Projet Développement

Mobile 2023

Une image contenant texte, clipart

Description générée automatiquement

Réalisé par Popadiuc Claudiu

Supervisé par Monsieur Riggio, Jonathan

Table des matières

[Introduction 4](#_Toc134728944)

[Description des technologie utilisées 5](#_Toc134728945)

[Sujet d’application 6](#_Toc134728946)

[Fonctionnalités 7](#_Toc134728947)

[Analyse 15](#_Toc134728948)

[Limitations et développement futur 16](#_Toc134728950)

[Conclusion 17](#_Toc134728951)

# **Introduction**

Dans le cadre du cours de Mobile, il nous a été demandé de réaliser une application mobile qui sera développée uniquement sur Android Studio en utilisant le langage Java. Pour ma part, j'ai réalisé une application de gestion des matchs de football destiné aux arbitres, comprenant un historique des matchs ainsi que les profils des joueurs.

Dans ce rapport, je vais commencer par une description des technologies utilisées ainsi que mes motivations. Ensuite, je rentrerai dans le vif du sujet en expliquant le sujet de mon application suivi par les fonctionnalités de mon projet, ainsi que tout ce qui a été implémenté d'un point de vue à la fois fonctionnel et technique, en mettant en avant les défis techniques et les solutions apportées. Les points d'intérêt concernant la structure de mon implémentation seront ensuite présentés sous forme de diagramme.

Pour donner suite à cela, il y aura une section UI/UX qui détaillera les points d'attention apportés à l'UI/UX, ainsi que la façon dont les besoins et les préférences des utilisateurs ont été pris en compte.

Je continuerai mon rapport avec les limites de mon application et je répondrai à des questions telles que dans quels cas d'utilisation mon application pourrait ne pas fonctionner comme prévu ? Y a-t-il des aspects techniques non traités ? Si j’avais plus de temps pour le projet, qu'aurais-je amélioré ?

Je finirai ensuite mon rapport avec une conclusion en résumant ce qui a été vu dans le rapport, ce que j'ai réussi à faire ou non durant le projet, et les apprentissages que j'en tire. Pour finir, je donnerai mon avis et mon ressenti sur le développement de mon application.

# **Description des technologie utilisées**

Pour réaliser ce projet, j'ai utilisé exclusivement Android Studio comme environnement de travail. Android Studio est un environnement de développement intégré (IDE) utilisé pour créer des applications mobiles pour la plate-forme Android. Les langages de programmation possibles sont Java et Kotlin. Android Studio est largement utilisé par les développeurs du monde entier pour créer des applications mobiles pour la plate-forme Android.

Pour mon projet, j'ai utilisé uniquement le langage Java. Java est un langage de programmation populaire utilisé pour créer une grande variété d'applications, notamment des applications de bureau, mobiles, et web. Il est orienté objet, portable et largement utilisé dans l'industrie pour sa fiabilité et sa sécurité.

Dans Android Studio, le code Java est associé au code XML qui est disponible lors de la création d'une activité. XML est un langage de balisage utilisé pour décrire la structure et le contenu des données de manière structurée et lisible par les machines. Il est largement utilisé dans le développement d'applications web et d'autres domaines pour échanger et stocker des données.

# **Sujet d’application**

De manière simple et structurée, voici les fonctionnalités de mon application :

Le but de mon application est de permettre à l'utilisateur d’avoir sa propre saison de football avec les joueurs qui y participent. L'application permettra de créer des matchs et d'enregistrer l'historique de tous les matchs précédents. Chaque joueur pourra avoir son propre profil avec des statistiques telles que le nombre de défaites ou de victoires.

Une notification sera envoyée pour le chronomètre qui décompte le temps restant lorsqu'un match est joué, de sorte que l'arbitre n'aura pas besoin de se connecter à l'application pour consulter le temps restant. Une autre notification sera envoyée à l'arbitre une fois que le match sera terminé.

L'application sera également compatible avec les tablettes. Il y aura également une API qui permettra d'indiquer la position des joueurs comme pour les joueurs professionnels (ceci est expliqué plus en détail dans la section sur l'API).

Une carte sera disponible lors de la création de chaque match par l'arbitre, qui pourra ajouter le lieu où le match est joué. Dans l'historique, l'emplacement exact où le match de football a été joué sera disponible.

