

Rapport de stage

Alexis Lapeze

BTS SIO 1^{er} année

Institut Limayrac



Sommaire :

Table des matières

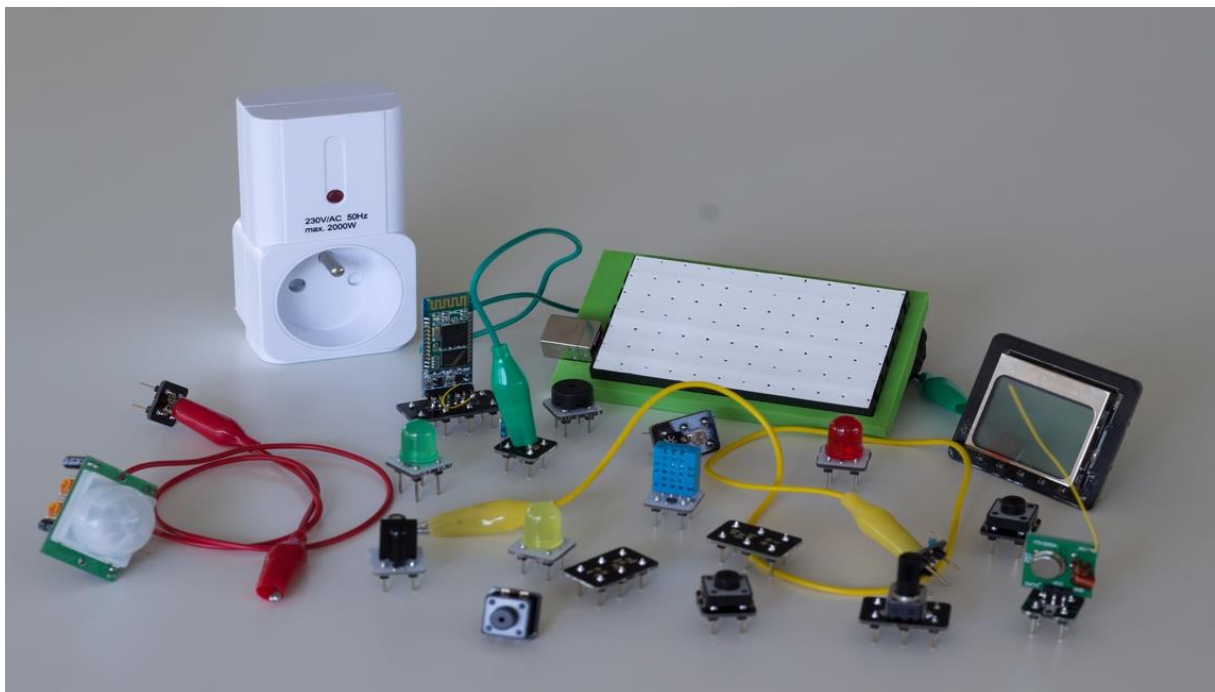
I.	L'entreprise :	3
	Présentation de l'entreprise.....	3
	Localisation de l'entreprise.....	4
2.	Contexte technique :	5
	Introduction.....	5
	Technologies étudiées.....	5
	Ionic, explication du framework.....	5
	Le MVVM (Modèle Vue Vue Modèle) avec Angular.....	6
	Le versioning avec Git et Bitbucket.....	7
3.	Missions réaliser :	9
	Benchmark.....	9
	Bluetooth.....	10
	Base de donnée.....	11
	Exemple de cas d'utilisation.....	12
	Remerciements.....	13
4.	Conclusion :	14

I. L'entreprise :

Présentation de l'entreprise :

Thingz - Mutiny est une start-up (SAS) créée en 2014 par Cyril Loucif-Durouge, qui vend des kits électroniques permettant aux enfants d'apprendre à coder. Il existe 3 kits différents en vente aujourd'hui avec plus ou moins de « briques » (modules électroniques). Le principe du produit est de brancher les briques sur une base et de venir les programmer directement sur le site internet de l'entreprise grâce à une interface de programmation ressemblant à « Scratch ».

Le produit est utilisable à partir de 8 ans, mais peut très bien être utilisé par des adolescents ou des adultes voulant découvrir la logique de la programmation et avoir l'autosatisfaction d'avoir créé un objet qu'ils auraient pu acheter dans le commerce. Il est ainsi possible avec les 14 briques que propose l'entreprise de réaliser des objets plus ou moins complexes allant d'une simple LED qui s'allume à la gestion de prises radio commander pour commencer à faire un de la domotique à la maison.



