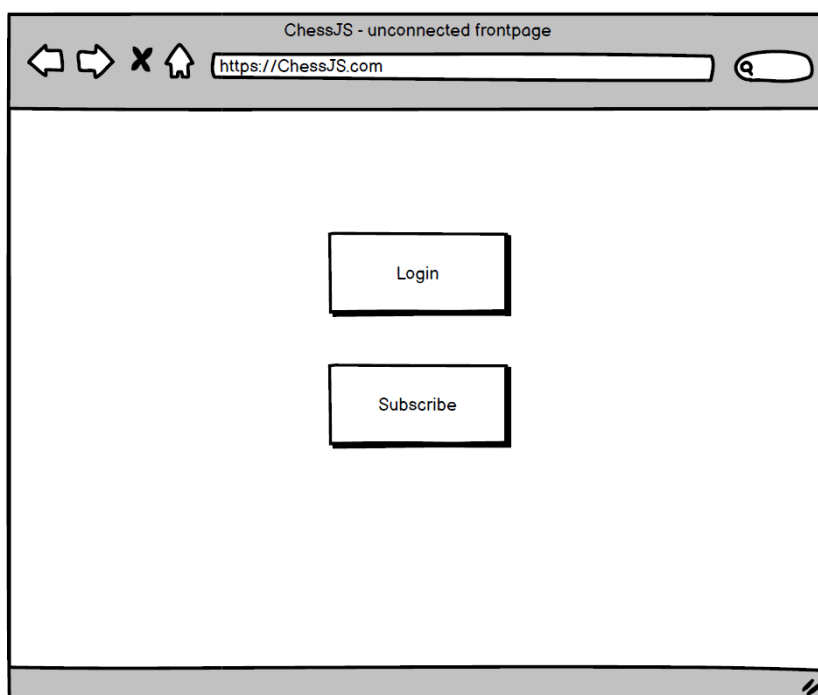
2.1 Diagramme de classe

Author : Maxime Borgeaud
Project : ChessJS
Last update : 01.02.2024

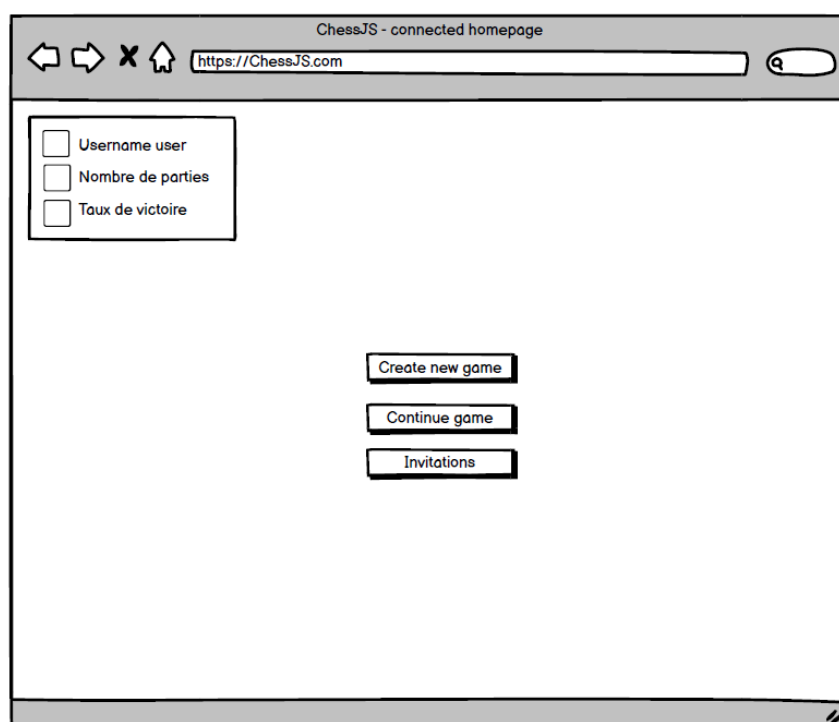


2.2 Maquettes

