

École Polytechnique de Montréal

Département de génie informatique

Hiver **2023**

INF8405 – Informatique Mobile

**POLYTECHNIQUE
MONTRÉAL**



Projet final :

Amélioration de l'application développée au TP2

Présenté par:

- **1933097** Pritam Patel
- **1955204** Batikan Iscan
- **2169358** Imad Berrahil
- **1961278** Simon Tran

Soumis à M : Mehdi Kadi

le: 23 Avril 2023



Table des matières

Introduction	3
Présentation	3
Fonctionnalités du TP2	3
Fonctionnalités du projet	4
Critère 1: Création d'un compte usager avec une photo de profil	4
Critère 2: Utilisation d'au moins deux capteurs	5
Critère 3: Stockage de données dans Firebase	5
Critère 4: Informations sur la batterie	5
Critère 5: Informations sur la consommation de bande passante de l'application	6
Critère 6: En cas de perte de communication réseau l'application doit garder son état et ne pas fermer la connexion Bluetooth	6
Critère 7: Support de 2 langues	6
Difficultés rencontrées	6
Critiques et améliorations	7
Conclusion	7
Annexe	9
Bibliographie	20



Introduction

Le but principal du deuxième TP est de développer une application Android de détection d'appareils Bluetooth. Notre application utilise l'API Bluetooth pour détecter les appareils Bluetooth à proximité et ensuite affiche les informations pertinentes de ces appareils dans une liste. De plus, l'application permet de situer les appareils Bluetooth actifs détectés sur une carte. Cette application est utile pour les utilisateurs qui souhaitent connaître les appareils Bluetooth à proximité tels que les enceintes Bluetooth, les écouteurs, les montres connectées, etc.

Présentation

Les fonctionnalités de l'application sont divisibles en deux catégories: celles qui ont été implémentées pour le TP2 et celles pour le projet final. Ainsi, la présentation de notre projet sera séparée en deux sections. Il faut noter que la partie "Fonctionnalités du TP2" a été rédigée avant la remise du TP2, les captures d'écran et certaines descriptions peuvent ainsi être obsolètes.

Fonctionnalités du TP2

L'application utilise la fonctionnalité Bluetooth de l'appareil Android pour détecter les appareils Bluetooth actifs dans le réseau local. Utilisant l'API Bluetooth, elle peut afficher les informations qu'elle peut recueillir des appareils, tels le nom de l'appareil, l'adresse MAC,etc.

Lorsque l'utilisateur démarre la détection des appareils, l'application sera en mesure de détecter les périphériques Bluetooth disponibles. Puis, elle les localise sur la carte via le service de localisation en ligne Google Maps [1]. Notre application a été développée sur Android Studio[2]. Pour que l'application soit fonctionnelle sur une grande partie des appareils, nous avons choisi la configuration par défaut sur Android 7 (API 24) afin de respecter les spécifications du laboratoire et versions supérieures [3].

Notre application utilise l'API Google Maps[1]. La carte est centrée sur la position de l'utilisateur lors du lancement et un marqueur est créé pour chaque appareil Bluetooth. La bibliothèque Bluetooth par défaut d'Android est utilisée. Afin de pouvoir stocker les informations des appareils Bluetooth de façon permanente, la base de données Firestore de Firebase a été utilisée.[7][8]

Lors du lancement de l'application, l'utilisateur est présenté par une splashscreen affichant des informations sur l'application et les étudiants (voir figure 1 en annexe). La page principale de l'application affiche une carte, un bouton pour démarrer ou arrêter le balayage, un bouton pour changer le thème, un bouton pour centrer la carte sur la position de l'utilisateur et finalement un bouton pour afficher la liste des appareils (voir figure 2 en annexe).

Il est possible d'afficher la liste des appareils détectés et de changer le thème pour un mode plus clair. La figure 3 présente une capture d'écran de ce mode.



En appuyant sur le marqueur sur la carte d'un des appareils, on peut afficher un menu qui affiche les informations de l'appareil ainsi que des boutons pour étoiler l'appareil, le partager ou obtenir un itinéraire vers la position de l'appareil. Les figures 4 et 5 présentent ces fonctionnalités.

Fonctionnalités du projet

Critère 1: Création d'un compte usager avec une photo de profil

Tout d'abord, plusieurs boutons et dialogues ont été ajoutés à l'interface d'utilisateur de l'application. Un bouton permet à l'utilisateur de se créer un compte afin de sauvegarder ses données dans une base de données Firebase. Ce bouton est situé en haut à gauche et a une icône d'une personne. La figure 6 en annexe présente cette nouvelle interface

Lorsque l'utilisateur appuie sur ce bouton pour la première fois, un dialogue lui demande s'il désire se créer un nouveau compte. S'il confirme, le système crée un nouvel utilisateur dans Firebase et l'application affiche le dialogue d'édition des informations du compte après quelques secondes. Il est possible d'observer ce flux dans la figure 7 en annexe.

Tel qu'illustré en annexe à la figure 8, l'utilisateur a alors la possibilité de prendre une photo de profil et d'entrer un nom d'utilisateur. Il doit appuyer sur le bouton "Save Changes" pour confirmer les changements et les sauvegarder dans la base de données. Sur la page principale, on voit alors le nom d'utilisateur et le bouton de compte comporte la photo de profil.

