

Design, conception et déploiement d'une solution full-stack de gestion de données de géolocalisation pour la startup NeoFinder

Travail de fin d'études présenté par

Martin Degeldt

en vue de l'obtention du diplôme de

Master en Sciences de l'Ingénieur Industriel orientation Informatique

REMERCIEMENTS

En premier lieu, j'aimerais adresser mes remerciements à Mme. Queenie Halsberghe, Research Unit Manager au CERDECAM. Grâce à elle et son support quotidien, j'ai pu débiter cette aventure entrepreneuriale et le projet NeoFinder a pu se développer.

Je remercie les efforts de M. Combéfis et Mme Flémal pour leur supervision, leur aide technique et leurs bons conseils.

J'aimerais aussi remercier M Manoel Da Silva pour avoir placé les bases du projet en début d'année et, avec M Sébastien d'Oreye, d'avoir conféré une bonne ambiance durant mes journées passées au CERDECAM.

Je souhaite adresser un merci spécial à Sam Bertrand, pour avoir été mon binôme durant 3 ans et demi. Sans lui, le projet n'aurait pas de raison d'exister. Il a réussi à tirer le projet vers l'avant grâce à sa rigueur et son organisation.

Un grand merci également à toute l'équipe de NeoFinder pour le développement de l'idée.

Je souhaite adresser mes remerciements à toute l'équipe d'accompagnement du StartLab ICHEC : Hélène Chochaux, Aurélie Mulowa et notre coach, Xavier de Poorter, pour nous avoir aidés dans la création de NeoFinder

Et enfin, je remercie de tout cœur ma maman et son amie Françoise pour leurs relectures de ce travail.

RÉSUMÉ

La moyenne d'âge de la population augmente et de plus en plus de personnes se retrouvent en maison de repos. De nouveaux challenges apparaissent et la sécurité de ces personnes en fait partie.

Ce TFE s'inscrit dans un projet entrepreneurial nommé NeoFinder. NeoFinder est un outil qui a pour but d'aider les entreprises se souciant de la sécurité de leurs clients, ou de leur public, les plus susceptibles de s'égarer. Cette solution prend la forme d'un dispositif embarqué communiquant les données de géolocalisation, de la personne le portant, à une plateforme cloud via LoRaWAN. Ces données sont ensuite présentées visuellement sur une carte dans une application mobile.

Le sujet principal de ce TFE porte sur la partie cloud et application mobile du produit. Il s'agit d'un serveur qui s'occupe de la gestion des données de géolocalisation et des données relatives aux entreprises utilisant la solution et d'une application mobile permettant de visualiser l'emplacement et l'état des dispositifs, ainsi que de déterminer un périmètre de sécurité. Ce travail couvre le cycle complet du développement d'un système logiciel, en partant des choix de l'architecture logicielle, des modèles de données, des techniques et technologies en passant par une analyse d'aspects de sécurité, afin de réaliser une solution multi-plateforme.

La seconde partie du rapport présente le travail de mise en place de l'entreprise. Les aspects abordés concernent le parcours réalisé au StartLab ICHEC et les différents dossiers de demandes de financement.

CAHIER DES CHARGES

Objectifs à atteindre :

- Un serveur doit être mis en place afin de récupérer et traiter les informations envoyées depuis les capteurs de géolocalisation.
- Une interface client doit être mise en place afin d'afficher les informations nécessaires au bon fonctionnement du service.
- Un dossier de candidature, pour répondre à un appel à projet de spin-off, doit être complété et rendu. Cet objectif est commun à toutes les personnes du projet NeoFinder.

Principales étapes :

1. Création du squelette de la plateforme :
 - Choix d'architecture et design de la plateforme
 - Mise en place du serveur
 - Choix d'un système de base de données
 - Affichage sur l'interface client
2. Implémentation de fonctionnalités :
 - Définition du périmètre de sécurité
 - Affichage des données de géolocalisation sur une carte
 - Notification lors du franchissement du périmètre de sécurité
3. Finalisation :
 - Sécurisation générale de la plateforme et des communications
 - Réception des données du réseau WAN
 - Déploiement de la plateforme
 - Documentation du fonctionnement et de l'utilisation de la plateforme
4. Entrepreneurat :
 - Formation et travail de création d'entreprise au StartLab ICHEC
 - Rédaction du dossier de candidature
 - Rédaction d'un cahier des charges et suivi de la conception mécanique du capteur.

TABLE DES MATIÈRES

Introduction	6
I. Contexte.....	7
I.A. Contexte entrepreneurial	7
I.A.1. StartLab ICHEC.....	7
I.A.2. Projet entrepreneurial.....	8
I.A.3. Equipe	9
I.B. Contexte technique.....	10
I.B.1. CERDECAM	10
I.B.2. Projet Technique	11
I.C. Objectifs	14
II. Technologies	15
II.A. Etat de l'art	15
II.A.1. Architecture générale.....	15
II.A.2. Serveur – API.....	16
II.A.3. Application utilisateurs	17
II.A.4. Base de données	19
II.B. Choix technologique	20
II.B.1. Serveur.....	20
II.B.2. Application utilisateurs	21
II.B.3. Environnement.....	22
III. Développement.....	23
III.A. Architecture.....	23
III.B. Implémentation.....	25
III.B.1. Sprint 1: création du squelette.....	25
III.B.2. Sprint 2: implémentation des fonctionnalités	27
III.B.3. Sprint 3: finalisation	31
III.C. Tests.....	33
III.C.1. La plateforme	33
III.C.2. L'application	35
III.D. Déploiement	36
IV. Discussions.....	38
IV.A. Résultats.....	38
IV.B. Perspectives.....	39
IV.B.1. Améliorations	39
IV.B.2. Développements futurs.....	40
V. Conclusions	41
VI. Bibliographie	42
VII. Annexes	43

INTRODUCTION

Le vendredi 15 mars 2019 se déroulait la première édition du Hack2Start, un hackathon organisé par le StartLab ICHEC et le centre de recherche et développement de l'ECAM (le CERDECAM). Le study case était :

« Comment peut-on faire pour retrouver des personnes perdues telles que des enfants dans un supermarché, des joggeuses dans un parc et des personnes âgées quittant leur maison de retraite. La solution a pour but d'être indépendante d'un téléphone portable. L'objectif n'est ni de les tracer ni de les encombrer. »

À l'issue de 3 jours de réflexion, de débats et de calculs, un jury multidisciplinaire a élu l'idée gagnante du concours : une solution spécialisée pour les maisons de retraite, permettant de retrouver les résidents qui s'égarer. Cette solution serait directement proposée aux homes, afin que ceux-ci puissent mettre en place l'infrastructure nécessaire à la sécurité de leurs pensionnaires.

Ayant envie de donner suite aux idées émises lors du hackathon, le CERDECAM a proposé aux participants du week-end de poursuivre l'aventure. Les étudiants de l'ECAM qui accepteraient seraient considérés comme *étudiants entrepreneurs*. Trois personnes ont répondu à cet appel : Sam Bertrand, Houda Hannouni et moi-même. La création du projet NeoFinder pouvait débuter. L'objectif du projet est, d'une part, de développer la partie entrepreneuriale de l'idée et, d'autre part, de réaliser la partie technique de la solution.

Ce Travail de Fin d'Étude (TFE) est réalisé à l'issue de 5 années passées à l'ECAM, afin d'obtenir le diplôme d'ingénieur industriel en informatique. Le travail qui suit s'attarde spécifiquement sur la partie technique du projet.

La technique du projet peut se séparer en deux parties : une partie hardware et une partie software. Ce travail présente la partie software, la partie hardware étant présentée dans le TFE de Sam Bertrand.

Depuis septembre, cette collaboration a permis de donner vie au projet. Les guidelines étaient de créer un système fonctionnel et modulable. Cet aspect est primordial car la solution doit s'adapter aux besoins du public cible.

Dans un premier temps, le cadre, l'encadrement et les objectifs du projet NeoFinder sont présentés.

Deuxièmement, un état de l'art des différentes technologies utiles au projet est dressé. La sélection de ces technologies est argumentée.

Ensuite, seront présentées les différentes étapes de la réalisation du travail.

Par la suite, l'état du projet et les perspectives seront abordés.

En conclusion, un bilan objectif sera dressé.

I. CONTEXTE

Pour donner suite à la première édition du Hack2Start, le CERDECAM décide de développer l'idée gagnante : une solution spécialisée pour les maisons de retraite, permettant de retrouver les résidents qui s'égarer. Cette solution peut être intégrée dans différents dispositifs tels qu'un bracelet.

Soucieux de faire mûrir cette idée, le centre de recherche a proposé aux personnes suivantes d'apporter leur aide à ce projet : Sam Bertrand, Houda Hannouni et à moi-même.

Le projet ayant un aspect entrepreneurial, l'équipe a eu l'occasion d'intégrer le StartLab ICHEC. Elle y a suivi des formations et des séances de coaching.

Ce projet spécifique est encadré par :

- une promotrice pour l'aspect entrepreneurial, Mme Queenie Halsberghe,
- un promoteur pour l'aspect technique, M. Manoel Da Silva. À la suite de son départ, ce rôle a été endossé par M. Sébastien Combéfis,
- une tutrice, Mme Flémal.

I.A. CONTEXTE ENTREPRENEURIAL

Ce paragraphe pose le contexte entrepreneurial. Le projet a eu la chance d'être encadré par le StartLab ICHEC.

En annexe, un document analogue présente le travail entrepreneurial. Différents sujets y sont abordés, tels que les candidatures de subsides, les étapes de la création d'entreprise, les coachings, etc.

I.A.1. STARTLAB ICHEC

Le StartLab ICHEC est une structure pour jeunes entrepreneurs bruxellois. Elle accompagne des porteurs de projet dans la création de leur entreprise en fournissant des formations, un hébergement, un suivi individuel avec un coach, une mise en réseau et l'accès à des experts. Le StartLab est ouvert aux candidatures tous les ans et accompagne une trentaine de projets. Cet incubateur est lié à l'ICHEC pour les reconnaissances académiques, les expertises et le réseau, mais est un programme ouvert à tous les étudiants et jeunes diplômés bruxellois.

L'équipe se constitue de :

- Hélène Cochaux : responsable du projet
- Aurélie Mulowa : responsable communication et de la communauté des porteurs
- Nathalie Degroote : point de contact pour les dispositions académiques

L'équipe se constitue de différents coachs aidant les projets au sein de l'incubateur. Le coach du projet NeoFinder s'appelle Xavier De Poorter

Durant la période de préparation du TFE, un dossier de candidature a été rempli pour intégrer le StartLab. NeoFinder a été reçu le 4 février. Depuis lors, l'équipe suit régulièrement des formations permettant d'acquérir des compétences en gestion, en finances et en communication. Ces séances ont aussi permis de challenger les hypothèses du projet.

Une série d'actions nommée « parcours gagnant » ont été effectuées. Celles-ci ont permis, tout comme les formations, de remettre en question l'idée du projet et d'améliorer la proposition de valeur pour les futurs clients.

En plus des formations et du travail régulier sur le parcours gagnant, l'équipe a également suivi des sessions de coaching. Ces réunions avaient pour but de guider l'aventure entrepreneuriale et de cadencer le travail en présentant l'avancement réalisé entre deux sessions.

I.A.2. PROJET ENTREPRENEURIAL

Lorsqu'une personne vulnérable s'égare, le plus inquiétant pour son responsable est de ne pas savoir où elle se trouve et si elle va bien. Les personnes perdues ont aussi besoin de se sentir rassurées, le temps d'être retrouvées. La solution proposée à ces problèmes est un dispositif offrant un service de géolocalisation.

Grâce à une puce de géolocalisation s'activant lorsqu'une personne s'égare, il sera plus facile pour un responsable de retrouver une personne perdue. Cette puce de géolocalisation est intégrable dans un bracelet, ou dans tout autre élément portable. Ceci permet de se sentir en sécurité grâce à un dispositif discret et léger. Si une personne, portant le dispositif, franchit une zone sûre, une notification sera envoyée à sa personne responsable. Nous appellerons cette dernière « le garant » pour la suite de ce travail.

Le client dispose d'un ensemble de dispositifs qu'il place sur des personnes vulnérables. Lorsque celles-ci s'égarent, les garants sont alertés via notification.

Un logiciel est nécessaire pour recevoir ces notifications et pour paramétrer la zone de sécurité. Celui-ci prend la forme d'une application mobile ou web. Le personnel gérant la maison de retraite peut se connecter sur l'application afin d'avoir un contrôle et paramétrer les fonctionnalités nécessaires.

Cela permettra au personnel des homes d'effectuer leur travail avec moins de stress. Ce stress est principalement lié à la fuite des personnes âgées souffrant de démence ou de désorientation.

Grâce à cette solution, un impact sociétal positif est créé.

I.A.3. EQUIPE

En début d'année, un appel à candidature a été lancé aux participants du Hack2Start pour leur permettre de rejoindre le projet. Trois personnes supplémentaires ont répondu :

- Raissa Kassenga, étudiante à l'ICHEC en gestion d'entreprise, 2^e bachelier,
- Cyril Le Ray, étudiant à l'ECAM en ingénieur industriel en électromécanique, 3^e bachelier,
- Quentin Vandenborre, étudiant à l'ICHEC et à l'ECAM en business analyst, 2^e master.

L'équipe s'est organisée de la façon suivante :

Pôle Marketing :

- Responsable : Houda Hannouni
- Suppléants : Raissa Kassenga et Cyril Le Ray

Pôle Gestion :

- Responsable : Quentin Vandenborre

Pôle Technique :

- Responsable hardware : Sam Bertrand
- Responsable software : Martin Degeldt
- Suppléant : Cyril Le Ray

I.B. CONTEXTE TECHNIQUE

Pour l'aspect technique, nous avons été encadrés par le CERDECAM qui nous a fourni les ressources et le support nécessaire à la réalisation de ce travail.

I.B.1. CERDECAM

Le CERDECAM est le centre de recherche et développement de l'ECAM. En novembre, l'équipe se constituait d'un manager, de deux chercheurs et de quatre étudiants débutant leur TFE.

Les membres sont :

- Queenie Halsberghe : manager
- Manoel Da Silva : chercheur sur une station de recharge sans fil pour de moyennes puissances
- Sébastien d'Oreye : chercheur sur un outil informatique permettant la visualisation de l'entourage de personnes à problème psychiatrique, visant leur réinsertion dans la société
- Fatima Nouhi : étudiante en TFE portant sur l'implémentation d'une fonctionnalité pour un robot assistant d'opérations chirurgicales
- Louis de Cockborne : étudiant en TFE et étudiant entrepreneur. Son TFE porte sur le prototypage du produit de son entreprise
- Sam Bertrand : étudiant en TFE et étudiant entrepreneur

Le travail de Sam Bertrand est en lien avec ce projet. Son TFE a pour but de créer la partie physique du produit qui sera appelé « le dispositif » pour la suite de ce document. Ses principaux objectifs sont de créer un circuit électronique permettant d'émettre sa position. Ce dispositif se géolocalise en utilisant la technologie GPS et communique sa position sur le réseau LoRaWAN.

