



a gamers meeting place

Compte rendu de réalisation de projet Android

Mise en contexte

Cette application mobile a été développée durant notre deuxième année de master développement logiciel et ingénierie des données au sein de l'Université Gustave Eiffel. Sous la tutelle de M.Chilowicz, enseignant chercheur au sein de l'Institut Gaspard Monges et également notre professeur de l'unité d'enseignement "Android front end".

Notre groupe, chargé de développer le projet Lootin, était composé de 6 membres à savoir : Pierre-Jean Besnard, Louis Billaut, Thomas Calonne, Jeanne Crohare, Armand Liegey et Thomas Juillard.

Ce rapport débute par la présentation du projet, les exigences initiales du corps enseignant, la présentation du sujet et les objectifs fixés par le groupe. Il décrit ensuite quels choix technologiques et structurels ont été faits afin de développer cette application. La troisième partie du projet s'articule autour du groupe, les méthodes de gestion de projets mises en place durant le développement de celui-ci et le rythme adopté. Enfin, ce rapport s'achève par une présentation de l'application, les fonctionnalités disponibles sur l'application finale ainsi que les dispositifs additionnels qui pourraient y être apportés. Une prise de recul est enfin décrite, afin de présenter les apports personnels et professionnels perçus par l'ensemble des membres du groupe.

I. Initialisation du projet	4
1. Rappel des exigences	4
2. Objectifs initiaux	5
3. Etude de l'existant	6
4. Sites déjà existants	6
5. Des applications de rencontre mobiles	7
6. Les rencontres, un marché de "niche"	8
7. Définition des features en versions et maquettage de l'application	9
II. Choix architecturaux et détails techniques	13
1. Architecture technique	13
2. Parcours des requêtes http au travers le service back-end	14
3. Mode d'authentification application - service	14
4. Schéma de la base de stockage	15
5. Utilisation des channels websockets via le protocole stomp	17
III. Développement du produit	19
1. Répartition du travail	19
2. Pattern de versioning git	22
3. Rythme de travail adopté	22
IV. Produit final	23
1. Fonctionnalités disponibles et services additionnels	23
2. Difficultés rencontrés	25
3. Améliorations possibles	26
4. Prise de recul	27
5. Liens et sources externes	27

I. Initialisation du projet

1. Rappel des exigences

Durant l'attribution des sujets de développement de l'unité d'enseignement Android, il était laissé libre aux élèves de travailler sur un sujet de leur choix, choisi en alternative au sujet proposé initialement par le professeur. Cependant, le sujet alternatif devait respecter un certain nombre de conditions afin d'être validé par le corps enseignant.

Le projet proposé devait donc :

- comporter une étude des applications existantes présentant des similarités avec l'application projetée
- introduire une part importante d'originalité
- prolonger un sujet de projet donné dans une autre matière

En terme de contraintes techniques, l'application devait également :

- comporter au moins deux activités
- posséder au moins un service
- avoir également au moins un *BroadcastReceiver*
- utiliser la géolocalisation et/ou les différents capteurs de l'appareil
- être développée en langage Java et/ou Kotlin en utilisant la bibliothèque standard Android tout en étant compatible avec le maximum d'appareils possible en utilisant le *minSdk* le plus bas possible
- être réalisée selon une approche native Android, c'est-à-dire ne pas codée à l'aide de technologies web (HTML5, JavaScript...)

2. Objectifs initiaux

Le marché des applications mobiles de rencontre étant en plein essor en 2021, le développement d'une application spécialisée dans un certain domaine nous parut être une idée intéressante tant le développement de celle-ci nous rapprocherait d'une application commercialisable et en phase avec le marché.

Lootin est donc une application de rencontre pour personnes au profil dit "gamer" c'est-à-dire adeptes des jeux vidéos. L'application se concentre donc autour de la niche du ludiciel, secteur très en vogue. Sa spécificité réside dans le fait de mettre l'accent sur la personnalité des profils avant leur physique. Pour cela, Lootin dont le nom dérive du verbe anglais très utilisé par la communauté gamer "loot" signifiant "piller", se focalise sur ce qui rapproche les gamers : les jeux qu'ils aiment pratiquer. Ainsi, les recommandations de profils devront s'effectuer en fonction des jeux vidéo en commun qu'auront les personnes inscrites sur l'application.

Nos ambitions pour l'application étaient tout d'abord de proposer à l'utilisateur de sélectionner les jeux auxquels celui-ci joue au moment de son inscription au service. Nous tenions donc à ce que l'application présente une interface de navigation entre les profils recommandés par l'application. Nous souhaitions également incorporer une option permettant aux utilisateurs ayant une attirance commune d'échanger des messages textuels et photos.

Enfin, toujours dans l'optique de proposer une application en phase avec le marché, un soin particulier devait être apporté à l'interface de l'application qui devait être ergonomique et responsive, la charte graphique (code couleur, logo) ainsi qu'un vocabulaire propre à l'univers de l'application.

3. Etude de l'existant

Afin de mieux situer l'état du marché concernant les applications de rencontre pour adeptes de jeux vidéo, une étude préalable a été effectuée sur le store Apple Play. Cette étude devait également permettre de définir le potentiel du marché ciblé, les features intéressantes à conserver pour le développement de l'application Lootin ainsi que les fonctionnalités manquantes sur les produits déjà disponibles sur le marché.

Introduction

Le marché du site de rencontre est un marché déjà bien implanté et très vaste en France: il existe près de 2000 sites. Il s'agit d'un secteur florissant, un quart des Français les utilisent de manière très régulière, et surtout les 18-34 ans, qui sont plus de 40 % à être inscrits et 25 % à avoir payé pour des offres premium sur certains sites.

Sites déjà existants

Selon l'INSEE, le nombre de célibataire est en perpétuelle croissance car les mœurs de notre société ont changé, les personnes se marient de plus en plus tard et on voit le nombre de divorces qui ne fait qu'augmenter. Malgré cela, il semblerait que le nombre d'inscrits sur ce genre de sites n'augmente pas à la même vitesse que le nombre de célibataires. On compte aujourd'hui en France environ 9 millions d'utilisateurs pour environ 2000 sites de rencontres.