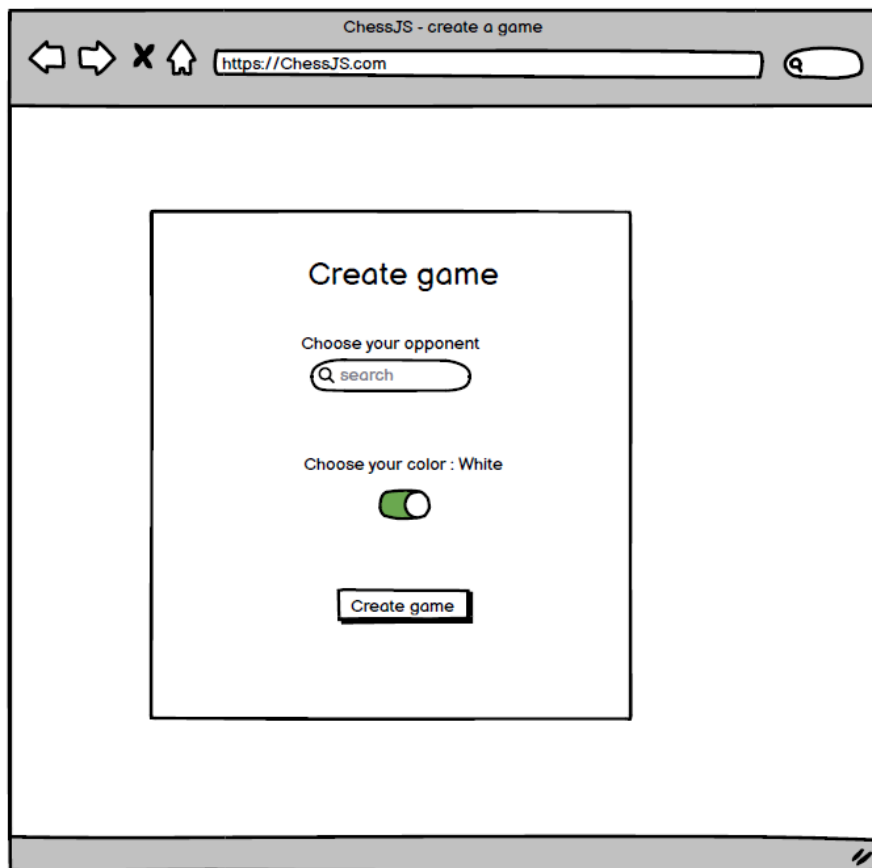
Page d'accueil « user non-loggé » :



Page d'accueil « user loggé » :

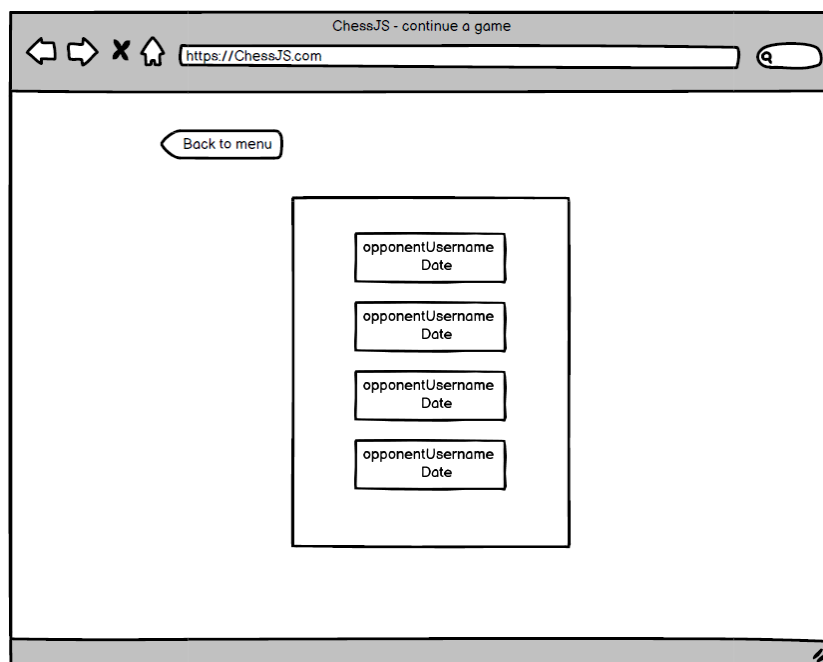


Page de création d'une partie :