Intégré à la stratégie de l'ECAM, le CERDECAM a pour but :

- **La Recherche appliquée**

Deux chercheurs sont engagés à temps plein afin de travailler sur des projets de recherche en collaboration avec des entreprises et des universités.

- **Le Service technologique aux entreprises**

Certaines entreprises n'ont pas la possibilité ou les capacités techniques de développer leurs propres projets. Le CERDECAM met alors à disposition des enseignants ou des étudiants en stage/TFE afin de pouvoir répondre à leur demande.

- **La Formation continue**

Le CERDECAM offre des formations continues à destination des entreprises. Des professeurs ou chercheurs offrent leur expertise dans des domaines précis comme le

MOD BUS, l'impression 3D ou la communication en milieu industriel. Les formations sont actuellement toutes des réponses à des demandes particulières intra-entreprises créées sur mesure. Le but dans le futur est de proposer des formations continues inter-entreprises.

- **Le Soutien à l'entrepreneuriat et aux projets innovants**

Le CERDECAM offre la possibilité à des étudiants sélectionnés de devenir « étudiant entrepreneur ». Les étudiants sélectionnés ont alors l'occasion de modifier légèrement leur programme de cours afin de développer leur entreprise. Les projets doivent relever un défi technologique.

Les locaux de l'entreprise se situent au dernier étage du bâtiment « Vinci 1 » de l'ECAM. Les chercheurs, stagiaires et étudiants en TFE travaillent dans un open-space, ce qui permet une communication transversale. Mme Halsberghe, quant à elle, dispose de son propre bureau, facilement accessible pour les autres membres de l'entreprise. Le CERDECAM dispose aussi d'une salle de réunion.

Les journées commencent à 9h et finissent à 17h. L'essentiel du temps de travail est passé dans l'open-space. Un tableau blanc est mis à disposition afin de communiquer et d'illustrer les idées.

I.B.2. PROJET TECHNIQUE

La partie technique du projet est d'offrir un objet connecté léger et portable permettant de localiser une personne si celle-ci sort d'une zone définie. L'information sur la position serait transmise sans fil à un serveur cloud. Ce serveur serait capable de notifier l'incident au garant.

Le projet technique peut se séparer en deux grandes parties :

- une partie hardware qui comprend le dispositif et la communication des données,
- une partie software qui comprend une plateforme cloud et une application d'affichage des données.

Les deux premiers mois de l'année scolaire ont été consacrés à un stage qui avait pour but de préparer le terrain pour ce travail de fin d'études.

Une architecture générale a été imaginée en se basant sur des recherches portant sur les architectures IoT :

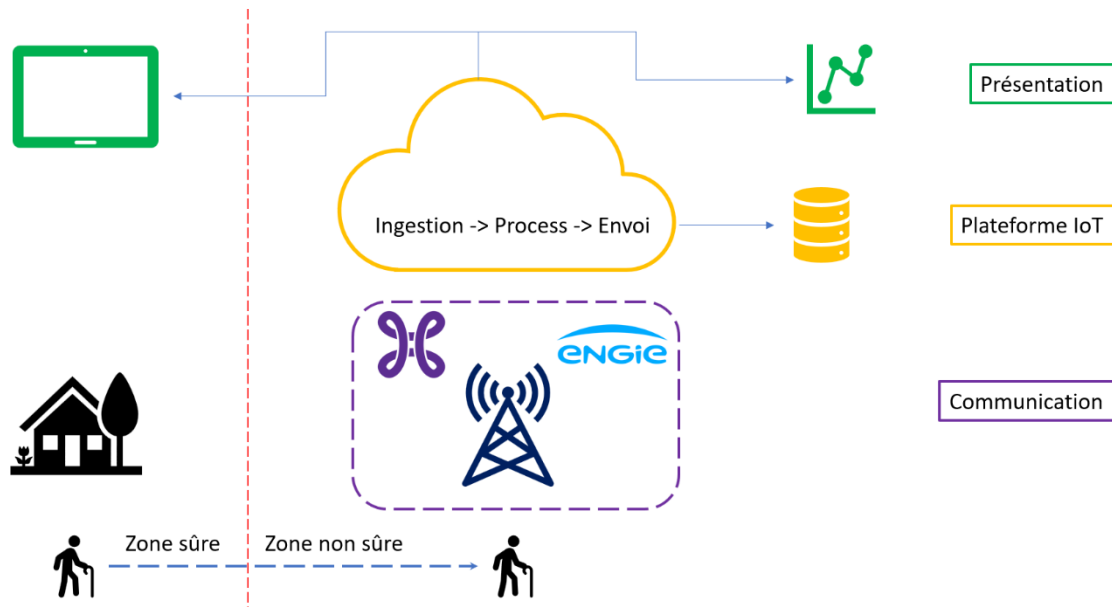


Figure 1 Architecture globale du projet

Tant qu'un dispositif reste dans la zone sûre, celui-ci est considéré comme passif.

Au moment où un dispositif quitte cette zone, le GPS s'active et les coordonnées sont envoyées via un réseau WAN¹. Ces informations sont récoltées par un serveur cloud. Ce serveur traite les informations et les stocke. Les données sont ensuite présentées au garant pour que celui-ci entame la recherche du dispositif égaré.

Un premier état de l'art a également été dressé. Celui-ci porte principalement sur la partie concernant l'objet connecté. Les différentes technologies qui ont été abordées portaient sur :

- la télécommunication des systèmes embarqués
- la géolocalisation
- les matériaux pour la conception

De plus, le dimensionnement de la consommation électrique du dispositif a été calculé. Le microcontrôleur a également été choisi.

Ce stage s'est conclu en sélectionnant les différentes technologies pour créer le prototype électronique.

Pour la communication, c'est la technologie LoRaWAN qui a été choisie, tandis que pour la géolocalisation, le choix s'est porté sur la technologie GPRS avec une puce Ublox.

Pour le microcontrôleur, la décision a été prise d'utiliser un PIC24.

Ce travail se concentre sur la partie software de la solution. Sam Bertrand quant à lui, consacre son travail de fin d'étude à la réalisation du hardware.

¹ Wide Area Network. Les réseaux les plus connus pour l'IoT sont LoRaWAN déployé en Belgique par Proximus et Sigfox étendu dans toute l'Europe par Engie

Un budget de 500€ a été alloué au projet par le CERDECAM. Ce budget est commun aux deux parties du projet. Etant donné le nombre de composants et l'achat d'un kit de développement pour la conception électronique du MVP², une décision a été prise : minimiser les coûts pour le développement du software afin de laisser plus de liberté au développement du hardware.

² Minimum Viable Product : produit non-utilisable disposant uniquement des fonctionnalités nécessaires au bon fonctionnement du service.

I.C. OBJECTIFS

L'objectif de ce travail est de créer la partie de traitement de données du projet NeoFinder. Cela comprend une plateforme cloud et une application utilisateur.

La plateforme doit être capable de

- offrir une interface pour réceptionner les données,
- stocker les données reçues,
- gérer les comptes clients,
- déterminer si une localisation se trouve dans la zone sûre du client,
- notifier le client sur son application mobile en cas de localisation en dehors de la zone.

L'application utilisateur doit être capable de :

- permettre au client de se connecter,
- établir la zone sûre du client,
- afficher les positions envoyées depuis la plateforme sur une carte,
- afficher le dernier état connu des différents dispositifs du client.

Un aspect de de sécurité doit être abordé.

Les communications entre la plateforme et l'application doivent être sécurisées, c'est-à-dire que les données doivent être transmises via HTTPS.

Pour accéder à la plateforme, les requêtes doivent présenter un token d'authentification.

Le traitement et le stockage des données doit aussi être sécurisé. Les requêtes effectuées au système de gestion de base de données (SGBD) doivent être chiffrées avec un certificat SSL. Le SGBD doit aussi être protégé par une authentification de l'utilisateur.

Les composants développés ont pour but d'être modulaires. L'objectif est de créer des services utilisables le plus indépendamment les uns des autres pour pouvoir modifier les différentes fonctionnalités de la solution après le retour de l'équipe de marketing ou des utilisateurs.

II. TECHNOLOGIES

Dans ce chapitre, les différentes technologies utilisées pour mener à bien ce projet sont présentées. Tout d'abord, l'état de l'art est établi. Les langages et bibliothèques choisis pour créer la plateforme et l'application utilisateurs sont présentés. L'environnement de développement est brièvement exposé.

II.A. ETAT DE L'ART

II.A.1. ARCHITECTURE GENERALE

L'architecture générale a été établie lors du stage et a permis d'identifier les différents composants de la plateforme.

Le dispositif peut communiquer soit en réseau local, avec une *gateway*, soit directement avec le réseau WAN. Ces informations sont envoyées au serveur, qui, en fonction du système de télécommunication, reçoit ou va chercher les informations de l'objet connecté. Le serveur traite ensuite les données et les stocke. Les données sont présentées au client ou sont utilisées afin de faire des statistiques ou du *big data*.

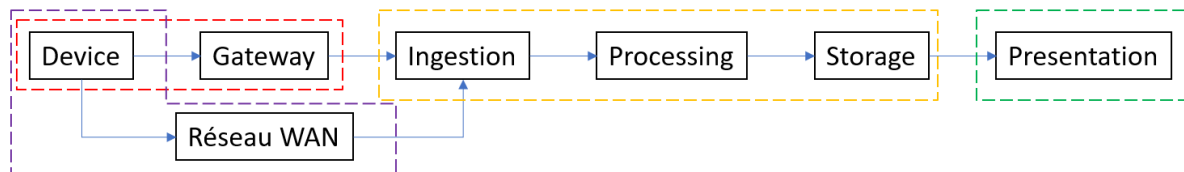


Figure 2 Architecture générale d'une application IoT

- En **rouge** : la communication en réseau local
- En **mauve** : la communication en réseau WAN
- En **orange** : la partie serveur/cloud
- En **vert** : la partie affichage de données

Ce travail se concentre principalement sur les parties serveur/cloud et affichage des données.

On peut distinguer :

- Un composant faisant l'interface avec le réseau WAN et permettant de récolter les informations

Il s'agit ici d'une API web (*Application programming interface*)

- Un composant permettant de traiter les informations

Il s'agit ici d'une application serveur

- Un composant permettant de stocker les informations

Il s'agit ici d'un système de gestion de base de données

- Un composant permettant d'afficher les informations

Il s'agit ici d'une application mobile ou web

Comme montré en orange, les composants API, serveur et base de données peuvent être rassemblés sur une plateforme cloud.

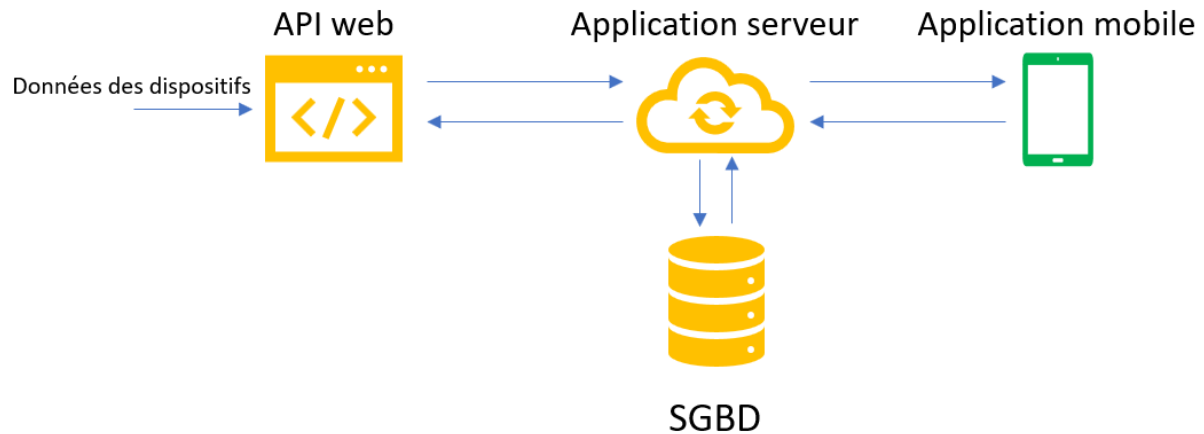


Figure 3 Agencement des composants de la plateforme cloud et du composant de présentation

II.A.2. SERVEUR – API

Afin de faciliter le développement, un langage de programmation doit être choisi. Le logiciel écrit avec ce langage permettra de recevoir des données depuis le réseau WAN, enregistrer ces informations dans une base de données et les transférer à l'application mobile.

II.A.2.a) PYTHON

« Python est un langage de programmation multi-paradigme favorisant principalement la programmation impérative structurée, fonctionnelle et orientée-objet. [...] Python est un langage interprété fonctionnant sur la plupart des plateformes informatiques. » (Combéfis, Python, 2020)

Ce langage s'utilise dans de nombreux domaines d'application. Ceci est renforcé grâce à son système de package (pip) permettant l'installation de bibliothèques dans l'environnement Python.

Python est très utilisé pour les applications liant la big data, et les solutions scientifiques.

II.A.2.b) NODE.JS

« En tant qu'environnement d'exécution JavaScript asynchrone et orienté événement, Node.js est conçu pour générer des applications extensibles. » (OpenJS Foundation, 2020)

Node.js a aussi son propre système de package, node packet manager (npm). Node.js est de plus en plus utilisé dans les entreprises, principalement dans les applications web.

II.A.3. APPLICATION UTILISATEURS

Dans le cas de la solution actuelle, l'application utilisateurs doit être disponible sur mobile et sur navigateur. Cette disponibilité permet de faciliter l'utilisation et de toucher un maximum d'utilisateurs.

