

Rapport de stage sur le développement de solutions applicatives pour smartphones



Stage effectué du 8 janvier au 23 février 2018
sous l'encadrement de Monsieur PRACA Damien

HighConnexion
365 Avenue Archimède
13290 Aix-en-Provence
04 42 24 58 24



HighCONNEXION

SOMMAIRE

Remerciements.....	3
1. Introduction.....	4
1.1 Histoire et présentation de la société HighCo.....	4
1.2 Histoire et présentation de la filiale HighConnexion.....	4
1.3 Organigramme de la société.....	6
1.4 Mission du stage.....	7
2. Développement.....	8
2.1 Ma place dans l'entreprise et analyse de l'état du projet.....	8
2.2 Objectifs et deadline de l'application Rouge.....	8
2.3 Résultat du travail réalisé.....	10
2.4 Extras : étude de faisabilité RTM.....	14
3. Conclusion.....	15
Annexes.....	16
Index.....	21
Bibliographie / Sitographie.....	22

Remerciements

Après mes 7 semaines vécues au sein de la filiale HighConnexion de la société HighCo, je tenais à remercier sincèrement l'ensemble de l'équipe éducative du lycée Marie Curie ainsi que les collaborateurs de l'agence ayant permis de rendre ce stage à nouveau possible et de l'avoir rendu si riche en apprentissage tout en m'offrant une vision toujours plus réelle du monde du travail et du quotidien de salariés exerçant le métier que je convoite en m'octroyant des responsabilités toujours plus importantes.

J'aimerais adresser une pensée toute particulière à Monsieur **VACHIAS Thibault**, manager de la branche développement chez HighConnexion, qui m'a permis de rendre ce stage de seconde année à nouveau possible dans les conditions les plus idéales que je pouvais espérer.

J'adresse également mes remerciements et mes pensées à Monsieur **PRACA Damien**, ingénieur et développeur sénior au sein de la filiale HighConnexion pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce à sa confiance, j'ai pu m'accomplir totalement dans la mission qui m'a été donnée. Il fut également d'une aide précieuse dans les moments les plus délicats et durant ma première semaine, pendant laquelle l'adaptation au développement mobile fut difficile.

Pour finir, je souhaite également accorder ma gratitude à Monsieur **GUENIOT Sylvain**, ingénieur analyste développeur au sein de HighConnexion pour le soutien et l'aide qu'il m'a également apporté. Son expérience, ses compétences et son côté très pédagogue dans le développement mobile fut extrêmement enrichissants.

1. Introduction

1.1 Histoire et présentation de la société HighCo

Côté à la bourse de Paris depuis 1996, HighCo est un groupe de solutions marketing fondé en 1990.

Spécialisé dans la grande distribution et la grande consommation, HighCo est présent sur 5 types d'activités : le couponing, l'échantillonnage, les points de ventes, le traitement des données et les conseils en communication.

Sa mission est de développer des offres toujours plus mesurables et plus performantes en terme de retour sur investissements. Cela passe notamment par l'utilisation des technologies web et mobiles.

Avec plus de 700 collaborateurs et 14 filiales réparties en Europe, HighCo, dont le siège social est situé à Aix-en-Provence, se positionne comme un acteur majeur et indépendant des solutions marketing dans le monde du numérique. En 2016, la société réalise un chiffre d'affaires de plus de 155,38 millions d'euros pour une marge brute de 81,06 millions d'euros.

Profil de la société HighCo :

Président : CHABASSIEU Didier

Création : 1990

Chiffre d'affaire (2016) : 155,53 millions d'euros

Résultat : 14,10 millions d'euros

Effectif : 700 collaborateurs

Implantations : France, Belgique, Benelux, Espagne, Italie, Émirats Arabe-Unis

1.2 Histoire et présentation de la filiale HighConnexion

High Connexion est une filiale appartenant à 51 % à la filiale HighCo Shopper, qui elle-même appartient au groupe HighCo. Elle développe et maintient des applications iOS, Android & Windows Phone de type natives ou customisées en fonction de l'ergonomie et du rendu souhaité et apporte son expertise technique, fonctionnelle et graphique pour la réalisation d'applications. L'organisme conçoit et met également à disposition des outils innovants pour informer, alerter, fédérer, divertir, promouvoir ou acheter.

Sa capacité de production est de 6 ingénieurs mobiles basés à Aix-en-Provence et à Lyon qui ont déjà réalisés plus de 100 applications distinctes dans des univers très variés et pour de grands comptes tels que Carrefour, l'Olympique de Marseille, La Poste, Oasis, Bouygues Telecom, Colis privé etc. et possède une forte expertise dans la gestion complète d'un projet et de ses spécifications jusqu'à sa publication sur les stores.

On peut relever des applications phares plus en détails comme :

- L'application **RTM** permettant, entre autres, d'obtenir les horaires des lignes et les places des parkings en temps réel.

- L'application **McDo France**, permettant de passer commande depuis son mobile, puis de retirer celle-ci dans le restaurant sélectionné au préalable.
- L'application **eJobber**, société d'intérim, permettant de se positionner facilement sur des missions disponibles en fonction d'une date souhaitée.

HighConnexion est une filiale du groupe HighCo qui a su, avec l'expertise de ses dirigeants, associer l'innovation technologique et les usages mobiles à la gestion de la relation client pour faire de cette branche un acteur reconnu, apprécié et leader sur le marché du marketing et du paiement mobile.

Profil de la filiale HighConnexion :

