



Rapport de stage
Licence 3 Informatique

**DEVELOPPEMENT D'UNE APPLICATION MOBILE
ET D'UNE API REST**

Entreprise d'accueil : AXELIFE

Maitre de stage : Franck MOUNEY

Encadrant académique de Stage : Bruno ZANUTINI

YABA BILONGO Michel Davel

2017/2018

REMERCIEMENTS

Que l'honneur et la gloire revienne au Dieu Tout Puissant qui, par sa grâce, j'ai eu le courage, la volonté et l'intelligence nécessaire pour réaliser ce travail.

Un tel travail ne saurait être l'œuvre d'une seule personne. C'est pourquoi, je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage.

J'adresse mes remerciements au Directeur de la Société AXELIFE, le Docteur Magid HALLAB, de m'avoir accueilli dans son entreprise. Je n'oublie pas non plus le directeur de la Normandie Incubation M. Laurent PROTIN qui m'a permis de postuler dans cette entreprise et trouver ce stage qui était en adéquation avec mes attentes.

Je tiens à remercier vivement mon maitre de stage, Franck MOUNEY, responsable recherche et développement au sein de la Société AXELIFE, pour le temps passé ensemble, le partage de son expertise au quotidien et son accueil durant mon séjour à Saint-Nicolas de Redon. Grâce à sa confiance j'ai pu accomplir les missions qu'il m'a assignées. Il fut d'une aide précieuse dans les moments délicats.

Mes remerciements vont aussi au corps enseignant de la faculté des Sciences de l'Université de Caen, pour la formation assurée, en particulier à monsieur Bruno ZANUTINI, mon encadrant académique de stage, pour son suivi et ses orientations.

Je ne manquerai pas de mentionner mon collègue de travail, Bocar TIMITE, avec qui j'ai travaillé pendant la durée de ce stage. Ce fut un réel plaisir de collaborer avec lui.

Je remercie également toute l'équipe d'AXELIFE, à savoir les autres stagiaires, et aussi l'équipe de la Normandie Incubation, pour leur accueil et pour m'avoir aidé à m'intégrer facilement dans leur environnement.

Enfin, je tiens à remercier toutes les personnes qui ont participé de près ou de loin, d'une manière ou d'une autre à l'aboutissement de ce travail : ma famille, mes amis de promotion, mes frères spirituels.

AVANT-PROPOS

L'Université de Caen Normandie intègre à la fin de son cursus de formation de Licence Informatique un stage de formation pratique d'une durée de deux mois minimale dans un laboratoire scientifique ou dans une entreprise.

Ce stage constitue une période d'activité durant laquelle un étudiant inscrit pour l'année universitaire en cours met en application les enseignements théoriques suivis, dans le cadre d'un projet réalisé dans un organisme d'accueil. Il fait l'objet d'une restitution et donne lieu à évaluation. Le stagiaire est sous l'encadrement d'un expert métier.

Le présent document représente l'aboutissement de deux (2) mois de stage pratique effectué dans la société AXELIFE et tient lieu à cet effet de notre rapport de fin de stage.

TABLE DES MATIÈRES

REMERCIEMENTS.....	I
AVANT-PROPOS.....	II
TABLE DES MATIÈRES	III
LISTE DES FIGURES ET TABLEAUX	V
INTRODUCTION GENERALE.....	1
PARTIE 1	2
PRESENTATION GENERALE	2
Présentation de la structure d'accueil et du sujet d'étude	3
Introduction	3
I. Structure d'accueil.....	3
I.1 Présentation	3
I.2 Organisation	3
I.3 Structure d'accueil.....	4
I.4 Localisation.....	4
II. Présentation du sujet.....	5
II.1 Contexte du sujet.....	5
II.2 Objectif du sujet.....	5
II.3 Problématique.....	5
II.4 Intérêt du sujet.....	6
Conclusion	6
Les concepts liés au sujet d'étude.....	7
Introduction	7
I. API.....	7
I.1 Les Types d'API.....	7
I.2 API REST	7
Conclusion	8
PARTIE 2.....	9
ANALYSE, CONCEPTION ET MISE EN ŒUVRE.....	9
Etude préliminaire & Méthodes	10
Introduction	10
I. Etude préliminaire	10
I.1 Description du système actuel	10
I.2 Limites du système actuel.....	10

I.3 Solution envisagée dans le cadre de ce projet	11
II. Choix d'une méthode d'analyse et conception	11
Conclusion	11
Analyse et conception	12
Introduction	12
I. ANALYSE DU BESOIN & CONCEPTION	12
I.1 Besoins fonctionnelles	12
I.2 Besoins techniques	12
I.3 Spécification fonctionnelle	13
I.3.1 Identification des acteurs	13
I.3.2 Identification des cas d'utilisation	13
I.3.3 Organisation des cas d'utilisation par acteurs	13
I.3.4 Organisation des cas d'utilisation par module	14
I.4 Conception de la base de données	15
I.5 Architecture globale de la solution	16
Conclusion	16
Mise en œuvre : Réalisation et déploiement	17
Introduction	17
I. Outils et technologies d'implémentation	18
II. Réalisation	19
II.1 Développement du Client	19
II.2 Développement de l'API REST	20
II.3 Application Web	21
III. Quelques captures d'écrans	21
Conclusion	22
PARTIE 3	24
GESTION DU PROJET ET BILAN	24
Gestion de Projet et Bilan	25
I. Conduite de projet	25
II. Bilan	25
III. Difficultés	26
IV. Perspectives	26
CONCLUSION GENERALE	27
BIBLIOGRAPHIE / WEBOGRAPHIE	28
ANNEXE	29