Parmi les sites les mieux implantés on peut compter :

- Meetic.fr
- Adopteunmec.com
- Gleeden.com
- Edarling.fr
- Attractiveworld.net

Tous ces sites sont basés sur le même principe: proposer un site de «dating» c'est-à-dire permettre aux célibataires de se rencontrer en leur proposant des services tels que des questionnaires dédiés à la recherche amoureuse. Les utilisateurs peuvent donc rédiger des annonces, ajouter des photos et répondre à certains critères afin de faciliter la rencontre entre des personnes qui pourraient se correspondre. Ce premier procédé correspond à la première

offre que l'on peut trouver sur ce marché, c'est-à-dire celle des sites de rencontres classiques où le but principal est de «chatter» avec des individus qui pourraient nous plaire.

Des applications de rencontre mobiles

Sur ce marché, on retrouve également les applications de rencontres qui proposent une interface sur mobile. Parmi les applications les plus utilisées en France, on peut compter :

- Badoo
- Tinder
- Happn
- Lovoo
- Adopte Un Mec

A titre d'exemple, au premier trimestre 2019, l'application de rencontre Tinder devenait l'app la plus rentable avec un chiffre d'affaires de 260 millions d'euros détrônant Netflix, en tête depuis trois ans, selon une étude menée par Sensor Tower.

Les jeunes sont de plus en plus séduits par les applications de rencontre : plus d'un jeune âgé de 18 à 25 ans sur quatre fréquente ces applications en 2016, contre 19 % des 26-30 ans et 13 % des 31-35 ans. L'arrivée des applications mobiles, qui facilitent la rencontre de par leur prise en main rapide, type Tinder, sur le marché a certainement joué sur leur attrait pour la rencontre en ligne.

Les rencontres, un marché de “niche”

De plus, il y a des sites de rencontres spécialisés qui proposent des offres beaucoup plus spécifiques qui ciblent une population bien précise. Souvent ces sites sont appelés «sites niches». Parmi ces niches on peut compter par exemple:

- Sites de rencontre gay
- Sites de rencontres pour senior
- Motorsport dating
- ...

Et Lootin ?

Lootin souhaiterait s'implanter sur un marché de niche, Lootin souhaiterait donc cibler une niche: celle du jeu vidéo, en proposant de rencontrer des "gamers". Quelques rares applications mobiles, centrées autour de la rencontre d'amis avec lesquels jouer sont déjà disponibles sur le Google Play Store :

- PLink
- Lanslot

Fonctionnalités retenues sur l'ensemble des applications

La plupart des applications proposent un système de navigation entre les profils potentiellement compatibles, ainsi qu'un système de partage de messages entre utilisateurs se plaisant mutuellement, cette fonctionnalité sera donc implémentée dans l'application Lootin. Des notifications sont envoyées à chaque message aux utilisateurs dans la plupart des applications citées précédemment, ce qui sera également implémenté dans notre projet.

Enfin, aucune application ne centre ses recommandations de profil autour des jeux en commun des utilisateurs, fonctionnalité manquante qui sera également présente dans Lootin.

4. Définition des features en versions et maquettage de l'application

L'application Lootin devait donc être composée d'un service back-end Java gérant les services de notifications, de recommandation des profils, et d'échange avec la base de données et d'une application mobile développée en Kotlin, permettant à l'utilisateur de profiter des fonctionnalités décrites ci-dessous.

Conformément aux attentes du professeur, ce service établira la jonction entre la matière Android et l'unité d'enseignement Back-End et respectera les

contraintes d'architecture REST facilitant, entre autres, la scalabilité du service back-end en cas de nécessité de montée en charge.

Un découpage en versions des fonctionnalités de l'application a été effectué afin de définir clairement les objectifs du groupe quant au développement de Lootin.

V1: 01/02/21

- inscription et connexion à l'application
- formulaire à remplir à l'inscription (informations afin de construire un profil et cibler les préférences de l'utilisateur : nom, âge, genre, jeux)
- possibilité de "looter" des profils qui nous ressemblent avec qui nous voudrions échanger ou de passer au profil suivant avec l'option "drop it"
- interface pour parler à nos matchs, possibilité d'envoyer des messages via un channel websocket

V2: 31/03/21

- notifications à l'utilisateur quand une correspondance avec son profil est détectée ou un message est reçu grâce à un Service Android connecté à un channel websocket
- ajout de la prise de photo ou d'une sélection depuis la galerie du téléphone à l'inscription
- modification des préférences, photo et goûts utilisateurs.
- envoi de photos prises avec le téléphone, et des photos qui sont déjà en mémoire sur le téléphone grâce aux protocoles REST et WebSocket
- proposition de mise à jour de la liste des matchs lorsqu'un nouveau est disponible à l'aide d'un broadcastReceiver

V3: 01/06/21

- déléguer à Facebook et Google la connexion à l'application, ce qui permet de faciliter l'inscription
- palette des couleurs variante sur l'application, variations en fonction des données du profil
- ajout des préférences par rapport aux signes astrologiques
- ajout de la fonction: "programmer une partie" permettant de proposer à

ses matchs des rendez-vous sur un jeu vidéo en particulier, si accepté, le rendez vous s'ajoute automatiquement dans l'agenda de l'utilisateur.

V4: fin 2021

- version IOS

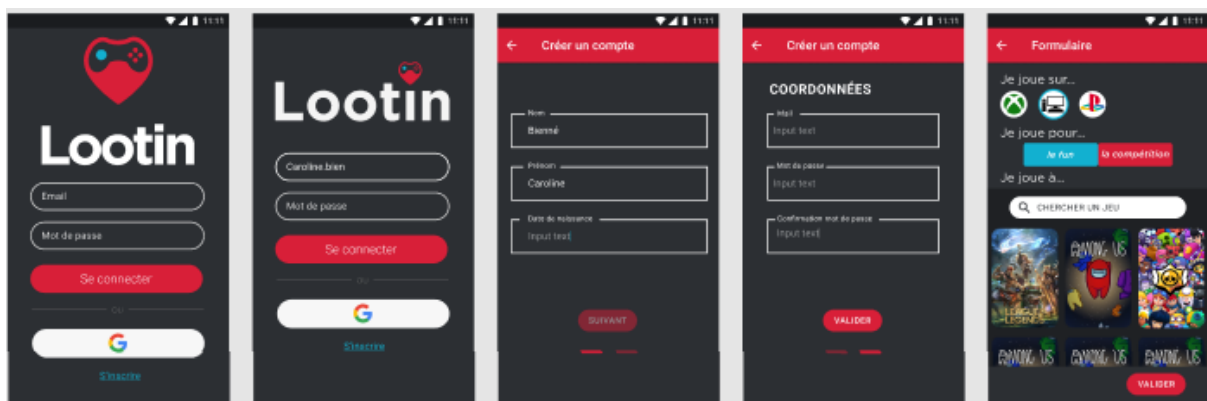
Le développement de Lootin ne se limitant pas nécessairement aux contraintes temporelles imposées (31/03/21), il était important pour le groupe de prévoir le contenu de potentielles versions à implémenter après la date de rendu du projet. Ces versions "d'après rendu" seront développées par les membres souhaitant continuer le développement de Lootin dans l'optique d'une mise en ligne sur les magasins d'applications en ligne des terminaux Android et IOS.