En mars 2020, les parts de marché des systèmes d'exploitation dans le monde se répartissent ainsi :

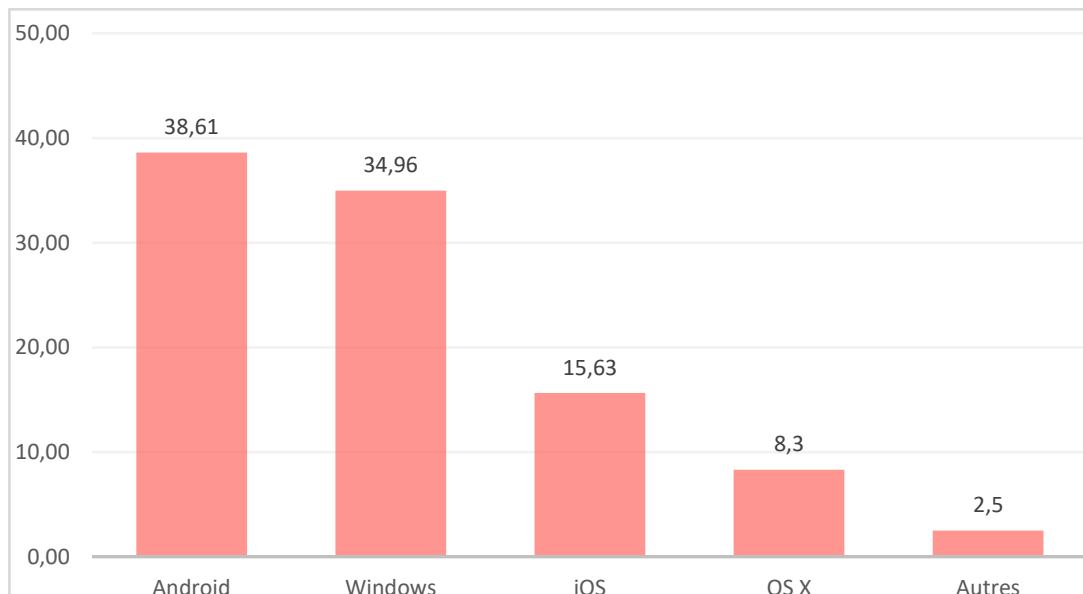


Figure 4 Parts de marché (%) des systèmes d'exploitation dans le monde (StatCounter, 2020)

Si on s'attarde uniquement aux parts de marché des systèmes d'exploitation mobiles, la proportion entre Android et iOS est d'environ 70/30 en faveur de l'OS de Google.

Choisir un seul système d'exploitation mobile n'est pas possible car chaque partie prenante du duopole possède une part de marché trop importante.

De plus, il est nécessaire de permettre aux futurs utilisateurs de pouvoir utiliser la solution sur un ordinateur classique.

Les différents outils disponibles pour le développement sont explorés ci-dessous.

II.A.3.a) XAMARIN

« Xamarin est une plateforme d'applications open source de Microsoft permettant de créer des applications iOS et Android modernes et performantes avec C# et .NET. » (Microsoft, 2020)

« C# (prononcé "Cé Sharp") est un langage de programmation moderne, orienté objet et type-safe. Le C# a ses racines dans la famille des langages C et sera immédiatement familier aux programmeurs C, C++, Java et JavaScript. » (Microsoft, 2020)

Xamarin a l'avantage de développer, en un seul langage, une application mobile Android et iOS. En revanche, Xamarin ne permet pas de créer d'applications utilisables sur un ordinateur classique. Certaines parties du code écrites en C# peuvent toutefois être utilisées pour écrire l'application « desktop ». Malgré cela, il existera deux versions de la solution à créer et à maintenir.

II.A.3.b) IONIC

« *Ionic Framework est un ensemble d'outils d'interface utilisateur open source permettant de créer des applications, mobiles et de bureau, performantes et de haute qualité en utilisant les technologies du web - HTML, CSS et JavaScript - avec des intégrations pour des frameworks populaires comme Angular et React.* » (Ionic, 2020)

Ionic, avec l'aide de Cordova, ou Capacitor, permet de faire des appels natifs au système d'exploitation.

« *Capacitor est un environnement d'exécution d'application multiplateforme qui permet de créer facilement des applications web qui fonctionnent en natif sur iOS, Android et le web. Nous appelons ces applications "Native Progressive Web Apps" et elles représentent la prochaine évolution au-delà des applications hybrides.* » (Ionic, 2020)

Cela permet d'écrire une application unique pour les différentes plateformes des utilisateurs. En plus d'une application mobile et d'une application web, Capacitor permet de compiler la solution pour une application desktop, avec l'aide d'Electron.

II.A.3.c) NATIF

Il est aussi possible de créer des applications dites natives.

Pour une application mobile sur Android, les langages sont le Java ou le Kotlin.

« *Java™ est un langage de programmation orientée objet. [...] Un des grands avantages de Java est sa portabilité, principalement due à l'existence de machines virtuelles permettant d'exécuter du bytecode Java et disponibles sur plusieurs systèmes d'exploitation.* » (Combéfis, Java, 2020)

« *Kotlin est un langage de programmation moderne statistiquement typé qui augmentera votre productivité et le bonheur de vos développeurs.* » (Google developers, 2020)

Kotlin se base sur la machine virtuelle Java et profite de sa portabilité.

Du côté iOS, le langage de programmation est l'Objective-C ou le Swift.

« *Swift est un langage de programmation puissant et intuitif pour macOS, iOS, watchOS, tvOS et au-delà. [...]* » (Apple, 2020)

L'Objective-C est de plus en plus déprécié.

Se baser sur un langage natif permet une meilleure expérience utilisateur. Cependant, cela demande de créer et maintenir différentes versions de la même solution.

II.A.4. BASE DE DONNEES

Dans la conception d'une solution software, le choix d'un système de base de données est primordial. Il existe deux grandes familles de bases de données : les bases de données relationnelles (appelées aussi bases de données SQL) et les bases de données non relationnelles (appelées aussi NoSQL)

II.A.4.a) SQL

Les bases de données SQL utilisent le principe ACID

- **Atomicity** – Atomicité : chaque transaction est soit entièrement réussie, soit entièrement annulée
- **Consistency** – Cohérence : les données écrites dans une base de données doivent être valables selon toutes les règles définies
- **Isolation** – Isolation : lorsque les transactions sont exécutées simultanément, elles n'ont pas d'impact entre elles et agissent comme si elles étaient exécutées de manière séquentielle
- **Durability** – Durabilité : une fois qu'une transaction a été engagée dans la base de données, elle est considérée comme permanente, même en cas de défaillance du système

Du point de vue de l'extensibilité, ce type de base de données s'étend verticalement. C'est-à-dire qu'il est nécessaire d'augmenter les capacités d'un serveur pour étendre la base de données. Celles-ci sont conçues pour fonctionner sur un serveur unique afin de garantir la cohérence des données.

Ce paradigme est un bon choix pour des données structurées et pour les systèmes orientés « transaction ».

II.A.4.b) NOSQL

Les bases de données NoSQL utilisent le principe BASE

- **Basic Availability** – Disponibilité basique : cela signifie que le système garantit la disponibilité, bien qu'il ne puisse pas obtenir certaines données demandées. Les données peuvent être incohérentes ou dans un état changeant.
- **Soft state** – État mou : la structure de la base de données peut évoluer dans le temps.
- **Eventual consistency** – Cohérence éventuelle : la base de données finira par devenir cohérente et les données se propageront partout à un moment donné dans l'avenir.

Du point de vue de l'extensibilité, ce type de bases de données s'étend horizontalement. C'est-à-dire qu'il est possible d'ajouter des serveurs à l'architecture pour étendre la base de données. Cela peut, en plus d'augmenter la taille du stockage, augmenter la disponibilité du système de base de données.

Ce paradigme est un bon choix lorsque la structure des données est incertaine et pour des systèmes flexibles.

II.B. CHOIX TECHNOLOGIQUE

II.B.1. SERVEUR

Version de Python : 3.6.9

L'API est écrite en Python. Python est un langage de programmation utilisé à l'ECAM depuis la première année d'études. À l'aide de bibliothèques, ce langage se révèle polyvalent et facilite la création d'applications.

II.B.1.a) FLASK

Version de Flask : 1.1.1

Flask est la bibliothèque Python permettant d'exposer un serveur web. Cette bibliothèque est très utilisée pour créer des serveurs de développement. Il est possible de créer un serveur de production en utilisant Flask. Ceci est réalisable en y ajoutant *NGINX*, *Gunicorn* et *Supervisor* sur un serveur Linux.

Flask a été choisi comme interface passerelle de serveur web (aussi appelé WSGI pour *Web Server Gateway Interface*) car il s'agit d'un micro-framework avec une grande flexibilité au niveau de la structure de l'application serveur et de ses interactions.

II.B.1.b) MONGODB

Version de mongoDB : 4.2.3

Les données de notre application sont structurées, mais il est possible qu'elles changent en fonction de la demande des clients, ce qui justifie le choix d'une base de données NoSQL pour une première version du projet.

MongoDB est une base de données NoSQL orientée document. Cette base de données est polyvalente et est utilisée par les plus grandes entreprises informatiques telles que Google ou Facebook. De plus, les bases de données orientées documents sont régulièrement utilisées pour les applications IoT.

II.B.1.c) FIREBASE

Firebase est un service Google permettant de gérer des applications iOS, Android et web. Ce service dispose d'un grand nombre de fonctionnalités, afin de faciliter le développement mobile. Dans le cadre de ce projet, Firebase est utilisé pour envoyer des messages aux applications. Cette fonctionnalité s'appelle *Firebase Cloud Messaging* (FCM).

II.B.2. APPLICATION UTILISATEURS

Lors de la création du projet, le développement s'est concentré sur l'application mobile. Ne disposant pas d'ordinateur sous macOS, ni de téléphones sous iOS, seule l'application Android a pu être développée et testée.

II.B.2.a) IONIC

Version de Ionic : 5.4.9

Version de Angular CLI : 7.1.1

Version de Node : 11.3.0

Version de NPM : 6.13.1

Ionic est un framework spécialisé pour la création d'applications mobiles. Ce framework, comme expliqué dans le paragraphe sur l'état de l'art II.A.3.B), permet de développer une application disponible sur plusieurs plateformes (web, mobile, desktop).

Ionic peut se baser sur différents frameworks JavaScript tels que Angular, Vue ou React. Dans le cadre de ce projet, l'application est écrite à l'aide d'Angular.

Angular base son architecture sur des modules appelés *components* et *services*.

Les *components* sont séparés en deux parties : la logique et l'affichage. La logique est écrite en JavaScript tandis que la vue du *component* est écrite en HTML/CSS.

Les *services* sont communs aux différents *components*. Un service est un objet ou une fonction utilisés à plusieurs endroits dans l'application. Par exemple, la plupart des applications Angular ont un service pour contacter une base de données.

Ces données récoltées par les services peuvent être mises dans des objets, afin de simplifier la structure et la compréhension du code.

II.B.2.b) CAPACITOR

Capacitor est un lien entre Ionic et le système d'exploitation permettant de faire des appels aux fonctionnalités natives des différentes plateformes.

Un autre lien de ce type, Cordova, existe. Les fonctionnalités sont globalement les mêmes. La principale différence est que Capacitor, lancé en 2018, utilise des API plus récentes, dont certaines ne sont pas disponibles avec Cordova, créé en 2009.

II.B.2.c) LEAFLET

Version de Leaflet : 1.6.0

Version de Leaflet.Draw : 1.4.0

Leaflet est une librairie JavaScript permettant de gérer les cartes *OpenStreetMap* et *GoogleMaps*. Ces cartes sont incluses dans des pages web. Dans le cadre de ce projet, ce sont les cartes fournies par *OpenStreetMap* qui ont été utilisées, la raison principale étant leur gratuité.

En addition avec Leaflet, Leaflet.Draw est un plugin permettant à l'utilisateur de dessiner des formes sur une carte.

II.B.3. ENVIRONNEMENT

Le projet a été développé sous Windows.

- OS : Windows 10 Pro 64-bit
- CPU : Intel(R) Core (TM) i7-6650U CPU @ 2.20GHz
- Stockage : 250Go SSD
- RAM : 8Go

Le *virtual private server* (VPS) de développement a été commandé chez OVH. Ces spécifications techniques sont :

- OS : Ubuntu 18.04 Server 64-bit
- CPU : 1vcore @ 2GHz
- Stockage : 20Go SSD
- RAM : 2Go

II.B.3.a) VISUAL STUDIO CODE

Visual Studio Code est un éditeur de texte distribué par Microsoft. Cet éditeur est à la fois léger et puissant. Il est possible d'installer des plugins et de personnaliser l'environnement de développement. De plus, cet éditeur dispose d'une invite de commande intégrée.

Pour la partie serveur, le plugin SFTP a permis de synchroniser les modifications faites en local avec le VPS. L'invite de commande de l'éditeur est connectée au serveur en SSH.

Pour la partie application mobile, l'invite de commande est utilisée pour *build* l'application et lancer le serveur Ionic.

II.B.3.b) POSTMAN

Postman est un programme permettant de créer des requêtes HTTP(S). Ce software a été très utile lors des différents tests de fonctionnalités de l'API.

Il est possible de modifier les *headers* des requêtes afin de présenter un *Authentication-Token* pour les routes sécurisées.

III. DEVELOPPEMENT

III.A. ARCHITECTURE

Maintenant que tous les éléments constitutants sont présentés, voici comment ces composants s'agencent ensemble. Le rectangle pointillé en mauve étant la partie du TFE de Sam Bertrand :

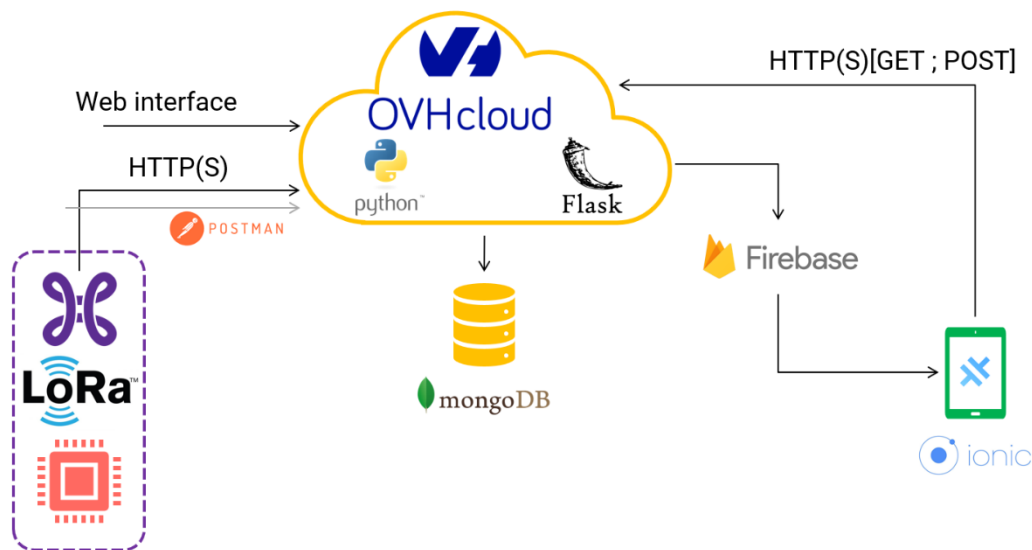


Figure 5 Agencement des différents composants de la solution

L'élément central est un serveur cloud hébergé par OVH. Sur ce serveur, une application Python utilise le framework Flask. Cette application permet de créer l'API et utilise MongoDB comme système de gestion de base de données.

Le serveur de réseau LoRaWAN, de Proximus, envoie les données des dispositifs sur l'API. Ce serveur utilise le protocole HTTP(S) pour envoyer ses requêtes. Durant le développement, ces requêtes sont simulées par le logiciel Postman.