Localisation de l'entreprise :

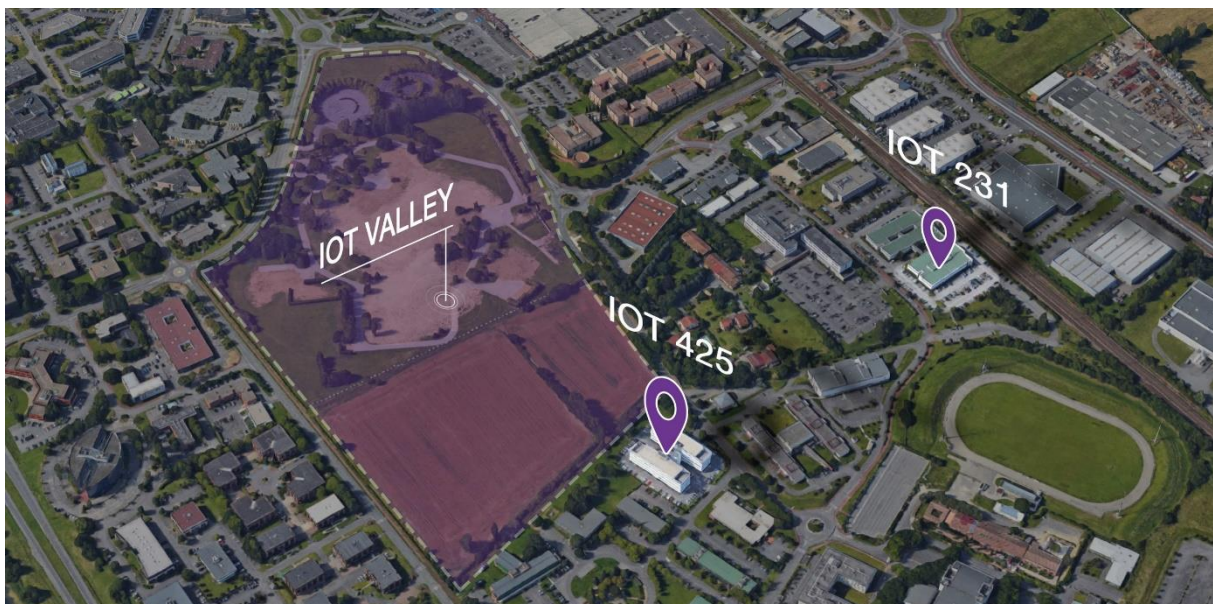
L'entreprise est située dans l'un des deux bâtiments de l'IOT Valley à Labège (Bâtiment 2 : 231 Rue Pierre et Marie Curie). L'IOT Valley est une association qui s'est donnée pour mission de développer l'excellence et la productivité dans l'Internet Des Objets. Créé en 2009 par 4 entrepreneurs, elle compte aujourd'hui plus de 4 000 membres et s'étend sur 10 000 m².

L'IOT Valley contient un accélérateur de start-ups qui permet aux nouveaux projets de gagner en maturité rapidement en les dédouanant au maximum de toute contrainte administrative et financière. Ils ont à leurs dispositions des ateliers complets (poste à souder, imprimantes 3D, etc...), comptables et des mentors qui leur donnent des conseils sur comment monter leurs entreprises et faire grandir leurs projets dans un laps de temps de 9 mois.

Il contient aussi un espace de coworking appelé « la passerelle » qui est à disposition des entreprises sortant du cadre de l'incubation, mais n'étant pas entièrement développé. Ces espaces ne sont plus gratuits comme pour l'incubateur, mais restent à des coûts bas pour continuer à les soulager financièrement.

Pour finir, les locaux contiennent aussi des bureaux pour les entreprises plus développées et ayant atteint une plus grande maturité ainsi qu'une plus grande autonomie.

L'IOT Valley étant en expansion constante, elle a pour ambition de créer un campus IOT à Labège. Les travaux débuteront bientôt et le campus ouvrira en 2019.



2. Contexte technique :

Introduction :

Mon rôle au sein de l'entreprise était de débiter le développement d'une application mobile pour l'entreprise qui lui permettrait de pouvoir communiquer avec son produit via Bluetooth. Le principe étant de se connecter au module Bluetooth de la carte et de pouvoir ensuite lui envoyer des messages et en recevoir par la suite : comme une application de messagerie, mais avec du Bluetooth pour la partie communication.

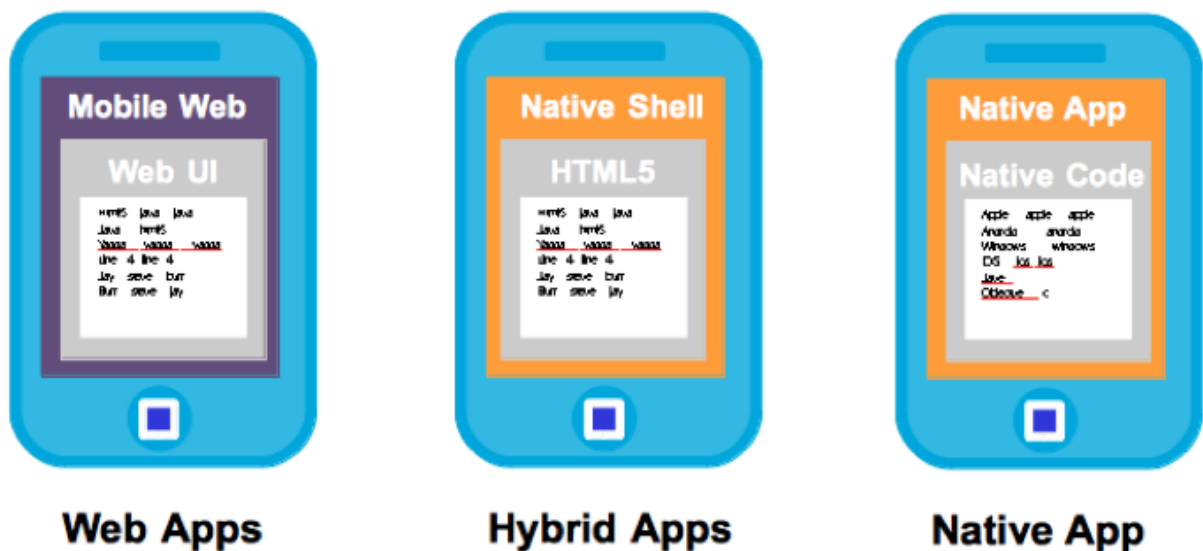
Technologies étudiées :