LISTE DES FIGURES ET TABLEAUX

Figure 1 - Organigramme AXELIFE.....	4
Figure 2 - Architecture du projet.....	5
Figure 3 - Fonctionnement d'une API.....	8
Figure 4 – Diagramme de cas d'utilisation Module Application Mobile et Serveur	14
Figure 6 - Architecture de la base de données.....	15
Figure 7 - Architecture Globale de la Solution	16
Figure 8 - Interface de l'application mobile	22
Figure 9 - Exécution de l'API REST	22
Tableau 1 - Besoins fonctionnels	12
Tableau 2 - Acteurs du Système.....	14

INTRODUCTION GENERALE

Dans le cadre de ma troisième année de Licence Informatique, j'ai effectué mon stage chez AXELIFE SAS. Il s'agit d'une entreprise spécialisée dans la commercialisation des dispositifs médicaux. J'ai souhaité réaliser mon stage dans une entreprise innovante parce qu'elle offre généralement à ses employés un environnement stimulant qui emmène ces derniers à se surpasser. Les missions liées à ce stage m'ont particulièrement attirées car elles me permettent de découvrir un domaine parmi d'autres de l'informatique et aussi à mieux faire le choix dans la poursuite de mes études.

AXELIFE SAS s'est fait connaître avec des produits comme « pOpmètre ». Ainsi, développant actuellement son second produit, « iHeartWatch », de nouveaux besoins se sont fait sentir. C'est dans ce contexte que s'inscrit ce stage dont le thème s'intitule « **Développement d'une application Android et d'une API REST dans le cadre du projet iHeartWatch** ».

Dans un premier temps, nous décrirons l'entreprise, le sujet d'étude et les concepts liés au domaine. Puis nous ferons une analyse, conception du travail demandé. Il s'ensuivra la présentation des étapes de développement de la solution avant de terminer par la gestion et le bilan du projet.

PARTIE 1

PRESENTATION GENERALE

Présentation de la structure d'accueil et du sujet d'étude

Introduction

Dans ce chapitre, nous allons présenter AXELIFE qui est la structure d'accueil, nous allons ensuite parler du contexte du sujet et de son intitulé, de la problématique du sujet et de son objectif puis, de l'intérêt du sujet.

I. Structure d'accueil

I.1 Présentation

AXELIFE SAS est une entreprise française créée en mars 2010 et spécialisée dans le domaine de la santé, notamment dans la prévention cardiovasculaire. Elle conçoit et développe des dispositifs médicaux.

Son premier produit, le pOpmètre est un appareil innovant qui mesure la rigidité artérielle (RA) via l'onde de pouls en seulement deux (2) minutes, soit en d'autres termes le vieillissement artériel. La RA étant un bio marqueur agréé par les sociétés savantes européennes et françaises comme meilleur indicateur de l'AVC et de l'infarctus, mieux que le diabète, la tension, ou le cholestérol. Ce paramètre médical du XXIème siècle est validé par plus de vingt-cinq (25) études de morbi-mortalité à travers le monde et quatre (4) méta-analyses concernant plus de cent milles (100 000) patients.

Le deuxième produit en cours de développement, iHeartWatch® est une montre connectée intelligente qui déclenche des alertes cadencées en cas de détection de signes de stress intense, comme par exemple la poussée de tension et le pouls qui s'accélère, et bien d'autres paramètres. A termes, l'objectif de l'iHeartWatch® est de détecter les signes prémonitoires d'un infarctus ou d'un AVC.

I.2 Organisation

AXELIFE SAS est organisée comme suit :

- ☞ Un conseil scientifique

- ☞ Un conseil stratégique
- ☞ Le service Recherche, Développement et Industrialisation (RD&I)
- ☞ Le service Qualité et Affaires Réglementaires (Q&AR)
- ☞ Le service Commercial
- ☞ Le service Production
- ☞ Un Support Conseil

D'où l'organigramme ci-après :