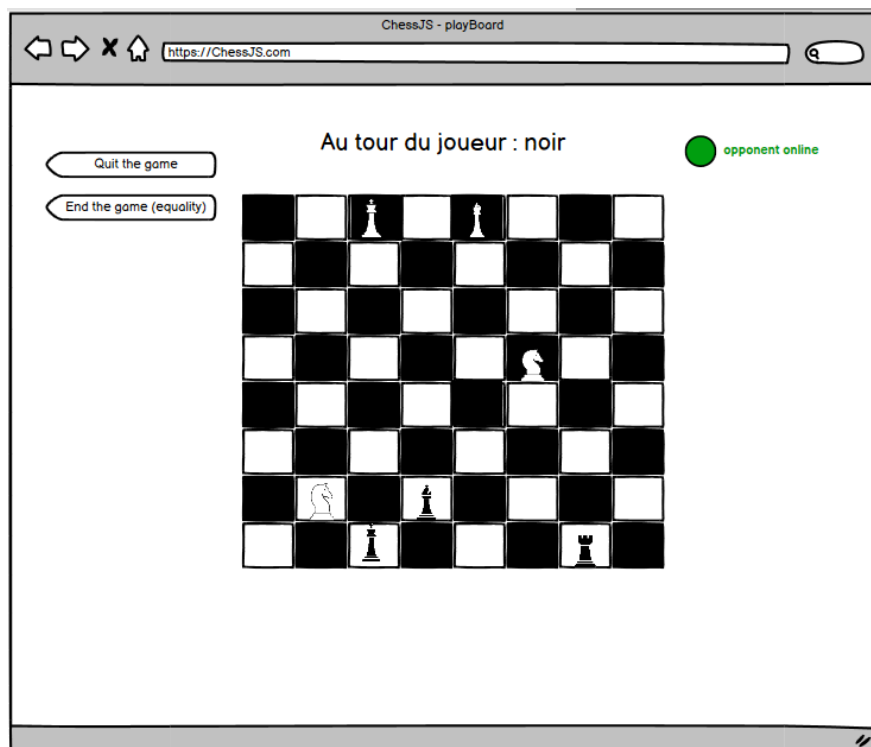


Le créateur de la partie choisit sa couleur

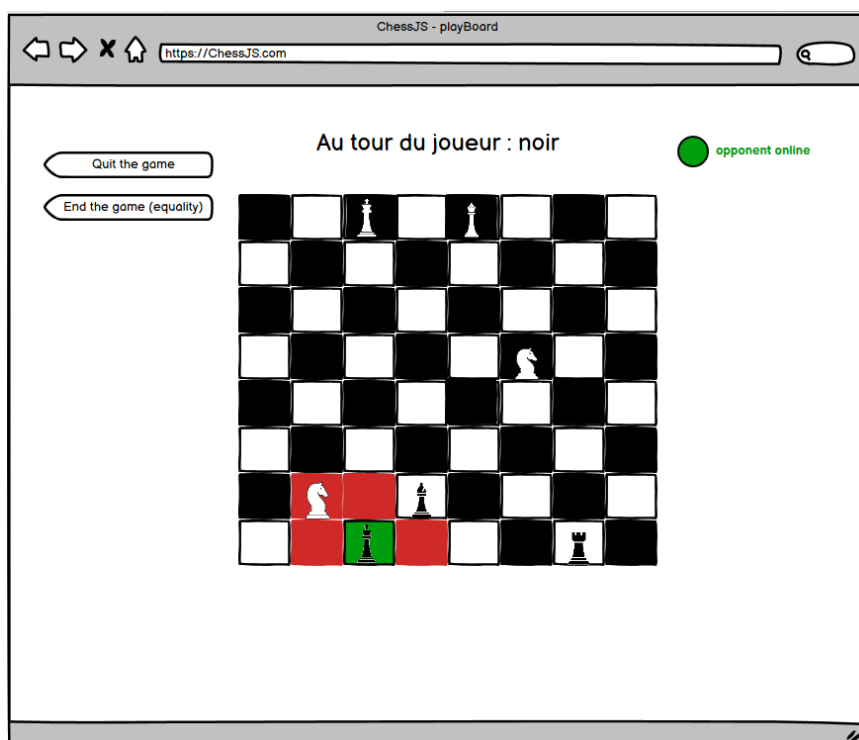
Page pour reprendre une partie commencée précédemment :