Voici le fonctionnement de l'application :

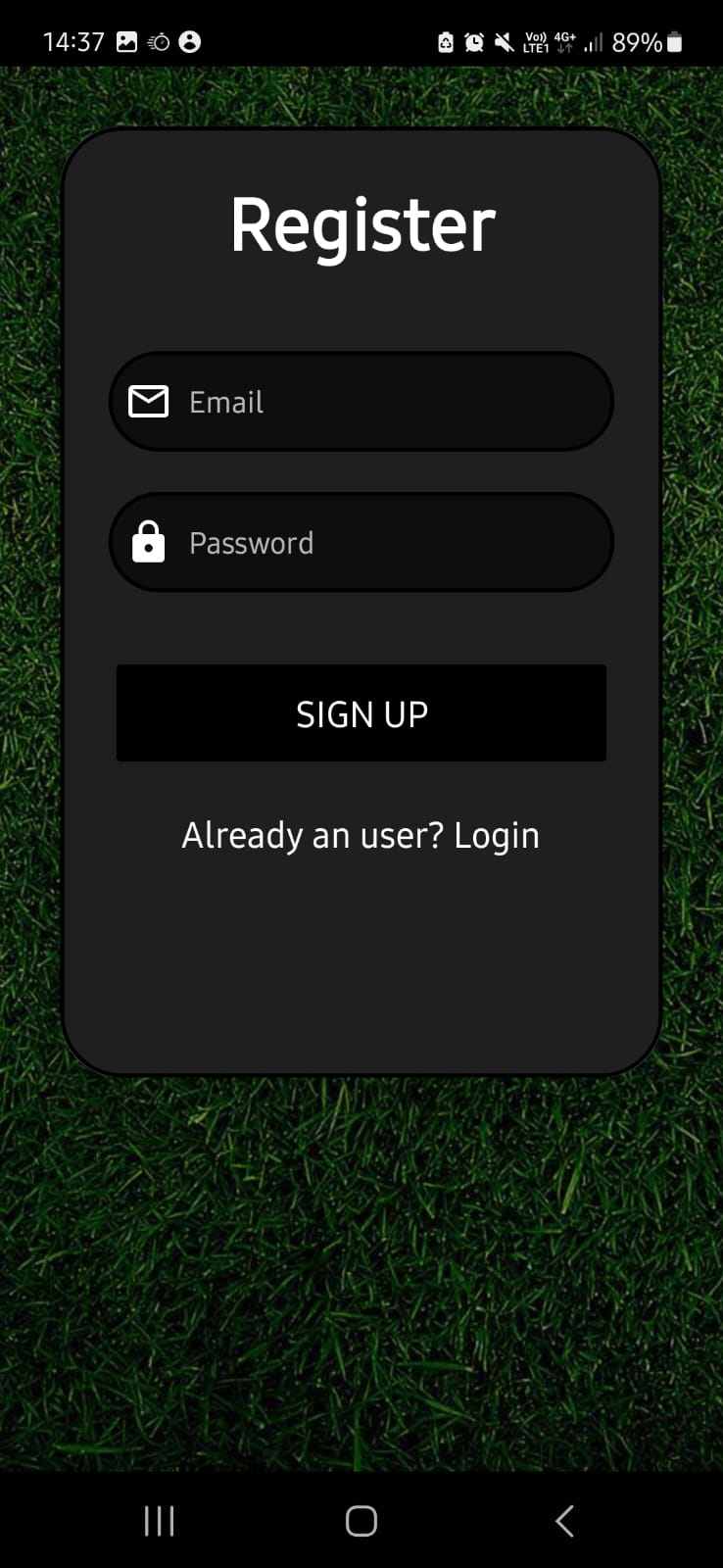
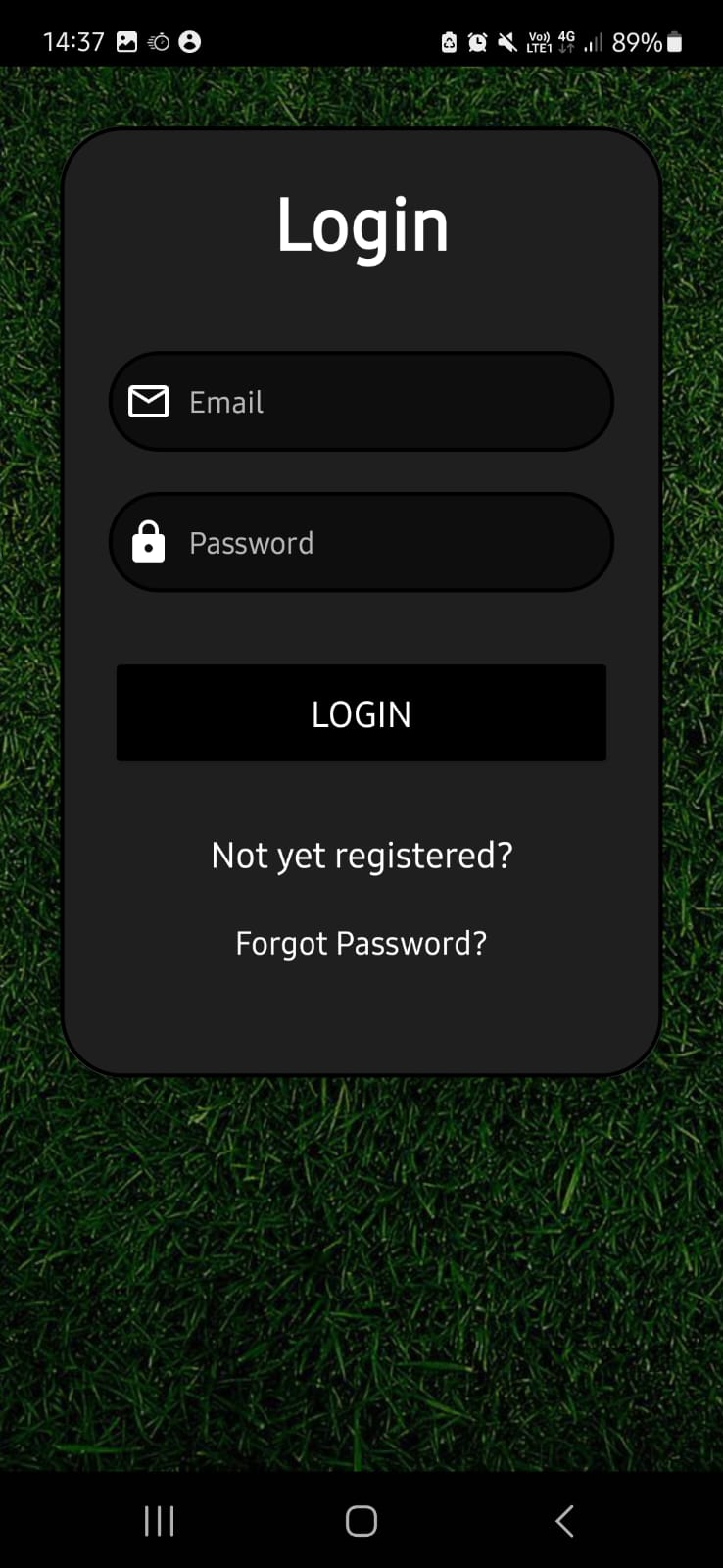
Un utilisateur poura s’inscire ou non dans l’application, Un utilisateur sera soit « Arbitre » soit « Joueur » S’il est arbitre, il pourra créer des profils pour les joueurs qui auront accès à l’application. L'arbitre pourra également créer des matchs qui seront enregistrés dans l'historique. La différence entre l’utilisateur « arbitre » et l’utilisateur « joueur » c’est que les joueurs ne pourront pas créer de matchs. Ils pourront seulement consulter l'historique des matchs ainsi que les profils des joueurs de la saison sans pouvoir modifier leurs statistiques. Seul l'arbitre pourra le faire.