Si l'utilisateur appuie sur le bouton "Delete Account Data And Logout", un dialogue de confirmation s'affiche. Si l'utilisateur confirme, il sera alors déconnecté et il ne pourra plus se reconnecter à ce compte. La figure 9 montre ce dialogue.

Critère 2: Utilisation d'au moins deux capteurs

Un bouton "Show Sensors" permet d'afficher un nouveau dialogue. Ce dialogue est illustré à la figure 14 en annexe. Après quelques secondes d'initialisation, le dialogue s'affiche et montre les données brutes provenant de l'accéléromètre et le gyroscope. Cependant, les tablettes Galaxy Tab A fournies par l'école ne possèdent pas de gyroscope, les champs seront donc vides. Pour la présentation, un appareil muni de tous les capteurs sera employé afin de bien démontrer toutes les fonctionnalités.

Critère 3: Stockage de données dans Firebase

Firestore de Firebase est utilisé pour stocker les informations des utilisateurs. Il y a une liste des appareils découverts que l'on filtre pour obtenir seulement les appareils de l'utilisateur. La figure 10 montre la collection "devices" dans Firestore.



Il y a aussi une liste des utilisateurs. Chaque élément de cette collection a un champ pour le nom d'utilisateur, il serait possible d'ajouter des champs dans le futur si on désire stocker plus d'informations. Cette collection est illustrée en figure 11.

Pour le stockage des photo de profil, nous utilisons Firebase Storage. Il y a un dossier "images" dans lequel l'application téléverse les photos, le nom de l'image est l'identifiant de l'utilisateur afin de pouvoir facilement y accéder. Une capture d'écran de Firebase Storage est disponible en annexe à la figure 12.

Finalement, pour la création des comptes, nous utilisons l'authentification anonyme de "Firebase Authentication". Ce choix est dû au fait que les requis ne demandaient pas de permettre à un utilisateur de choisir un mot de passe ou tout autre forme d'authentification. L'authentification anonyme était l'option la plus simple qui remplissait tous les critères. La figure 13 illustre les comptes.

Critère 4: Informations sur la batterie

L'application fournit désormais des informations détaillées sur la batterie, notamment l'évolution du niveau de charge de celle-ci pendant l'utilisation de l'application. Par exemple, il est maintenant possible de calculer et d'afficher la consommation d'énergie depuis le lancement de l'application, ce qui permet aux utilisateurs de mieux comprendre l'impact de l'application sur la durée de vie de la batterie de leur appareil. En plus de calculer et d'afficher la consommation d'énergie depuis le lancement de l'application, celle-ci informe également l'utilisateur de la quantité d'énergie (niveau de batterie) utilisée pour la recherche de réseaux Bluetooth à proximité. Cette fonctionnalité permet aux utilisateurs de mieux comprendre l'impact de la recherche de réseaux Bluetooth sur la durée de vie de leur batterie, et de prendre les mesures nécessaires pour optimiser l'utilisation de l'application. Pour accéder aux informations sur l'utilisation de la batterie, il suffit de cliquer sur le bouton « Afficher les analyses ». Cette action ouvre une boîte de dialogue qui contient les informations souhaitées, comme illustré sur la figure 14.

Critère 5: Informations sur la consommation de bande passante de l'application

Les informations sur la consommation de bande passante sont affichées en tout temps sur la page principale de l'application. Les vitesses de téléchargement et de téléversement sont affichées en bas à gauche de l'écran et mises à jour chaque secondes. La figure 15 illustre cette interface.

Critère 6: En cas de perte de communication réseau l'application doit garder son état et ne pas fermer la connexion Bluetooth

Si le Bluetooth est désactivé pendant la découverte, cette dernière est mise sur pause. Un message est affiché en bas de l'écran et le bouton pour basculer la découverte est



désactivé. Lorsque le Bluetooth est réactivé, un nouveau message apparaît et la découverte des appareils recommence automatiquement. Ce flux est illustré à la figure 16.

Puisque nous utilisons Firebase pour la persistance des données, celui-ci gère automatiquement le comportement en cas de perte de réseau. Les données seront téléchargées automatiquement lorsque le réseau sera à nouveau disponible.

Critère 7: Support de 2 langues

L'application supporte entièrement l'anglais et le français. Un bouton en bas à droite permet de basculer la langue. Des captures d'écran montrent l'application en français en figure 17 de l'annexe.

Difficultés rencontrées

Une difficulté majeure a été la gestion de l'authentification et des différents comptes utilisateurs. En effet, nous avons dû repenser notre base de données, puisqu'à la base nous pensions avoir une liste commune pour tous les utilisateurs. Il a fallu qu'on filtre les appareils pour n'afficher que ceux qui appartiennent à l'utilisateur. La gestion des différents utilisateurs fut bien plus complexe que tous les autres requis.

De plus, il fut difficile de gérer le flux lorsque l'utilisateur désactive le Bluetooth. Nous avons dû intercepter les événements reliés à cela et mettre la découverte sur pause.

Critiques et améliorations

Nous avons trouvé le TP très intéressant d'un point de vue technique, puisque nous avons pu toucher à plusieurs fonctionnalités d'Android. Par contre, il aurait été intéressant d'avoir un but concret pour l'application au lieu d'une liste de critères vagues sans lien entre eux. En effet, il était parfois difficile de faire des choix concernant la bonne implémentation de certaines fonctionnalités, puisque nous ne savions pas à quoi elle servait.