Le nom de l'opposant et la date de création de la partie s'affichent
Page de jeu :

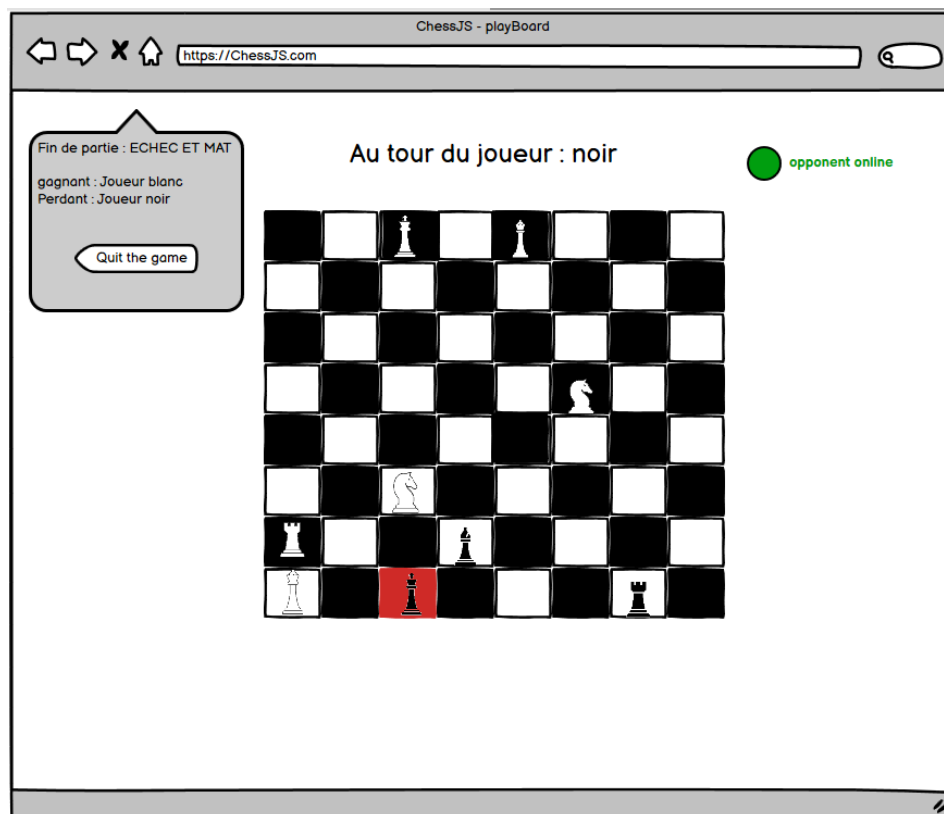


Page de jeu avec « ClickActif » :