# **Fonctionnalités**

**Le processus d’inscription/Connexion.**

Page d’acceuil : Page d’inscription : Page de connexion :

Une image contenant texte, arbre, plante, signe

Description générée automatiquement  

Sincèrement, je n'ai pas rencontré de difficultés pour implémenter le system de connexion et d’inscription. J'ai suivi un tutoriel sur YouTube qui m'a permis de comprendre et de développer cette fonctionnalité dont je vais parler plus tard. Au début, j'avoue que c'était compliqué et j'ai dû tester cela dans un nouveau projet avant de pouvoir l'implémenter dans mon projet actuel. Cependant, après quelques heures, j'ai réussi à faire fonctionner l'inscription et la connexion. Mais je vais rentrer dans les details pour cela.

Avant tout, il est important de préciser que mon application n'exige pas nécessairement une inscription. Cependant, si l'utilisateur décide de ne pas s'inscrire, certaines fonctionnalités de l'application lui seront inaccessibles (ce qui serait dommage, d'ailleurs). Par exemple, sur la page d'accueil, les boutons "Historique" et "Statistiques" ne seront disponibles qu'en étant connecté. Par conséquent, il sera obligatoire d'avoir un compte utilisateur pour profiter pleinement de l'application. Toutefois, si l'utilisateur souhaite réellement éviter de s'inscrire, il pourra créer un match qui ne pourra pas être sauvegardé.