Conclusion

Au cours de ce travail pratique, nous avons eu l'opportunité de nous familiariser avec l'environnement de développement intégré (IDE) Android Studio, qui est un outil incontournable pour le développement d'applications mobiles Android. Nous avons également révisé les concepts de programmation orientée objet et d'interfaces graphiques en langage Kotlin, qui sont des compétences fondamentales pour développer des applications Android.

En outre, nous avons exploré l'intégration de modules d'accès aux périphériques Bluetooth et GPS dans notre application mobile, ce qui nous a permis de mieux comprendre les fonctionnalités de ces périphériques et leur utilisation dans une application mobile. Nous avons toutefois rencontré certaines difficultés dans la gestion de l'accès à ces périphériques, ce qui nous a demandé un travail supplémentaire pour les résoudre.

Puis, nous avons également exploré l'utilisation de Google Maps dans notre application mobile, ce qui nous a permis d'ajouter des fonctionnalités de localisation et de cartographie



à notre application. Toutefois, cela a également présenté des défis, notamment en termes de configuration et d'utilisation des API de Google Maps, qui ont nécessité un certain temps pour les comprendre et les intégrer correctement.

Finalement, une tâche particulièrement difficile a été d'intégrer la base de données Firestore pour persister la liste des appareils détectés. Cela nous a permis de mieux comprendre Firebase et la gestion des données.

Nous avons trouvé cette expérience très enrichissante et nous sommes convaincus que cela nous sera utile pour nos futurs projets.