De l'autre côté, le service Cloud Messaging de Firebase permet d'adresser des messages à des instances ciblées de l'application utilisateurs. Cette application Ionic utilise le protocole HTTP(S) pour établir une communication avec la plateforme.

L'API dispose également d'une interface web qui présente l'équipe et le projet. Il est aussi possible de créer un compte client depuis cette interface.

La solution de NeoFinder est un bracelet de géolocalisation permettant aux garants de retrouver les pensionnaires égarés. Ils reçoivent une notification sur leur smartphone lorsque ceux-ci quittent un périmètre défini. Voici le diagramme de séquence reprenant les éléments nécessaires au bon fonctionnement du système d'alerte.

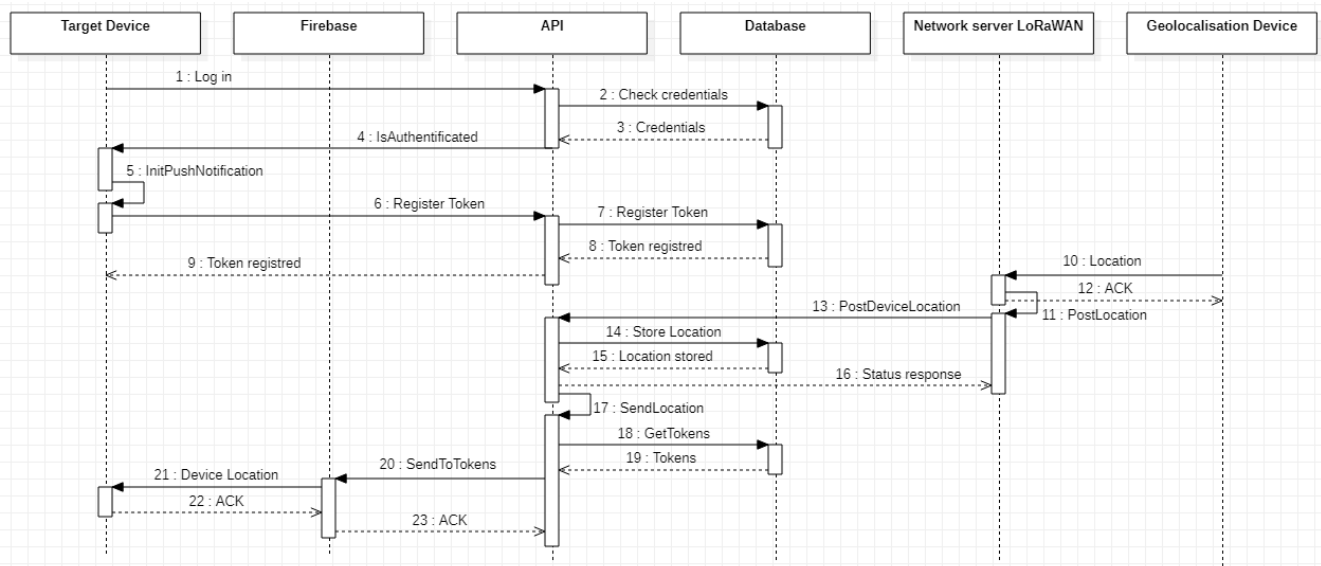


Figure 6 Diagramme de séquence pour la notification d'une localisation

Pour commencer la séquence, l'utilisateur doit s'identifier afin d'être notifié.

Une fois identifié, le smartphone génère un *token* qui va permettre au service de FCM de lui adresser un message. Ce *token* est envoyé vers le serveur et stocké dans la base de données, avec les informations du client.

Quand le dispositif de géolocalisation envoie une position, cette donnée passe par le serveur de réseau LoRaWAN de Proximus. Ce serveur transfère la donnée vers le serveur-API de la solution. Le serveur enregistre cette position dans la base de données. Etant donné que l'utilisateur s'est enregistré auparavant, l'API est capable de trouver le *token* FCM du client propriétaire du dispositif de géolocalisation.

Pour finir, l'API fait une requête vers le service Firebase, avec un message et le *token* correspondant à l'application utilisateurs ciblé, afin que le client puisse être notifié.

III.B.IMPLEMENTATION

Le développement des deux parties de la solution, la partie serveur et la partie application mobile, a été effectué en parallèle. Cela a permis un développement agile, réalisé en plusieurs itérations.

III.B.1. SPRINT 1: CREATION DU SQUELETTE

Ce sprint a pour but d'implémenter le squelette de l'architecture. L'objectif, en fin de sprint, est d'avoir un message envoyé à l'API qui le transfère vers l'application mobile.

En premier lieu, le serveur est commandé pour un an, jusqu'au 20 novembre 2020. L'hébergeur OVH propose différentes offres, les performances du serveur de base sont suffisantes pour développer la plateforme de NeoFinder.

Pour configurer le serveur, une connexion SSH est établie depuis un terminal PowerShell.

Certaines règles de bonne pratique sont suivies :

- Modification du mot de passe fourni par OVH
- Création d'un utilisateur autre que *root*. Cet utilisateur a le droit d'utiliser la commande *sudo*, afin de réaliser des opérations d'administrateur
- Mise en place d'un firewall : Uncomplicated Firewall (UFW) afin de ne laisser que le port SSH et le port de Flask ouvert.

Une fois l'administration du système terminée, la librairie Flask est installée.

Une application « Hello, World » est lancée sur le serveur avec l'adresse host '0.0.0.0', pour utiliser l'adresse IP du serveur, et sur le port 5000. Une connexion au serveur depuis un navigateur est établie afin de tester si le serveur est en place et accessible.

Le serveur étant fonctionnel, le développement de l'application mobile débute. L'application est créée avec Ionic. Grâce à cette instruction écrite dans une invite de commande, la base de l'application mobile est créée avec Capacitor :

```
$ ionic start NeoFinderApp sidemenu -capacitor
```

Le template **sidemenu** permet d'avoir un bouton de type « burger » en haut à gauche.

Appuyer sur ce bouton ouvre le menu et permet de naviguer entre les différentes pages. Ces pages sont nommées **Home**, **Login**, **List** et **Zone**.

Afin d'envoyer des messages depuis l'API vers l'application mobile, la solution utilise Firebase. Ces messages sont appelés des *push notifications* dans la documentation de Capacitor et sont supportées nativement par le framework.

Il est nécessaire d'enregistrer l'application mobile dans la console Firebase, afin de récupérer le fichier `google-services.json`. Ce fichier est à insérer dans le dossier `android` des sources de l'application.

La manipulation pour l'application iOS est plus complexe et n'a pas été effectuée. Cette manipulation requiert un compte développeur et un ordinateur Apple.

L'application est désormais capable de générer le *token*. Ce *token* est unique par objet connecté et permet d'identifier vers quel appareil envoyer un message. Ce *token* est envoyé à l'API afin de l'enregistrer dans les données du client. Cela a pour but d'identifier les différents dispositifs d'un client à notifier.

Comme expliqué dans le chapitre consacré aux technologies, la plateforme utilise une base de données MongoDB pour enregistrer les différentes données liées à l'application.

La base de données MongoDB et la librairie Python pymongo sont installées. Tout comme le serveur Flask, la base de données est configurée sur l'adresse host '0.0.0.0' pour exposer MongoDB à l'adresse du serveur au port 27017.

L'API détient désormais des *tokens* pour les *push notifications*. Afin d'adresser une requête au service de message Firebase, le serveur a besoin de deux éléments :

1. La librairie Python firebase_admin,
2. Un fichier `admin_sdk.json`. Ce fichier est trouvable sur la console Firebase. Il donne une clé permettant d'autoriser la requête au serveur Firebase.

Le serveur est désormais capable d'adresser des messages à des instances spécifiques de l'application mobile.

Pour finir le premier sprint, une route est créée sur l'API afin de recevoir les données de géolocalisation. Ces données ont été envoyées par le logiciel Postman durant la période de développement. Ces données sont transmises au service de cloud messaging de Firebase pour qu'elles atteignent un appareil spécifique. Cette fonctionnalité remplit les objectifs du premier sprint.

III.B.2. SPRINT 2: IMPLEMENTATION DES FONCTIONNALITES

Ce sprint a pour but d'implémenter les fonctionnalités de l'application. L'objectif, en fin de sprint, est de trouver les localisations affichées sur une carte et de permettre à l'utilisateur de créer les zones sûres. En fin de sprint, la plateforme sera capable de déterminer si une coordonnée GPS se trouve dans cette zone.

Désormais, les données du dispositif de géolocalisation sont envoyées sur l'application mobile. Afin de les afficher sur une carte, l'application utilise la librairie JavaScript *Leaflet*, qui permet de créer des cartes interactives. La page principale, nommée la page **Home**, a été modifiée pour accueillir cette carte.

Un *Listener* est utilisé pour mettre à jour cette carte. Un *Listener* est un objet de type *Subject* qu'on expose en tant qu'*Observable*. Lorsque la méthode *subscribe* est utilisée sur un *Observable*, celle-ci « s'active » quand un nouvel élément est envoyé. Cela a pour effet d'exécuter une fonction qui dispose de ce nouvel élément en paramètre.

Ce *Listener* est mis à jour à chaque fois qu'une *push notification* est reçue. La fonction déclenchée a pour but de placer un marqueur sur la carte aux coordonnées géographiques reçues.

Voici à quoi cela ressemble :

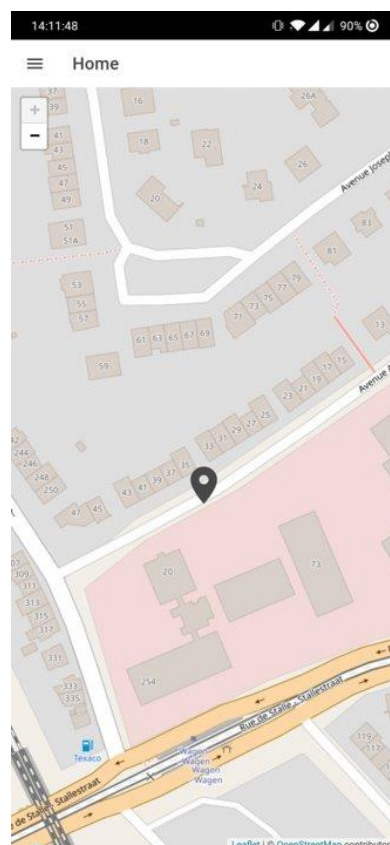
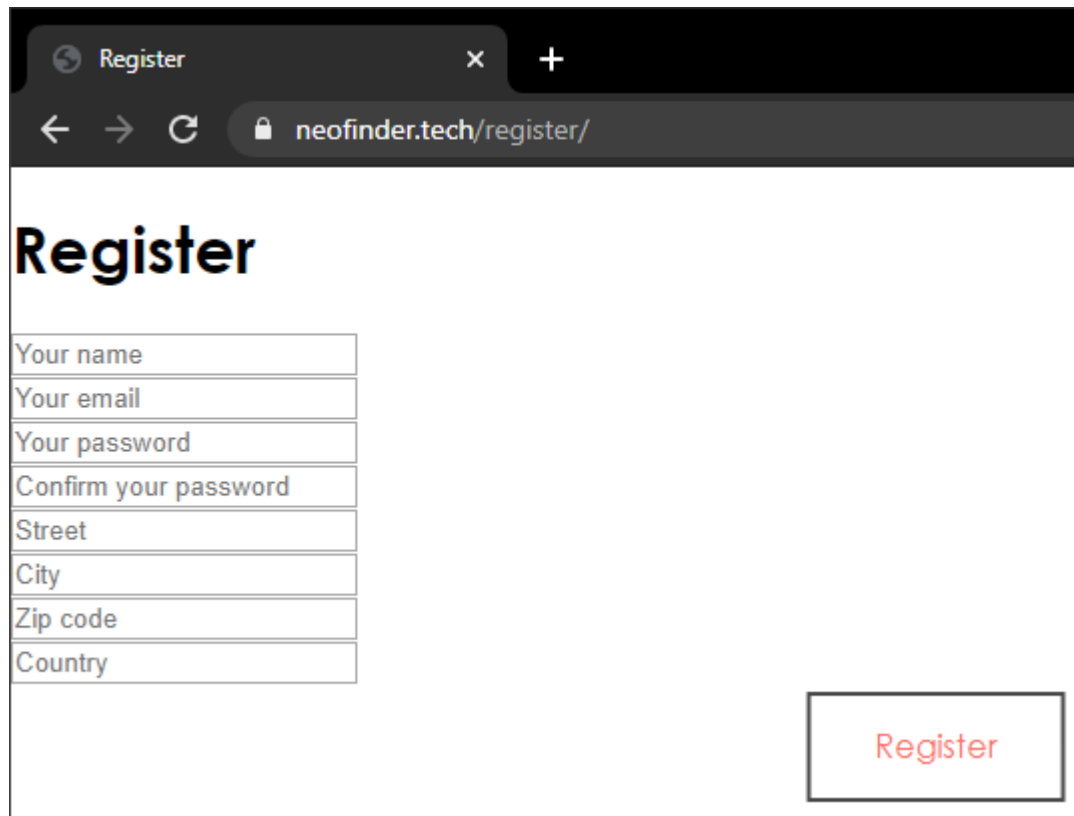


Figure 7 Visuel de la page Home ayant un marqueur de position

L'objectif suivant est de permettre à l'utilisateur de définir sa zone sûre. Pour ce faire, il est nécessaire d'identifier les différents clients.

Premièrement, une route est créée afin de permettre à un futur client de se créer un compte. Cette route affiche le formulaire suivant :



The screenshot shows a web browser window with the address bar displaying "neofinder.tech/register/". The page title is "Register". The form contains the following fields:

- Your name
- Your email
- Your password
- Confirm your password
- Street
- City
- Zip code
- Country

A red "Register" button is located at the bottom right of the form.

Figure 8 Formulaire de création d'un compte client

Les vérifications nécessaires au bon fonctionnement de ce formulaire HTML se font au niveau du serveur-API. Une fois que l'utilisateur a rentré ses coordonnées, elles sont enregistrées dans la base de données. Cet enregistrement est appelé le document du client. Les mots de passe sont chiffrés avec un sel (élément ajouté lors du chiffrement afin que 2 valeurs n'aient pas le même chiffrement) à l'aide de la librairie *Bcrypt*.