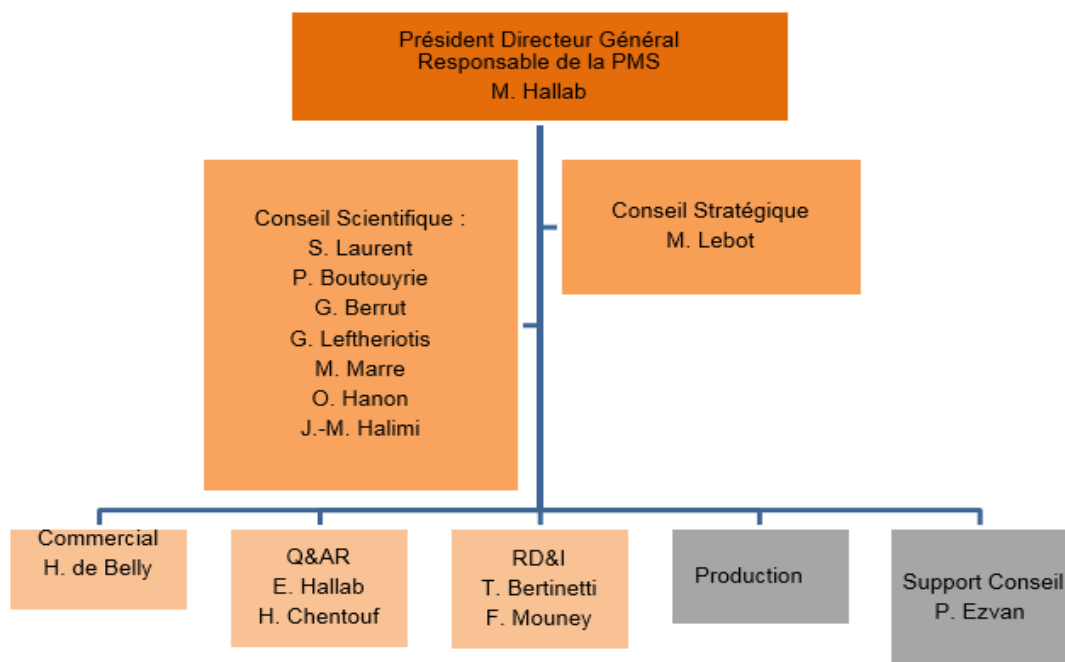


Figure 1 - Organigramme AXELIFE

I.3 Structure d'accueil

J'ai effectué mon stage au service Recherche, Développement et Industrialisation sous la responsabilité de Franck MOUNEY, ingénieur Recherche et Développement. Ce service a pour mission de répondre à tous les besoins techniques de l'entreprise.

I.4 Localisation

Le siège social de la société AXELIFE se situe au 17 Place La Fontaine, à Saint-Nicolas de Redon. En effet, Saint-Nicolas-de-Redon est situé sur la rive gauche de la Vilaine, à un kilomètre au sud-est de Redon (Ille-et-Vilaine), dans une agglomération à la jonction de deux régions, la Bretagne et les Pays de la Loire, et de trois départements, la Loire-Atlantique, l'Ille-et-Vilaine et le Morbihan.

II. Présentation du sujet

II.1 Contexte du sujet

Dans le cadre du développement du second produit, à savoir iHeartWatch, de nouveaux besoins sont apparus. En particulier, le développement d'une application mobile Android/iOS, le besoin de développement d'un système d'information sur Cloud sécurisé et de page web.

II.2 Objectif du sujet

L'objectif de ce projet est de développer la partie de collecte de ces valeurs et de stockage dans une base de données externalisée. Il est nécessaire de basculer les données acquises sur le système Android dans une base de données adaptée située sur un serveur web. Par la suite, les données sont traitées via un algorithme permettant de générer des bio-marqueurs et les résultats sont renvoyés au patient sur son application Android.



Figure 2 - Architecture du projet

II.3 Problématique

La problématique du sujet soumis à notre étude a des volets multiples. Vu les tâches à accomplir pour répondre au cahier de charge, nous nous sommes posés les questions suivantes :

- ☛ Comment récupérer les données (fichier CSV) issues du téléphone ?

- ☞ Comment stocker les données récupérées ?
- ☞ Quelles technologies utilisées pour arriver à bout de ces tâches ?

II.4 Intérêt du sujet

Le sujet d'étude revêt un intérêt capital pour la structure d'accueil et pour l'étudiant en fin de cycle. En effet, la réalisation de ce projet va permettre à AXELIFE de centraliser les données sur un serveur de l'entreprise pour plusieurs raisons :

- sauvegarder les données médicales patientes récoltées ;
- utiliser ces données pour l'amélioration du système expert de prédiction des problèmes cardiovasculaire ;
- permettre au médecin de consulter l'état de santé des patients en ligne via une page web.

Conclusion

Ce chapitre nous a permis de présenter la structure dans laquelle nous avons effectué notre stage et de poser les bases nécessaires à la compréhension de notre sujet. Le chapitre suivant est réservé à la présentation des concepts généraux liés à notre étude.

Les concepts liés au sujet d'étude

Introduction

Pour réussir et atteindre les objectifs de notre projet, il fallait appréhender les différents concepts y afférent. Nous présenterons dans ce chapitre, de manière plus ou moins brève, les divers concepts liés au domaine d'étude.

I. API

API : Application Programming Interface

En d'autre terme, c'est une interface qui permet à une application (mobile ou web) d'interagir et d'échanger les données avec d'autres applications.