Ma période de formation à commencer avec une étude de cas pour déterminer quelle technologie sera utilisée pour créer cette application mobile. J'ai donc effectué un benchmark des différentes technologies existante et les ai confrontés entre elles.

Ionic, explication du framework :

La solution retenue lors de ce benchmark est Ionic. C'est un framework open-source créé en 2013 par Max Lynch, Ben Sperry et Adam Bradley. La version actuelle d'Ionic est la 3, elle se base sur le langage Angular et TypeScript qui sont des technologies JavaScript.

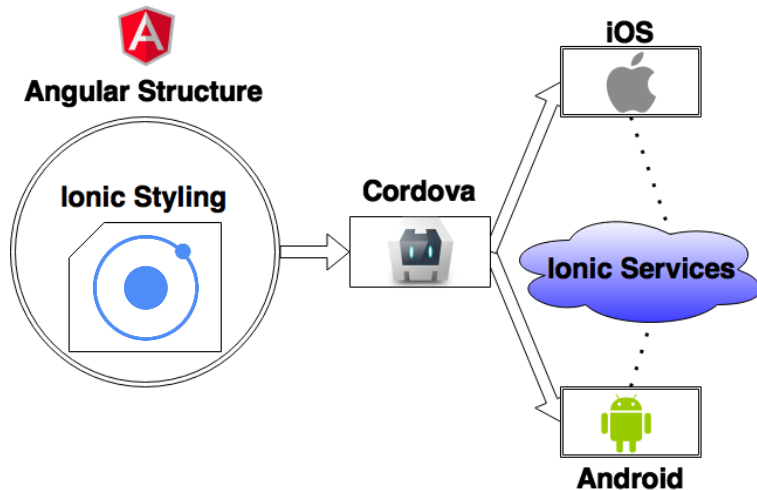
Il existe trois types d'application mobile :



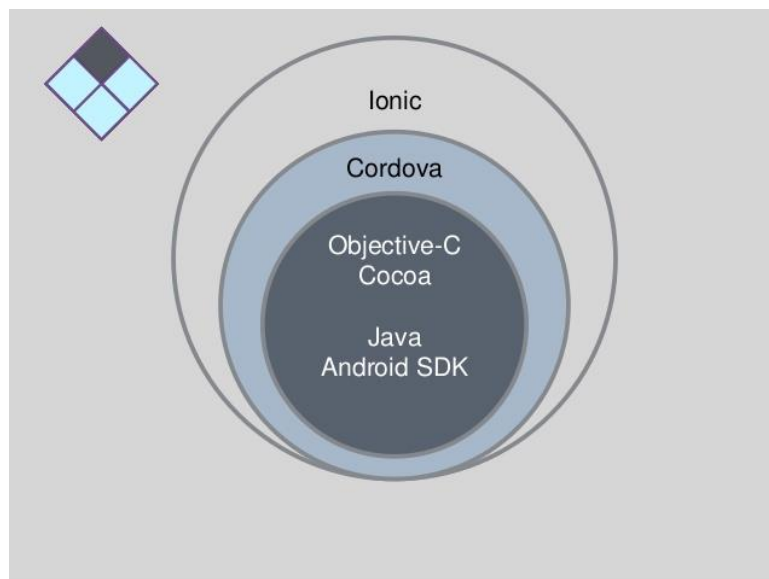
- Une « web apps » est une application mobile développée en HTML CSS, elle n'est exécutable que par le biais du navigateur et n'est pas téléchargeable sur un store d'applications. Elle n'a pas de différences avec un site internet classique.
- Une « hybrid apps » est une application mobile développée en HTML CSS, mais émulée par le smartphone lui-même. Elle se base sur le navigateur du téléphone pour l'affichage, mais permet d'accéder à des composants natifs du smartphone (camera, micro, Bluetooth, etc.)
- Une « native apps » est une application mobile développée en langage de programmation compilé. Elle permet d'accéder nativement à tous les composants du smartphone de façon plus optimisée, qu'une application hybride, mais une application est développée pour un seul système d'exploitation.