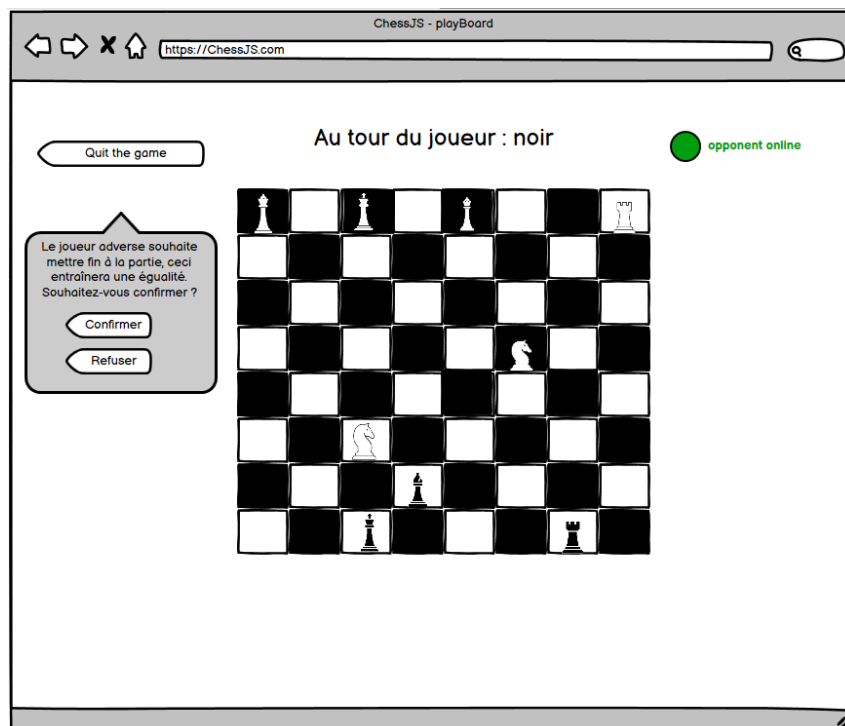


Le joueur noir clique sur son roi pour afficher ses déplacements possibles

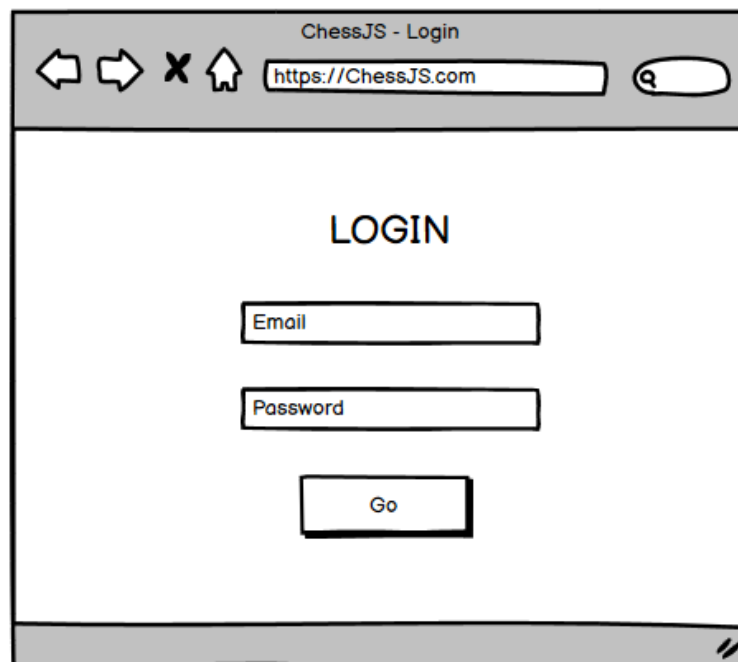
Page de jeu « echec & mat » :



Page de jeu « demande de fin de partie (égalité) » :

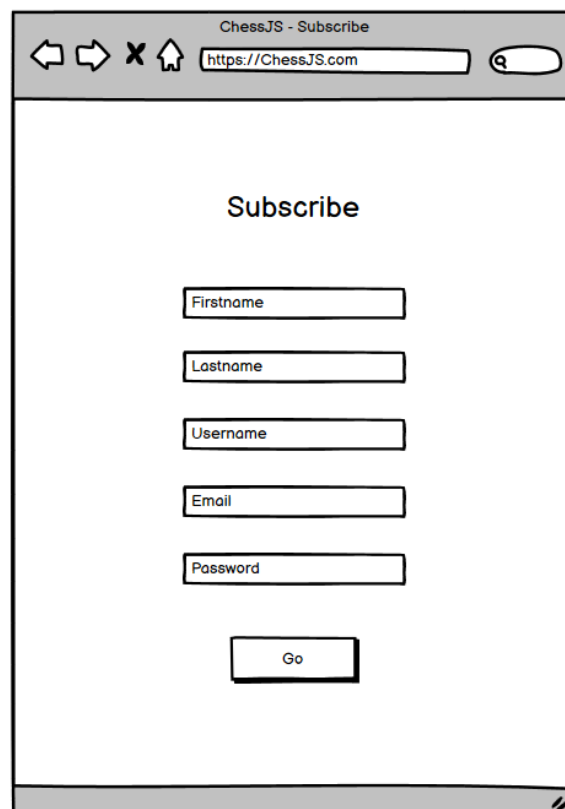


Page de login :



A screenshot of a web browser window titled "ChessJS - Login". The address bar shows "https://ChessJS.com". The page content is centered and features the heading "LOGIN". Below the heading are two input fields: "Email" and "Password". At the bottom of the form is a "Go" button. The browser window has standard navigation buttons (back, forward, stop, home) and a search icon.