Pour pouvoir accéder aux données reçues via une API, il faut bien sûr élaborer ou construire des requêtes.

I.1 Les Types d'API

Il existe deux grands types d'API : REST et SOAP. Dans le cadre de ce projet, nous allons nous appuyer sur les API REST.

I.2 API REST

REST : "Representational State Transfer". Sa principale particularité réside dans sa polyvalence concernant les types d'échange d'information entre applications. Les API REST sont basées sur le protocole http (Hypertext Transfer Protocol). Il sied, néanmoins, de signaler que tous les API ne sont pas basées sur http. Mais contrairement à SOAP les données peuvent être dans un format au choix. Le format JSON (JavaScript Object Notation) est souvent utilisé, mais XML, CSV, ou même RSS sont aussi valables. Comme le web, les API REST s'appuient sur le principe client-serveur.

Le principe client-serveur définit les deux entités qui interagissent dans une API REST, à savoir le client et le serveur. Le client envoie une requête, et le serveur renvoie une

réponse. Ce dernier doit avoir le plus d'informations possible sur le client, car il est important qu'ils soient capables de travailler indépendamment l'un de l'autre.

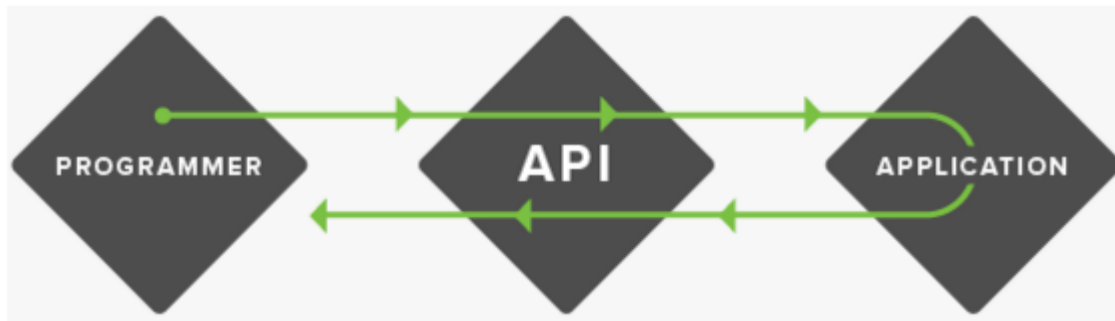


Figure 3 - Fonctionnement d'une API

Conclusion

Ce chapitre nous a permis de présenter les concepts liés à notre projet. La partie suivante est réservée à la présentation de l'analyse, la conception et la méthode que nous allons choisir pour bien mener à terme notre projet.

PARTIE 2

ANALYSE, CONCEPTION ET MISE EN ŒUVRE

Etude préliminaire & Méthodes

Introduction

L'objectif de cette partie est de mettre en lumière la situation actuelle du projet « iHeartWatch », nous ferons ressortir ses limites puis proposer des solutions. Enfin, nous allons faire un choix sur la méthode d'analyse de conception.

I. Etude préliminaire

I.1 Description du système actuel

Le projet « iHeartWatch » concerne la détection précoce des AVC et crises cardiaques à l'aide du bracelet connecté développé par la société AXELIFE. Ce bracelet permet la mesure continue de différentes constantes physiologiques (onde de pouls, fréquence cardiaque, oxymétrie....). L'objectif est de « remonter » ces données en continue et, par l'analyse de celles-ci, de lever des alertes le cas échéant. Une application Android a été développée afin de récolter les informations sur le mobile de chaque patient. A l'heure actuelle, l'application permet l'enregistrement des données acquises dans un fichier csv sur le portable de l'utilisateur.

I.2 Limites du système actuel

L'enregistrement des données se faisant dans un fichier csv sur le portable de l'utilisateur du bracelet, ceci ne constitue pas une solution viable à long terme pour la commercialisation du bracelet.

En effet, l'un des inconvénients de ces types de fichiers c'est le manque de sécurité car les données sont en clair et on peut les lire à partir de tous supports. Ces fichiers peuvent donc être lus facilement à l'insu de l'utilisateur. Ce qui constitue un souci vu la sensibilité de ces données.

I.3 Solution envisagée dans le cadre de ce projet

Vu les limites constatées ci-dessus, L'entreprise a envisagé de développer une application mobile Android qui charge un fichier csv depuis le téléphone puis l'envoie vers l'API qui va traiter ledit fichier en extrayant les données et les insérer dans une base de données. Ces données devront être cryptées par la suite.

II. Choix d'une méthode d'analyse et conception

La bonne réalisation d'un projet dépend de la méthode choisie. En effet, chaque projet est adapté à un type de méthode bien spécifique.

Plusieurs familles de méthodes existent : fonctionnelles, systémiques, formelle, orientés objets, etc. La méthode Orienté Objets correspondait le mieux à notre projet par sa simplicité de mise en œuvre et la proposition des langages de modélisation. Notre choix s'est fixé sur le processus 2TUP (processus unifié) et Le langage objet UML.