Le framework Ionic permet donc grâce à Angular et TypeScript de créer des applications dites hybrides. L'avantage d'une telle solution est avant tout le temps de développement qui est considérablement réduit grâce à des fonctionnalités prédéfinies et des « composants » fournis par le framework a intégré directement dans le projet. Ionic permet aussi grâce à Cordova qui est lui-même un framework développer par la fondation Apache, qui permet de développer des applications pour différentes plateformes : Android, iOS, Firefox OS, Windows 8, Ubuntu OS, etc...

Ionic fonctionne donc ainsi :



Le développeur développe donc son application avec le langage Angular, et Ionic s'occupe de la mise en pages des éléments et les transmet à Cordova qui les compile pour la bonne plateforme.

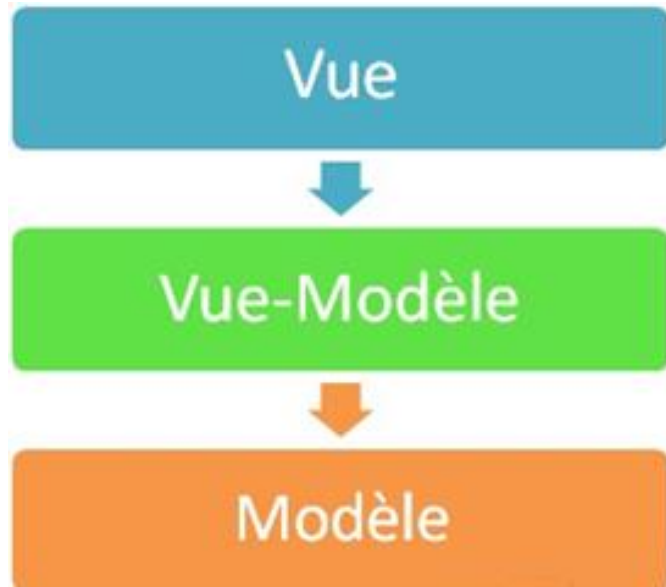


Ionic est donc un framework proposant des composants de style prédéfinis aux développeurs ainsi que des composants dit « natifs » par le biais de Cordova qui s'occupera de la compilation de ses composants sur les bonnes plateformes.

Le MVVM (Modèle Vue Vue Modèle) avec Angular :

Le MVVM est un pattern (patron de conception) qui a été spécialement conçu pour améliorer la séparation entre les données et la vue qui les affiche. Le lien entre la vue et le modèle de données est fait par des mécanismes de binding (un binding est un mécanisme qui permet de faire la liaison entre des données de manière dynamique). En finitude, ce qui est modifié dans le modèle se répercute dans la vue de façon automatique et dynamique.


Dans le principe du MVVM, le Modèle contient les données, la Vue est ce qui est affiché à l'utilisateur, et la Vue-Modèle s'occupe de faire le lien entre le Modèle et la Vue. C'est ici qu'intervient le binding. Dans ce cas, la Vue ne doit jamais traiter les données, mais uniquement se charger de les afficher à l'utilisateur. Le Vue-Modèle aura en charge les conversions et les accès au modèle.




En conclusion, le MVVM, se repose sur le DataBinding et l'évènementiel. La vue est couplée aux données (stocker dans le Modèle) via les mécanismes de binding et invoque les méthodes du Vue-Modèle. Le Vue-Modèle lui invoque les méthodes du modèle qui contient les données.

Angular étant malgré tout une technologie JavaScript, il n'utilise pas le DOM (Document Object Model) comme les technologies javascript standards, mais le MVVM qui est une méthode beaucoup plus structurée.

Le Versioning avec Git et Bitbucket :

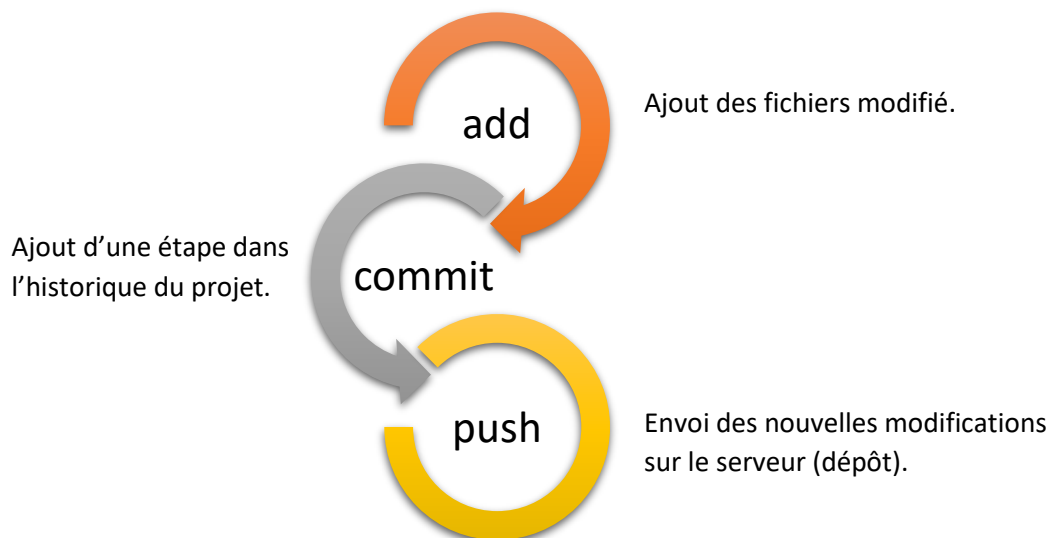
 Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de version Git. Il est gratuit pour les dépôts publics et payants pour les dépôts privés.