Président : LAURENT Bruno

Création : Février 2008

Chiffre d'affaire (2015) : 37,8 millions d'euros

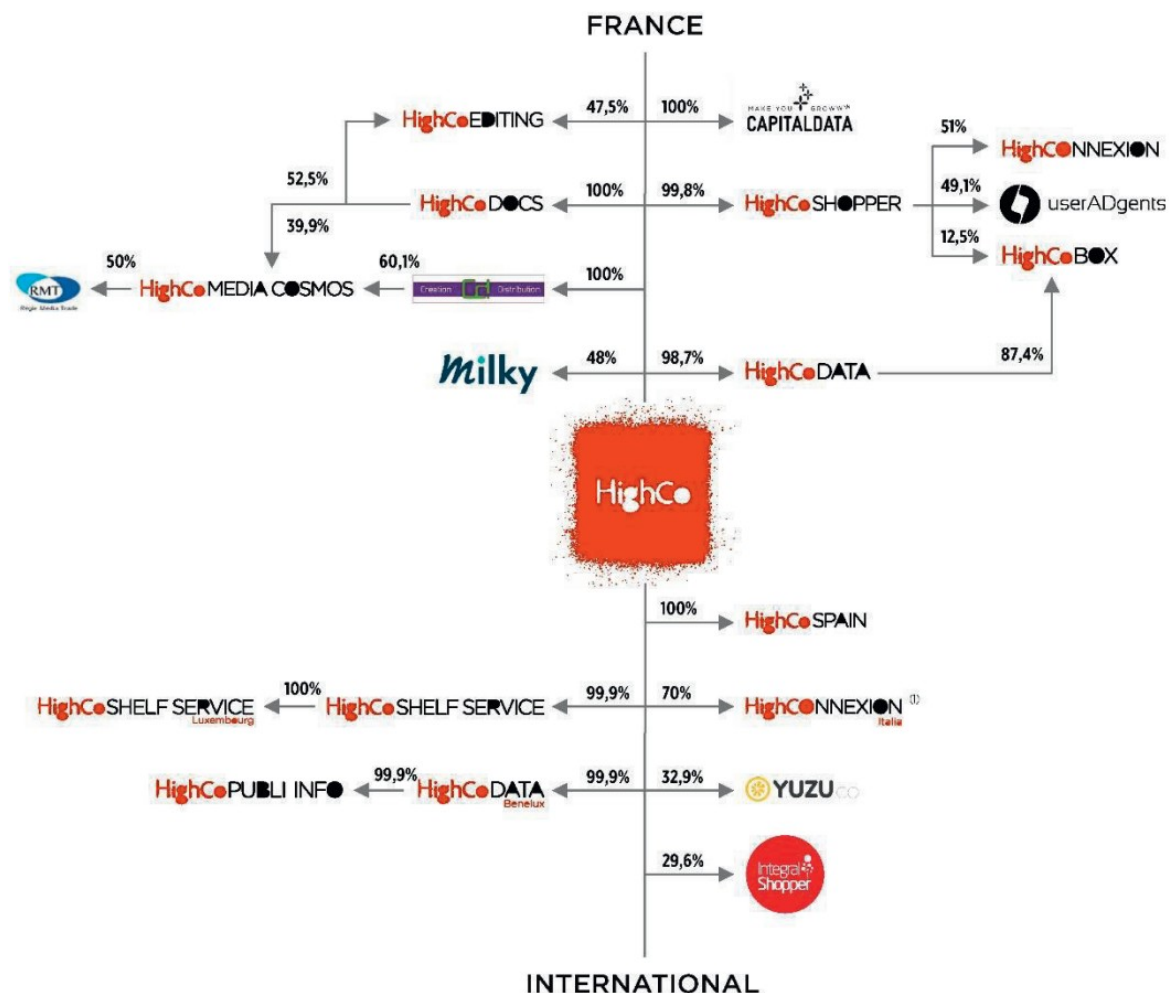
Résultat : 3,1 millions d'euros

Effectif : 30 collaborateurs

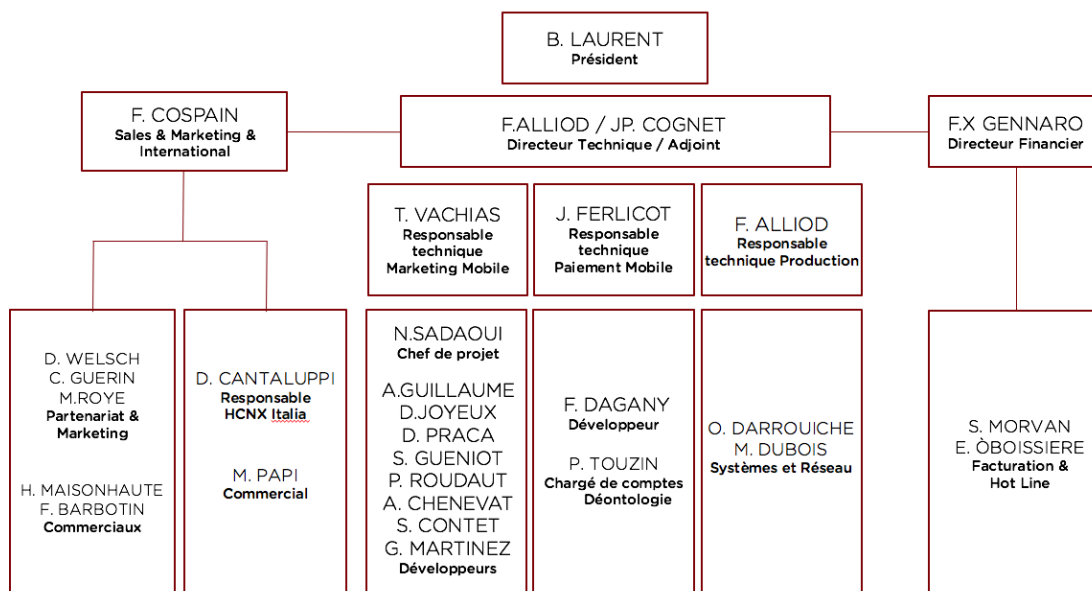
Implantations : Lyon, Paris, Aix-en-Provence, Côme (Italie)

1.3 Organigramme de la société

Organigramme de la société HighCo :



Organigramme de la filiale HighConnexion :



1.4 Mission du stage



Pour mon retour chez HighConnexion, il m'a été demandé de développer une nouvelle application mobile Android, pour un client cette fois-ci : le groupe Rouge, une radio/TV Suisse.

Ce projet consiste en une refonte totale 2.0 de l'application de Rouge, déjà disponible en version 1.6 sur le Play Store. Celle-ci doit apporter une ergonomie revisitée, ainsi qu'une interface plus respectueuse des nouvelles guidelines d'Android. Cette nouvelle version doit également implémenter le support de la TV live et des webtvs, ainsi que des webradios. Des fonctionnalités qui étaient absentes de la version précédente, trop ancienne et sur une base trop instable pour envisager une simple évolution plutôt qu'une réécriture complète de l'application.

2. Développement

2.1 Ma place dans l'entreprise et analyse de l'état du projet

J'ai travaillé durant ma période de stage en qualité de développeur Android, me plaçant donc sous l'encadrement de Monsieur **VACHIAS Thibault**, manager de la branche développement à Lyon et de Madame **SADAoui Noémie**, chef de projet de la section développement à Aix-en-Provence.