Conclusion

Dans ce chapitre nous avons fait une étude préliminaire du projet. Pour la méthode de conception, notre choix s'est porté sur le processus de développement logiciel 2TUP que nous avons couplé au langage de modélisation objet UML. Dans le chapitre suivant nous détaillerons l'analyse et la conception de notre système suivant le langage UML et le processus 2TUP.

Analyse et conception

Introduction

L'objectif de cette partie est de comprendre par l'analyse l'ensemble des éléments qui rentrent dans le processus de développement de notre application. Ensuite, nous allons nous servir des résultats de cette analyse pour bâtir une conception.

I. ANALYSE DU BESOIN & CONCEPTION

Il est indispensable dans tout type de projet de bien comprendre les besoins des futurs utilisateurs de l'application. Nous présenterons dans cette partie les besoins fonctionnels et techniques du système à concevoir.

I.1 Besoins fonctionnels

Le tableau suivant illustre les besoins en fonctionnalité.

BESOINS	FONCTIONNALITÉS
Application Android	<ul style="list-style-type: none">- chargement d'un fichier csv depuis le téléphone- Envoie du fichier chargé vers le serveur
Serveur	<ul style="list-style-type: none">- Extraction des données du fichier csv- Enregistrement des données extraites dans une base de données

Tableau 1 - Besoins fonctionnels

I.2 Besoins techniques

- ☞ **l'ergonomie** : le système doit présenter des interfaces claires, nettes et précises.
- ☞ **La performance**: L'application doit être simple d'utilisation et avoir des temps de réponses relativement courts.
- ☞ **Le langage de programmation** : Android Studio, Flask, HTML, CSS, JavaScript

I.3 Spécification fonctionnelle

I.3.1 Identification des acteurs

Un acteur représente une entité (personne ou un système) extérieure qui interagit directement avec le système étudié.

Le recueil des besoins fonctionnels nous a permis d'identifier les acteurs.

L'acteur principal étant celui qui déclenche la plupart du temps le cas d'utilisation. Dans le cas de ce projet, nous avons comme acteurs primaires :

- **Médecin** : Il a accès aux données envoyées par le patient au serveur à travers une interface qui lui sera destiné.

Par opposition à l'acteur primaire, l'acteur secondaire est un acteur sollicité par le système seulement lors du cas d'utilisation. Dans le cadre de notre projet, nous avons :

- **Application mobile**: c'est l'application qui charge le fichier csv et l'envoie au serveur.
- **Serveur** : Traite la demande en provenance du patient.

I.3.2 Identification des cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable, intéressant pour un acteur particulier. Il exprime les interactions acteurs/système et apporte une valeur ajoutée considérable à l'acteur concerné.

Ci-dessous les différents cas d'utilisation du système à mettre en place :

- ☞ Charger fichier
- ☞ Envoyer fichier
- ☞ Traiter fichier
- ☞ Enregistrement des données dans une base de données
- ☞ Consulter les données enregistrées dans la base

I.3.3 Organisation des cas d'utilisation par acteurs

N°	ACTEURS	CAS UTILISATIONS
1	App mobile	<ul style="list-style-type: none"> - Charger fichier - Envoyer fichier
2	Serveur	<ul style="list-style-type: none"> - Traiter fichier - Enregistrement des données dans une base de données
3	Médecin	<ul style="list-style-type: none"> - Afficher les données enregistrées dans la base