 Git est un logiciel de gestion de version décentralisée. Créé en 2005 par Linux Torvald (créateur du noyau Linux), il est open-source et permet de pouvoir collaborer à plusieurs sur un seul et même projet en évitant les conflits de modification d'un même fichier dans un projet par deux développeurs.

L'entreprise a ouvert un dépôt privé sur le service web Bitbucket pour pouvoir y stocker et y sauvegarder différentes versions de leurs logiciels en développement. C'est le cas de l'application Mobile que j'ai été amené à réaliser.

Travaillant avec un cycle de développement itératif, chaque « push » effectué sur le dépôt correspondait à une fonctionnalité de l'application, ce qui impliquait que toutes les versions de l'application hébergée sur le dépôt étaient fonctionnelles et utilisables.

Voici comment fonctionne un ajout de version avec git :



3. Missions réaliser :

Benchmark :

Comme dit précédemment dans ce rapport, lors de cette période de formation j'ai été amené à réaliser une application mobile pour les besoins de l'entreprise. J'ai donc commencé par lister et confronter les différentes technologies existantes pour développer une application sous la forme d'un benchmark :

	Native Script	React native	Ionic + cordova	Android Studio	xCode
Avantages	Développement tout support	Développement tout support	Développement tout support	Développement uniquement sous Android	Développement uniquement sous IOS
	Prototypage rapide avec des connaissances en JS	Prototypage rapide avec des connaissances en JS	Prototypage rapide avec des connaissances en JS	Grande possibilités de personnalisations mais longue (graphie)	Très bon support de xCode par Apple
	nombre conséquent de composants proposer	Nombre conséquents de composants de base	Beaucoup de composants déjà développer (facile de trouver un code déjà fait)	Application native plus performante qu'avec ionic	Application native plus performante qu'avec ionic
	documentation complète	Documentation conséquente	Grande de documentation + grande communauté		
	plus performant que ionic et react native sur le plan natif (100% d'accès au code facilement transportable de ionic a nativescript (grace a angular)	Plus performant qu'ionic sur le plan natif	Graphique plus jolie et plus fluide qu'une application native		
Inconvénients	Faire la graphie des composants (boutons ...) temps de developpement plus long	Faire la graphie des composants (boutons ...)	Developement d'une application web ou hybride sulement	Temps de développement long (surtout en personnalisation graphique)	Documentation swift pas ouf
	besoin de télécharger des plugins (bluetooth)	Certaines fonctionnalités non officiels, il faut passer par des plug-ins (plug-in bluetooth)	Personnalisation d'un élément assez compliquer a réaliser	Tout a faire de A à Z donc developpement assez long	Tout a faire de A à Z donc developpement assez long
		Nécécite de connaître objective C, swift et java pour certain probleme	l'app n'est pas aussi performante qu'une app native		
Langage	AngularJS / css / HTML	JS / css / HTML	AngularJS / css / HTML	JAVA	Swift

Travaillant avec un cycle de développement itératif, le cahier des charges arrivait oralement au fur et à mesure de l'avancement du projet.

La première contrainte a étudié était que la technologie puisse me permettre facilement d'introduire du Bluetooth dans mon application. Il fallait ensuite pour une raison de coût de développement, mais aussi de rapidité que la technologie soit portable sur différentes plateformes, notamment iOS et Android. Et enfin, que la technologie soit le plus facile et rapide à prendre en main.

Le choix s'est donc tourné vers Ionic qui propose initialement beaucoup plus de possibilités d'intégration de composants (aussi bien natifs que non-natif) que les autres solutions. Tout ou presque est incluse dans Ionic : l'accès aux composants tels que le Bluetooth, la caméra, le gyroscope, l'accéléromètre, etc..., que des composants graphiques (UI components) prêt à l'emploi.

Bluetooth :

Le but de cette application était de pouvoir faire communiquer la brique Bluetooth avec le smartphone de l'utilisateur. Il fallait donc que l'application soit en mesure de découvrir les appareils Bluetooth à proximité, de pouvoir s'y connecter, d'envoyer et recevoir des messages et de s'y déconnecter. Il a été convenu que l'application serait un dérivé d'une application de messagerie standard, avec des conversations et des messages.