Pas d'inscription :



A screenshot of a web browser window titled "ChessJS - Subscribe". The address bar shows "https://ChessJS.com". The page content is centered and features the heading "Subscribe". Below the heading are five input fields: "Firstname", "Lastname", "Username", "Email", and "Password". At the bottom of the form is a "Go" button. The browser window has standard navigation buttons (back, forward, stop, home) and a search icon.

2.3 Concepts

Voici les principaux éléments principaux qui composent le projet

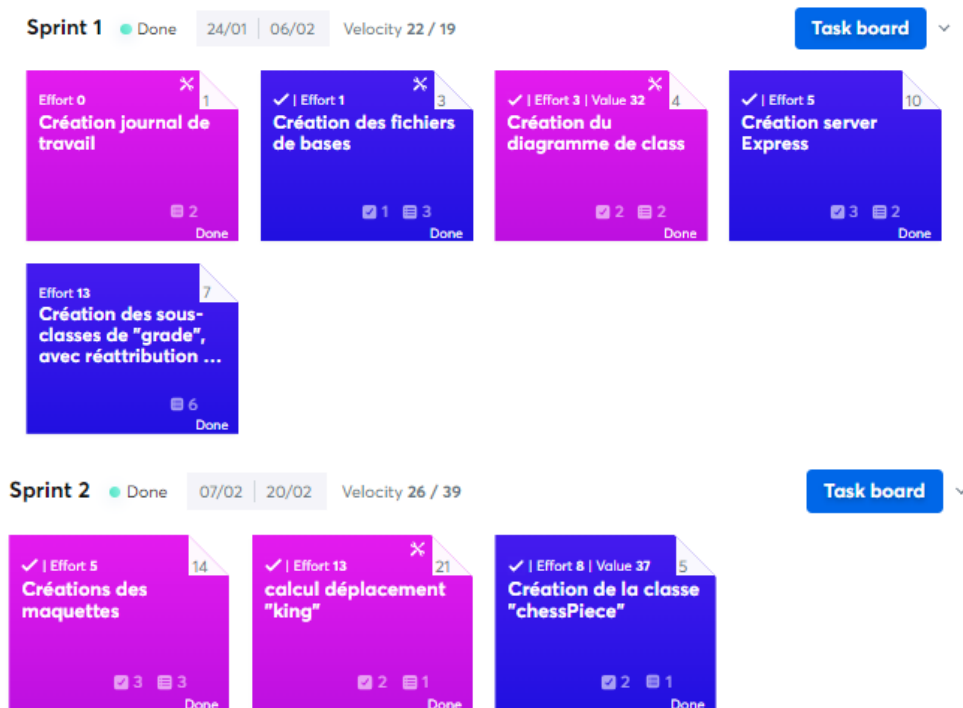
- ⇒ Un serveur web NodeJS Express
- ⇒ Une base de donnée MariaDb
- ⇒ Un serveur WebSocket NodeJS
- ⇒ Un système d'authentificationJWT
- ⇒ Stockage des mots de passe encryptés (hash)
- ⇒ Modèle MVC
- ⇒ POO