Ainsi, si l'utilisateur désire d’avoir un avec un maximum de sensation sur l'application, il devra créer un compte. Comme vous pouvez le constater sur ma page d'inscription, il y a seulement deux champs à remplir : l'adresse e-mail et un mot de passe pour créer un compte. Sur mon application, c’est tres limité car l’application n’est pas poussée pour faire autre chose de plus complexe.

Pour mettre en place cette fonctionnalité, j'ai suivi un tutoriel disponible ici : <https://www.youtube.com/watch?v=TStttJRAPhE&t=650s>

J’ai dû utiliser Firebase qui est disponible directrement dans android studio dans les Tools, j’ai conencter mon projet avec Firebase, dans la partie authentification pour cette partie d’inscritpion avec mail et mots de passe :

Une image contenant texte

Description générée automatiquement



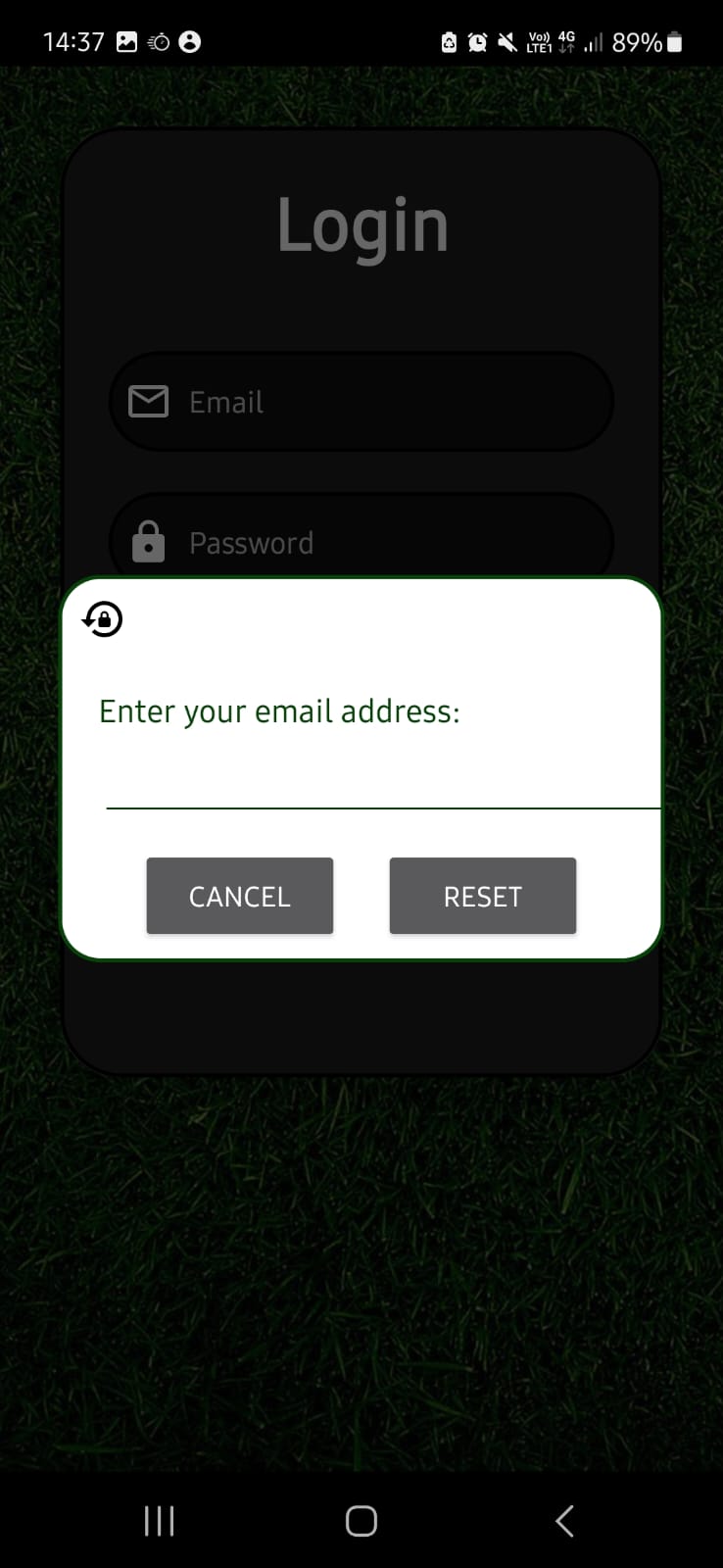
\*Afin de rester profesionnel, j’ai voulu respecter la vie privée de mes utilisateurs alors j’ai baré en bleu leur adresse mail.