L'entreprise utilisant un module Bluetooth serial, il m'a fallu utiliser le plugin correspondant dans la documentation officiel de Ionic. Ce plugin contient toutes les fonctions nécessaires avec les possibilités d'usage : connexion, déconnexion, découverte alentours, recevoir des messages envoyés des messages, etc...

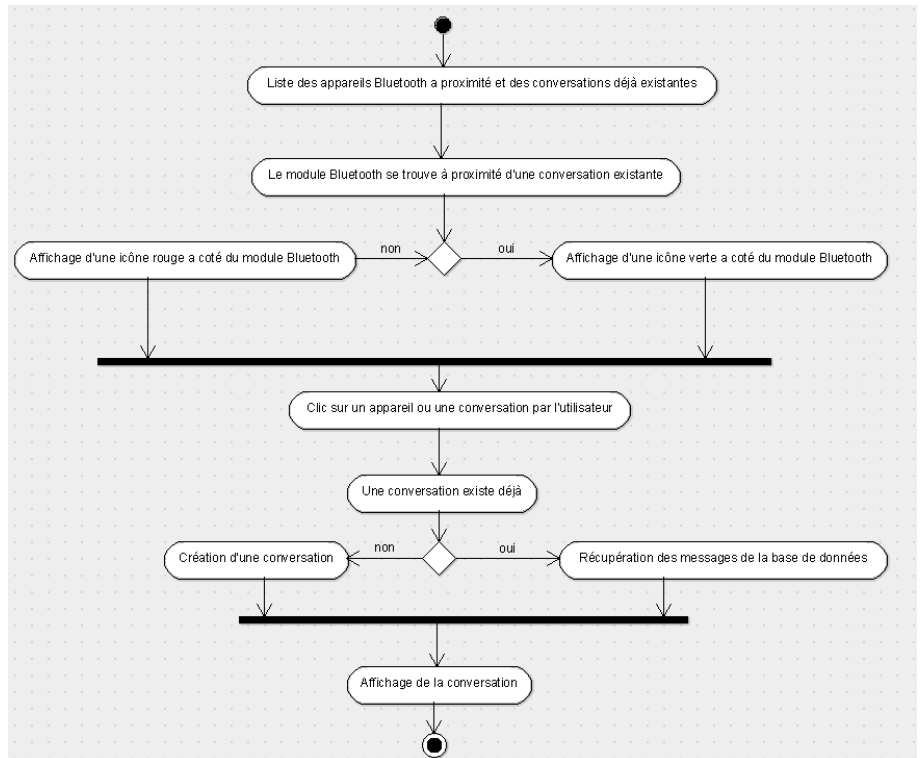
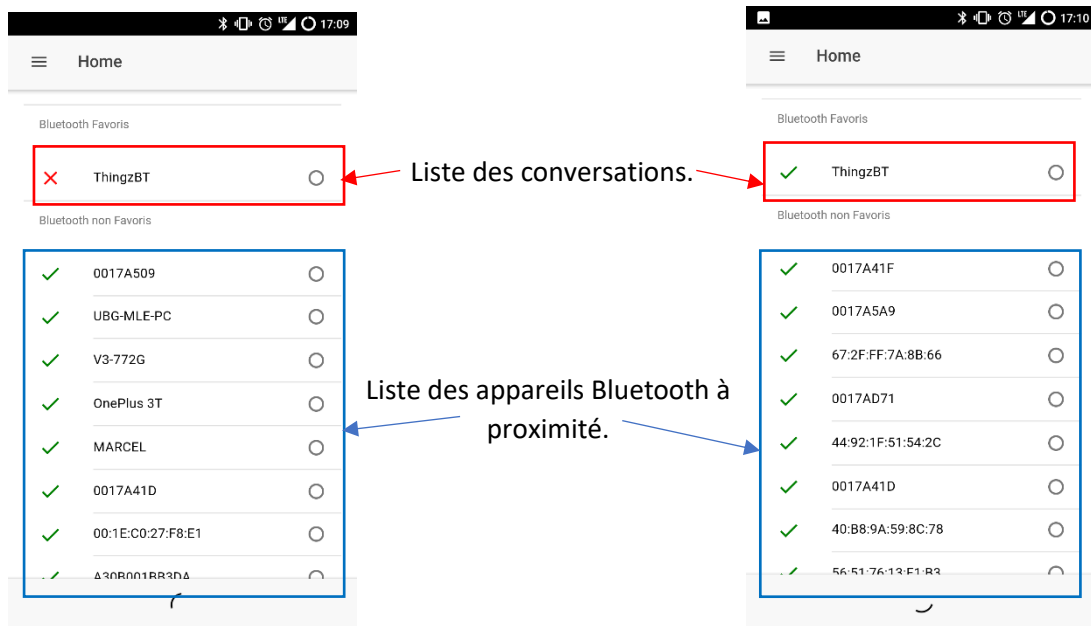
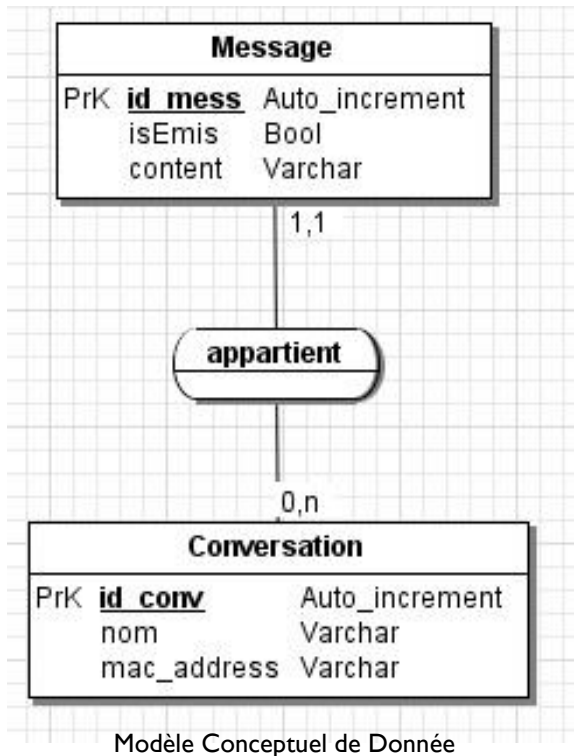


