

**Rayane Rachid Kennaf et Lemar Andar**

Projet synthèse

Groupe 01

Projet synthèse :

**LOCs**

Travail présenté à

**Jean-Christophe Demers**

Cégep du Vieux-Montréal

02 février 2023

Voici LOCs, l'application qui vous expose au gens autour de vous. Discutez, partagez et échangez avec eux. Que ce soient des photos, de la musique ou vos pensées, notre réseau social permettra un utilisateur de parler dans le chat public de sa localisation. De plus, votre identité est anonyme jusqu'au moment d'accepter l'échange d'information avec un autre utilisateur. Dites maintenant adieu à l'ennui et bonjour aux nouvelles connexions avec LOCs.

L'utilisateur rentre dans notre application et peut voir directement le chat public du lieu où il se situe. Si vous êtes dans le Cégep du Vieux-Montréal, Vous avez le droit de communiquer dans la discussion. Au cas où vous changez d'endroit, alors il ne sera plus possible de bavarder dans le chat public du Cégep du Vieux-Montréal. Vous avez accès à n'importe quel chat public autour de vous, mais pour y participer il faut s'y rendre comme dans Pokémon Go. Ces lieux peuvent être des restaurants, des bibliothèques, des festivals, des cinémas, etc. Vous comprenez alors que cette application est fondamentalement basée sur le système de géolocalisation. Pour faciliter la découverte de ces lieux, notre page principale vous affiche les endroits autour de vous. Avec la fonctionnalité de préférence de distance, la page vous affiche donc les localités selon votre choix. De plus, une carte vous permet de rechercher les fils de discussion partout dans le monde. Celle-ci permet aussi de voir en temps réel l'achalandage (moyenne de personnes qui utilise l'application sur place). Ce système vous permettra de joindre et d'éviter les lieux achalandés.

Donc une de nos idées phares est l'instauration de nos profils publics/privés. Un profil public est comme ce que son nom dit est un profil que tout le monde peut voir. Ce profil n'est qu'un avatar avec un nom et trois centres d'intérêts. Il n'y a pas d'informations personnelles sur ce profil. À l'encontre, un profil privé est un profil qui contient beaucoup plus de détails de la personne (photo du visage, nom et prénom, âge, photos, emploi, etc.). Ce profil ne peut qu'être vu seulement si un utilisateur est intéressé par ce profil et que la personne propriétaire de ce profil veut aussi voir le profil de l'utilisateur intéressé. Ceci est le but de notre application et on a appelé cette action le Delocs. Vous allez vous dire pourquoi ne pas avoir seulement un profil public avec tous nos informations personnelles afin d'aller directement parler à la personne ? Le problème avec ça est qu'une personne qui n'attire pas autant que d'autres personnes auront peu vu pas de succès dans des applis de rencontre classique, car c'est surtout le profil physique qui apparaît en premier. Dans notre app, on veut que ça soit la personnalité et les intérêts qui prime sur le physique, donnant ainsi lieu à une égalité de chance de rencontrer quelqu'un entre nos utilisateurs.

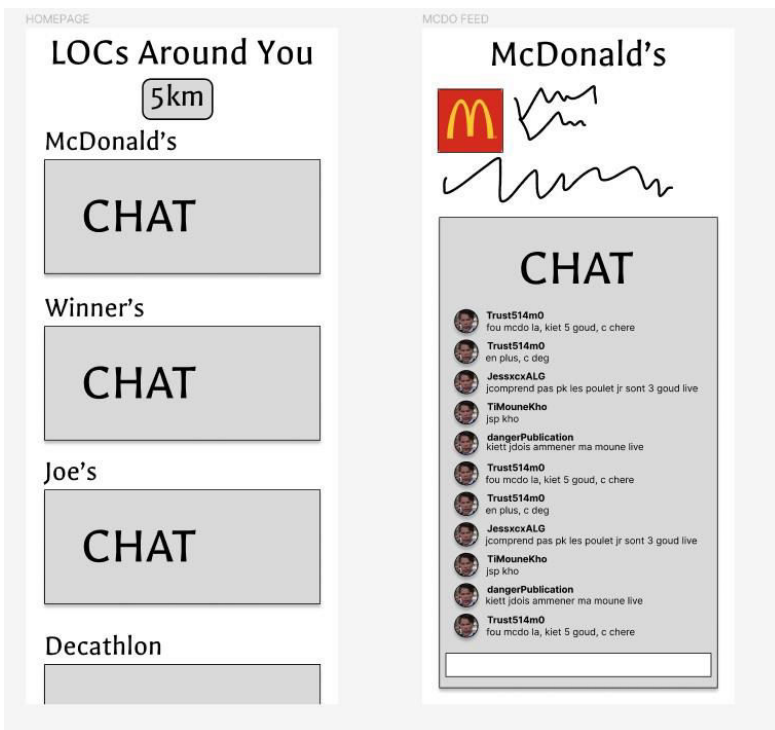
Lors de l'installation de l'appli LOCs, l'utilisateur devra s'inscrire avec un courriel et un mot-de-passe pour s'authentifier. Par la suite, ce dernier devra choisir parmi une dizaine d'avatar celui qui représente le mieux sa personnalité. Chaque avatar détient une description qui cerne certains traits de personnalité (Énergique, sociable, Enthousiaste, Modeste, Fiable, Réserve, Analytique, etc.). Par la suite, ce dernier, à l'aide d'hashtag, devra trouver 3 centres d'intérêt afin mieux décrire le genre de personne qu'il est. On finit l'inscription par un choix de nom d'utilisateur et l'inscription du profil public est terminée. Le profil privé ne serait pas

immédiatement nécessaire à inscrire au début, car tant que l'utilisateur n'est pas proche d'être en Delocs, alors son profil privé ne pourra pas être vu.