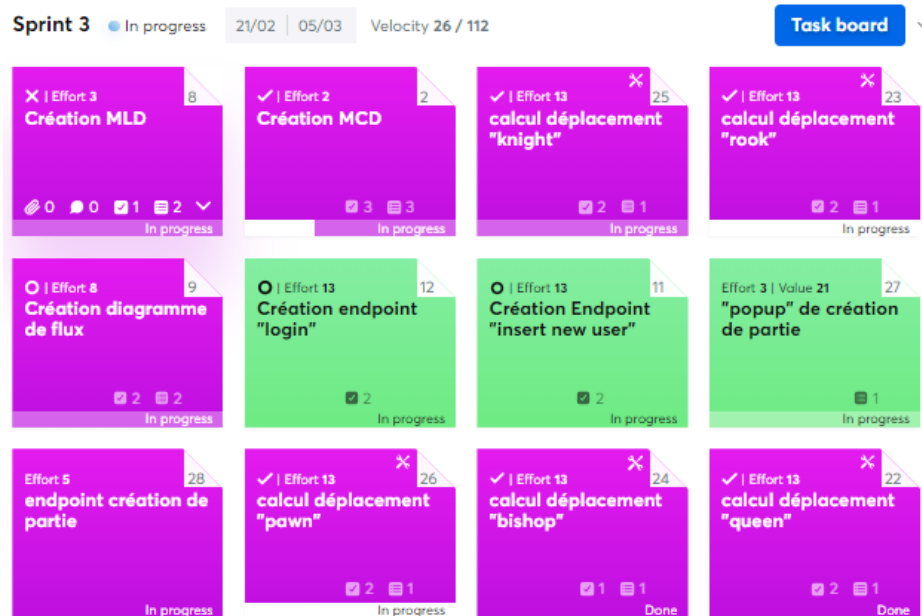
2.4 Méthode de gestion de projet

J'utilise la méthode Agile. J'ai défini 5 sprints de deux semaines dans lesquels j'ai mis les tâches à faire, qui sont détaillés dans la section ci-dessous

2.5 Plannification finale

La planification se trouve dans le IceScrum du projet que je vous remets ci-joints : <https://icescrum.cpnv.ch/p/CHESSJS/#/planning>





A l'origine, tous les calculs de déplacements des pièces étaient dans le sprint précédents. Ils ont dû être repoussés car le temps n'a pas été suffisant pour terminer ces tâches à 100%. (90% de ces tâches étaient complètes à la fin du sprint 2)



2.6 Stratégie de test

J'ai pris le parti de ne pas faire de test unitaire et au contraire de tester mon code de façon « manuelle » avec des « console.log() ». J'ai conscience que ce n'est pas la façon la plus propre de réaliser un projet de développement mais la raison de ce choix et mon manque de temps et de connaissance en la matière. Les tests unitaires en Javascript ne faisant pas partie des cours que nous avons suivi. J'ai préféré axer mes recherches d'acquisitions de connaissances dans les WebSocket, POO en JS et JWT plutôt que les tests unitaires. Voici quelques exemples de tests que j'ai réalisé durant le projet afin que vous compreniez mon mode de fonctionnement :

2.7 Risques techniques

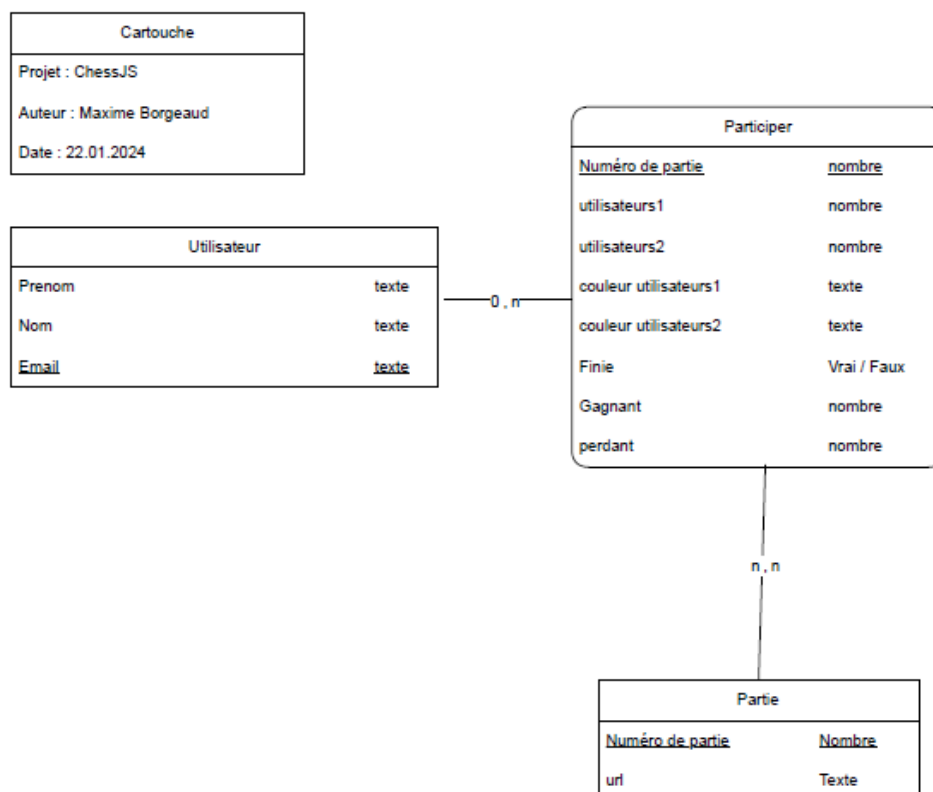
Le principale risque est mon manque de connaissance en matière de WebSockets. Etant un sujet que j'apprend en cours de projet, il n'est pas impossible que certaines difficultés ralentissent grandement mon avancement du projet.