À mon arrivée, il est important de noter que l'application Rouge 2.0 était totalement inexistante, puisque mon stage consistait en un développement intégral de celle-ci. Contrairement à mon stage précédent où l'application était déjà entamée, avec une architecture globale réalisée, j'ai donc dû créer les dépôts Github et le projet Android « from scratch » en me basant sur les maquettes du graphiste pour concevoir le squelette de l'application. Une expérience plus difficile, puisque toutes les librairies de gestion des requêtes, des remontées d'erreurs dans nos interfaces etc. n'étaient pas implémentées, mais plus enrichissante au niveau personnel.

2.2 Objectifs et deadline de l'application Rouge

Après analyse des spécifications techniques réalisées par **SADAoui Noémie**, ainsi que de la maquette réalisée par le graphiste et approuvée par le client, voici les différentes tâches qui m'incombait de réaliser. Toutes les captures d'écran illustrant mes propos sont disponibles dans les annexes du rapport, page 21.

Dans la partie « Radio » :

Cette partie est axée autour de deux sous-menus : « Rouge Live » et « Podcasts ».

Dans la partie « Rouge Live » :

L'en-tête de cette partie contient le player radio, avec le nom de l'artiste et du titre, ainsi que l'illustration du programme en cours de diffusion. Sous celui-ci, on peut retrouver les titres précédemment joués à l'antenne, ainsi qu'un bandeau en bas de page permettant d'indiquer la fréquence de la radio dans votre zone géographique en fonction de votre position dans le cas où vous souhaiteriez l'écouter directement sur votre poste radio et non depuis l'application.

Dans la partie « Podcasts » :

Une liste affiche tous les programmes disponibles en rediffusion. Un clic sur l'un d'eux ouvre une nouvelle page listant toutes les émissions classées de la plus récente à la plus ancienne disponible pour une lecture en playlist.

Dans la partie « TV » :

Cette partie est axée autour de deux sous-menus : « Rouge TV Live » et « Replays ».

Dans la partie « Rouge TV Live » :

Celle-ci fonctionne de la même manière que le player radio de Rouge et reprend également son interface pour une cohérence graphique. La couverture affiche donc l'illustration de l'émission ou l'affiche du film en cours et la partie « Précédemment » informe l'utilisateur des derniers programmes. Un clic sur le bouton « Lecture » lancera un lecteur vidéo natif intégré et géré directement dans l'application.

Dans la partie « Replays » :

L'utilisateur pourra retrouver dans ce sous-menu une sélection de films et de programmes en rediffusion réalisée par Rouge et classée dans trois grandes catégories contenant chacune trois vidéos : « Les plus populaires », « Notre sélection » et « Pour les plus cinéphiles ».

Dans la partie « Web » :

Cette partie est axée autour de deux sous-menus : « Webradios » et « Webtvs ».

Dans la partie « Webradios » :

Une liste de toutes les webradios gérées par Rouge est disponible et un clic sur l'une d'elle ouvre la page radio précédemment décrite.

Dans la partie « Webtvs » :

Une liste de toutes les webtvs gérées par Rouge est disponible et un clic sur l'une d'elle lance directement un lecteur vidéo sans passer par la page TV, tout simplement car les webtvs ne possèdent pas d'historique et de visuels sur les programmes en cours.

Dans la partie « Jeux » :

Cette catégorie est censée répertorier tous les jeux-concours en cours réalisés par Rouge à l'aide de l'outil « MGS » (**M**edia **G**ame **S**olution) de HighConnexion. Ceux-ci peuvent être des jeux gratuits via internet à l'aide d'un formulaire à remplir, ou bien un jeu de type « vote » nécessitant l'envoi d'un SMS à un numéro surtaxé pour valider sa participation.

Dans la partie « Plus » :

Cette partie est composée de deux sections : « Fréquences » et « Nous contacter ».

Dans la partie « Fréquences » :

Dans cette partie, une carte Google Maps affichant les antennes avec leur fréquence associée, ainsi que leur zone d'émission est mise à disposition. À l'aide de celle-ci, un utilisateur pourra aisément trouver la fréquence couvrant une ville en particulier depuis la barre de recherche.

Dans la partie « Nous contacter » :

Une redirection automatique vers l'application de messagerie par défaut est réalisée, avec le champ du destinataire et le sujet déjà pré-rempli, ainsi que toutes les informations relatives au téléphone, à la version de l'application ou à la configuration des notifications pushes précisées dans le corps du message afin de pouvoir mieux cibler le problème de l'utilisateur si celui-ci vient de l'application.

Pour finir, la partie « Plus » comporte également deux switchs permettant d'activer ou de désactiver la localisation GPS afin de respecter le choix de l'utilisateur dans le cas où celui-ci ne souhaiterait pas/plus être traqué par l'application, ainsi qu'un autre switch permettant de couper les notifications pouvant être émises par Rouge.

La deadline pour le développement de l'intégralité des fonctions explicitées ci-dessus était fixée pour le 31 janvier. Mon mois de février, quant à lui, devait servir à traiter les retours de **SADAUI Noémie**, ainsi que ceux de la cliente pour fournir d'ici le 23 février une version de production correspondant parfaitement aux attentes du groupe et prête à être mise sur le Google Play Store.

2.3 Résultat du travail réalisé

Au terme de mes 7 nouvelles semaines passées et partagées avec les développeurs mobiles de la filiale HighConnexion, je suis à présent en mesure de revenir sur l'avancée et le travail réalisés sur l'application Rouge. Étant revenu plusieurs fois sur certaines parties afin d'y apporter des correctifs ou de modifier certaines fonctionnalités suite à des changements de dernière minute, il me sera difficile de quantifier réellement le temps passé sur chacune d'elle.

Partie « Radio » :

Pour la partie « Rouge Live », celle-ci m'a donné un peu de fil à retordre non pas sur son aspect fonctionnel, mais sur son interface. En effet, ce design étant commun à la Radio/TV Live ainsi qu'aux webradios, il a fallu lui apporter une attention particulière et l'agencement des différents fonctionnements n'est pas aussi simple et intuitif que l'on pourrait l'imaginer. Il a également fallu prendre en compte les cas où l'artiste ou le titre de la chanson est trop grand pour tenir sur une ligne, le cas où le serveur ne nous renverrait aucune image d'album pour les artistes de niche ou en cas de problème avec le chargement de celle-ci.

Concernant le bandeau de fréquence en bas de l'écran, afin d'éviter de consommer trop de batterie et étant donné que la localisation est une tâche gourmande en énergie, la page radio Live se contente d'afficher la fréquence disponible pour l'utilisateur en fonction de la dernière localisation du téléphone. La géolocalisation est donc uniquement demandée dans la page « Fréquences ».