Diagramme d'activité de la page d'accueil



Base de données :



Comme une application de messagerie standard, l'application devait pouvoir stocker des conversations ainsi que des messages.

Les smartphones utilisent des bases de données SQLite, qui a pour particularité de ne pas établir le schéma habituel client-serveur des autres solutions, mais d'être directement intégré aux programmes. Ainsi, l'intégralité de la base de données est stockée dans un fichier.

Ionic propose un plugin facilitant la manipulation de ce SGBD (Système de Gestion de Base de Donnée).

Comme le montre le MCD ci-contre, un message appartient à une seule conversation et une conversation contient aucun ou plusieurs messages.

Chaque message a un identifiant unique qui permet de l'identifier, un booléen qui permet de savoir si c'est un message envoyé ou un message reçu, et un contenu qui est le message en lui-même.

Chaque conversation contient donc aussi un identifiant unique qui permet de l'identifier, un nom qui correspond au nom du module Bluetooth associé et une adresse mac qui permet la connexion au module.

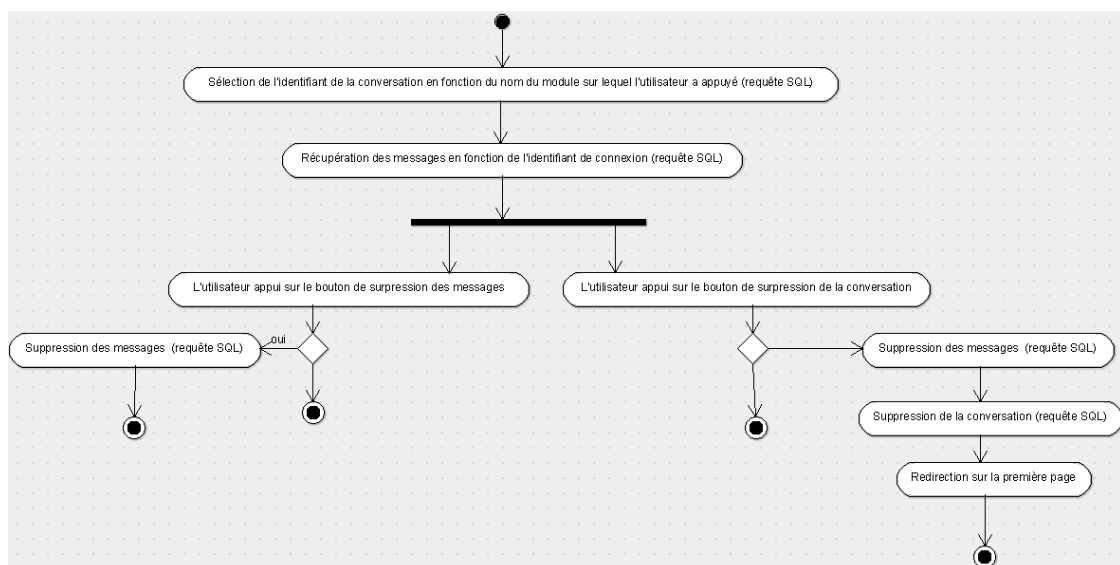


Diagramme d'activité lors du choix d'une conversation

L'application devait donc être en mesure de récupérer les messages stockés dans la base de données, mais aussi de pouvoir laisser à l'utilisateur le choix de supprimer les messages et la conversation sélectionnée.