Ce qui a été complique pour realiser cette fonctionalité c’est que j’aurais voulu utiliser la Real Time Database pour avoir mes utilisateurs dedans mais je n’ai pas réussi a le realiser, je suivis plusieurs tuto, j’ai couper mon proejt en plusieurs parties, mais je n’ai malheureusement pas réussi a faire cela, alors j’ai preferer rester sur le system de authentification qui est largement suffisant dans le cadre de mon application.

Pour rentrer dans les details, c’est lorsque j’appuie sur le bouton « Sign Up » que toute la partie de code se fera. Je vais d’abord verifier si le champs mail correspond bien à la strucutre mail avec « @ », je verifie aussi si le champs n’est pas vide ou alors si pour le mots de passe il ny’a pas au moin 8 caractere et si tout cela est bon alors on va ajouter un utilisateurs avec une adresse mail et un mots de passe dans le system de auhtentification.

Pour la connexion le principe sera legerement different car il verifiera si l’email et le mots de passe encoder fait bien parties des utilisateur disponible dans le system de authentification sinon il renvera un Toast.

Si l’utilisateur n’arrive plus a se connecter car il possiblement oublié sont mots de passe, un system de recuperation de mots de passe est disponible dans la page de connexion, a voir dans la page. Voici ce qui apparaitra lorsque j’appuyerais sur « Forgot password ? » :

 Une image contenant texte

Description générée automatiquement

Un email de récuperation de mots de passe sera alors envoyé, apres avoir tester cette fonctionalité, ça fonctionne bel et bien.

**Les fonctionnalités propres à mon application**

**Création de match**

TeamSelection : Team : Live :

 Une image contenant texte

Description générée automatiquement 

Voici le fonctionnement de la création de match :

Après avoir appuyer sur le bouton « New Game » qui est dans la page d’acceuil, on arrive sur la premiere page qui demande tout simplement quel type d’equipe desirer vous, si c’est random alors on arrivera sur la page Team, et dans cette page ci, on devra simplement selectionner des joueurs qui sont disponible dans mes statistiques, si on clique sur un des joueurs ca va l’ajouter dans le cadre « Players » et se suprimera de la liste des joueurs disponible pour indiquer a l’utilisateur qu’il a été selectionner et pour pas avoir de doublons, si l’utilisateur desir ajouter un joueur qui ne se trouve pas dans les statistiques pour ce match, alors il poura l’ajouter manuellement en appuyant sur le cadres « Players » et en ecrivant le nom du joueurs suivi d’une virgule comme indique juste au dessus du cadre. Une fois cela fait, l’utilisateur devra ajouter un temps de jeux pour chaque mi temps et pour la pause, qui ne pourra pas depasser 2 chiffres, donc une mi temps ne peux pas durer plus de 99 minutes, meme chose pour la pause, afin d’eviter des nombres abérant comme 45125 minutes pour une pause.

Si on n’ajoute pas au minimum 2 joueurs, et qu’on n’indique pas le temps de jeux pour chaque champ, alors le match ne peux pas commencer ce qui est logique, il n’ya aps de match sans temps de jeux et pas d’equipes sans au moin 1 joueurs dans chaque equipes.

Une fois tout cela bien rempli, nous arrivons sur la derniere partie qui est le Live du match, c’est ici que l’arbitre poura ajouter le nombre de goal en cliquant sur les balons de football pour chaque equipe, ou par exemple si les equipes qui ont été tirer au sort ne sont pas asse bien alors il y a le bouton « Refresh » en bas a guache qui permet de relancer le sort et faire 2 nouvelles equipes avec les joueurs qui on été choisi pour jouer le match, et remets les valeurs des goals a 0. Si tout est bon le match peux commencer en cliquant sur le bouton « Start » qui la est mis sur « Pause » car le match est en train de se dérouler et le temps défile déjà, et on peux mettre le chronometrer sur pause si on desire mettre sur pause le match.

Le match se deroule calmement, une fois que le match est terminé, il poura etre enregister en cliquant sur le bouton en bas a droite, qui demandera via un Builder s’il est sur d’enregistrer le match maintenant, car si l’utilisateur le souhaite il peux enregistrer un match avant que le temps de jeux soit terminée.