Maquettage de l'application

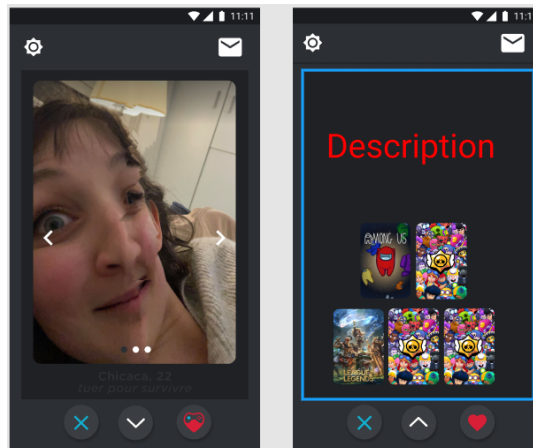
Comme explicité plus haut, un travail attentif fut effectué afin d'obtenir une application dotée d'une charte graphique cohérente avec l'esprit Lootin. De ce fait, des maquettes de l'application ont été créées afin d'orienter les développeurs dans le design de leurs activités et fragments.

Ainsi, plusieurs maquettes furent produites :

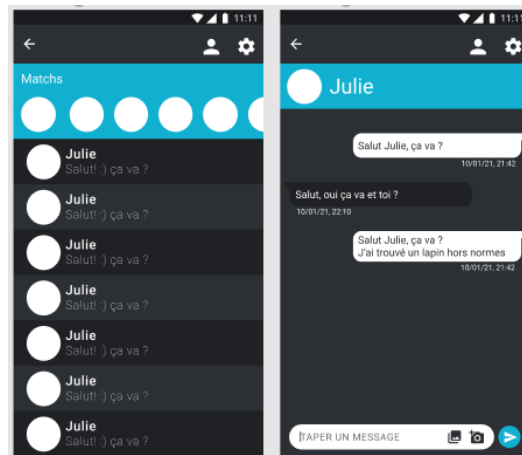
Ecrans d'inscription :



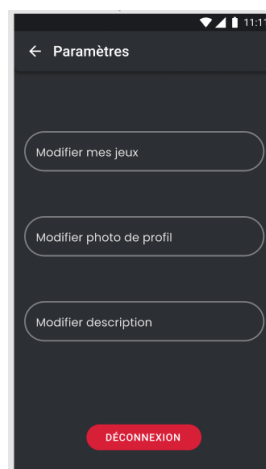
Ecrans d'activité de navigation entre profils :



Écrans de visualisation de matchs et d'envois de messages :

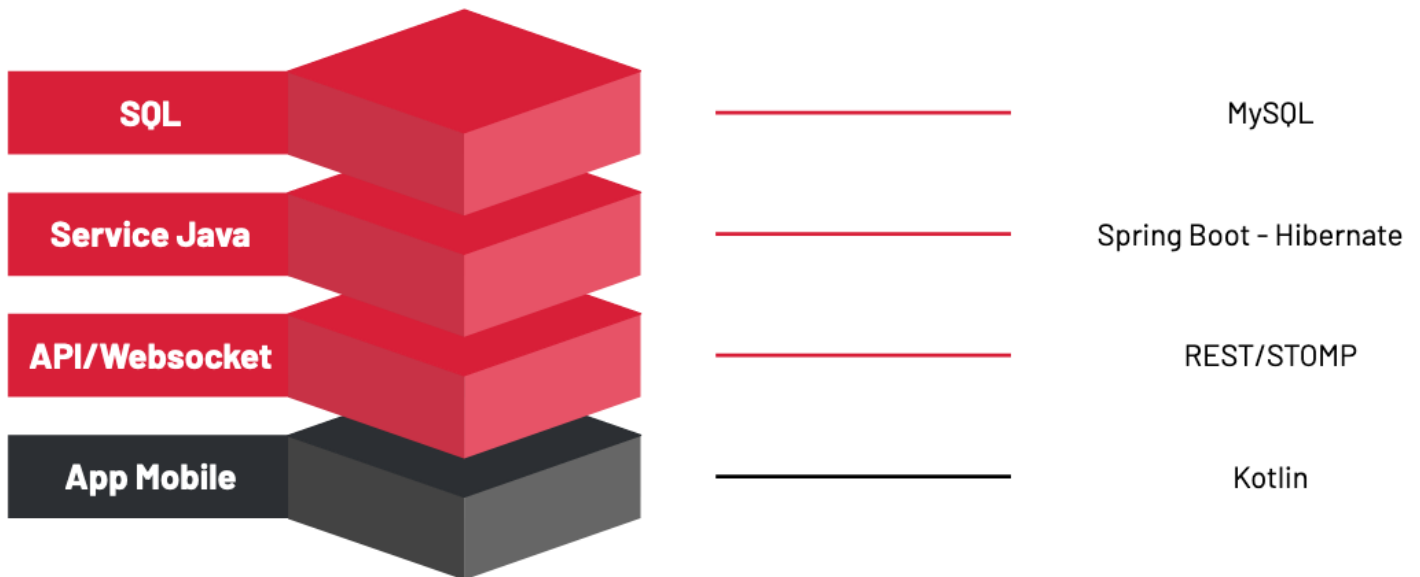


Ecran de modification des paramètres utilisateur :



II. Choix architecturaux et détails techniques

1. Architecture technique



Le schéma suivant décrit les choix architecturaux de l'application Lootin.