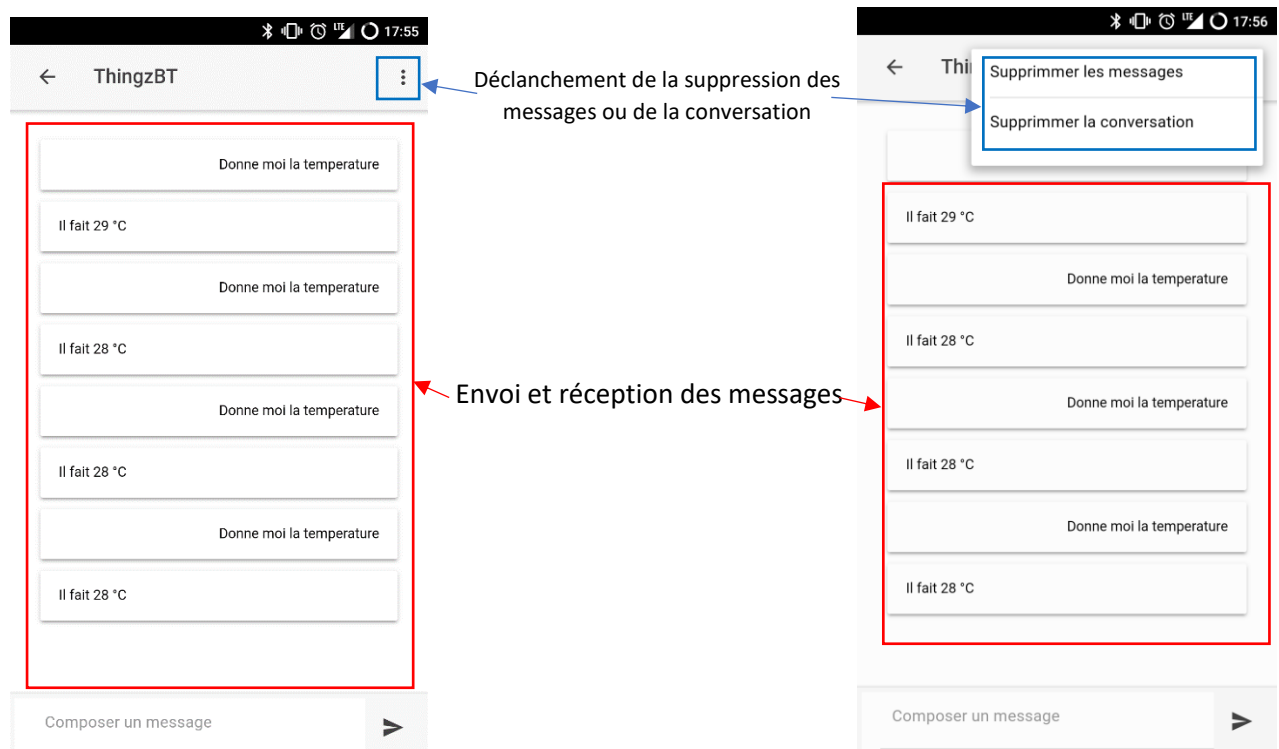
Exemple de cas d'utilisation :

Voici un exemple de cas d'utilisation de l'application :

J'ai créé un objet, qui me renvoie via Bluetooth la température quand je lui envoie le message suivant :
« Donne moi la température ». (A noter que le '\n' à la fin de la ligne 19 me sert de délimiteur dans l'application pour récupérer les messages qui sont stocker dans un buffer)

```
1 #include "Bluetooth.h"
2
3 #include "Temperature.h"
4
5 Bluetooth monBluetooth1;
6
7 Temperature meteo;
8
9 void setup()
10 {
11     monBluetooth1.acceptConnection("ThingzBT");
12 }
13
14
15 void loop()
16 {
17     if (monBluetooth1.dataAvailable()) {
18         if (monBluetooth1.receive() == "Donne moi la temperature") {
19             monBluetooth1.send(String("Il fait ") + String(String(meteo.temperature()) + String(" °C\n")));
20         }
21     }
22 }
23 }
```

Code de l'objet



Remerciements :

Je tiens à remercier tout particulièrement :

- Cyril Loucif-Durouge, CEO de l'entreprise et tuteur de mon stage, pour son aide ses conseils et son temps lors de mes différentes phase de questionnements pour toutes les réponses qu'il a pu m'apporter et l'aide qu'il a pu me fournir.
- Clément Amblard, développeurs en systèmes embarqués, pour tous ses conseils et ses astuces de développement apporter lors de mes phases de developpements.

4. Conclusion :

Lors de cette période de formation, j'ai eu l'occasion suivre une veille technologique constante en lisant les documentations sur les différentes technologies que j'utilisais. Ce stage m'a donc permis d'enrichir mes connaissances techniques mais aussi théoriques sur différentes techniques de programmations. Travailler sur un projet réel et concret apporte énormément de rigueur dans les tâches à réaliser, mais aussi beaucoup de compréhension dans la nécessité de l'apprentissage par le biais d'études ou d'autodidactie pour porter à bien un projet quel qu'il soit.

J'ai pu poursuivre mon apprentissage dans le domaine de la technique, mais aussi apprendre dans le domaine de l'entrepreneuriat, de la création et de la gestion d'entreprise car l'IOT Valley en général regorge d'entrepreneurs avec des entreprises de différentes tailles. Cela m'a permis de comprendre plus clairement les points abordés en cours sur l'économie, le droit et le management d'entreprise.