Les podcasts, quant à eux, ont nécessité des modifications côté application pour le rendu des programmes : les coins arrondis, le bandeau rouge suivant l'arrondi des images, le nom de chaque programme.. tout ceci ne fait pas partie de l'image mais est bien géré dynamiquement par l'application. Le serveur nous renvoyait également les podcasts pour chaque programme sans tri réalisé et c'est au niveau applicatif que la chronologie se fait à la volée.

Partie « TV » :

Le webservice nous renvoyant les informations de la TV Live de Rouge étant pratiquement identique à celui de la radio, il a été assez facile à implémenter. Cependant, la lecture des vidéos devait se faire dans un lecteur natif à l'intérieur de l'application et Android ne propose pas de lecteur par défaut. Le développement complet d'un lecteur vidéo fiable et robuste dans un délai aussi court n'étant pas envisageable, j'ai donc commencé à chercher une librairie de lecteur vidéo libre pour Android sur Github. Après une matinée de recherche, j'ai finalement pu trouver un lecteur soutenu par la communauté, très personnalisable et compatible avec un grand nombre de format, dont le .m3u8 utilisé par Rouge, qui est courant dans les flux TV et permet d'envoyer plusieurs formats de qualité différente dans un flux afin que l'appareil, en fonction de sa résolution et de sa qualité de connexion, sélectionne automatiquement le plus adapté.

Les replays, au format .mp4 quant à eux, sont également lus par le lecteur natif et n'ont pas posé de problèmes particuliers, le mp4 étant un format très répandu. Cependant, mes problèmes ont été une nouvelle fois liés à l'interface de cette page. En effet, les angles gauches des miniatures pour les replays sont arrondis et suivent la courbe de l'encadré rouge. Il était donc nécessaire de recouper l'image à la volée et uniquement sur un côté, tout comme le filtre noir transparent se trouvant au-dessus de celle-ci.

Partie « Web » :

La partie « Webradios » et « Webtvs » se limitant à une simple liste d'éléments redirigeant vers la partie Radio ou TV, la mise en place globale de ce menu n'a pas été d'une très grande difficulté. Cependant, un des souhaits de la cliente était que la lecture se lance automatiquement à l'ouverture d'une webradio. Il fallait donc que cela soit harmonieux et ne relance pas la radio si vous retournez sur celle qui est déjà en cours de lecture. De plus, cette partie utilisant le même code que la partie « Rouge Live », il ne fallait pas non plus que le lecteur lance automatiquement la radio live quand vous y accédez depuis le menu du bas et pas depuis la partie webradio. Des contraintes qui ne sont pas forcément difficiles à intégrer de manière bancal, mais qui nécessitent un minimum de réflexion pour éviter des vérifications à foison pas très esthétiques, ni optimisées.

Partie « Jeux » :

Les jeux-concours sont le résultat de l'implémentation de la librairie « MGS », développée par HighConnexion et en charge de la gestion des jeux. Celle-ci a été très facile à mettre en place, car elle se destine avant toute chose à des clients souhaitant intégrer la solution de la société dans leurs produits. Il fallait donc que son intégration soit la plus facile possible pour que son adoption soit importante. Un développeur n'aura donc aucun mal à l'ajouter dans son application.

Partie « Plus » :

Attardons-nous dans un premier temps sur la section « Fréquences », qui a sans nul doute été une des deux parties les plus difficiles de mon stage (la suivante sera détaillée plus tard). En effet, l'implémentation des pins et des polygones est issue au départ d'un simple fichier texte au format KML, dérivé du XML qui est un langage d'exportation et importation de données à balises. Les pins étaient donc définis par une latitude et une longitude, et les zones d'émission par plusieurs coordonnées latitude/longitude, déterminant les contours d'un polygone. Il a donc été nécessaire dans un premier temps de convertir ce fichier KML afin de le rendre compréhensible par le système Android et par la carte. Il était également demandé à ce que la carte ouvre les informations de la pin, zoome sur un polygone, et que celui-ci change de couleur de manière dynamique si l'utilisateur se trouve dans cette zone d'émission. Le problème fut également de trouver un bon ratio précision/consommation de batterie, car il est nécessaire d'être fidèle à la position de l'utilisateur sans pour autant demander une géolocalisation toutes les secondes, sans quoi l'autonomie de l'appareil sera fortement réduite. Il a donc été décidé de mettre en place une actualisation s'opérant toutes les 60 secondes et uniquement si l'utilisateur s'est déplacé de plus d'un kilomètre. Une barre de recherche est également à la disposition de l'utilisateur, avec un système d'auto-suggestion se basant sur Google Places et limité uniquement à la Suisse pour éviter que les propositions soient impertinentes.

Concernant la section « Nous contacter », celle-ci ne nécessite pas de s'y attarder très longtemps : elle demande juste au système d'ouvrir l'application de messagerie par défaut, tout en lui fournissant les champs du destinataire, le sujet, et le message afin de les pré-remplir.

Partie « divers » : service audio, notifications, CheckVersion, interstitiel, WhatsApp...

L'application contient également d'autres fonctionnalités qui ne se rattachent pas à une section particulière. J'ai donc décidé de les traiter dans cette partie supplémentaire.

Commençons par le service audio, qui est une des sections nécessaires, voire fondamentales pour une application radio, mais complètement transparentes aux yeux de l'utilisateur. Celui-ci est par ailleurs la deuxième partie qui m'a posé le plus de problèmes avec la partie « Fréquences », puisqu'il est censé gérer la lecture audio, la pause, le player audio en notification, le fonctionnement en arrière-plan et bien-sûr, il doit être capable de notifier à l'application entière quelle source audio est en train d'être jouée, car

dans le cas inverse, si vous jouez la radio live, puis que vous quittez la section avant d'y revenir, le lecteur audio n'aura aucun moyen lors de son chargement de savoir que vous êtes en cours de lecture et qu'il doit donc afficher le bouton « pause » et non pas le bouton « lecture » si le service audio ne lui notifie pas que c'est bien sa source audio qui est jouée.

De plus, le service audio ne doit pas fonctionner de la même manière lorsqu'il joue une radio ou un podcast, puisqu'un podcast doit pouvoir être mis en pause pour une reprise plus tard si l'utilisateur le souhaite, tandis qu'à la pause d'une radio, celle-ci doit se stopper et non pas se mettre en pause pour économiser la batterie et le forfait internet de l'utilisateur, puis reprendre sur le direct lors de la reprise. Une complication supplémentaire pour un service déjà complexe.

Il était aussi nécessaire de mettre en place une manière intuitive d'arrêter le lecteur audio depuis la notification : celle-ci peut donc être « swipée » uniquement si la piste de lecture est en pause pour éviter toute mauvaise manipulation. Cette action doit aussi cacher le bouton « mute » de la barre d'action, puisque l'utilisateur n'utilise plus aucune source audio. Des contraintes multiples et difficiles qui m'ont demandé finalement plusieurs jours de travail.

