



**Application Mobile - BLOC 2 SLAM**

**Jeu d’échec en ligne**

**Documentation technique**

*Rédigé le 13/02/2022 par Vann Vincent et Biegas Hugo*

*Dernière mise à jour : 30/03/2022*

**Présentation du projet**

L’équipe est composée de deux étudiants en BTS SIO2 SLAM, M.Vann et M.Biegas. Nous allons réaliser une application mobile Android hébergeant un jeu d'échecs multijoueur en Java avec une base de données distante. L’intégration d’une API comprenant le système de tour par tour en ligne sera mise en place.

https://github.com/mordeur/EchecMultiApp

**Equipe du projet**

| Nom – Prénom | Rôle dans le projet | E-mail |
| --- | --- | --- |
| Biegas Hugo | Développeur | [hugo.biegas@gmail.com](mailto:hugo.biegas@gmail.com) |
| Vann Vincent | Développeur | [vincentvann93@gmail.com](mailto:vincentvann93@gmail.com) |

**Contexte du projet**

Durant notre cursus scolaire, nous devons réaliser une application mobile Android portant sur le thème de notre choix et devant être validé par notre professeur M.Vinciguerra.

Nous serons amenés au choix à inclure une base de données locale ou distante ainsi que la mise en place d’au moins une API pour nous familiariser avec ces outils que nous serons amenés à utiliser en tant que futur développeur.

Nous allons pour ce faire utiliser un IDE pour composer votre code Java qui sera Android Studio qui est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle.

Nous avons fait le choix d’avoir opté pour le kit de développement (SDK) Android 5.0 (Lollipop) qui est un ensemble d'outils logiciels destinés aux développeurs, facilitant le développement d'un logiciel sur une plateforme donnée. Ce choix se traduit par une volonté de maximisation de compatibilité et rentabilité

**Description fonctionnelles**

Accueil et connexion

L’application doit être accessible selon les modalités suivantes : création de comptes directement sur l’application par personne pour stocker les données utilisateur (victoire, défaite, nom, email, mdp…). Si cela est possible nous mettrons en place l’accès en tant qu’invité et la mise en place de token temporaire pour inviter en partie privée sera aussi envisagée.

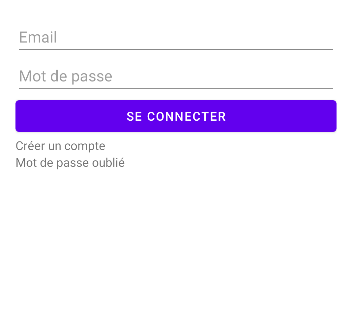
**Fonctionnalités :**

Fonctionnalité n°1 : Pouvoir se connecter

Objectif :

Mettre en place une page de connexion sécurisée avec un système de clé accompagné d’un algorithme de chiffrement pour les mots de passe de l’ensemble des utilisateurs.

Niveau de priorité : Haute

****

Fonctionnalité n°2 : Le jeu d’échec

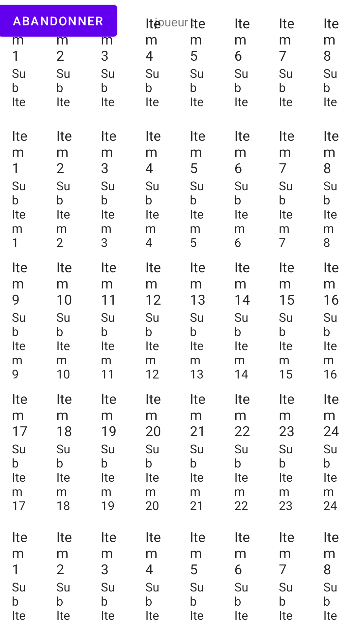
Objectif :

Mettre en place un plateau et des pièces d’échecs avec lequel l’utilisateur va pouvoir interagir pour pouvoir jouer ayant pour contrainte les règles d’échec traditionnel

Description :

L’utilisateur sera amené à pouvoir bouger les pièces d’échecs sur un plateau

Niveau de priorité : Haute

****

Fonctionnalité n°3 : Le multijoueur

Objectif :

Mettre en place la possibilité de jouer en multijoueur

Description :

Les utilisateurs vont pouvoir s’inviter pour jouer une partie à travers une liste d'amis,une liste de partite créer ou s’ils sont seuls de rentrer dans une file pour affronter un joueur dans la même file ayant rejoint plus tôt.