Tableau 2 - Acteurs du Système

I.3.4 Organisation des cas d'utilisation par module

Module Application mobile et Serveur

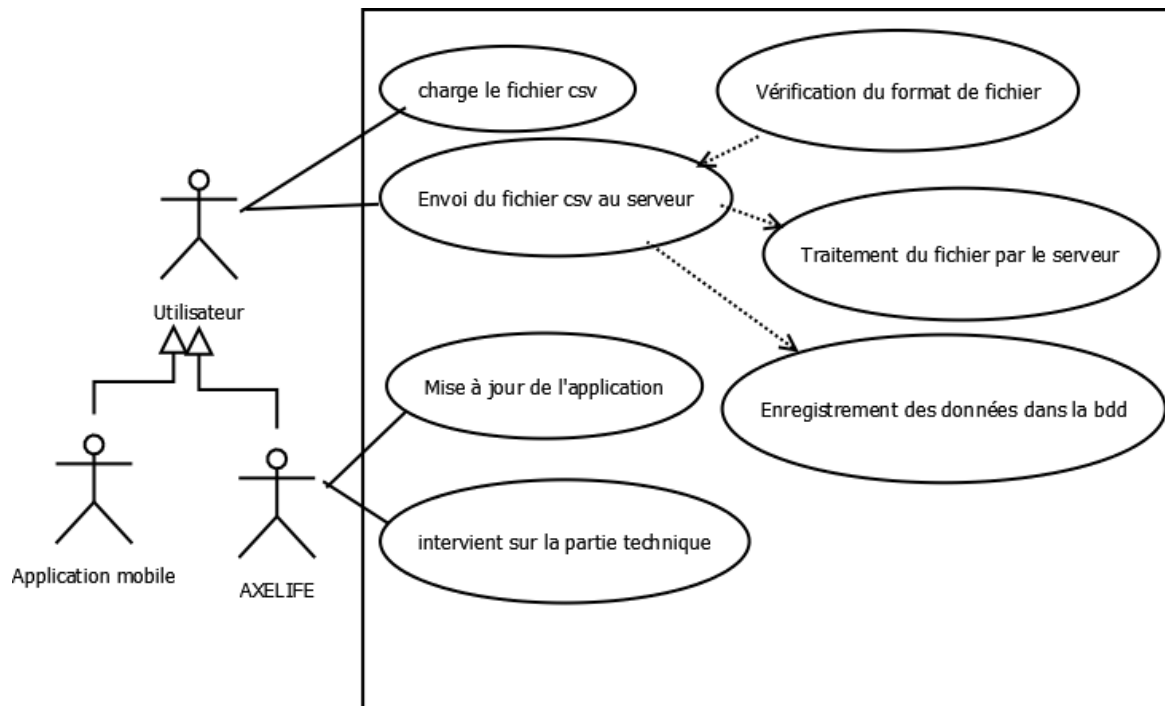


Figure 4 – Diagramme de cas d'utilisation Module Application Mobile et Serveur

L'expression des besoins a donné lieu à une modélisation par les cas d'utilisation. Nous avons ainsi identifié les acteurs et les cas d'utilisation.

Après avoir défini les bases fonctionnelles de notre application, nous allons à présent voir comment le serveur va communiquer avec la base de données.

I.4 Conception de la base de données

Une base de données est un ensemble d'informations archivées à l'aide d'un Système de Gestion de Base de Données (SGBD). Une base de données ici est nécessaire dans le but de stocker les données se trouvant dans le fichier csv que le l'application mobile va envoyer au serveur.

La conception d'une base de données est guidée par des principes. Le premier c'est la présence de données redondantes, car elles consomment de l'espace et augmentent la probabilité d'erreurs et les incohérences. Le second c'est l'importance de l'exactitude et de l'intégralité des données présentes sur la base de données. Une conception appropriée de base de données doit, par conséquent :

- Diviser les informations en tables pour réduire les données redondantes.
- Offrir un accès avec les informations nécessaires pour joindre les informations dans les tables par rapport aux besoins.
- Aider à la prise en charge et à la garantie de la précision et l'intégralité des données.
- Prendre en charge le traitement des données en rapport.

En respectant ces principes et en fonction des données qui se trouvent dans le fichier csv, voici l'architecture correspondante à notre base de données :

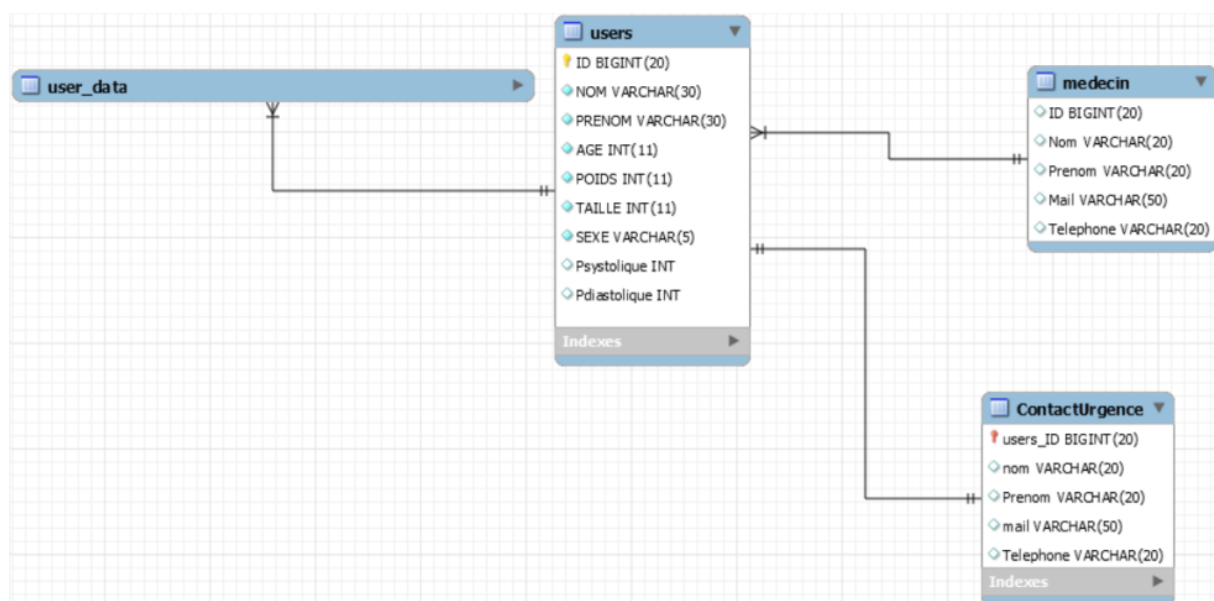


Figure 5 - Architecture de la base de données

Sur le schéma ci-dessus la table centrale est celle du patient, celle-ci est identifiée par un identifiant (ID) c'est la clé primaire de la table. Cette table a de nombreuses connexions, un Patient est lié à ses données brutes, mais les données sont liées à un seul patient. De plus chaque patient est lié à un seul contact d'urgence et à un seul médecin. Cependant un médecin peut avoir plusieurs patients.