L'application est divisée en deux parties: la partie back-end (en rouge sur le schéma) et la partie front-end (en gris sur le schéma).

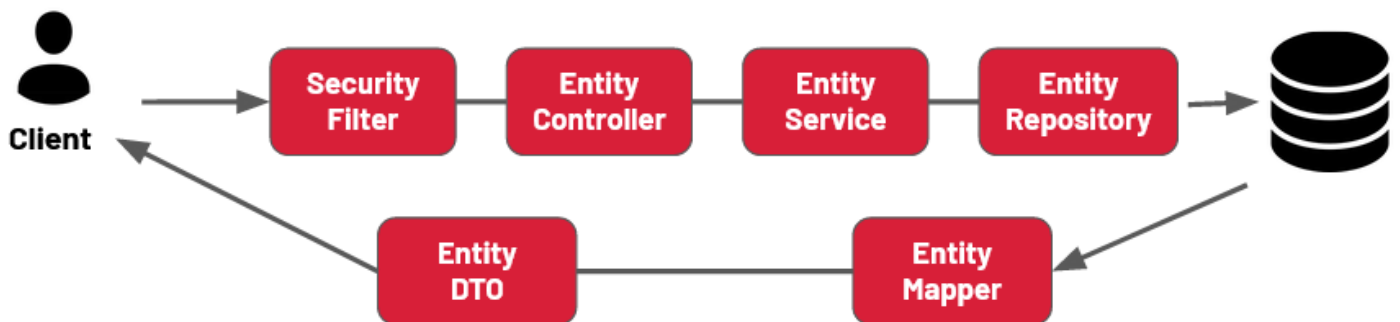
Concernant la partie back-end, nous l'avons développé en Java, avec le framework Spring-boot ainsi que le framework Hibernate. Cette application back-end permet la communication entre une API Rest ou des Websockets et la base de données.

Cette base de données est codée en MySQL. Quant à l'API, elle suit une architecture RESTful. L'application est également accessible via un protocole STOMP. Celui-ci permet, entre autres, de gérer les messages de l'application. Concernant l'application mobile, elle peut être utilisée sur des appareils Android et est développée en Kotlin.

2. Parcours des requêtes http au travers du service back-end

Le parcours d'une requête utilisateur depuis le client, c'est-à-dire notre application Lootin sur terminal Android, effectué via l'api REST et à travers les différentes étapes clés du micro service complémentaire à l'application Lootin est décrit par le schéma ci-dessous.

Les détails à propos de l'identification de l'application au sein du service back-end sont décrits par la section suivante.



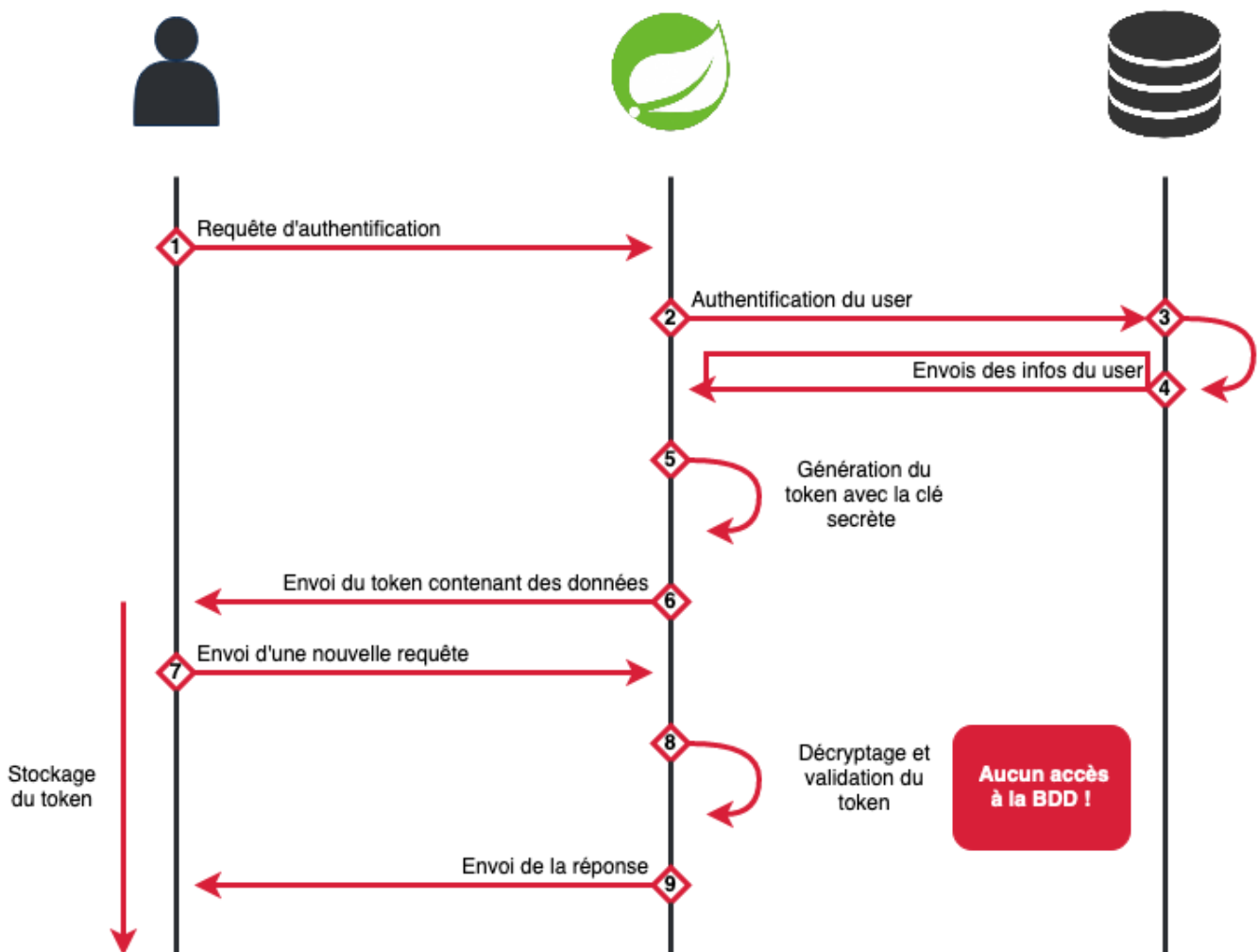
3. Mode d'authentification application - service

La méthode d'authentification utilisée par notre application back-end est le JWT Token. Cela permet d'avoir un système d'authentification "stateless", les informations des sessions ouvertes côté application Android ne sont pas stockées côté serveur. Cela nous permet, dans un premier temps, un gain de mémoire mais cela permet également aux utilisateurs de pouvoir se connecter sur plusieurs appareils et enfin, d'assurer la montée en charge de l'application.

sans problèmes liés à un potentiel système d'authentification par sessions. Les JWT de notre application contiennent donc l'id et le login.