Un autre risque technique est le fait que mon serveur web communique avec des requêtes http, Google chrome bloque automatiquement ces requêtes. Il faut espérer qu'il n'y ait0.

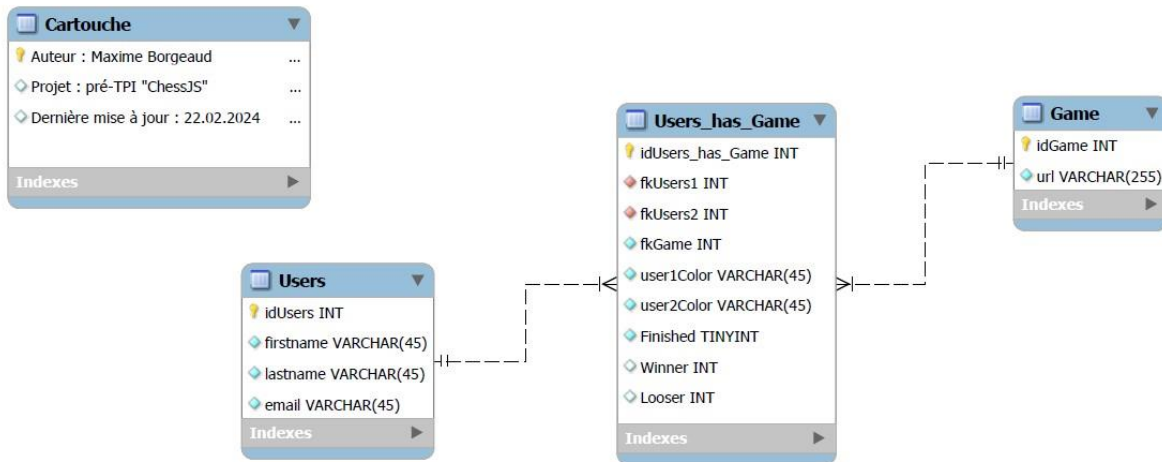
pas de mise à jour du browser durant le projet.

2.8 Base de données

2.8.1 MCD



2.8.2 MLD



2.9 Dossier de conception

Fournir tous les document de conception:

- le choix du matériel HW
- le choix des systèmes d'exploitation pour la réalisation et l'utilisation
- le choix des outils logiciels pour la réalisation et l'utilisation
- site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...
- bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.
- programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...

Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

3 Réalisation

3.1 Dossier de réalisation

Décrire la réalisation "physique" de votre projet

- les répertoires où le logiciel est installé
- la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)
- les versions des systèmes d'exploitation et des outils logiciels

- la description exacte du matériel
- le numéro de version de votre produit !
- programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.

NOTE : Evitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)
- tests sans preuve: fournir au moins une description

3.3 Erreurs restantes

S'il reste encore des erreurs:

- Description détaillée
- Conséquences sur l'utilisation du produit
- Actions envisagées ou possibles

3.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- le rapport de projet
- le manuel d'Installation (en annexe)
- le manuel d'Utilisation avec des exemples graphiques (en annexe)
- autres...

4 Conclusions

Développez en tous cas les points suivants:

- Objectifs atteints / non-atteints
- Points positifs / négatifs
- Difficultés particulières
- Suites possibles pour le projet (évolutions & améliorations)

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, ... dans une fourre en plastique "

Classes :

Première idée de classes :