En haut a droite du Live il y a une icone qui permet de aller directement dans l’historique des matches joué, si l’utlisateur va dans cet historique le match en cours sera suprimer si il n’est pas enregistrer, alors un builder est aussi envoyer pour indiquer si il veux quiter le live sans enregistrer par contre si le match est enregistrer, le builder n’est pas envoyer.Il faut savoir que pour le system d’equipes « Selected » reprend le meme principe que pour le « Random «  sauf que ca ne va pas melanger mais, il y aura 2 champs dans la séléction, un pour chaque equipe et les joueurs pouront etre ajouter de la meme facon que pour le random.Pour le live du match « Selected » le bouton en bas a gauche ne permetera pas de re randomiser les equipes comme pour les equipes Random car les equipes seront déjà choisi par l’arbitre lui-même, donc pas besoin de faire ca, par contre si on appuie sur ce bouton les goals seront aussi remis à 0.

Meme si on ne peux pas le voir comme cela, mais il y a des animations entre les activiter par exemple on peux appuyer sur Random ou slider vers la gauche comme la fleche l’indique de meme pour Slected.

D’un point de vue thecnique tout cela n’est pas bien compliqué a comprendre, Pour la partie Random le champs d’equipe est un edit Text, pour les joueurs a selectionner c’est une listView avec comme Item tout les joueurs de pas base de donnée REAL TIME :