Voici un visuel d'un document de client :

```
_id: ObjectId("5e74a688ea52bcb880842c6c")
name: "Palmyra"
street: "Zennebeemd 15"
city: "Drogenbos"
zipcode: "1620 "
country: "Belgium"
hpassword: Binary('JDJiJDE0JHNnRERuNwdGZ0pjZ3BIe1EzbmRKTGVHSTJBb0N3QXJ5a3B4ZnZoQWVvWS5JUUhPUktya05x', 0)
email: "contact@palmyra.be"
```

Figure 9 Exemple d'un document client

Ensuite, une page **Login** est ajoutée à l'application mobile. Celle-ci constitue un formulaire demandant une adresse email et un mot de passe. Une route est ajoutée à l'API afin que le formulaire, sur l'application mobile, puisse demander au serveur une identification. Les données envoyées, via ce formulaire, ne sont pas chiffrées au préalable car la connexion finale se fera avec le protocole HTTPS.

Lorsque le serveur reçoit une demande d'authentification, celle-ci est traitée grâce à la librairie *Bcrypt*. La réponse du serveur à l'application mobile contient le verdict de l'authentification.

Du côté de l'application mobile, se connecter enclenche le mécanisme de *token* pour les FCM réalisés dans le sprint 1. Ces *tokens* sont désormais enregistrés dans le document du client.

Sur l'application mobile, une page **Zone** est créée. Elle a pour but de permettre à l'utilisateur de définir sa zone sûre. Une carte *Leaflet* y est ajoutée ainsi qu'un module permettant de dessiner, modifier et supprimer des polygones. Ces polygones ont pour but de modéliser la zone sûre. Ceci est rendu possible grâce à *Leaflet.Draw*. *Leaflet.Draw* est un plugin qui permet de dessiner différentes formes comme des cercles et des polygones sur une carte Leaflet.

Voici le résultat de la page **Zone**, avec un polygone dessiné :



Figure 10 Visuel de la page Zone ayant une zone sûre dessinée

Deux nouvelles routes sont créées. La première permet de stocker le polygone précédemment créé dans la base de données MongoDB. Les zones des clients sont stockées dans leur document respectif. Une requête est réalisée vers cette route lorsque l'utilisateur quitte la page **Zone**.

La seconde route permet de récupérer les polygones afin de les afficher sur les pages **Home** et **Zone**. Grâce à cette mécanique, les zones sont stockées sur le serveur.

La dernière étape réalisée dans ce sprint est la capacité à déterminer si une position géographique se trouve dans cette zone sûre ou non. Pour réaliser cela, le serveur utilise une librairie nommée *shapely*. *Shapely* est une librairie Python permettant de gérer des formes.

Grâce à cette librairie, il est possible de déterminer si un point se trouve dans un polygone en utilisant la fonction *contains*.

Quand une nouvelle donnée arrive d'un dispositif de géolocalisation, la plateforme va chercher les zones du client qui détient ledit dispositif. Elle regarde alors si le point se situe dans une des zones ou non. Les sommets des polygones étant des coordonnées géographiques, la librairie *shapely* est capable de reconstituer la forme. La dernière donnée de localisation est comparée aux polygones créés et la plateforme est finalement capable de déterminer si cette donnée est dans la zone sûre du client ou non.

III.B.3. SPRINT 3: FINALISATION

Ce sprint a pour but de finaliser la solution. L'objectif, en fin de sprint, est de sécuriser les routes de l'API et la base de données ainsi que de finaliser des dernières fonctionnalités.

Au début du sprint 3, toutes les fonctionnalités nécessaires au bon fonctionnement de la solution sont implémentées.

La première implémentation de sécurité est celle de la base de données. Des utilisateurs ont été ajoutés afin d'administrer la base de données : un utilisateur de type **administrateur** et un utilisateur de type **utilisateur**. L'administrateur a le droit de gérer tout le système de base de données tandis que l'utilisateur n'aura accès qu'à la base de données *neofinder* et ses différents documents. C'est cet utilisateur qui est sollicité pour établir une connexion à la base de données depuis le script de l'API.

La seconde implémentation de sécurité est la sécurisation des routes de l'API. Pour ce faire, la plateforme utilise un système de *token* d'authentification. Ce token, pour le différencier de celui utilisé par le système de FCM, sera appelé *authToken*. Pour la gestion des *authTokens*, en collaboration avec *Flask*, la librairie *flask_jwt_extended* est utilisée.

JWT est l'acronyme de JSON Web Token. Un JWT est un *token* envoyé à l'utilisateur et est capable de retenir toutes sortes d'informations, comme une adresse email, un identifiant ou d'autres données relatives à l'utilisateur. De plus, ces *tokens* peuvent avoir une date de fraîcheur et une date de péremption. Il est formellement déconseillé d'enregistrer des données sensibles telles que les mots de passe dans ces tokens. Une personne expérimentée en cybersécurité est capable de lire ces informations.

La stratégie utilisée dans ce projet est de fournir un de ces *tokens* lors de l'authentification du client. Le token est alors valide durant deux jours et est enregistré chez l'utilisateur. Cela permet à l'utilisateur d'utiliser la plateforme sans devoir se connecter tous les jours. Toutes les routes sont sécurisées par cette méthode³. C'est-à-dire que pour effectuer une requête à l'API, il est nécessaire de présenter un token frais. La fraîcheur du token est de 1 jour. Si, lors d'une requête, le token est valide mais pas frais, l'API va en informer l'application et celle-ci va effectuer une demande afin de rafraichir son token. Grâce à cette mécanique, tant que le client utilise l'application, il ne devra pas se connecter chaque fois. Si l'utilisateur n'effectue aucune action en deux jours, il verra son compte se déconnecter de l'application. L'API, qui n'accepte que des tokens frais ou valides, enverra une instruction de déconnexion à l'application si jamais celle-ci effectue une requête avec un token périmé.

Les objectifs pour la réalisation de la partie logicielle du projet sont sur le point d'être atteints. Afin de clôturer la partie technique, il est nécessaire de faire un lien entre les objets connectés et la plateforme. Un script a été écrit afin de réaliser ce lien. Ce script a pour but de modifier le programme à téléverser sur le dispositif pour que celui-ci ait un ID unique. Pour réaliser cela, une collection est créée dans la base de données. Cette collection reprend les différents dispositifs et leur état. Créer un document reprenant ces informations génère un ID unique

³ Mis à part la page d'accueil, la page de création de compte et la route de login.

dans la base de données. Le script crée un nouveau document et récupère l'ID fraîchement créé, pour l'assigner au dispositif.

Après cette manipulation, un *template* du code est modifié afin que le dispositif garde cet ID créé en base de données.

Finalement, pour transmettre les données des senseurs de géolocalisation vers l'API, Un système de « transaction » est nécessaire, étant donné que ce sont des données sensibles.

Tout d'abord l'API doit autoriser la requête du serveur de réseau LoRaWAN. Ce serveur se connecte avec un identifiant et un mot de passe. L'API lui donne un token valide durant 1 seconde, le temps d'effectuer une transaction.

Une dernière fonctionnalité a été ajoutée. Avec la nouvelle collection de dispositifs, lorsque la plateforme reçoit une nouvelle information de localisation, elle met à jour en base de données l'état de l'objet connecté.

Pour que ces états soient visibles sur l'application, une page **List** est créée. Elle reprend le pourcentage de batterie et la dernière localisation de l'objet connecté. Cette dernière location est aussi affichée sur la page **Home**.

III.C. TESTS

L'API web et l'application mobile ont été testées. Pour l'API, les tests se sont portés principalement sur le comportement des routes exposées. Pour l'application utilisateurs, ce sont les pages et leur comportement qui ont été testés.

Les tests ont été effectués de manière manuelle, systématique et méthodique. Lorsqu'une nouvelle fonctionnalité est implémentée, celle-ci est directement soumise à un test à l'aide des logiciels décrits ci-dessous. Ces tests se sont portés sur le comportement des fonctionnalités fraîchement développées et leurs interactions avec les modules déjà existants.

Le temps pour les tests a été sous-évalué. Aucun test n'a été automatisé. Le temps passé sur le projet a principalement été dédié au développement des fonctionnalités.

III.C.1. LA PLATEFORME

Pour tester la plateforme, le logiciel Postman a été utilisé. Postman comporte plusieurs avantages.

Ainsi, durant la phase de développement, le logiciel a permis de simuler l'arrivée de données de géolocalisation sur l'API.

Deuxièmement, le logiciel est adapté pour tester les routes de l'application et leurs réponses. En faisant des requêtes et en utilisant la console Python, il est possible d'analyser le comportement de l'API.

De plus, il est possible de configurer une requête avec un *token* d'authentification. Ceci est nécessaire pour vérifier les routes sécurisées.

III.C.1.a) EXEMPLE

Pour illustrer le fonctionnement de Postman, voici comment la dernière fonctionnalité implémentée a été testée. Cette fonctionnalité consiste à recevoir le dernier état enregistré des objets connectés.

- Une requête POST sur la route `/api/login` est envoyée afin de recevoir le *token* d'authentification.
- Ce token est ajouté dans le header de la prochaine requête Postman. Il est visible sur la figure 11.
- Une requête GET est réalisée sur la route `/api/getlist` afin de récupérer la liste des dispositifs du client authentifié au format JSON. Le client est identifié en présentant le *token*.
- La réponse de la requête, visible sur la figure 11, affiche la liste des dispositif d'un client. La figure 12 affiche le document de la base de données. On peut remarquer que les dispositifs sont bien envoyés.

Cette méthode est principalement une manière rapide de déterminer si un nouveau module implémenté fonctionne. Une automatisation des épreuves est nécessaire pour des tests de plus grande ampleur.

Chapitre III : Développement

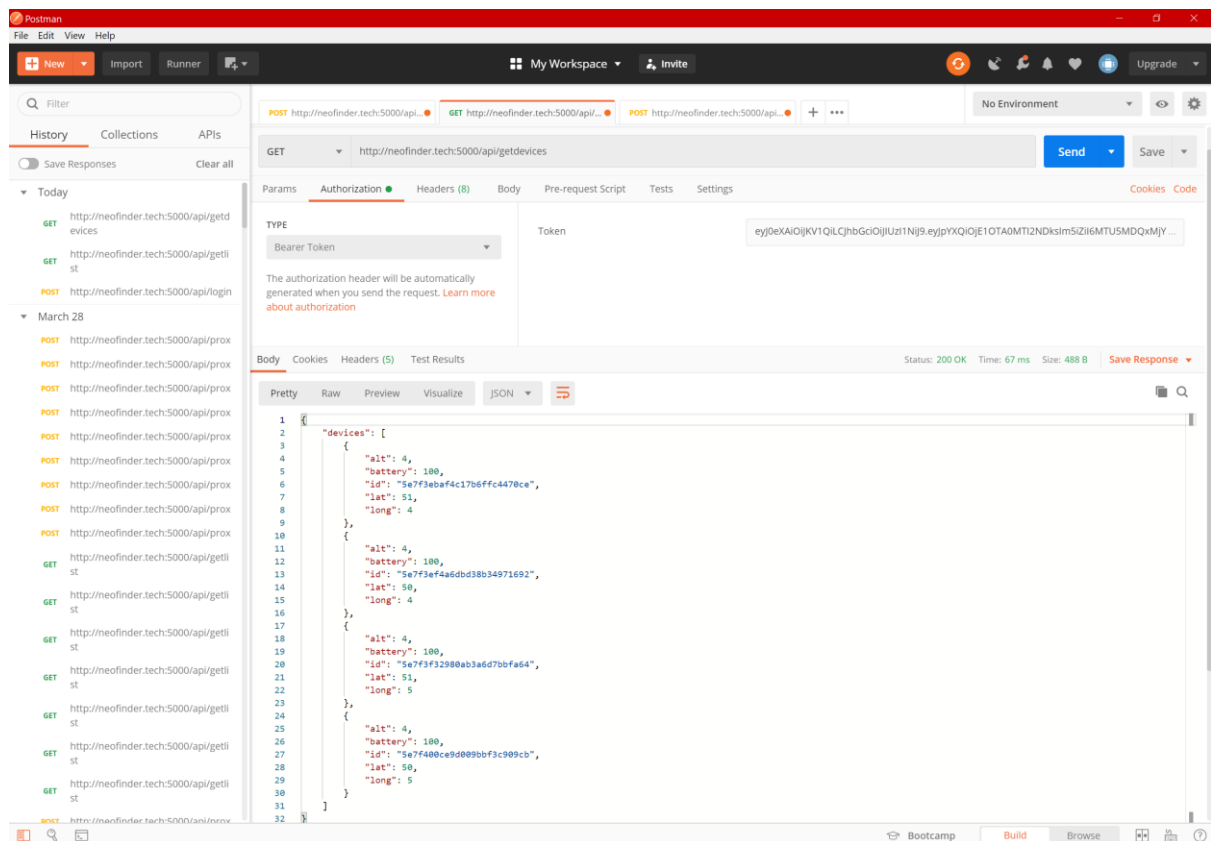


Figure 11 Test de la plateforme : requêtes pour avoir le dernier état des dispositifs

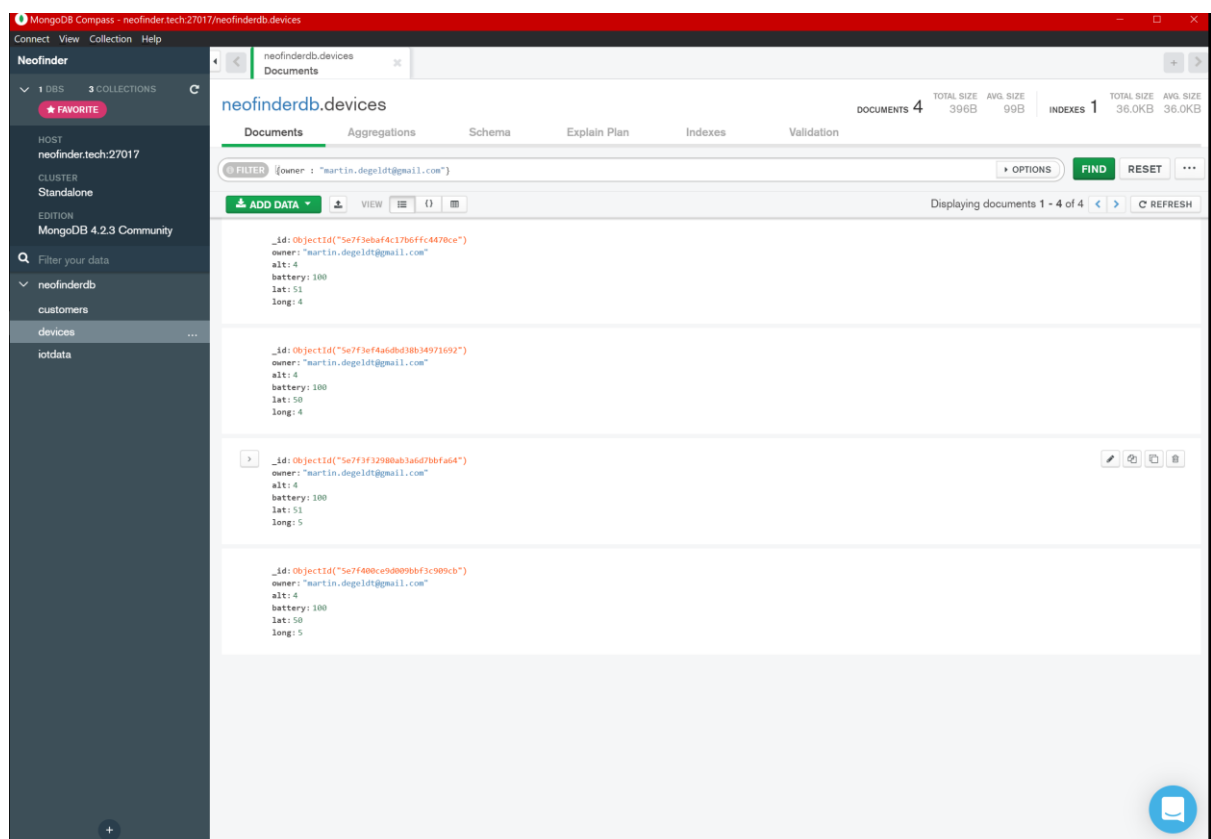


Figure 12 Liste des dispositifs pour un utilisateur dans la base de données

III.C.2. L'APPLICATION

Trois méthodes ont été utilisées afin de tester le fonctionnement de l'application.