Annexe

Figure 1: Splash screen de l'application



Figure 2: Page principale de l'application

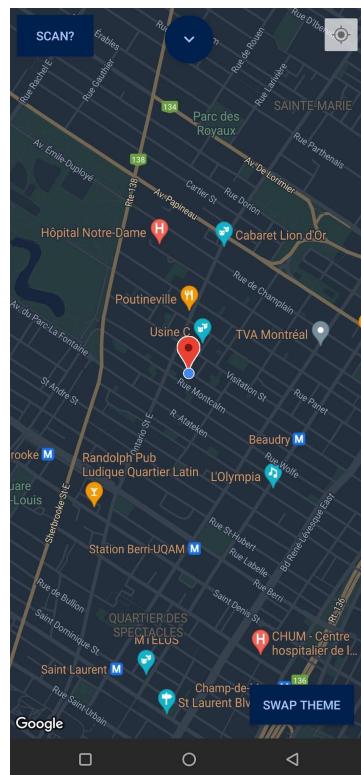




Figure 3: Liste des appareils et mode clair

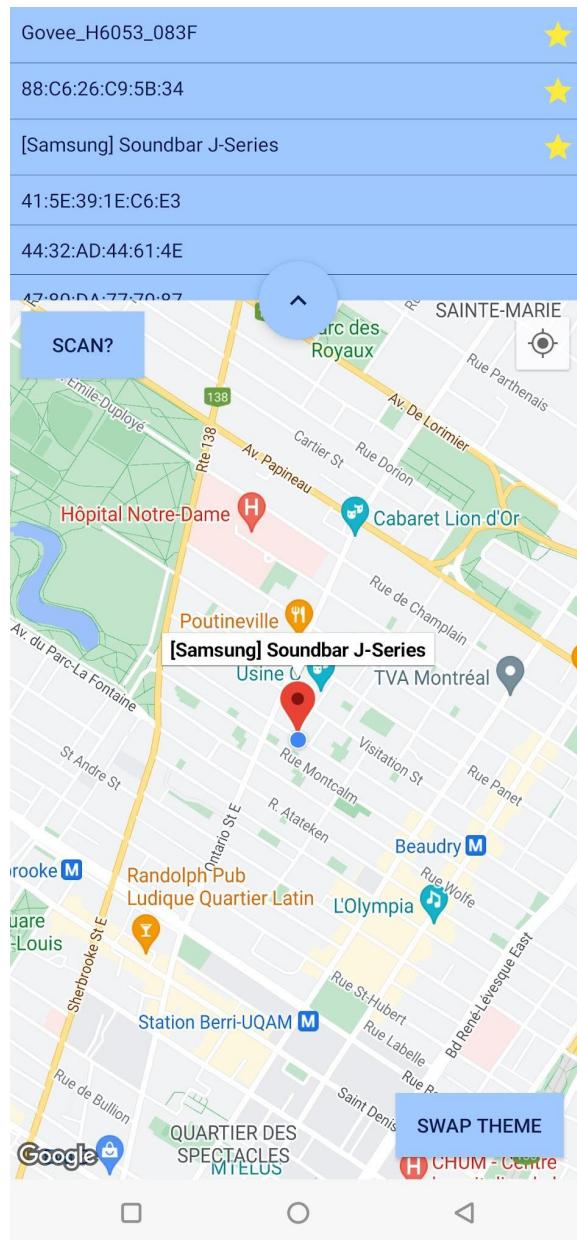




Figure 4: Menu détaillé d'un appareil

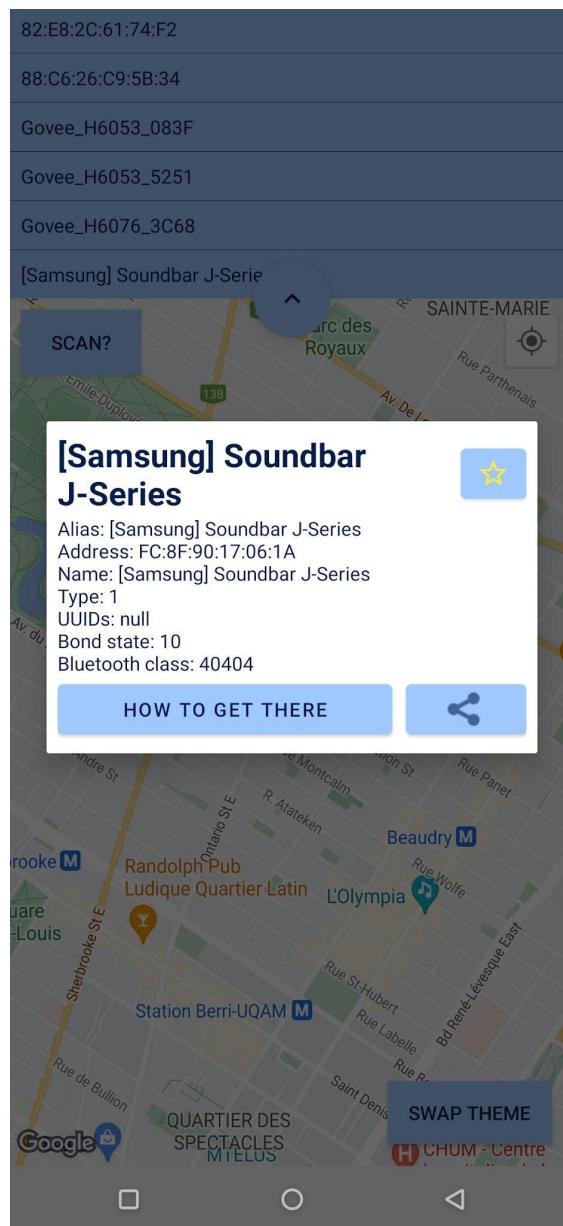




Figure 5: Menu "Partager"