Niveau de priorité : Haute

****

**APi utiliser et diverse logicielle:**

on vas utiliser le service en ligne Firebase pour gérer la BDD et pour gérer le

multijoueur grâce à leur api

Firebase est considéré comme une platform as a service, en outre cette plateforme développée par Firme et appartenant à Google déployé depuis 2011, c’est outil polyvalent comportant entre autres une variété de fonctionnalités.  
On peut citer en l’occurrence ceux que l’on a implémenter :

* Firebase Realtime Database
* Firebase Cloud Firestore
* Firebase Authentication

Pour être concis, l’installation des dépendances à été largement facilité avec notre IDE (Android Studio) qui nous proposent de connecter la plateforme de Firebase du site <https://console.firebase.google.com/> au projet actuellement sur l’IDE.

Ainsi Realtime Database, est un modèle de base de données en temps réel c’est à dire que les tables ainsi que les champs et occurrences ne sont pas prédéfini et variable, or par exemple les champs on très avoir deux occurrences ne sont pas obligés d’avoir les mêmes champs. Cela nous a permis de manière sécurisée de mettre en place un système de création de salon dans lequel plusieurs utilisateurs (ici deux) peuvent rejoindre et interagir selon ce que l’on a mis dans le layout.

Ensuite Firebase Cloud Firestore peut être considéré comme mySQL, en effet ce dernier à contrario de Realtime Firebase doit avoir des champs et des tables prédéfinis sinon des erreurs de champ null par exemple peuvent être rencontrées. Nous l’utilisons entre autre pour garder en mémoire les utilisateurs couplé à Firebase Authentication qui est permet d’ajouter un système d’authentification sécurisé avec le mot de passe hashé, cela revient à une utilisation optimal de l’API.

**Bonus(si possible):**

Fonctionnalité n°4 :

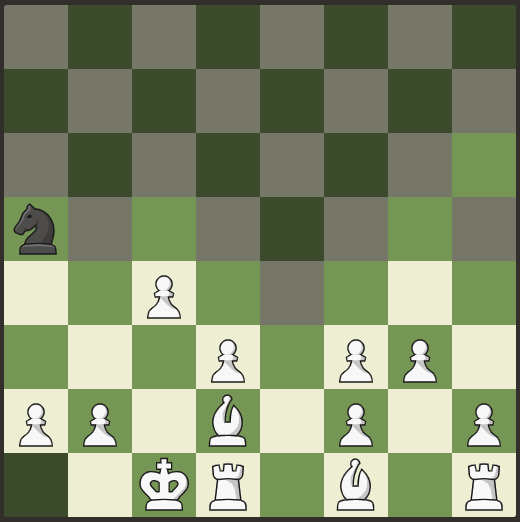
Objectif :

Mettre en place des mode alternative d'échec (brouillard de guerre)

Description :

L' utilisateur pourra jouer à différents modes pour varier les plaisirs.

Niveau de priorité : bas



Fonctionnalité n°5 :

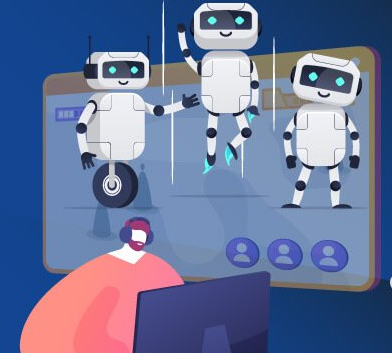
Objectif :

Mettre en place un bot

Description :

L' utilisateur pourra jouer contre un bot .

Niveau de priorité : bas



Fonctionnalité n°6 :

Objectif :

Appli Responsive

Description :

Avoir une application responsive pour s’adapter à chaque téléphone

Niveau de priorité : moyen



**Source/Documentation utile :**

<https://www.youtube.com/watch?v=nj-trslVFPc>

<https://firebase.google.com/docs/android/setup>

<https://firebase.google.com/docs/database/android/start>

<https://console.firebase.google.com>

<https://devstory.net/10525/comment-ajouter-des-bibliotheques-externes-a-android-project-dans-android-studio>

<https://firebase.google.com/docs/database/security/get-started?authuser=0>

<https://openclassrooms.com/fr/courses/4872916-creez-un-backend-scalable-et-performant-sur-firebase/4982770-ameliorez-la-gestion-des-utilisateurs-avec-firestore>