Le but de ce programme est que lorsqu'un nouveau patient s'enregistre via l'application mobile, ses données personnelles comme son nom, son prénom, son poids, sa taille, mais aussi des informations sur le contact d'urgence et le médecin de ce patient soient automatiquement enregistrées dans la base de données.

I.5 Architecture globale de la solution

Nous avons opté pour l'architecture suivante :

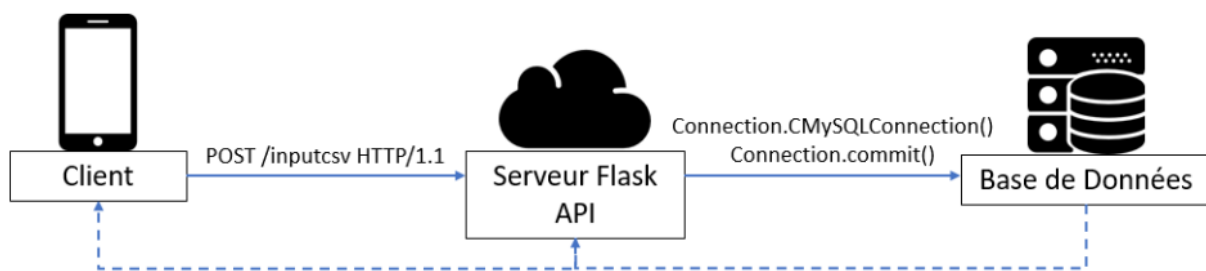


Figure 6 - Architecture Globale de la Solution

Conclusion

Ce chapitre nous a permis de présenter les phases d'analyse et de conception de la solution informatique à mettre en place.

Partant des besoins exprimés, nous avons modélisé (Diagrammes de cas d'utilisation) ladite solution à partir de la méthode que nous avons choisie.

L'analyse et la conception de la solution étant faite, il nous faut alors la mettre en œuvre.

Mise en œuvre : Réalisation et déploiement

Introduction

Dans ce chapitre nous allons présenter les technologies qui ont permis l'implémentation de la solution.

I. Outils et technologies d'implémentation



Dia est un logiciel libre de création de diagramme développé en tant que partie du projet GNOME conçu par Alexander Larsson.

Il propose aussi la génération de code PHP5, C++, Java, Python, etc. directement depuis le diagramme UML.



Android Studio est un environnement de développement pour développer des applications Android. Il permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.



XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web confidentiel, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.



Sublime Text est un éditeur de texte générique. A présent elle est à sa 3^{ème} version, que j'ai utilisée. Sublime Text intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l'auto complétion, un système de plugins...



Flask est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates. Flask contient un serveur de développement et un débbuger, supporte les tests unitaires, utilise le moteur de template Jinja2, supporte les cookies sécurisés (session), est entièrement

compatible avec WSGI 1.0, se base sur l'Unicode, dispose d'une documentation complète, est compatible avec Google App Engine, permet de créer des extensions.

Ajax combine JavaScript, les requêtes de type XMLHttpRequest, les manipulations du DOM, ainsi qu'un format de données (XML ou JSON).



Il est utilisé pour la communication asynchrone : envoyer les requêtes vers le serveur et déclencher des opérations lors de la réception de réponses de celui-ci.). Il est utilisé pour structurer les informations envoyées par le serveur Web).

HTML



HyperText Markup Language, généralement abrégé **HTML**, est le langage de balisage conçu pour représenter les pages Web. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS).

II. Réalisation

II.1 Développement du Client

Dans le cas de notre projet, le Client est une application mobile.

Pour développer ce client, nous nous sommes appuyés sur un client web qui a été au préalable développé. En effet, le Client web comprend deux parties :

- **Front-End** : C'est la vitrine du Client Web. Il comprend un formulaire contenant un deux boutons : le bouton de téléchargement d'un fichier CSV et le bouton d'envoi du fichier au serveur. Les technologies utilisées sont HTML/CSS.
- **Back-End** : C'est la partie cachée du Client Web. Cette partie intervient quand l'utilisateur clique sur le bouton d'envoi. En effet, elle s'occupe de récupérer le fichier CSV envoyé par le patient puis l'envoie grâce à une requête AJAX à

l'adresse du serveur. L'adresse du serveur correspond à l'adresse IP de la machine sur laquelle tourne le serveur. Pour que la requête puisse avoir lieu entre les deux entités (le client et le serveur), ils doivent être dans un même réseau. Cela se fait grâce à la technologie AJAX utilisant JavaScript.