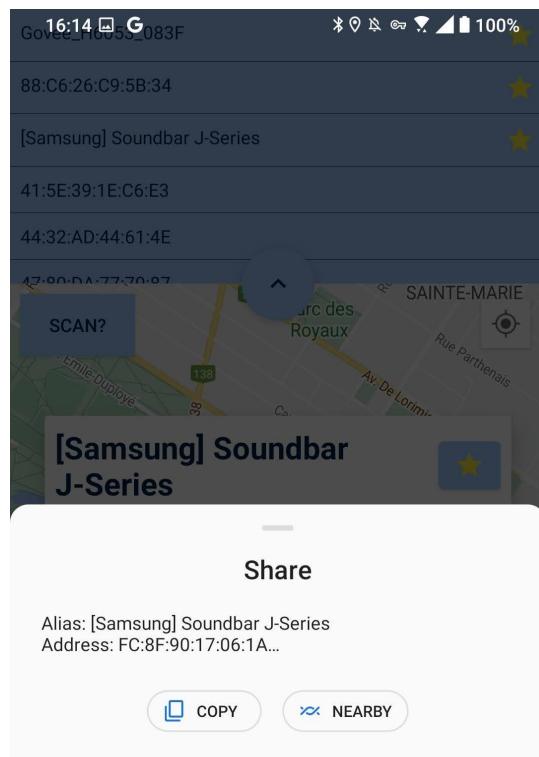




Figure 6: Nouvelle interface principale

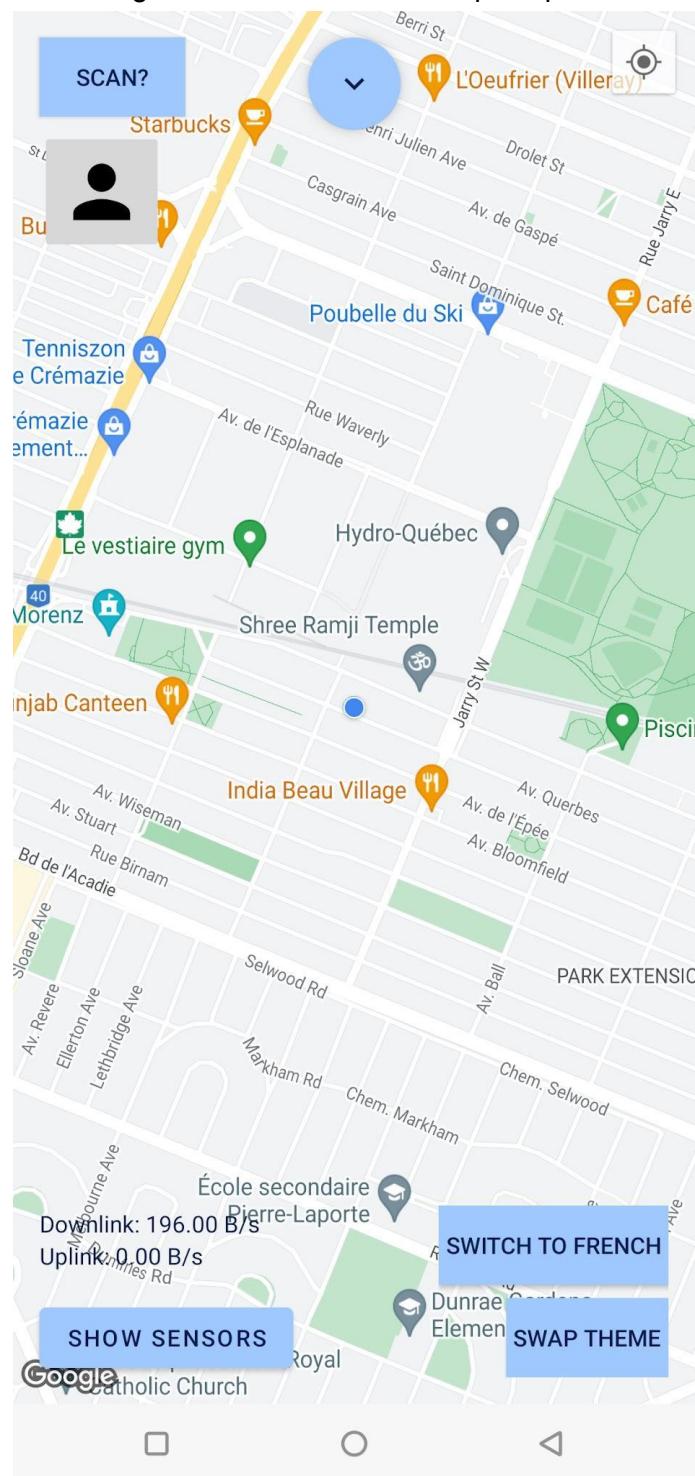




Figure 7: Flux de création de compte

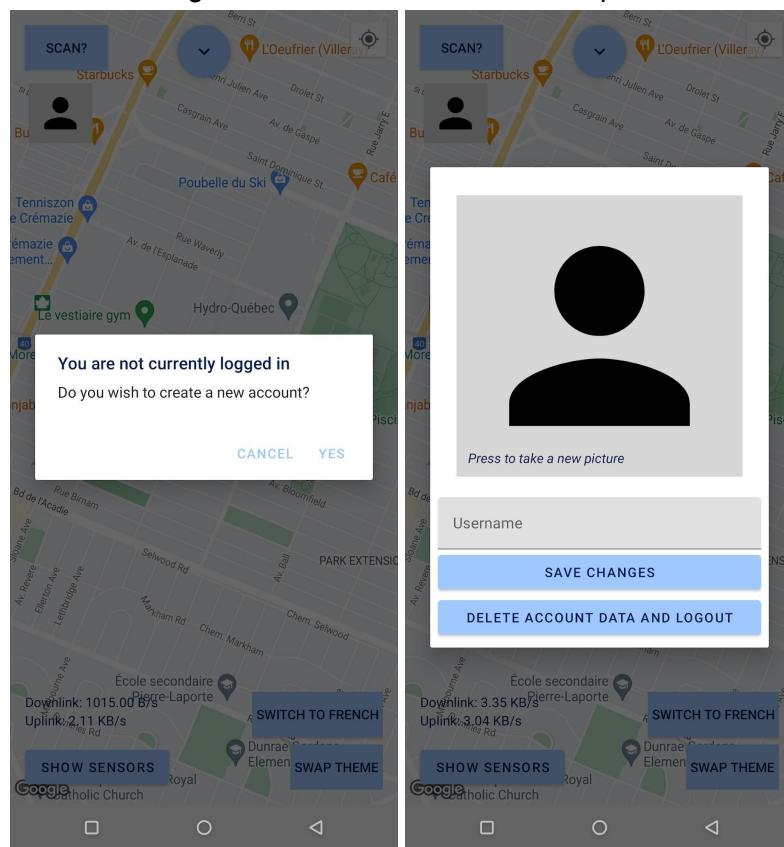


Figure 8: Édition des informations du compte et téléversement d'une photo de profil