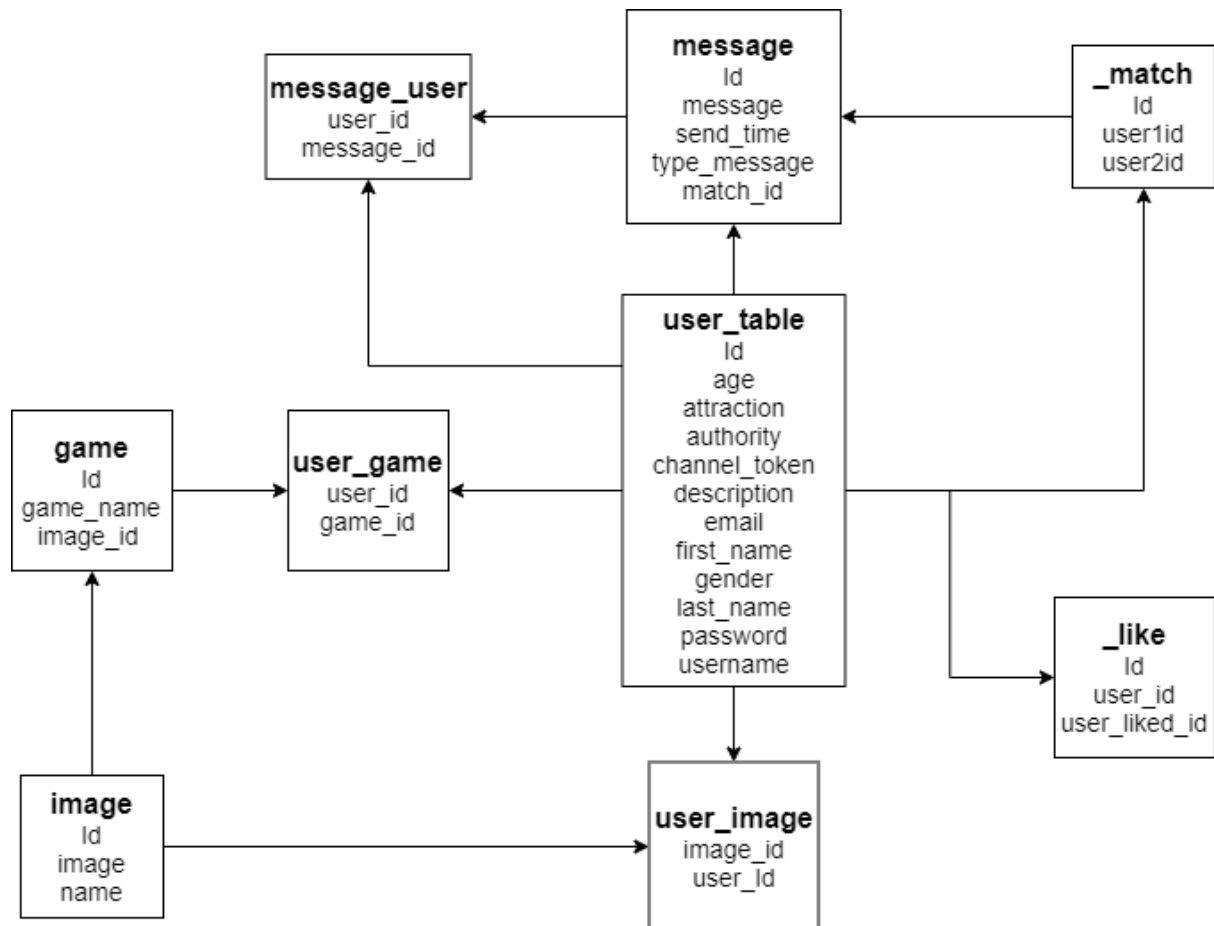
Ces JWT sont envoyés en header de chaque requête envoyées vers l'API REST nécessitant une authentification. Le serveur n'aura donc pas à faire un accès en BDD afin de récupérer les informations de nos utilisateurs.

Le schéma suivant illustre son fonctionnement dans notre application back-end:



4. Schéma de la base de stockage

Voici ci-dessus le schéma de notre base de données. Sur ce schéma toutes les flèches représentent les relations entre les tables. Par exemple, on peut observer que la table “message” dispose d’un “match_id” et d’un “sender_id”.



Ces deux champs correspondent au match lié au message en question et l'utilisateur ayant envoyé ce message. Nous avons besoin de la table de jointure “user_game” car chaque utilisateur peut jouer à plusieurs jeux et que tous les utilisateurs peuvent jouer aux mêmes jeux. De plus les tables “_match” et “_like” commencent par un underscore. Like et match sont des mots réservés par MySQL on ne peut donc pas utiliser ces mots comme nom de table.

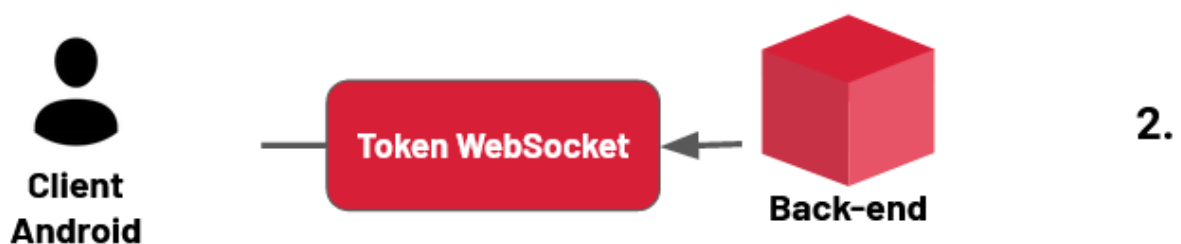
Dans une utilisation pratique de l'application les tables des user, des images ainsi que leur table de jointure seront remplis au moment de l'inscription. Celles des “_match” et “_like” seront remplis au moment du swipe des profils et les messages au moment de leur envoi par les utilisateurs. Par contre la table game ne pourra pas être remplie par les utilisateurs elle pourra seulement être renseignée par des administrateurs. Il faudra donc à la sortie de nouveau jeux les renseigner pour que nos utilisateurs puissent les choisir également.