La première méthode est d'utiliser Ionic en mode serveur. Cela permet de créer un serveur local sur l'ordinateur de développement afin d'analyser le comportement de l'application. Cette fonctionnalité est très efficace car il est possible de lire les logs dans la console du navigateur.

La seconde est d'utiliser Ionic Lab. Tout comme la première méthode, celle-ci permet de créer un serveur local. Cette fonctionnalité permet d'afficher le rendu de l'application Android et iOS, depuis un navigateur.

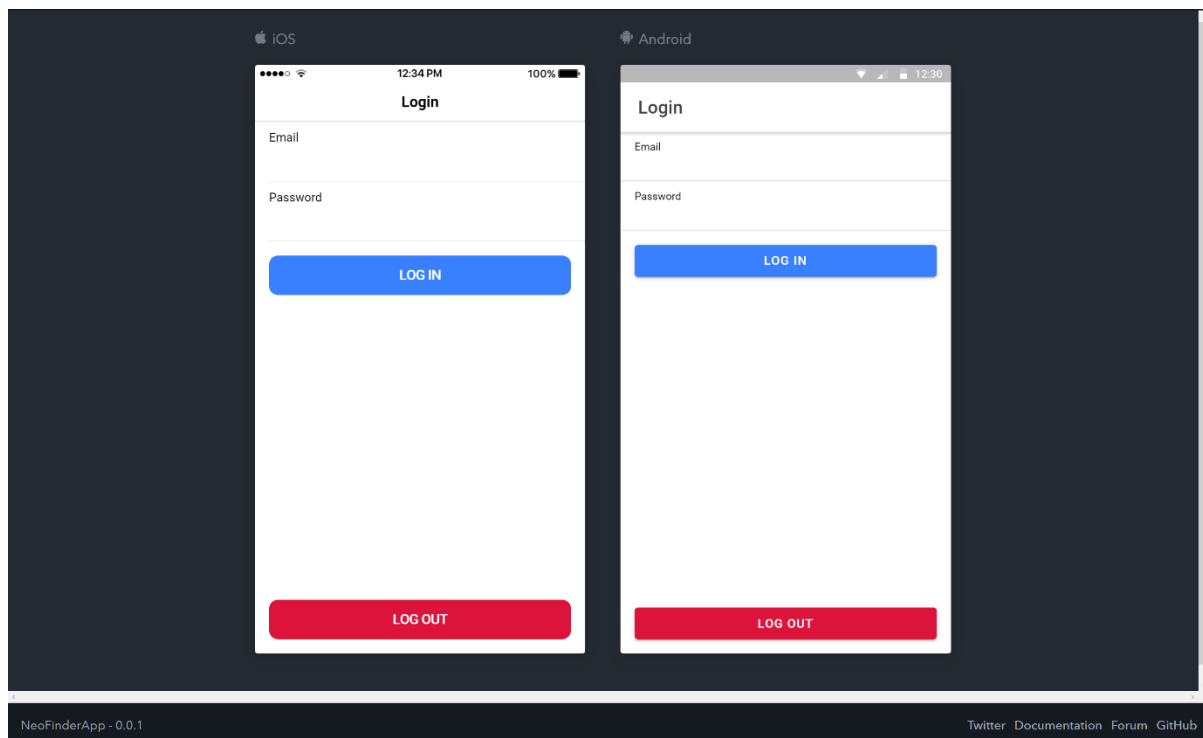


Figure 13 Ionic Lab

La troisième est de compiler l'application fraîchement créée sur un smartphone. Cela permet de tester le comportement de l'application sur Android et iOS. N'ayant pas de dispositif iOS à disposition, les tests ont été effectués uniquement sur Android. Il est aussi possible de lire les logs de l'application, en utilisant la fonctionnalité *remote devices* du navigateur *Google Chrome*.

III.D. DEPLOIEMENT

Pour déployer la plateforme, *NGINX* est utilisé comme serveur HTTP. Ce serveur va permettre de gérer les requêtes statiques, comme les fichiers CSS. En aucun cas, *NGINX* ne gère le code Python. C'est *Gunicorn* qui s'en charge, étant donné que *Flask* n'est pas conçu pour faire de la production. Une fois que *NGINX* est configuré, il est nécessaire de faire fonctionner *Gunicorn*. Pour cela, une simple commande suffit :

```
$ gunicorn -w 3 server:app
```

serveur.py étant le nom du script Python qui exécute le serveur *Flask*.

Le paramètre `-w` permet de déterminer le nombre de *workers* (*w*) qui s'occuperont de gérer les requêtes. Ce nombre est déterminé à partir du nombre de cœur (*c*) que détient la machine :

$$w = 2 * c + 1$$

La machine sur laquelle tourne le serveur possède un cœur, ce qui résulte en 3 *workers* *Gunicorn* fonctionnant désormais en local sur le port 8000.

Cette commande est utilisée dans le fichier de configuration de Supervisor :

```
[program:apiNF]
directory=/home/artoens/server
command= gunicorn -w 3 server:app
user=root
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
stderr_logfile=/var/log/apiNF/apiNF.err.log
stdout_logfile=/var/log/apiNF/apiNF.out.log
```

Supervisor est un logiciel permettant de s'assurer qu'une tâche est en train de s'effectuer. Ce logiciel est capable de lancer la tâche au démarrage de la machine et de la redémarrer quand elle a un problème. Supervisor est aussi capable d'enregistrer les logs de sortie et d'erreur dans les fichiers indiqués dans le fichier de configuration.

Afin de chiffrer les routes et de passer du protocole HTTP au protocole HTTPS, il est nécessaire de demander un certificat SSL. Ce type de certificat peut être obtenu par une Autorité de Certification de confiance, mais cela reste onéreux. Heureusement, OpenSSL fournit des certificats gratuits. Bien qu'ils n'inspirent pas la confiance que peut délivrer une Autorité de Certification, ils sont suffisants pour chiffrer les communications faites avec l'API.

Un certificat OpenSSL s'obtient facilement grâce à un logiciel appelé *certbot*. De plus, une commande est spécialement disponible pour les serveurs utilisant *NGINX* :

```
$ sudo certbot --nginx
```

Cette commande permet de recevoir et d'installer le certificats SSL.

Un script *bash* a été écrit pour renouveler de certificat de OpenSSL tous les premiers du mois.

Une fois que tout cela est effectué, voici à quoi ressemble le fichier de configuration de *NGINX*:

```
server {
    server_name neofinder.tech;
    root /home/artoens/server;

    location /static {
    }

    location / {
        proxy_pass http://localhost:8000;
        include /etc/nginx/proxy_params;
        proxy_redirect off;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/neofinder.tech/fullchain.pem; #
managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/neofinder.tech/privkey.pem;
# managed by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
server {
    if ($host = neofinder.tech) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name neofinder.tech;
    return 404; # managed by Certbot
}
```

Le déploiement pour l'application mobile n'a pas été réalisé.

IV. DISCUSSIONS

IV.A. RESULTATS

Ci-dessous sont présentés un bilan des objectifs et la manière dont ceux-ci ont été réalisés.

La plateforme est désormais capable de :

- Offrir une interface pour réceptionner les données

Grâce à l'API Flask, le serveur expose une route pour recevoir les données de géolocalisation. Devant garantir l'intégrité de ces données, cette réception se fait sous forme de transaction. Le serveur envoyant les données doit s'identifier et recevoir un *token* pour permettre d'envoyer une position à l'API.

- Stocker les données reçues

Ces données sont stockées dans la base de données MongoDB. Pour rappel, les requêtes faites vers la base de données et leurs réponses sont chiffrées avec un certificat SSL.

- Gérer les comptes clients

Les clients sont capables de se créer un compte à l'adresse neofinder.tech/register. L'API fournit au client qui se connecte un *token* permettant d'utiliser les routes de l'API.

- Déterminer si une localisation se trouve dans la zone sûre du client

Cet objectif a été réalisé à l'aide de la librairie *shapely*.

- Notifier le client sur son application mobile en cas de localisation en dehors de la zone

Les applications mobiles sont notifiées avec la mécanique de Firebase Cloud Messaging.

L'application d'affichage est désormais capable de :

- Permettre au client de se connecter

Un formulaire de connexion a été créé sur l'application mobile. Cet objectif est fortement lié à l'objectif « Gérer les comptes clients » des conditions requises de la plateforme.

- Afficher les positions envoyées depuis la plateforme sur une carte

Les cartes sont gérées avec la librairie *Leaflet* au niveau de l'application. Des marqueurs de positions de différentes couleurs ont aussi été créés, afin de mieux différencier les différents dispositifs.

- Établir la zone sûre du client

Une page sur l'application mobile permet de dessiner un polygone sur une carte définissant la zone sûre. Cela a été réalisé à l'aide de *Leaflet.Draw*.

- Afficher le dernier état connu des différents dispositifs du client

Une page sur l'application mobile permet d'afficher le dernier état connu des dispositifs d'un client connecté à l'application. De plus, les localisations de ces objets connectés sont aussi visibles sur la carte de la page d'accueil.

Les communications entre la plateforme et l'application sont sécurisées, c'est-à-dire que les données sont transmises via HTTPS et que les requêtes faites à la plateforme présentent un *token* d'authentification. C'est aussi le cas pour les communications avec la base de données.

IV.B. PERSPECTIVES

IV.B.1. AMELIORATIONS

Avant de clôturer le projet, quelques améliorations notables peuvent être mentionnées.

Tout d'abord, le besoin d'implémenter des tests automatiques. C'est un besoin primordial pour le bon fonctionnement du service, ainsi que pour s'assurer de la robustesse de la plateforme et de l'application avant leur commercialisation. Il est aussi nécessaire d'étendre ces tests. Il n'est pas suffisant de ne tester que les fonctionnalités et le comportement de l'application, les tests doivent aussi couvrir la performance des logiciels.

Deux fonctionnalités ont encore des comportements aléatoires :

1. Un conflit réside entre les cartes Leaflet lors de la transition entre les pages **Home** et **Zone**
2. Les notifications ne sont pas toujours activées lorsque l'application est fermée

Le système de FCM ne fonctionne que sur l'application mobile, une alternative devra être développée pour l'application web.

Certaines fonctionnalités ont été évoquées durant la période de développement mais n'ont pas été implémentées :

- Créer une alerte du niveau de batterie du dispositif
- Générer un nom lisible pour le dispositif, au lieu d'afficher un ID. Ce nom serait généré sur base de l'ID de l'objet et permettrait au client de plus facilement les différencier
- Prévoir un système de secours si les notifications ne sont pas envoyées
- Améliorer l'esthétique de l'application en se basant sur la charte graphique de NeoFinder
- Proposer une personnalisation de l'application mobile, comme un thème sombre ou permettre aux utilisateurs de retrouver le logo de leur entreprise en se connectant.

IV.B.2. DEVELOPPEMENTS FUTURS

Il existe plusieurs pistes pour continuer à peaufiner ce qui a été réalisé lors de ce travail de fin d'études, dans le cadre du projet NeoFinder.

Premièrement, l'application iOS doit être développée. Bien que l'application utilisateur ait été créée avec le framework Ionic, il est obligatoire d'avoir un compte développeur et un ordinateur Apple pour fournir l'application iOS. Certaines manipulations, comme les mécaniques de FCM, sont à retravailler pour porter l'application utilisateurs vers ce système d'exploitation.

Une fois que l'application sera assez robuste et prête à être commercialisée, il sera nécessaire de déployer une infrastructure plus complète. Dans le cadre de ce travail, les trois composants *cloud* (API web, application serveur et système de gestion de base de données) sont centralisés sur une seule machine. Ces composants peuvent fonctionner sur des machines différentes, ce qui permet de diminuer le couplage des modules.

Si une telle infrastructure est mise en place, un système de *monitoring* et de *load balancing* devra être configuré pour s'assurer du bon fonctionnement de l'installation.

V. CONCLUSIONS

Au terme de ce travail, une plateforme qui gère des données de géolocalisation et une application mobile permettant de les afficher ont été développées.

Pour revenir sur ce document, dans un premier temps, le contexte a permis de placer le cadre et les enjeux du projet NeoFinder.

Deuxièmement, l'état de l'art a permis de comprendre et d'argumenter le choix des technologies utilisées dans ce travail.

Ce document s'est poursuivi par une explication du développement en trois itérations :

- la création du squelette,
- l'implémentation de fonctionnalités,
- la sécurisation et l'achèvement du développement.

À court terme, les modifications pour l'application iOS devront être effectuées. Le système de FCM ne fonctionnant pas pour l'application web, une alternative devra être trouvée. Les tests seront étendus. Une version « alpha-test » sera distribuée aux membres de l'équipe, avant de proposer une version « bêta-test » pour les entreprises qui souhaiteraient tester notre solution.

À moyen terme, il sera possible de distribuer l'application aux utilisateurs en même temps que la vente d'un produit. Il sera aussi possible de créer une application desktop, pour répondre aux besoins du client. Cela rentre dans une stratégie de développement continu en utilisant les retours des utilisateurs. Une alerte du niveau de batterie du *device* est une autre amélioration envisageable, si le besoin est exprimé.

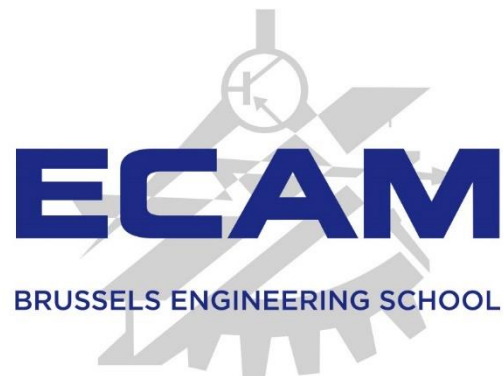
Comme mentionné dans l'annexe entrepreneuriale, une nouvelle idée de proposition de valeur a été émise : un système de géolocalisation à l'intérieur des bâtiments.