Concernant les notifications, exceptée celle liée au service lors de la lecture d'une source audio, il est important de savoir que la radio a la possibilité de lancer des campagnes de notification push auprès de ses utilisateurs afin d'informer de l'arrivée d'un nouveau podcast, de la fin imminente d'un jeu-concours ou pour toute autre raison. Ces campagnes se planifient depuis une interface web réalisée par HighConnexion et sont gérées par l'application à l'aide d'une autre librairie nommée « Hello » et qui, comme MGS, fait partie d'une plus grande librairie appelée « HCNX » (pour **HighCoNneXion**). Tout comme pour les jeux-concours, cette librairie est très simple à intégrer pour favoriser l'adoption de celle-ci par les clients et ne m'a donc pas demandé un temps très important pour son intégration.

Il en est de même pour la troisième et dernière librairie de HighConnexion intégrée qui se nomme « CheckVersion » et qui permet d'afficher une pop-up à l'ouverture de l'application pour informer l'utilisateur lorsqu'une nouvelle version est disponible sur le store. Il est même possible, dans des cas spécifiques et souhaités, de bloquer totalement une ancienne version en obligeant la mise à jour par l'utilisateur pour pouvoir continuer à utiliser l'application.

Un interstitiel a également été mis en place au lancement de l'application. Celui-ci s'affiche à un intervalle de X minutes renvoyé par le serveur. Il comporte une image, une couleur de fond et un lien de redirection si l'utilisateur clique sur celui-ci. Si le lien est de type « http:// », on ouvre le navigateur. S'il est de type « rouge:// », on redirige vers la partie de l'application souhaitée. Par exemple : « rouge://rouge_tv_live » redirige vers la section TV.

Pour finir sur une note plus légère, le logo en haut à droite visible sur toutes les sections de l'application hormis la carte est le logo « WhatsApp », qui permet d'ouvrir l'application directement sur une conversation avec la radio Rouge pour les utilisateurs possédant la messagerie et sur le site internet de WhatsApp invitant de télécharger l'application pour communiquer avec le numéro de Rouge si celle-ci n'est pas trouvée sur le smartphone de l'utilisateur.

2.4 Extras : étude de faisabilité RTM

Lors de mes deux dernières semaines de stage, l'application Rouge était en grande partie terminée et il ne restait plus que quelques retours de la radio à traiter avant une mise sur le store. Ces retours n'étant pas forcément journaliers ou très long à traiter, il me restait assez de temps pour traiter d'une autre problématique : dans le but de proposer une éventuelle fonctionnalité de réalité augmentée à la RTM dans le cadre d'une évolution de l'application Android actuellement disponible, il m'a été demandé de réaliser une étude de faisabilité du concept pour savoir si la visualisation des arrêts de bus/tramway et des stations de métro en réalité augmentée était un concept viable sans grandes contraintes fonctionnelles et de compatibilité.

Il m'a également été demandé de chiffrer, avec l'aide des développeurs Android, le temps nécessaire à la réalisation d'une telle fonctionnalité.

Mon étude a consisté dans un premier temps en l'analyse des différentes solutions de réalité augmentée disponibles sur le marché pour Android, en détaillant leurs avantages (facilité d'intégration, licence gratuite et libre, support des anciennes versions d'Android...) ainsi que leurs inconvénients (abandon du support officiel par la société, licence au prix prohibitif, solution pertinente mais comportant des risques de rachat...).

Dans un second temps, j'ai également dû réaliser des POC (**P**roof **O**f **C**oncept), c'est-à-dire des applications de démo en situation réelle. La première fut basée sur un projet Github et m'a demandé une journée de travail pour que celle-ci fonctionne avec la base de données RTM. Le résultat n'était pas des plus probants, car le point déterminant la position d'un arrêt bougeait quelque peu en fonction des mouvements de la caméra et était donc d'une précision discutable.

J'ai donc décidé de réaliser un second POC basé sur un framework léger et que j'ai rapidement pu m'approprier. Il m'a cependant demandé lui aussi une journée de travail, car le code avait tout de même 4 ans et nécessitait une remise à niveau pour qu'il soit fonctionnel sur les derniers smartphones. Le résultat final était probant, mais ma déception fut immense au moment de l'enregistrement sur le site internet du framework afin de récupérer une API key : c'est seulement à ce moment-là que j'ai pu apprendre que le support du framework avait été abandonné en avril 2017, sans libération du code source. Aucune autre partie du site n'informait le visiteur de cet abandon.

Un troisième POC a donc été réalisé par mes soins, basé sur un autre projet Github découvert après plusieurs heures de recherche : NavAR. Celui-ci a pour avantage d'être open-source, sous licence libre Apache et éprouvé par une large communauté de développeurs ayant formulé de très bons retours à son égard. De plus, il permet même de suivre son itinéraire vers le lieu souhaité en réalité augmentée. Un effet incurvé est également présent sur les points d'intérêt, leur donnant une allure plus soignée et réaliste. Des fonctionnalités complexes qui sont donc déjà prises en charge par le projet et qui ne seront qu'à reprendre pour les adapter en fonction des besoins plutôt que de partir d'une feuille blanche, qui aurait coûté beaucoup plus de jours de développement.