5. Utilisation des channels websockets via le protocole stomp

L'utilisation du websocket fut requise afin d'établir le système d'envoi de messages instantanés entre utilisateurs (V1) puis de notifications de nouveaux loot et messages (V2). Le protocole stomp permettant d'échanger des messages instantanés fut choisi.

Ci dessous sont décrit les différentes étapes d'identification via l'interface REST puis les différents channels dédiés aux envois de messages et à la réception des notifications utilisateur :

- Récupération du token utilisé pour l'authentification websocket auprès de l'api REST.



- Ouverture de connexion et souscription aux channels, identifiée grâce au token récupéré précédemment



- Récupération des messages et des notifications en temps réel



III. Développement du produit

1. Répartition du travail

Voici la répartition des tâches entre les différents développeurs de Lootin.
Pour le développement nous avons chacun essayé de sortir de notre zone de confort en nous attribuant des tâches à l'opposé de nos points forts respectifs.

Code couleur:



Front - Application
Android



Back-end



Tâches diverse



Activité principale
(navigation entre profils)
Service de notifications



Construction de
l'architecture websockets



Structuration du back-end
Développement du
back-end de l'application



Formulaire d'inscription
Réglages
Gestion des capteurs
photo



Activité principale
(navigation entre profils)
Service de notifications



Implémentation des JWT
Token
Co-développement du
back-end



Construction de
l'architecture websockets



Ecran de connexion



Gestion des images en
back-end



Documentation technique



Gestion des chats et des
matches

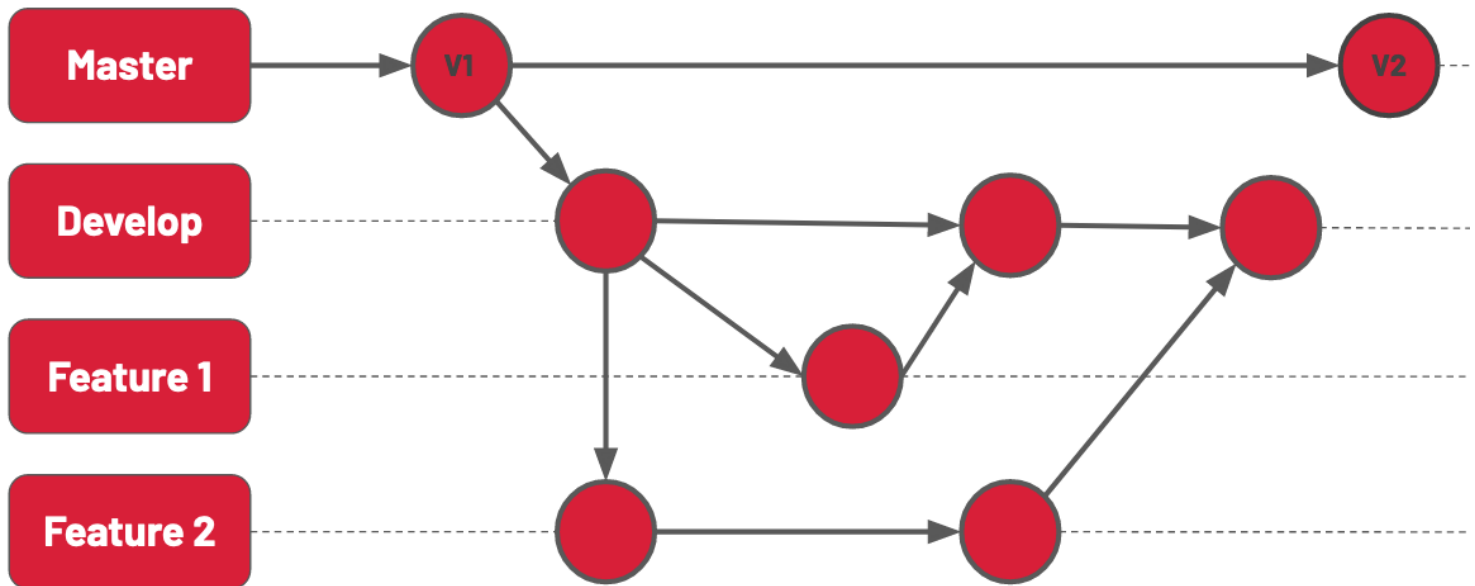


Création du dump BDD



Création des maquettes

2. Pattern de versioning git



Nous avons opté pour un pattern de versioning utilisant une branche par fonctionnalité.

Le groupe étant constitué de 6 développeurs, ce pattern nous semblait plus optimisé et plus adapté qu'un pattern découpant le travail en une branche par développeurs.

Lorsque nous avons terminé le développement d'une fonctionnalité, celle-ci était "merge" vers la branche develop.

Enfin, lorsque la version disponible sur develop était entièrement fonctionnelle et comportait toutes les fonctionnalités d'une version, nous la mergions vers master.

3. Rythme de travail adopté

Concernant le rythme de travail, des réunions hebdomadaires étaient organisées. Nous voulions notamment adopter un rythme de travail respectant les principes d'une méthode de gestion de projet Agile SCRUM.

Nous faisons donc chaque semaine, afin de respecter un rythme de sprint, une réunion qui nous permettait de faire un point sur la semaine passée, le travail réalisé, et les différentes difficultés rencontrées.

Nous fixions ensuite les objectifs pour la semaine suivante.