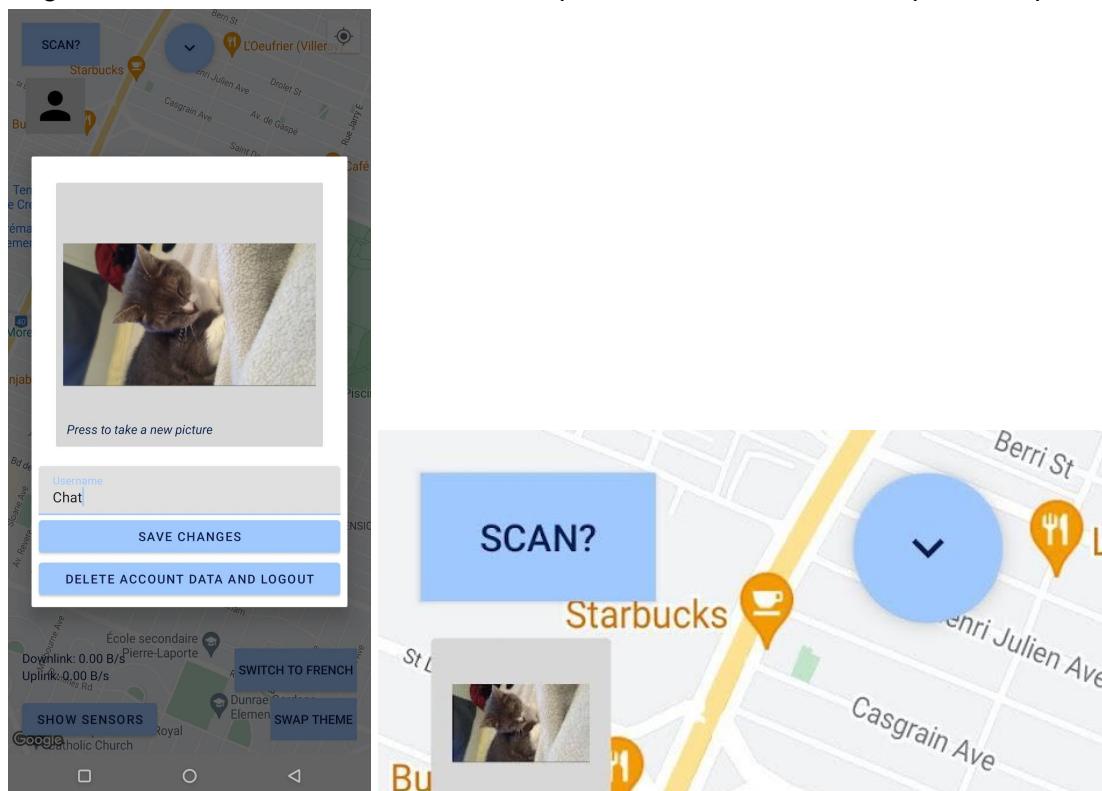




Figure 9: Confirmation de la déconnection

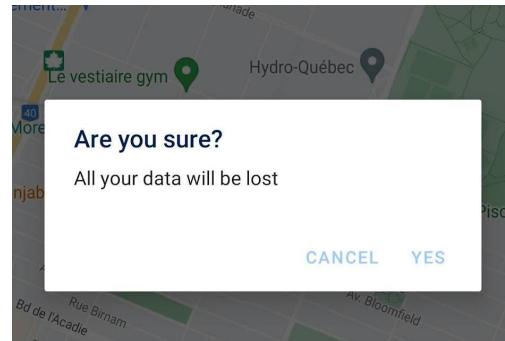


Figure 10: La collection "devices" dans Firestore

inf8405-bluetoothmap	devices	48:E1:FE:84:6F:14
+ Start collection	+ Add document	+ Start collection
devices >	48:E1:FE:84:6F:14 >	+ Add field
users	58:2E:08:A1:19:C4 5B:E6:07:BF:A3:BE 5F:B3:31:6B:DC:69 60:CB:68:8D:30:75 67:D2:AA:62:1D:A7 68:EE:42:BD:CE:2D 6F:D8:F5:B7:81:F8 76:AA:45:79:77:F8 B4:0E:DE:84:73:9D	address: "48:E1:FE:84:6F:14" alias: "LE-La Muse" bluetoothClass: "1f00" bondState: 10 deviceInfo: "Alias: LE-La Muse Address: 48:E1:FE:84:6F:14 Name: LE-La Muse Type: 2 UUIDs: null Bond state: 10 Bluetooth class: 1f00" latLng latitude: 45.5328368 longitude: -73.6376687 name: "LE-La Muse" starred: false type: 2 user: "XGX9UoeDJMfmml2R6rnihmDpzBp2" uuids: null

Figure 11: La collection "users" dans Firestore

inf8405-bluetoothmap	users	EuM9Zot5CRY4UqXfKCuOD16InPx1
+ Start collection	+ Add document	+ Start collection
devices	EuM9Zot5CRY4UqXfKCuOD16InPx1 >	+ Add field
users >	XGX9UoeDJMfmml2R6rnihmDpzBp2 tKidsorgizUPG8CTKrBd18zr1ny1	username: "Chat"



Figure 12: Photos de profil dans Firebase Storage

gs://inf8405-bluetoothmap.appspot.com > images

Upload file

Name	Size
EuM9Zot5CRY4UqXfKCuOD16InPx1	63.6 KB
tKidsorgizUPG8CTKrBd18zr1ny1	53.93 KB

 EuM9Zot5CRY4UqXfKCuOD16InPx1

Name: [EuM9Zot5CRY4UqXfKCuOD16InPx1](#)

Size: 65,126 bytes

Type: image/png

Created: Apr 20, 2023, 1:27:27 PM

Updated: Apr 20, 2023, 1:27:27 PM

File location

Storage location: gs://inf8405-bluetoothmap.appspot.com/images/EuM...

Access token: [Revoke](#)
33956c30-4d05-4b24-b359-88bfc9020214
[Create new access token](#)

Other metadata

No metadata found



Figure 13: Les comptes dans "Firebase Authentication"

Authentication

Users Sign-in method Templates Usage Settings Extensions NEW

Identifier	Providers	Created	Signed In	User UID
(anonymous)	👤	Apr ...	Apr ...	EuM9Zot5CRY4Uq...
(anonymous)	👤	Apr ...	Apr ...	k3aa4NsRdVRRjG...
(anonymous)	👤	Apr ...	Apr ...	XGX9UoeDJMfmm...