3. Conclusion

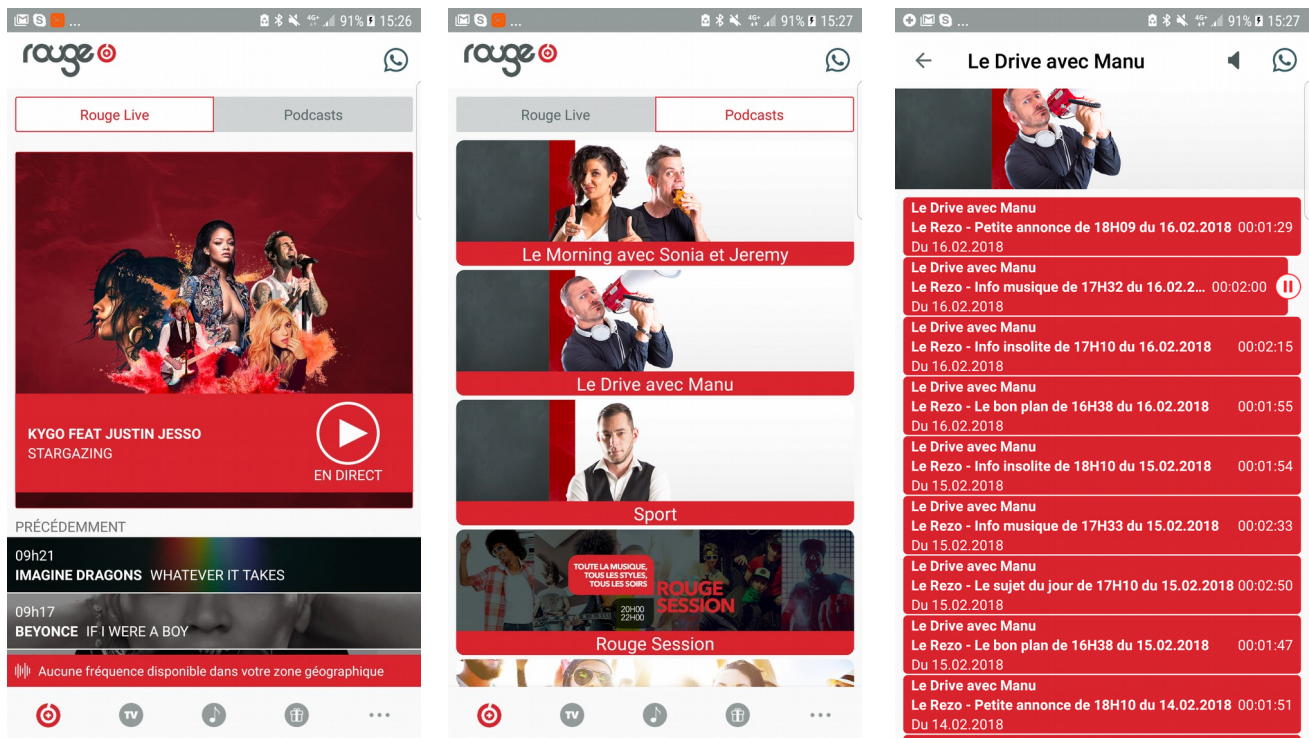
Pour conclure, j'aimerais m'attarder sur ce que mon second stage chez HighConnexion, que je considère comme une réussite, m'a apporté tout au long de ces 7 semaines.

Dans un premier temps, j'ai pu perfectionner mes connaissances en Java, le langage le plus utilisé pour le développement Android que nous n'avons pas vu durant les deux années mais qui se rapproche énormément du langage C#, m'ayant permis une adaptation rapide. J'ai pu également renforcer mon savoir sur le développement d'applications mobiles pour l'écosystème Android, qui devient de plus en plus facile à comprendre pour moi techniquement parlant au fil du temps, avec une évolution constante et des mécaniques qui lui sont propres. J'ai également mieux compris le champ lexical du monde du développement : commiter, pusher, merger, switcher sur le mock server, travailler en recette, les différentes branches dev, master, debug, preprod, release d'un git... des notions qui ne m'étaient pas complètement étrangères, mais qui restaient parfois encore assez floues

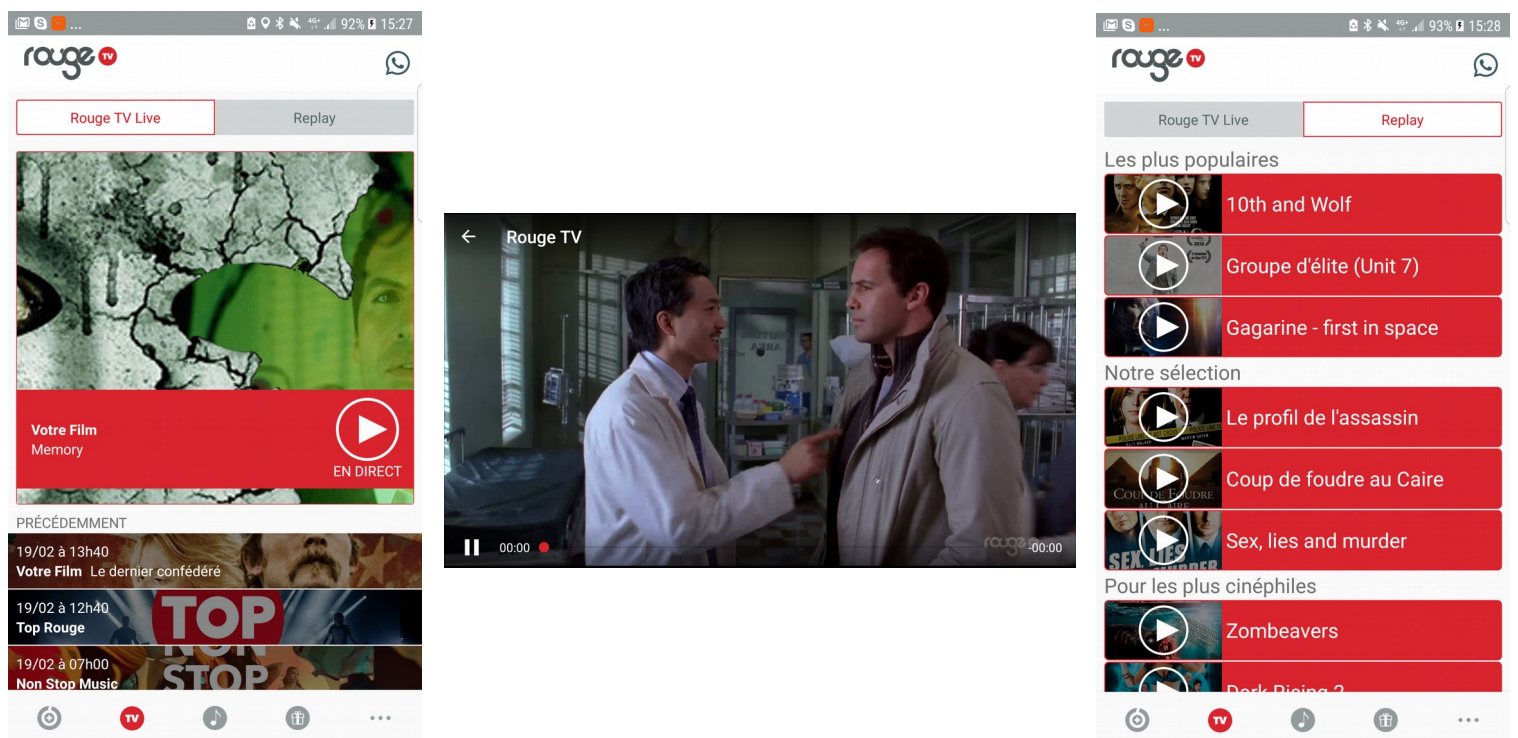
J'ai également continué mon perfectionnement sur des outils tels que Bitbucket, SourceTree, APLary, Postman, Monday ou encore MarvelApp, permettant un travail collaboratif et morcelé. Les enseignements que j'en ai tiré sont ma rigueur et mes automatismes concernant la gestion d'erreurs, qui se sont très fortement améliorés, car rien ne doit être laissé au hasard sur un projet d'une telle ampleur. Il est donc nécessaire d'essayer de se représenter toutes les possibilités d'utilisations et d'actions que l'utilisateur peut avoir afin de réduire au maximum le risque de crashes et d'exceptions. J'ai également eu l'impression que la très grande responsabilité que l'on m'a accordé sur ce projet, sur lequel j'étais seul à travailler, m'a aidé sur ma réflexion et ma vigilance, car il est nécessaire d'anticiper les problématiques sur lesquelles nous risquons d'être confrontées avant de s'engouffrer dans la réalisation d'un algorithme qui peut rapidement s'avérer être une impasse, car la réflexion précédant son écriture n'avait pas été importante.