Sur le long terme, si le projet pivote tout en gardant le principe de géolocalisation, les modules développés pourront être réutilisés dans d'autres contextes. De la même manière, si l'équipe marketing définit de nouvelles exigences utilisateur, la solution pourra facilement s'adapter dans ce sens.

VI. BIBLIOGRAPHIE

- Apple. (2020). *Swift*. Récupéré sur Apple Developer: <https://developer.apple.com/swift/>
- Combéfis, S. (2020). *Java*. Récupéré sur Uko: <https://www.ukonline.be/cours/java>
- Combéfis, S. (2020). *Python*. Récupéré sur Uko: <https://www.ukonline.be/cours/python>
- Google developers. (2020). *Kotlin and Android*. Récupéré sur Android Developers: <https://developer.android.com/kotlin>
- Ionic. (2020). *Capacitor: A Cross-platform App Runtime*. Récupéré sur Capacitor: <https://capacitor.ionicframework.com/docs/>
- Ionic. (2020). *Ionic Framework*. Récupéré sur Ionic Documentation: <https://ionicframework.com/docs/>
- Microsoft. (2020). *A Tour of C# - C# Guide*. Récupéré sur Microsoft Docs: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- Microsoft. (2020). *What is Xamarin?* Récupéré sur .Net: <https://dotnet.microsoft.com/learn/xamarin/what-is-xamarin>
- OpenJS Foundation. (2020). *A propos*. Récupéré sur Node.js: <https://nodejs.org/fr/about/>
- StatCounter. (2020). *Operating System Market Share Worldwide*. Récupéré sur StatCounter Global Stats: <https://gs.statcounter.com/os-market-share>

VII. ANNEXES



Rapport entrepreneurial de la startup NeoFinder

Travail de fin d'études présenté par
Sam Bertrand et Martin Degeldt

En vue de l'obtention du diplôme de
Master en Sciences de l'Ingénieur Industriel orientation Informatique

REMERCIEMENTS

En premier lieu, nous remercions Mme Queenie Halsberghe, Research Unit Manager au CERDECAM. Grâce à elle, nous avons pu débiter cette aventure entrepreneuriale avec tout le support nécessaire.

Nous tenons à remercier Houda Hannouni pour sa participation au travail d'équipe et l'énergie qu'elle dégage.

Nous remercions aussi Quentin Vandenborre pour la gestion de l'équipe et son travail de stratégie lors de la prise de décisions.

Un grand merci à Cyril Le Ray, qui s'occupe de la conception mécanique, pour l'aide qu'il a apportée au projet.

Merci à Raissa Kasenga Mania pour son aide au travail réalisé et sa bonne humeur.

Nous aimerions aussi gratifier les efforts de M. Combéfis et Mme Flémal pour leur supervision et leur aide technique au projet.

Nous souhaitons adresser nos remerciements à toute l'équipe d'accompagnement du StartLab ICHEC, et principalement à Hélène Chochaux pour l'incroyable travail derrière l'incubateur, Aurélie Mulowa pour nous tenir informés des dernières actualités de la communauté et Xavier de Poorter, notre coach, qui nous oriente tout au long de notre aventure.

Et pour finir, merci à tous les porteurs du StartLab ICHEC pour l'ambiance lors des formations !

CHANGEMENTS DUS AU COVID-19

Dans le cadre de la crise *coronavirus*, le gouvernement belge a établi une distanciation sociale. Les mesures prises dans les hôpitaux et les maisons de retraite limitent les contacts avec le personnel médical et les aide-soignants qui se retrouvent en première ligne.

Ces personnes défendent la vie de nos aïeux et nous avons fait le choix de ne pas les déranger. Cela a eu comme impact que nos contacts marketing ont dû s'arrêter prématurément. Nous avons avancé avec les informations récoltées au préalable et des questions stratégiques ont été posées pour répondre au mieux à cette crise.

Ce temps de pause a aussi permis de faire évoluer l'idée du pivot.

À cause de la distanciation sociale, le calendrier des dernières formations du StartLab ICHEC est modifié et celles-ci sont données à distance jusqu'à nouvel ordre.

RÉSUMÉ

Ce document est en rapport avec le projet entrepreneuriale NeoFinder. Les cas de la maladie d'Alzheimer augmentent tous les ans et le nombre d'aide-soignants reste le même. Au vu de cette situation, l'idée de NeoFinder est née. L'entreprise a pour but de créer un bracelet de géolocalisation connecté via IoT (l'internet des objets) à une plateforme web et mobile. Cette application permettra au personnel des établissements s'occupant de personnes démentes ou désorientées d'être notifiés lorsque ces personnes sortent d'une zone définie au préalable. Ce document a pour but de vous faire partager les avancées au niveau entrepreneurial du projet, a contrario des travaux de fin d'études de Martin Degeldt et de Sam Bertrand auxquels ce document est lié. Le document communique les avancées faites au sein du StartLab ICHEC, en termes de demandes de financement et l'évolution concrète de l'idée ces derniers mois. Les avancées faites au StartLab sont divisées en parcours gagnant, formations et coaching. Au cours de l'année académique 2019/2020 deux demandes de financement ont été introduites, la demande chez Innoviris Brussels ainsi que la demande d'accès à la bourse Albert Vanhée.

TABLE DES MATIÈRES

Introduction	6
I. Contexte.....	7
I.A. Contexte entrepreneurial	7
I.B. Objectifs	8
II. Parcours.....	9
II.A. Parcours gagnant.....	9
II.B. Formations	10
II.C. Coaching	11
II.D. Financements.....	13
II.D.1. Spin-off d'innoviris.....	13
II.D.2. Albert Vanhee	14
III. Discussion	15
III.A. Résultat	15
III.B. Perspectives.....	16
IV. Conclusions	17
V. Bibliographie	18
VI. Annexes	19
VI.A. Teams.....	19
VI.B. Parcours gagnant.....	20
VI.C. Parcours Start-test du StartLab.....	22

INTRODUCTION

Le vendredi 15 mars 2019 se déroulait la première édition du Hack2Start, un hackathon organisé par le StartLab ICHEC et le centre de recherche et développement de l'ECAM (le CERDECAM). Le study case était assez simple :

« Comment peut-on faire pour retrouver des personnes perdues, telles que des enfants dans un supermarché, des joggeuses dans un parc et des personnes âgées quittant leur maison de retraite. L'objectif n'est ni de les tracer ni de les encombrer. La solution a pour but d'être indépendante d'un téléphone portable. »

À l'issue de 3 jours de réflexion, de débats et de calculs, un jury a décidé de l'idée gagnante du concours : une solution spécialisée pour les maisons de retraite, permettant de retrouver les résidents qui s'égarer. Cette solution serait directement proposée aux homes, afin que ceux-ci puissent mettre en place l'infrastructure nécessaire à la sécurité de leurs pensionnaires.

Souhaitant donner suite aux idées émises lors du hackathon, le CERDECAM a proposé aux participants du weekend de poursuivre l'aventure. Les étudiants de l'ECAM qui accepteraient seraient considérés comme *étudiants entrepreneurs*. Nous sommes 3 à avoir répondu à cet appel : Sam Bertrand, Houda Hannouni et moi-même.

En septembre 2019 nous sommes rejoints par 3 autres membres : Cyril Le Ray, Raissa Kasenga et Quentin Vandenborre.

L'objectif du projet est, d'une part, de développer la partie entrepreneuriale de l'idée et, d'autre part, de réaliser la partie technique de la solution.

Nous réalisons ce travail à l'issues de 5 années passées à l'ECAM. Le travail qui suit s'attarde sur la partie entrepreneuriale du projet.

Nous avons collaboré tout au long de l'année, afin de donner vie à notre projet. Nous nous sommes organisés et nous avons réparti les tâches afin d'avancer efficacement, tout en étant un groupe de plusieurs personnes.

Nous allons tout d'abord commencer par poser le contexte de notre parcours entrepreneurial, en y décrivant nos objectifs. Nous allons ensuite passer en revue les différentes actions entreprises pour faire avancer notre projet, ainsi que les formations et les coachings que nous avons suivis. Pour finir, nous discuterons de l'état du projet avant de conclure.

I. CONTEXTE

I.A. CONTEXTE ENTREPRENEURIAL

Six membres se lancent dans le projet entrepreneurial en septembre 2019. Ceci demande un engagement de chacun pour la communication et le fonctionnement du projet. La première étape du projet a été de connaître l'implication de chacun. Ceci pour se répartir les tâches au mieux en fonction des capacités et du temps de chacun. Des réunions se sont tenues aussi souvent que possible pour que tous soient informés des avancées de chacun dans les diverses activités que le projet implique. Les activités consistent en l'avancée technique, le suivi des formations et les actions qui en découlent.

Afin d'organiser le travail au mieux, chaque membre du projet a un rôle. Ce rôle est une responsabilité de supervision. Le travail est fait par tous sur tous les éléments du projet mais le responsable s'occupe d'organiser le travail dans ce domaine. Martin Degeldt et Sam Bertrand sont responsables de la technique. Houda Hannouni est responsable du marketing avec Cyril Le Ray et Raissa Kasenga Mania comme aide principale. Quentin Vandenborre est quant à lui responsable de la gestion.

Des rôles de réunions pour permettre un bon fonctionnement ont été créés. Sam Bertrand est responsable du temps et secrétaire de réunion, c'est lui qui prend note, publie les CV et recentre le sujet lorsque c'est nécessaire. Les notes sont postées sur Teams dans le fichier PV de réunions. Quentin Vandenborre est responsable de l'animation, c'est lui qui sollicite les personnes qui parlent moins pour être sûr d'avoir l'avis de tous les participants de la réunion. Houda Hannouni est ambianceur, c'est-à-dire qu'elle fait attention à la cohésion d'équipe et règle les litiges au sein du groupe.

Le projet est encadré pour pouvoir le mener à bien. Queenie Halsberghe est responsable du CERDECAM, elle aide dans ses domaines d'expertise, surtout les demandes de subsides. Xavier De Poorter est coach assigné par le StartLab. Il aide par son expertise et ses grandes connaissances dans le domaine de l'entrepreneuriat. Hélène Cochaux et Aurélie Mulowa, le staff du StartLab, sont d'une grande aide en termes de mise en réseaux et d'organisation de formations.

Pour l'organisation des communications, des outils ont été mis en place. Le moyen de communication principal est Teams, la plateforme professionnelle de Microsoft. Cette plateforme permet d'organiser le travail via un KANBAN, un gestionnaire de fichier et un calendrier. Messenger par Facebook est utilisé pour la communication au jour le jour. La communication avec les pairs se fait surtout via mail. La plateforme Wikipreneur est utilisée pour la formation StartLab.²

I.B. OBJECTIFS

Les objectifs du travail entrepreneurial pour mener à bien ce projet sont multiples et différents pour chaque membre. Ci-dessous, sont décrits les objectifs de l'équipe. Nous pouvons recenser trois objectifs principaux :

- Un objectif de création d'entreprise

C'est l'objectif général du travail entrepreneurial. Cet objectif reprend différents sous-objectifs. On peut les séparer en trois grandes catégories : Les objectifs de gestion, de marketing et les objectifs techniques. La liste de toutes les étapes sont reprises dans l'annexe « Parcours gagnant »

- Un objectif d'apprentissage

En tant qu'étudiant, l'objectif est d'acquérir le savoir et les compétences utiles au monde professionnel.

Ce projet a permis d'étendre notre champ d'apprentissage. Ce dernier se compose de formations, de séances de coaching, de réflexions et de remises en question mais aussi de bilans et de rétrospectives sur les actions menées avec succès ou non.

- Un objectif de levée de fond

Une levée de fond est nécessaire afin de pouvoir réaliser les objectifs techniques, ainsi que rendre le projet pérenne. Le CERDECAM a explicitement demandé de postuler pour le subside Spin-off d'Innoviris, qui fut le premier objectif de cette catégorie.

II. PARCOURS

II.A. PARCOURS GAGNANT

Le parcours gagnant est un outil mis à disposition par le StartLab ICHEC. Cet outil permet de lister les actions à entreprendre pour faire évoluer une idée de solution ou de service vers une entreprise.

Ces actions sont divisées en différentes étapes. Ces étapes ont été identifiées aux backlogs du Kanban. Ces backlogs sont ensuite divisés en sprints de deux semaines.

Au début de chaque sprint, les tâches sont séparées. Cette méthode a été suivie jusqu'au confinement. La liste des tâches est disponible en [annexe](#).

Ces actions furent notre quotidien entrepreneurial. Tous ces travaux ont été documentés et sont disponibles sur le [Teams](#).

La première section de tâches a permis de mettre à plat les motivations et l'implication des membres de l'équipe. Cela a soudé l'équipe et permis d'identifier les objectifs de chacun pour cette expérience.

Le deuxième sprint a permis de déterminer la désirabilité du projet. Cette section a permis de formuler le problème que la solution semble résoudre, la différenciation et une proposition de valeur de la solution. Durant cette section, une étude des concurrents a aussi été réalisée. C'est durant ce sprint que l'impact sociétal du projet a commencé à évoluer. C'est devenu une des motivations profondes pour mettre le service sur le marché.

La troisième section a permis de déterminer la faisabilité du projet. Des outils ont été complétés tels que le Lean canvas, le value Business Model Canvas (BMC) et le Rieskiest Assumption Test (RAT).

À l'issue du troisième sprint, l'étude de marché a débuté. Des appels vers des homes ont été effectués. Cela a permis d'identifier l'utilité de la solution. La proposition de valeur qui était offerte n'était pas suffisante mais assez importante pour susciter l'intérêt des interlocuteurs.

Une idée de pivot, ou d'augmentation de la proposition de valeur a été discutée. Sur base des premiers retours de l'étude de marché, l'idée d'une géolocalisation aussi à l'intérieur des bâtiments a émergé.

Le quatrième sprint a permis de déterminer la viabilité du projet. Dans cette section, une équation de création a été chiffrée. Après certaines hypothèses, cette équation de création a permis de confirmer la viabilité du modèle.

C'est environ après la quatrième section que le travail a commencé à rencontrer des difficultés dues au COVID-19.

En confinement, un pitch a été écrit et filmé. Il a été publié au sein de la communauté du StartLab ICHEC et a reçu une bonne appréciation.

Un travail sur le choix de la technologie à adopter pour l'idée de pivot a aussi été réalisé durant la période de confinement. Par la suite, une réflexion stratégique a été menée pour connaître la position de la future entreprise sur divers points.

Ceci représente nos avancées de février à mai 2020.