Après un certain moment, le chat public se remplira de discussion. Si l'utilisateur commence à porter un Intérêt à quelqu'un d'autre alors il pourra cliquer sur son profil afin d'afficher son avatar ainsi que ses intérêts. Si ce dernier n'est pas intéressé alors il peut revenir au chat public, mais sinon il peut lui envoyer une demande de Deloc, une demande lui permettant de parler en privé avec une personne afin de mieux connaître la personne et de surtout connaître la vraie personne. Si cette personne accepte la demande de Delocs, alors les deux personnes seront dirigées vers un chat privé. Dans ce cas de figure, les deux utilisateurs disparaîtront du chat public, ne pouvant plus interagir pour le moment avec le chat public.

Maintenant, je vais décrire comment un utilisateur peut arriver à se Deloc, qui est le but de notre application. Comme ce qui a été dit au dernier paragraphe, à la fin de l'inscription l'utilisateur entrera dans notre application. Si ce dernier est dans un lieu permettant d'accéder à un chat public, alors il y aura instantanément accès. Sinon, l'application lui ouvrira notre carte afin de lui montrer les lieux les plus proches qui eux ont un chat public. Lorsqu'on est dans un lieu permettant l'accès à un chat public, l'application affiche tous les messages qui ont été écrits sur ce lieu dans une intervalle de 24h. Cela veut dire que si vous écrivez un message à 12h15 alors il restera dans le chat public du lieu jusqu'à 12h15 au lendemain. Il sera par la suite supprimé. L'onglet chat affichera aussi les utilisateurs connectés au même lieu que vous. L'utilisateur pourra écrire dans le chat public tout ce qu'il lui passerait par la tête. Il peut se présenter, faire une blague, donner une opinion sur un sujet et même parler d'un événement qui c'est passé récemment dans ce lieu. Par exemple : si le chat public est situé dans un McDonald's et qu'il y a eu un incident récent, alors les utilisateurs présents uniquement dans ce lieu peuvent en parler. C'est un principe similaire à l'application Twitter, mais ici seuls les gens présents dans le lieu d'évènement peuvent en parler.

Dans la page d'accueil de l'application, vous pouvez voir les différents chats autour de vous. Par exemple : Si vous êtes au McDonald's de Saint-Denis, alors vous pourrez voir les différents chats des restaurants autour de vous. Vous pouvez également changer le rayon de recherche de chat à l'aide d'un bouton. Lorsque vous êtes dans un endroit et qu'il y a un chat de discussion, alors ce dernier sera au top des chats de la page d'accueil. Lorsque vous cliquez sur ce chat, alors vous serez redirigé vers un autre onglet. Ce dernier vous affiche les informations concernant ce lieu ainsi que le chat qui lui prendra la majeure partie de votre écran.



En bas du chat, il y a une barre permettant d'écrire dans le chat. Si l'utilisateur est intéressé par un certain profil, alors il peut cliquer sur ce dernier et il sera rediriger vers un autre onglet. Cet onglet affichera les détails du profil public de l'utilisateur. De plus, le profil affichera un chat avec la conversion en public que vous avez eu avec.

C'est par ce même chat que vous pouvez écrire en message privée avec cet utilisateur. Tous ce qui sera écrit dans ce chat ne sera vu qu'entre vous 2. Le chat public ne publiera pas vos conversations. En appuyant sur le bouton DeLoc, cela envoie une notification.

Si l'utilisateur accepte l'invitation, il s'ajoutera dans la liste de DeLOC et son profil privé sera afficher. C'est l'étape finale de l'application.

Concernant les contraintes applicatives du projet, celle qui est la plus évidente est sans doute l'implémentation du langage react-native à notre application. C'est l'option que nous avons prise, car on voulait que notre application soit multiplateforme en un seule codebase pour notre application.

Également, on voudrait que le chat soit fluide et qu'il y ait peu de latence dans les conversations entres utilisateurs, car un chat avec beaucoup de latence pourrait irriter les utilisateurs. Il faudrait donc faire un système de conversation qui n'atteigne pas plus de 100ms, qui est considéré comme bon pour un système de chat, mais plus le chiffre descendrait et mieux serait notre application.

Aussi, l'application doit respecter les droits de confidentialités des utilisateurs. On ne voudrait pas que les chats privés entre utilisateurs ne soit qu'accessible que par eux-mêmes. De



même, l'application doit assurer une certaine norme de sécurité dans la mesure où les utilisateurs doivent inscrire un profil privé qui leur contient des informations privées.

De plus, une contrainte serait de garder dans une base de données les chats envoyés et de pouvoir rapidement faire des requêtes afin de récupérer ces chats. De plus, l'application ne doit perdre en fluidité lors de ces requêtes.

Dernièrement, la contrainte de la scalabilité en est une, car notre application doit gérer une multitude d'utilisateurs et de messages et si ce dernier n'est pas bien géré, alors il peut y avoir une performance d'application qui est très lente, des pertes de données, une mauvaise expérience utilisateur. Par conséquent, il est important de s'assurer que notre app soit conçue pour être évolutive dès le départ, afin qu'elle puisse gérer un nombre croissant d'utilisateurs et de messages sans sacrifier les performances.

En conclusion, notre application cible les plateformes mobiles telles que iOS ainsi qu'Android. La version minimum est iOS 11.0 et Android 5.0 (API 21).