Pour finir, ce stage m'a assuré que mon orientation future se tournera vers du développement Android, car la mobilité et la modularité du système apportent des problématiques inédites et un terrain sur lequel de nombreuses innovations et bonnes idées restent encore à découvrir.

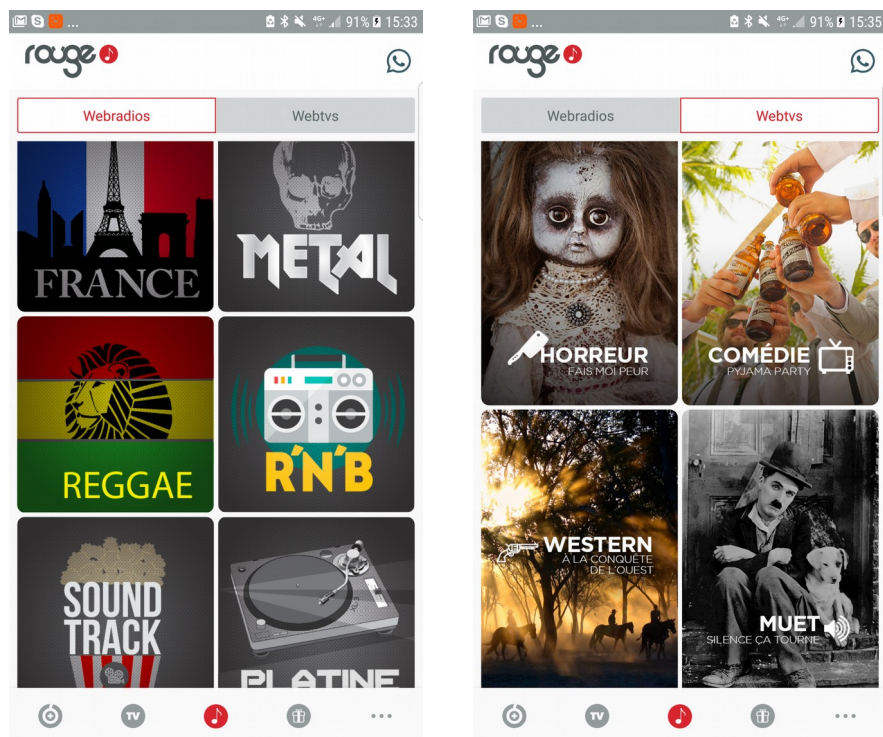
Annexes



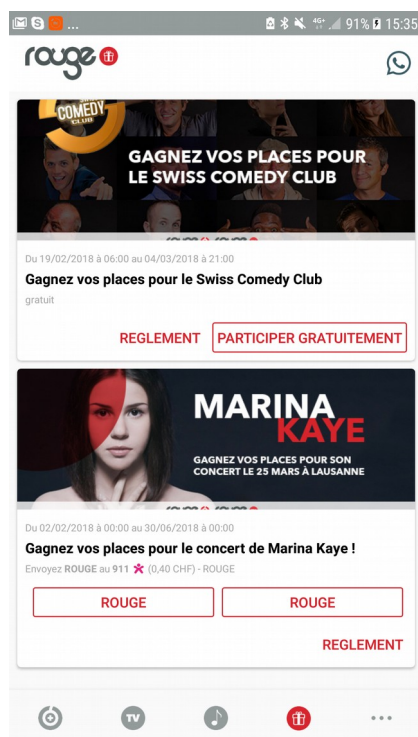
Captures d'écran de la section « Radio »



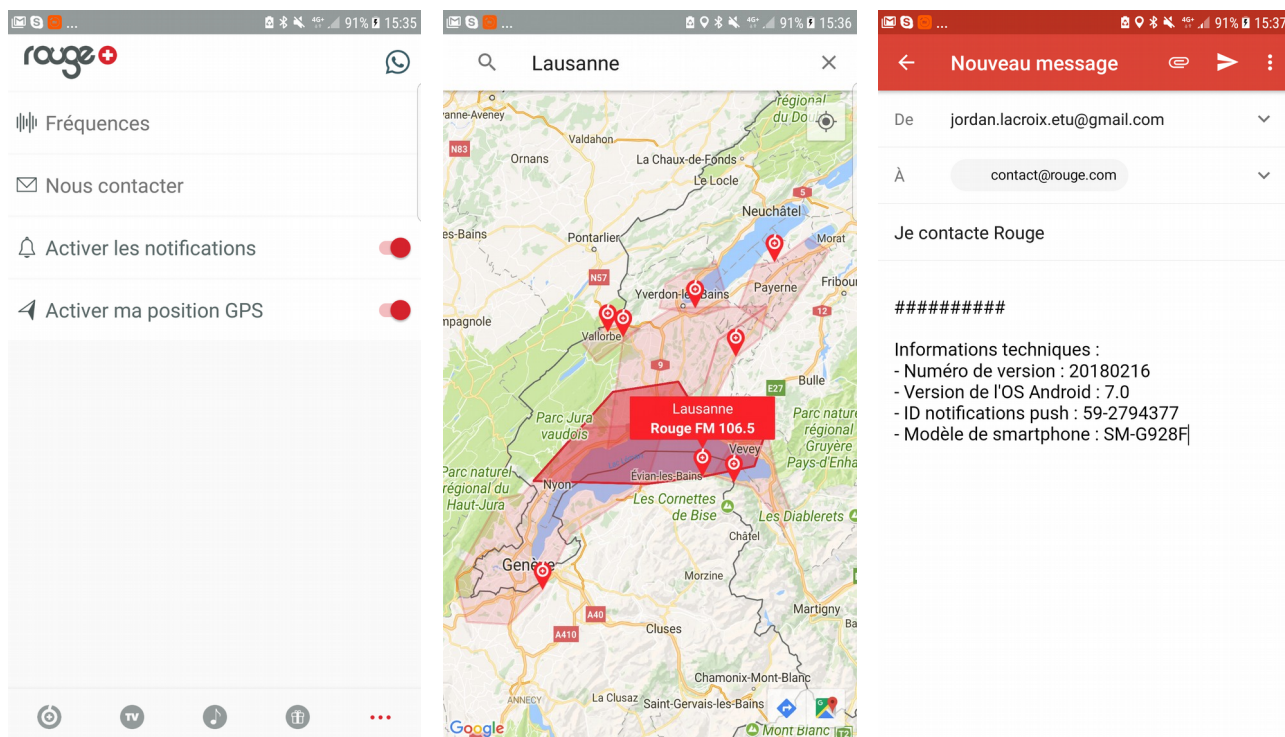
Captures d'écran de la partie «TV»