II.B. FORMATIONS

Commençant en février 2020, les formations du StartLab ICHEC ont permis à l'équipe d'apprendre les bases de l'entrepreneuriat. La formation est divisée en ateliers qui ont lieu entre février et mai 2020. Vous trouverez le parcours des formations en annexe.

Ci-dessous sont détaillées toutes les formations et leur contenu, en bref. Les notes complètes prises lors des formations sont disponibles sur le Teams de l'équipe.

« S'approprier le Lean canevas (positionnement sur son marché) » a eu lieu le 8 février 2020. Une première prise en main du Lean canevas a eu lieu. L'équipe a pu commencer à la remplir pour fixer la proposition de valeur de NeoFinder.

« Organiser la gestion du mon projet » (gérer un projet entrepreneurial) a eu lieu le 11 février 2020. Utiliser la méthode agile pour fonctionner en équipe comme un KANBAN et une évolution itérative est important. Répartir les tâches et structurer les réunions les rendent plus efficaces.

« Établir son équation de création » (finance) a eu lieu le 18 février 2020. L'équation de création est théorique et modulable, il ne faut pas hésiter à la modifier pour tester des hypothèses. Il y a trois manières de fixer un coût :

- par le coût,
- en fonction de la concurrence,
- par la valeur estimée par le client.

Il ne faut pas hésiter à mettre un prix de produit élevé pour tenir la route, quitte à le diminuer pour les premiers clients.

« S'approprier le Lean canevas (suite) » (positionnement sur son marché) a eu lieu le 3 mars 2020. Une introduction au RAT (« Rieskiest Assumption Test ») a été partagée et les règles du Tam Sam Som ont été expliquées et comprises.

« Augmenter l'impact sociétal positif de mon projet » (co-développement) a eu lieu le 7 mars 2020. La formation avait comme but d'expliquer et de mettre en place l'arbre à problèmes sociétaux. Celui-ci permet de voir l'impact sociétal du projet.

« Faire connaître mon projet (communication) » a eu lieu le 10 mars 2020. L'identité d'entreprise est importante pour en constituer la vision, les valeurs, les bénéfices émotionnel et fonctionnels ainsi que les attributs de marque.

Les formations suivantes ont dû être adaptées au confinement à la suite du Covid-19.

« Les bases du droit de propriété intellectuelle » (légal) a permis de comprendre quels sont les choix légaux à prendre lors de la création d'une entreprise.

« Challenger mes hypothèses » (tester son marché) a eu lieu en ligne les 21 et 28 mars 2020. Il est important de segmenter les clients interrogés. Le client est le centre du projet, s'il n'aime pas, il n'achètera pas.

« Pitcher mon projet avec confiance (Communication) » a eu lieu en ligne début avril 2020. Pour pitcher correctement son projet, il faut garder en tête les attentes du public. La formation insiste aussi sur la répétition pour que l'énoncé du pitch devienne automatique.

« Retour sur pitch (co-développement) » a été déplacé à septembre.

« Comprendre mes besoins financiers (Finance) » a eu lieu en ligne le 25 avril 2020. Le thème de la formation était la construction d'un budget et l'utilisation de quels outils pour y arriver. Il faut limiter les coûts fixes pour plus de souplesse dans l'utilisation du budget.

« Mettre en place ses premiers outils de gestion » (finance) a été déplacé à septembre.

« Choisir ma structure juridique » (légal) a eu lieu en ligne le 28 avril 2020. Cette formation était centrée sur les étapes concrètes de la création d'entreprise au niveau légal. Il est très important pour toutes les étapes légales de création d'entreprise d'établir un plan financier qui tient la route.

« Moi à ta place & évaluation » (co-développement) a été déplacé à septembre.

II.C.COACHING

Tous les mois depuis l'entrée au StartLab en février ont eu lieu des réunions de Coaching avec Xavier De Poorter. Il aide dans la création du projet et permet d'ajouter une vision externe et professionnelle. Xavier De Poorter est diplômé ingénieur commercial. Il est le fondateur d'ICHEC- PME, a fait toute sa carrière dans le monde de l'entrepreneuriat, et est digne de confiance pour aider à la réalisation du projet.

Une réunion de coaching mensuelle a eu lieu pour partager l'avancée de NeoFinder et avoir les retours de Mr. De Poorter. Les comptes rendus de réunions sont lisibles au complet sur [Teams](#).

Le 28/01 a eu lieu la première réunion. Après une prise de connaissance et une présentation rapide du projet, les bases du projet ont été mises en place ainsi que les objectifs prochains.

Le 21/02 a eu lieu la deuxième réunion de coaching. La séance a permis de partager les rôles et le fonctionnement tout en clarifiant les objectifs personnels et globaux. Une interrogation de fond de l'idée a permis de bien définir le projet. L'objectif actuel est défini : tester le marché.

La réunion du 27/03 a eu lieu à distance suite au confinement dû au Covid-19. Nous avons partagé les premiers résultats de notre étude de marché. Le sujet de l'évolution du projet a été abordé. Le nouveau but est de prendre le temps de continuer à développer la technologie pour pouvoir géolocaliser à l'intérieur en plus de l'extérieur. Le travail de l'équipe du prochain mois sera de bien définir ce pivot.

La réunion du 24/04 a aussi eu lieu à distance. La réunion a commencé par le partage du choix de la technologie pour la géolocalisation à l'intérieur ainsi que des choix de stratégie générale pour l'entreprise. La conclusion prise est, que pour avancer dans la création de l'entreprise, il

faut absolument essayer de trouver le premier client et partenaire pour développer la technologie.

La dernière réunion avant la fin des formations de cette année à lieu de 29 mai.

II.D. FINANCEMENTS

Ce paragraphe détaille le cheminement d'application des deux subsides que nous avons postulés : le subside SPIN-OFF d'Innoviris et la bourse Albert Vanhee de la Fondation pour les Générations Futures.

II.D.1. SPIN-OFF D'INNOVIRIS

Le programme SPIN-OFF soutient la création de nouvelles entreprises dans la Région de Bruxelles-Capitale afin de valoriser économiquement des résultats issus de la recherche scientifique. Il permet le financement de projets dont les objectifs sont :

- De finaliser la mise au point d'un produit, procédé ou service innovant basé sur des résultats acquis au cours de recherches préalables ;
- D'étudier les conditions d'exploitation industrielle et commerciale des résultats obtenus en vue de la création d'une nouvelle activité économique dans la Région de Bruxelles-Capitale.

Les projets financés doivent encore présenter des enjeux de développement expérimental lors de leur démarrage. Ils doivent avoir un impact favorable sur l'économie, l'emploi et l'environnement de la Région de Bruxelles-Capitale.¹

Dans le cadre de l'application de cette bourse, une recherche de parrains a été entamée. En plus du StartLab ICHEC qui aide le projet du point de vue marketing, Proximus a accepté de nous aider pour la technique, étant donné qu'ils sont gestionnaires du réseau LoRaWAN.

Un descriptif du projet a été écrit. Il reprend l'origine du projet et son évolution.

Un budget sur deux ans a aussi été réalisé. Celui-ci compte un ingénieur travaillant sur le développement du projet et le matériel dont celui-ci a besoin.

Ce travail de recherche a été planifié sur les deux ans et divisé en Work Package (WP), qu'ils soient pour la technique ou pour la création de l'entreprise.

Un travail sur le potentiel de réalisation et la stratégie de valorisation a été réalisé. Cela a permis de comprendre le marché dans lequel le projet évoluera et comment celui-ci s'intégrera dans ce marché.

L'objectif de l'application de ce subside est multiple. C'est le premier travail que l'équipe a réalisé ensemble. Cela a permis d'accorder les différents membres et de créer une cohésion de groupe. D'un autre côté, cela a permis de répondre à beaucoup de questions laissées en suspens, principalement celles concernant la demande du marché. Finalement, si l'idée était promue, le CERDECAM aurait pu engager un ingénieur pour travailler sur la conception du projet.

Malheureusement, la candidature n'a pas été retenue, étant donné que le projet n'était pas issu d'un projet de recherche au sens où l'entend Innoviris.

II.D.2. ALBERT VANHEE

La Fondation pour les Générations Futures a initié en 2019 un nouveau programme visant à soutenir les étudiants-entrepreneurs qui développent un produit, un service ou une technique porteuse d'impact sociétal positif.¹

L'innovation technique doit avoir un impact positif sur au moins deux dimensions du développement durable - People, Planet, Prosperity, Participation.

- **Planet** : l'impact sur la planète, sa biodiversité, son climat et ses ressources naturelles (ex. gestion dynamique des déchets, réduction de la pollution, usage rationnel de l'espace, des ressources, de l'énergie, etc.).
- **People** : l'impact sur le bien-être et l'équité de l'être humain, que ce soit ici ou ailleurs (ex. renforcement des liens sociaux, santé, solidarité, cohabitation, contribution aux compétences et aux connaissances, etc.).
- **Prosperity** : l'impact économique et la gestion économique de l'initiative (ex. favoriser l'autonomie et la transparence, bonne gestion financière des moyens disponibles, l'initiative contribue à la prospérité de la société, développe un modèle économique alternatif, etc.).
- **Participation** : la bonne gouvernance et l'implication des parties prenantes (ex. favoriser le fonctionnement démocratique, nouveaux partenariats, cogestion, résolution pacifique de conflit, etc.).

Afin de mener à bien l'application de cette bourse, la description du projet a été reprise de la bourse SPIN-OFF. Il y a été ajouté les enjeux sociétaux auxquels le produit répond.

La Fondation pour les Générations Futures demandait l'état d'avancement et les perspectives d'avenir du projet. Cela reprend en résumé le travail effectué du côté technique et une stratégie de mise sur le marché.

Ensuite, le plan du prototypage a été décrit. On y retrouve un budget pour la création de prototypes et les différents éléments pour que la création de ceux-ci soit menée à bien.

Pour finir, les impacts sociétaux positifs et négatifs ont été décrits, en expliquant quel impact aurait notre solution sur les différentes dimensions sociétales.

Cette bourse permettrait, d'une part, d'avancer sur la conception d'un prototype et, d'autre part, de lier des premiers contacts commerciaux pour la création de la solution finale.

L'idée a été primée par la bourse, et nous avons reçu 5000€ pour poursuivre le développement du prototype, ce que nous allons pouvoir faire au sein du CERDECAM.

III. DISCUSSION

III.A. RÉSULTAT

Ce document est écrit en mai 2020 et le projet a commencé en septembre 2019. Durant cette période les tâches suivantes ont été accomplies :

- Le dossier de candidature au StartLab ICHEC a été rempli et le projet a été accepté.
- Un dossier de candidature à Innoviris Brussels pour la création d'une Spin-off a été rendu.
- Le suivi des formations du programme start-test du StartLab a eu lieu. Celui-ci recouvre l'entrepreneuriat sous toutes ses facettes et a été partagé avec toute l'équipe, de la manière de préparer et gérer son budget à la manière de communiquer convenablement personnellement, avec le pitch, ou globalement, grâce à la communication d'entreprise.
- Le « Parcours Gagnant » proposé par le StartLab a permis d'accomplir des avancées concrètes pour l'entreprise. Ces éléments comptant parmi eux une équation de création et une analyse de marché.
- Un dossier de candidature à la bourse Albert Vanhée pour le financement de projets à impact positif a été rendu. Une bourse de 5000€ pour la création du prototype a été octroyée au projet à la suite de cette candidature.

Les bases saines pour la création de NeoFinder ont été posées. L'équipe est devenue soudée et professionnelle au fil des mois. En dépit du non-aboutissement de la demande de subsides à Innoviris, l'équipe a reçu des retours positifs et formateurs sur le projet. Ceci a permis la création d'un meilleur dossier pour la bourse Albert Vanhée, débloquant ainsi un budget de base pour le prototypage. NeoFinder a des bases solides pour la suite de la création d'entreprise.

III.B.PERSPECTIVES

NeoFinder et son équipe ont des perspectives ambitieuses. Néanmoins nous devons rester réalistes et franchir les obstacles un par un. Les perspectives sont à court, moyen et long terme. Celles-ci représentent les objectifs actuels et peuvent changer dans le futur.

Les objectifs à court terme sont surtout de terminer l'étude de marché une fois la crise sanitaire actuelle terminée. Ensuite finir le premier prototype et avec celui-ci trouver un client pilote pour aider à la finalisation du produit. L'équipe veut également passer à l'étape suivante dans le parcours StartLab et évoluer au mieux dans leur programme de formation.

À moyen terme, la technologie du bracelet pour qu'il fonctionne également à l'intérieur devrait être développée. L'équipe espère pouvoir s'installer sur le marché belge et commencer à rendre l'entreprise rentable.

À Long terme les objectifs sont de poursuivre la recherche et le développement pour toujours créer le meilleur produit possible. NeoFinder espère également s'étendre sur le marché international. Enfin, pourquoi pas ?, faire évoluer la clientèle vers les particuliers.

IV. CONCLUSIONS

Voici le bilan de la réalisation des objectifs

- L'objectif de création d'entreprise

Cet objectif n'a pas été totalement accompli. La création en tant que telle de l'entreprise n'a pas été réalisée mais les actions effectuées nous rapprochent de l'objectif. Grâce aux travaux réalisés sur les demandes de subsides et aux appels effectués dans les maisons de repos, une image et la structure du marché ont été décrites et notre compréhension de ce marché a évolué. Par ailleurs, avec l'aide des sessions de coaching, les idées ont été remises en question et cela a permis de préparer un pivot de notre proposition de valeur.

- Un objectif d'apprentissage

Cet objectif a été accompli d'une part grâce aux formations du StartLab, qui ont contribué à un apprentissage entrepreneurial. D'autre part, le travail de groupe sur les différents subsides nous ont permis d'améliorer notre travail en équipe.

- Un objectif de levée de fonds

Le projet a été primé de la bourse Albert Vanhee de la Fondation pour les Générations Futures, ce qui a permis de lever des fonds pour la création du prototype. Malgré le refus du subside SPIN-OFF d'Innoviris, il y a une motivation de réitérer la demande. Le sujet pourrait être la technologie de géolocalisation intérieure à développer.

Bien que le travail de marketing ait été compliqué, en raison de notre manque de compétence dans ce domaine, l'équipe a atteint la plupart des objectifs en ayant une cohésion et en développant de bonnes synergies.

En somme, ce fut une expérience très enrichissante et une extraordinaire première aventure entrepreneuriale.

V. BIBLIOGRAPHIE

- [1] Fédération pour les generations futurs. (2020, février). soutient l'entrepreneuriat étudiant à impact ! Fédération pour les generations futurs.
- [2] ICHEC-PME. (2017). Retrieved from Wikipreneur: www.wikipreneurs.com
- [3] Innoviris Brussels. (2020). *Règlement Spin-off 2020*. Retrieved from Innoviris: https://innoviris.brussels/sites/default/files/documents/guidelines_spin_off_2020_0.pdf