Rows per page: 50 1 – 3 of 3 < >

Figure 14: Information des capteurs et de l'usage de la batterie dans l'application

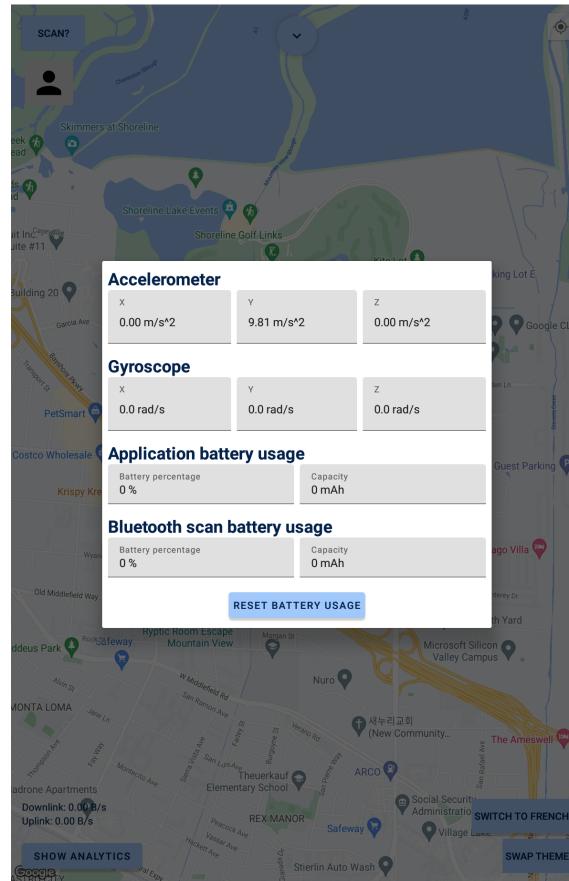




Figure 15: Informations sur la consommation de bande passante



Figure 16: Flux de la désactivation et réactivation du Bluetooth

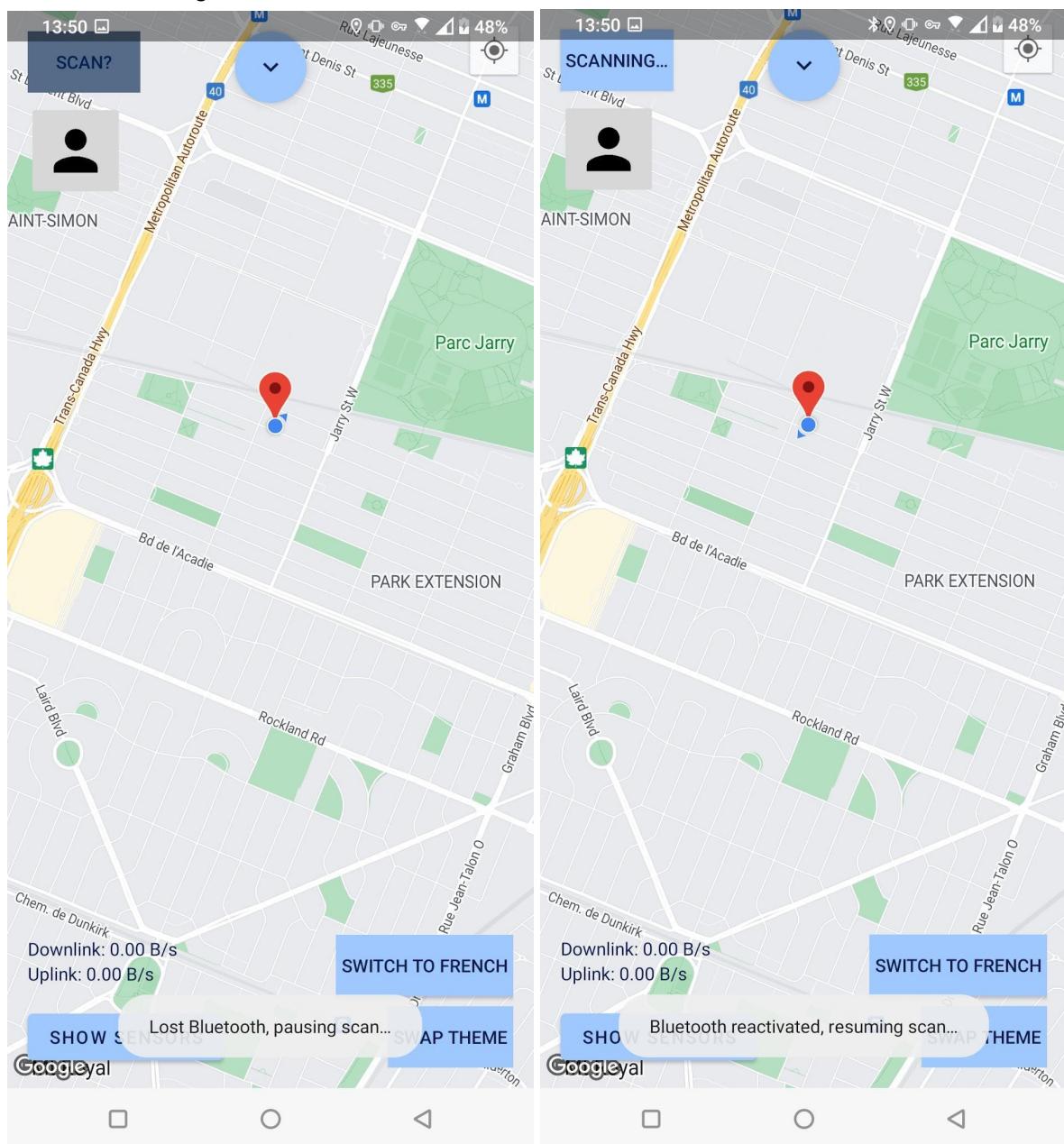
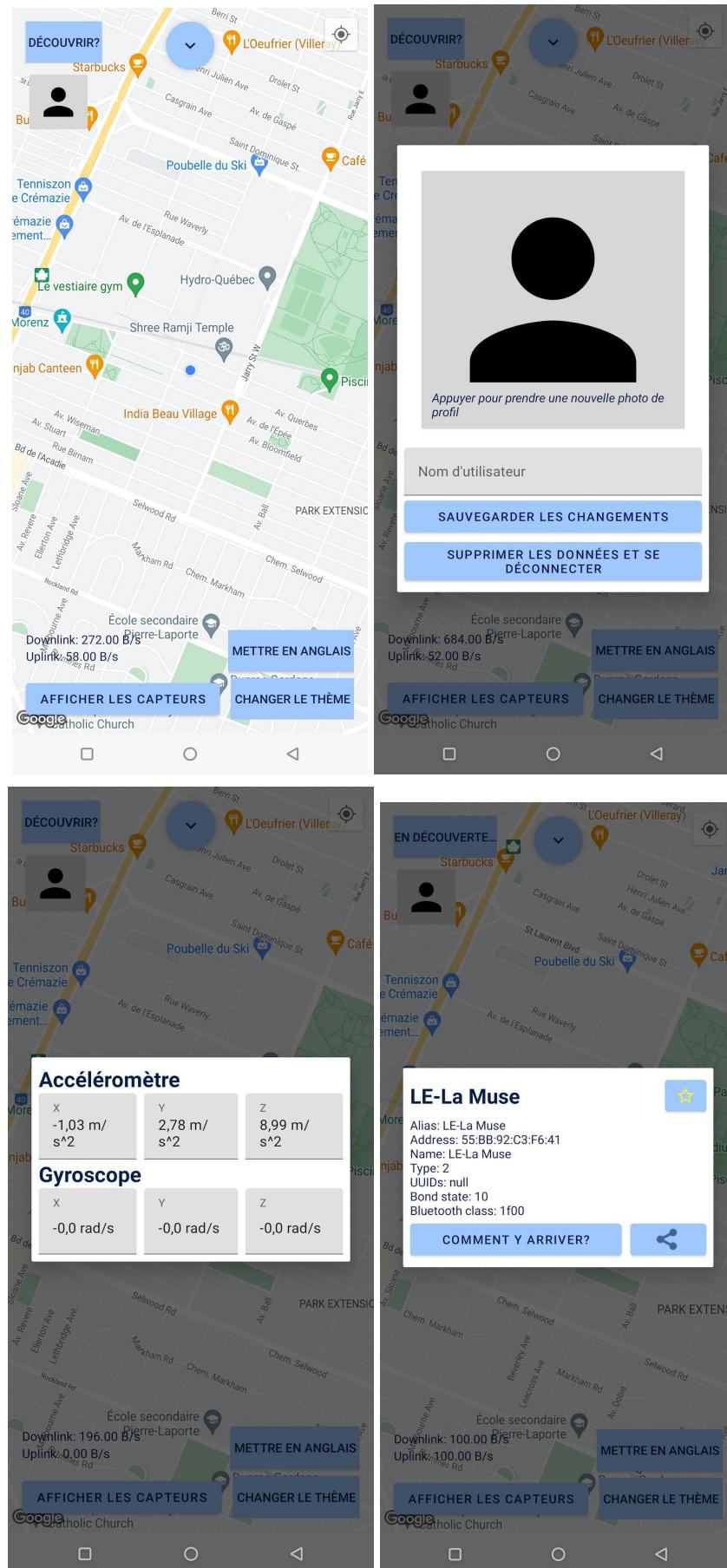