Captures d'écran de la partie «Web»



Capture d'écran de la partie «Jeux»



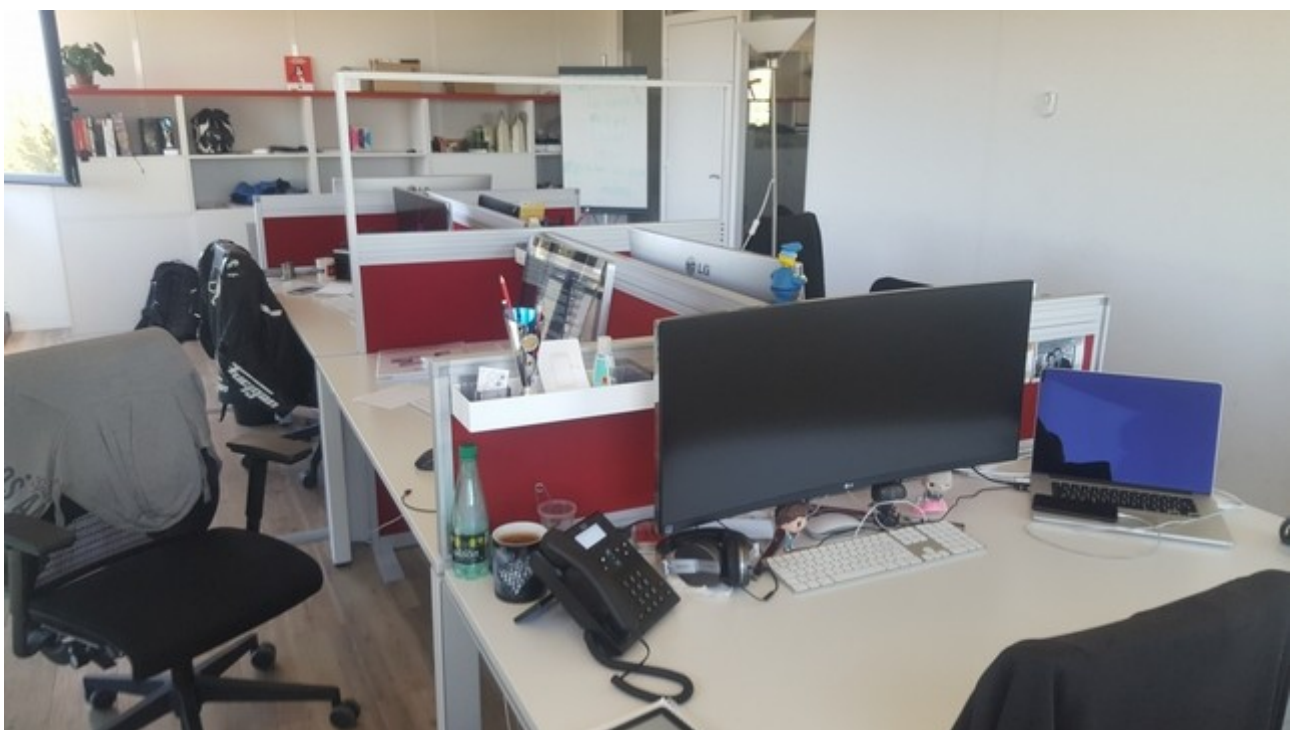
Captures d'écran de la partie « Plus »



Locaux de HighCo – Extérieur



Locaux de HighCo – Accueil



*Locaux de HighCo – Open space de HighConnexion
(mon bureau fut celui en premier-plan à gauche)*



Locaux de HighCo - Réfectoire



Locaux de HighCo – Salle de repos

Index

Client/serveur : L'environnement client/serveur désigne un mode de communication à travers un réseau entre plusieurs programmes : l'un, qualifié de client, envoie des requêtes. L'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'appareil sur lequel est exécuté le logiciel client, et le serveur, l'appareil sur lequel est exécuté le logiciel serveur.

Deadline : Date d'échéance à laquelle il faut rendre un projet.

Natif / Native : Un programme informatique en code natif (ou langage machine) est composé d'instructions directement reconnues par un processeur.

Notifications Push : Une notification push est un message, lié à une application installée sur l'appareil qui n'est pas obligatoirement ouverte, envoyé à l'utilisateur d'un smartphone ou d'une tablette. À la différence d'un SMS qui a un coût de par sa liaison à l'infrastructure d'un opérateur, la notification push est gratuite.

POC : POC, ou « Proof Of Concept », est une réalisation expérimentale concrète et préliminaire, courte ou incomplète, illustrant une certaine méthode ou idée afin d'en démontrer la faisabilité.

Web service : Le terme "web service" fait référence à des fonctionnalités de programmes informatiques permettant à des applications et des systèmes de fonctionner et d'échanger des informations à distance.

Bibliographie / Sitographie

Bibliographie :

- « Influencer le comportement du shopper », livre promotionnel réalisé et distribué par HighCo à tous les collaborateurs (*Aucune restriction de confidentialité sur celui-ci qui m'empêcherait de dévoiler certaines informations par inadvertance*)
- Rapport annuel de 2016 de la société HighCo : <https://www.highco.com/rapport-annuel/2016/index.html>

Sitographie :

- Site internet de HighCo : <https://www.highco.com/>
- Site internet de HighConnexion : <http://www.highconnexion.com/>
- Page Wikipédia de HighCo : <https://fr.wikipedia.org/wiki/HighCo>