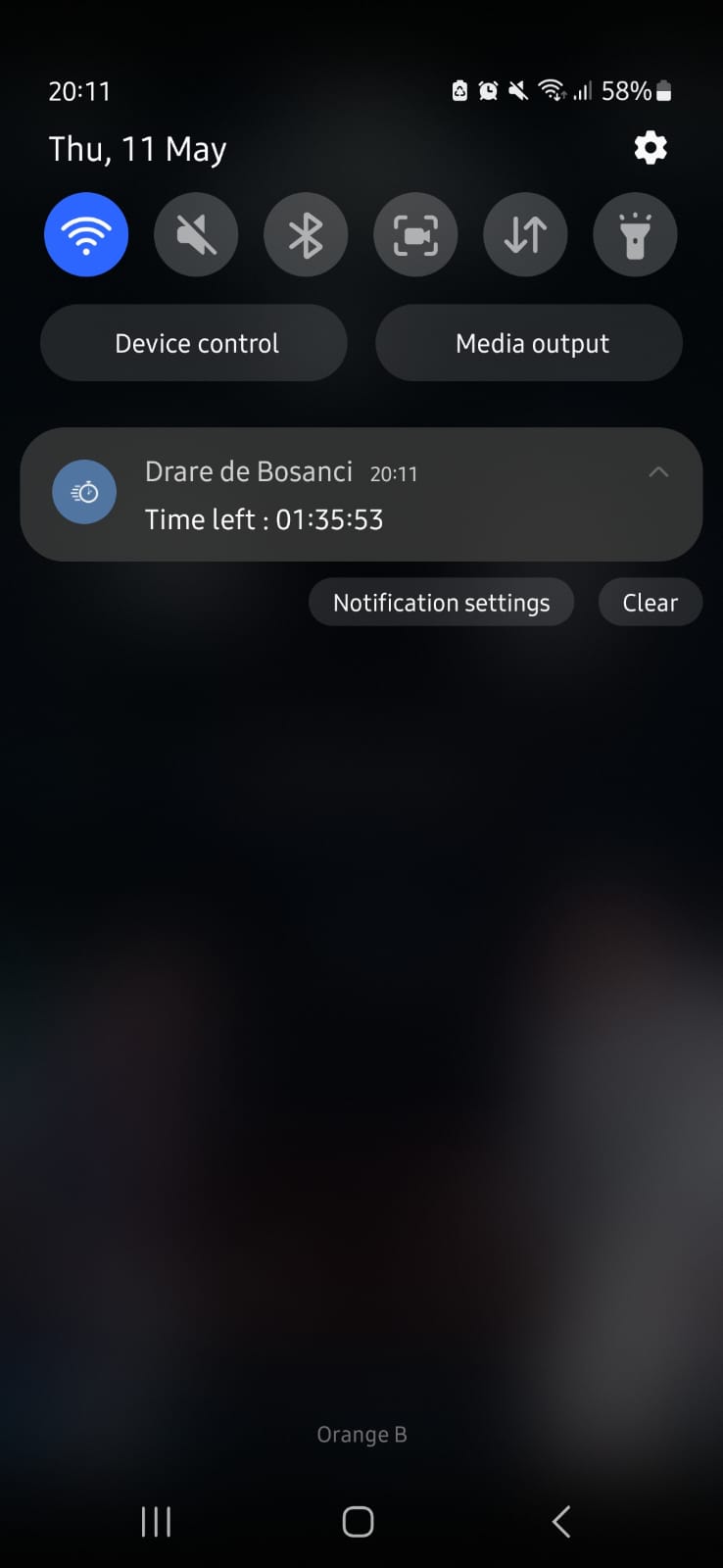
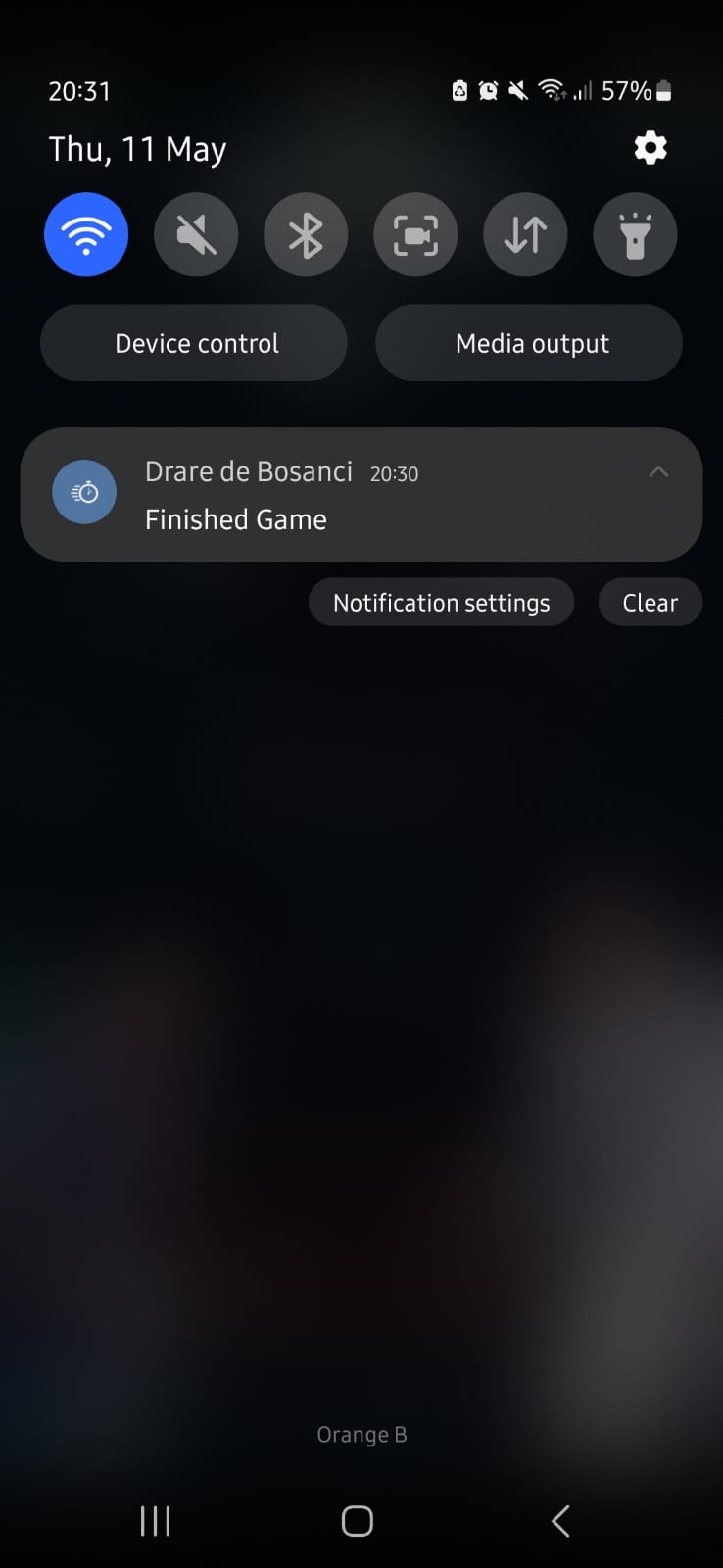
Une image contenant texte