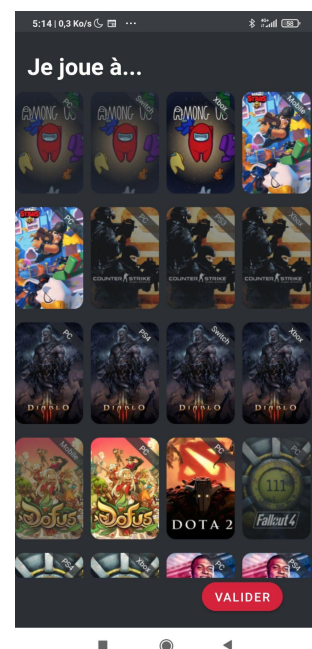
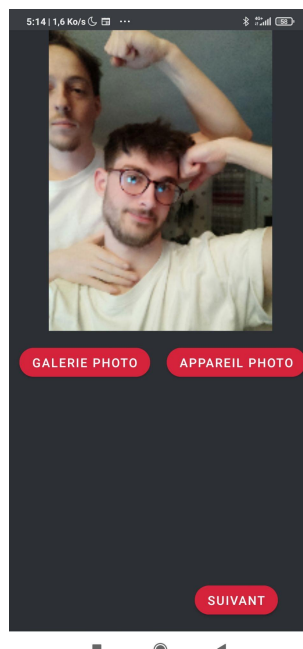
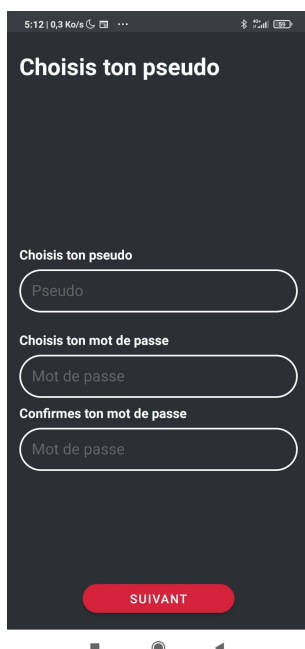
La communication au sein des différents acteurs du projet se faisait principalement via un serveur discord avec des channels de discussions organisés en sections afin d'avoir un accès à l'information plus efficace et une communication plus fluide.

Si des difficultés étaient rencontrées par un membre de l'équipe, les autres développeurs se rendaient disponibles au mieux afin de l'aider à surmonter ses éventuelles difficultés.

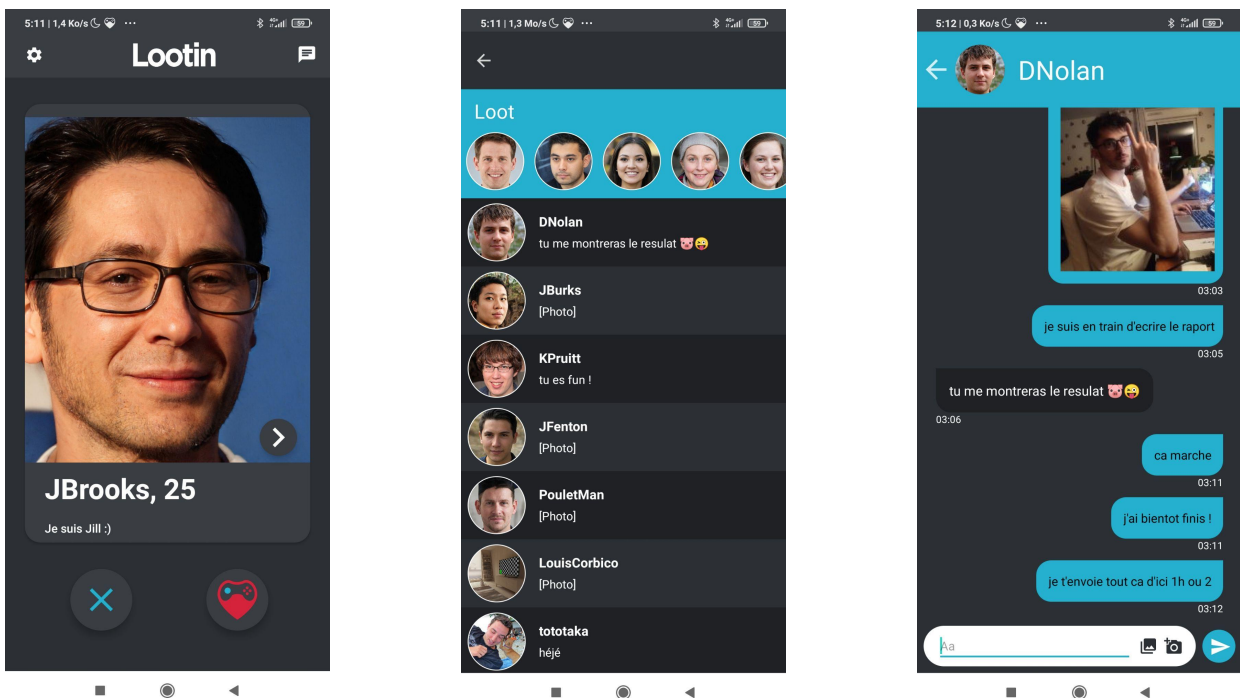
IV. Produit final

1. Fonctionnalités disponibles et services additionnels

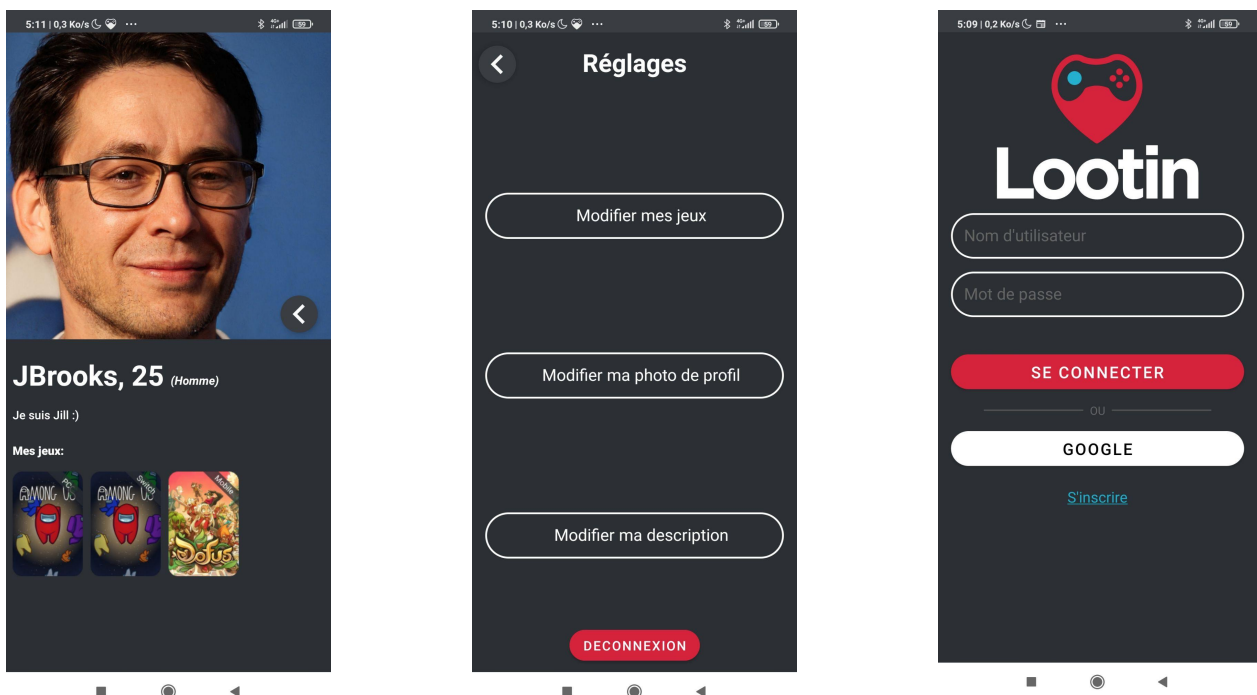
Quelques captures d'écran du formulaire d'inscription:



Ci-après des captures d'écran de l'activité principale, de la gestion des chats et du chat:



Ci-après, la visualisation complète d'un profil, les réglages et la connexion:



L'application comporte les fonctionnalités qui avaient été définies dans le cahier des charges, à savoir:

- Inscription: l'utilisateur peut définir son pseudo, mot de passe, son mail, son nom et prénom, sa date de naissance, son sexe, son attirance, définir une description, sélectionner une photo de profil, et définir les jeux auxquels il joue.
- Connexion: l'utilisateur entre son pseudo et mot de passe afin de se connecter à l'application.
- Activité principale, navigation entre profils: l'utilisateur peut "Looter" ou "dropper" les différents utilisateurs de l'application qui répondent à ses critères. Il a également la possibilité de cliquer sur le profil afin d'avoir une description complète du profil de l'utilisateur.
- Réglages: l'utilisateur a la possibilité de changer sa description, sa photo de profil et ses jeux. Il peut également se déconnecter de l'application.
- Gestion des loots et des chats: l'utilisateur peut visualiser ses "loots" et les différents utilisateurs avec qui il a une discussion en cours.
- Chat: l'utilisateur peut envoyer des messages et des photos à un utilisateur avec qui il a "looté"
- Notifications : l'utilisateur est notifié grâce à un service dédié, se lançant automatiquement au démarrage du téléphone via un broadcast receiver lorsqu'un nouveau match est détecté ou lorsqu'un autre utilisateur lui envoie un message.
- Notifications bis : l'utilisateur se voit proposer, à l'aide du service dédié et d'un second broadcast receiver, un rechargement des messages et des matchs, si celui-ci se situe dans le fragment d'affichage des matchs.

2. Difficultés rencontrées

Aucune difficulté majeure n'a été rencontrée au cours du développement de ce projet. Néanmoins, le développement du projet a quelquefois été ralenti par certains facteurs liés notamment à :

- la compréhension des websockets, non abordée durant le cours de d'Android, beaucoup de travaux de recherche et de documentation sur le web ont été nécessaires afin de pouvoir mettre en place ce système, nécessaire à la mise en place de nombreuses options proposées par l'application.
- la bonne coordination de tous les membres du groupe, la mise en place de méthodes de gestion de projet ayant eu un rôle majeure dans le bon développement du projet

3. Améliorations possibles

Comme précisé dans la section "Définition du cahier des charges final et maquettage de l'application", des versions ultérieures à la version présentée dans ce rapport ont été définies durant la phase d'élaboration du projet Lootin. Néanmoins, des fonctionnalités optionnelles peuvent également être envisagées pour le futur de l'application à savoir :

- L'utilisation d'une base de donnée NoSQL de type document comme MongoDB pour le service back-end, mettant l'accent sur l'accessibilité des données et étant plus adaptée à la scalabilité et la montée en charge de l'application.
- La mise en place d'un système de recommandation à l'aide d'une base de données graph comme Neo4j, permettant d'avoir un calcul plus rapide et plus pertinent des recommandations des utilisateurs.
- L'implémentation de jeux vidéo intégré à l'application, afin de permettre aux joueurs de s'affronter directement via Lootin.

4. Prise de recul

Cette section représente une prise de recul des membres du projet vis à vis de cette expérience de développement.

La totalité des membres du groupe sortent grandis de cette expérience, la volonté de s'investir dans un projet ambitieux et conséquent, en phase avec le marché, fut un élément clé de la réussite de ce projet. Le fait d'avoir distribué des tâches légèrement hors du champ de compétences de chacun fut bénéfique pour l'ensemble des acteurs du projet.

En termes de développement Android, notre culture et savoir ont grandement évolué. Aborder des notions complexes du système Android via le développement d'un projet tel que Lootin n'a pas son pareil en termes d'expérience acquise.

La gestion de projet fut également au centre du développement de ce projet, les méthodes mises en place, et adaptées au fur et à mesure de l'évolution du projet ont été comprises et appréhendées grâce aux leçons tirées de nos échecs.

5. Liens et sources externes

Voici la liste des langages et frameworks utilisés pour le développement du projet :

- [Kotlin](#)
- [Java](#)
- [Spring-Boot](#)
- [MySQL](#)

Le projet à été développé avec les dépendances suivantes:

- [Material Components](#)
- [Volley](#)
- [Gson](#)
- [Fragment](#)
- [NaikSoftware StompProtocol](#)
- [RxAndroid](#)
- [CicleIndicator](#)
- [CircleImageView](#)

Le projet Front (Application Android & Kotlin) est disponible sur [Github](#).

Le projet back-end (Application java Spring-Boot) est disponible sur [Github](#).

Ci-contre une liste des profils github des développeurs du projet:

- [BESNARD Pierre-jean](#)
- [BILLAUT Louis](#)
- [CALONNE Thomas](#)
- [CROHARE Jeanne](#)
- [JUILLARD Thomas](#)
- [LIEGEY Armand](#)