Figure 17: Interface de l'application en Français





Bibliographie

- [1] - "Google Maps — Wikipédia". [En ligne]. Disponible à:
https://fr.wikipedia.org/wiki/Google_Maps. [Consulté le: 15-mars-2023].
- [2] - "Android Studio release notes". [En ligne]. Disponible à:
<https://developer.android.com/studio/releases>. [Consulté le: 15-mars-2023].
- [3] - "SDK Platform release notes | Android Developers". [En ligne]. Disponible à:
<https://developer.android.com/studio/releases/platforms>. [Consulté le: 15-mars-2023].
- [4] « android.bluetooth | Android Developers ». [En ligne]. Disponible à:
<https://developer.android.com/reference/android/bluetooth/package-summary>. [Consulté le: 15-mars-2023].
- [5] « android.bluetooth | Android Developers ». [En ligne]. Disponible à:
<https://developer.android.com/reference/android/bluetooth/package-summary>. [Consulté le: 22-mars-2023].
- [6] « BluetoothClass.Device | Android Developers ». [En ligne]. Disponible à:
<https://developer.android.com/reference/android/bluetooth/BluetoothClass.Device>. [Consulté le: 22-mars-2023].
- [7] « Get Started with Cloud Storage on Android | Firebase ». [En ligne]. Disponible à: <https://firebase.google.com/docs/storage/android/start> . [Consulté le: 12-févr-2023].
- [8] « Get started with Cloud Firestore ». [En ligne]. Disponible à:
<https://firebase.google.com/docs/firestore/quickstart>. [Consulté le: 22-mars-2023].
- [9] « Permissions overview | Android Developers ». [En ligne]. Disponible à:
<https://developer.android.com/guide/topics/permissions/overview>. [Consulté le: 22-mars-2023].