Description générée automatiquement

Comme vous pouvez le voir ce sont tout les joueurs de mes statistiques, y seront ajouter dans ma list, et chaque item pourra etre cliquer afin d’etre repris dans l’edit text, ma liste a un Listener afin d’ecouter lorsque une modification a la liste est realsier et modifier la liste, donc quand j’appuie sur l’item il va suprimer le nom de la personne et grace a l’ecouteur, cela va fonctionenr sans problemes, enfin pour chaque item selectionner le text de l’item est simplement ajouter a l’edit text suivi d’une virgule, pourquoi ? car ensuite je vais mettre ce qui’il dans l’edit text dans une liste et la virgule me servira a diviser le nombres et les noms des joueur, cela sera donc fait lorsque on appuyera sur « Start », il mettra tout les joueurs dans une liste, et cette liste sera melanger grace Collection.Shuffle et ensuite on va diviser tout simplement en 2 listes la liste de base il y aura donc une liste avec les joueurs random par equipes . Le meme principe est applique pour les equipe predefini mais il n’yaura pas de random, ce sera directement le champs equipe 1 avec sa propre liste et de meme pour l’equipe 2, chaque equipe aura sa propre liste. Et une fois que le match commence chaque liste sera simplement afficher dans un TextView qui est celui en dessous du temps, la ou se trouve les noms, dans la page Live dans le grand cadre, ainsi chaque joueur sera affiche dans son equipe et comme vous pouvez le voir dans le screen si il y a un nombre impaire cava quand meme créer des equipes avec 3 et 4 joueurs et pas faire des 2 et 6 joueurs, ce qui est pratique pour garder des equipes equitable si c’est random.

Pour les champs du temps, j’ai ajouter une contraite qui permet d’ajouter que 2 chiffres maximum grace a un ecouteur de temps qui verifie a chaque fois que un chiffre est ajouter dans le champs la longeur de la chaine qui doit etre inferieur a 2, j’ai egalement ajouter un input de type chiffre dans ces champs la, ca veux dire que les lettres ne seront pas permise dans ces champs la.

**Notifications**

Mon system de notification est tres simple, une notification est envoyée lorsque l'arbitre a créé un match et qu’on arrive dans la page du Live. Un chronomètre est alors décompté dans la notification. Lorsque le match est terminé, l'arbitre reçoit une notification indiquant que le match est fini, j’ai aussi ajouter le fait que si on appuie sur le bouton pause, le chronometre se mets en pause aussi dans la notification que dans la l’affichage de l’application.

Par contre dit comme ça, en effet on dirait ca aurait pris 1h pour le faire mais c’est l’une des fonctionalite obligatoire qui m’a pris le plus de temps car j’ai un plusieurs problemes, Tout d’aboir il fallait afficher le chronomètre dans la notification, celui que j’avais mi dans le textView en grand, et il fallait reussir a reprendre la valeur qui était dans ce textView la, et la reprendre et l’envoyer dans une notification, et une fois que cela a fonctionner, les problemes arrivait.

C'était très compliqué et cela m'a pris beaucoup de temps. En effet, lorsque le téléphone reçoit une notification, c'est simple, mais pour que la notification change à priori chaque seconde, il fallait envoyer une notification chaque seconde. Cependant, cela générait un grand nombre de notifications non-stop. Pour bien contextualiser, ça veux dire que j’envoyais une notification des que le chronometre changait donc vu que il changait toute les secondes alors chaque secondes une notification était envoyer, ce qui peux deranger enormement l’utilisateur

Finalement, j'ai réussi à résoudre ce problème avec une méthode appelée "onTick". Cette méthode était appelée à chaque fois que le minuteur était décrémenté, ce qui permettait de mettre à jour l'affichage de la notification sans la renvoyer sur le telephone ni faire vibrer ou sonner le téléphone à chaque seconde, la methode fonctionne de facon a seulement changer le text qui est dans la notification, et donc a chaque seconde la notification allait afficher le temps restant, mais ce qu’il fallait surtout bien faire, c’est que il fallait pas recuperer le TextView avec le temps chaque secondes car c’est ce que je faisait et donc des que y avait un changement dans le textview avec un ecouteur de temps, alors je renvoyais une notification. J’ai donc dû changer cela afin que la notification elle-même recupere ces valeurs afin que ce soit la notification qui gere la decrementation du temp et qui est ensuite similaire pour le TextView pour afficher le temps restant dans le Live.

J’ai aussi eu quelque souci avec la pause du chronometre, car j’avais essayer d’implementer une pause au cas ou le match devrait se mettre en pause, pour faire cela j’ai du

**UI/UX**

# **Analyse**

# **Une image contenant diagramme Description générée automatiquement**

https://lucid.app/lucidspark/08d9e82b-cc1e-4dfa-9072-0f2c53099b23/edit?invitationId=inv\_90f57c65-f4d5-495a-a05d-52dd38b079d1#

# **Limitations et développement futur**

# **Conclusion**