Etant donné que notre client est une application mobile Android et comme un client web a déjà été développé, il a juste suffi pour moi de l'intégrer dans le projet Android.

La démarche que j'ai suivie est la suivante :

- J'ai créé un projet dans Android Studio
- J'ai ajouté au Layout un widget appelé « WebView ». En effet, Android WebView est un composant de système proposé par Chrome qui permet d'afficher les contenus Web dans les applications Android. Ce composant est préinstallé sur un appareil Android.
- Pour intégrer le client web dans Android Studio, plusieurs possibilités s'offrent à nous mais celle que j'ai choisie consiste à générer un dossier « Asset » puis y intégrer le projet correspondant au Client Web, à savoir :
 - Le fichier HTML qui représente le Front-End
 - Le fichier JavaScript qui représente le Back-End
- Une fois le client Web intégré dans le projet Android, nous avons chargé le client web dans « WebView » défini dans le Layout grâce à « loadUrl » en précisant le chemin d'où se trouve notre client web. Après cela, on a activé JavaScript grâce à « setJavaScriptEnabled » puis l'accès aux fichiers du téléphone grâce à « setAllowFileAccess ». Enfin, le reste du code qui lie le client web à l'application Android est intégré dans « setWebChromeClient ».

II.2 Développement de l'API REST

Comme dit précédemment, c'est avec le framework Flask et Python 3 que j'ai développé l'API REST.

Comme notre API REST interagit avec la base de données, nous avons donc au préalable créé une base de données vide. Pour y arriver, nous avons eu besoin de XAMPP.

Pour développer notre serveur, c'est au travers d'un programme de type « Python » que nous l'avons fait.

Pour arriver au bout de notre tâche, on a eu besoin de télécharger certains modules puis les intégrer dans notre projet. Il s'agit de :

- **Flask** : permet d'importer le module Flask
- **Flask –mysql** : c'est une extension pour Flask qui nous permet d'accéder à une base de donnée MYSQL
- **Flask-restful** : c'est une extension pour Flask qui nous permet de créer rapidement une API REST
- **Flask-connector-python**

Le programme utilisé pour créer notre API REST comprend deux fonctions principales :

- ☞ la première fonction se lance à partir de la route principale (« / ») et permet de créer les tables dans la base de données créé ;
- ☞ la seconde fonction se lance à partir de la route (« /inputcsv »). C'est sur cette route qu'est envoyé le fichier csv. Elle permet donc de récupérer le fichier csv, extrait les données de ce fichier puis les enregistre dans la base de données.

Après cela, il y'a un script qui permet d'exécuter le programme. Par défaut, les programmes Flask s'exécutent en local (127.0.0.1). Mais étant donné que le serveur est utilisé dans un réseau, nous avons modifié cette adresse pour mettre l'adresse 0.0.0.0 pour permettre l'accès au serveur avec n'importe qu'elle adresse.

Pour utiliser le serveur il faudrait tout d'abord créer les tables dans la base de données.

II.3 Application Web

C'est une interface web par laquelle le médecin va consulter les données enregistrées dans la table. Cette interface a été réalisée en Flask.

III. Quelques captures d'écrans

Dans cette partie seront présentés quelques écrans de l'application.

Client

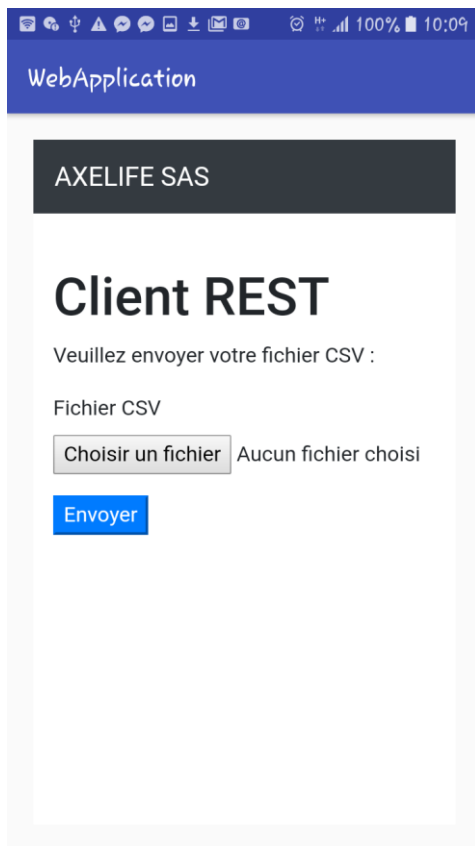


Figure 7 - Interface de l'application mobile

Serveur

```
yaba@yaba-HP-Pavilion-g4-Notebook-PC:~/Documents/Livrables Serveurs/Programmes/Api_rest$ sudo lampp restart
Restarting XAMPP for Linux 5.6.36-0...
XAMPP: Stopping Apache...ok.
XAMPP: Stopping MySQL...ok.
XAMPP: Stopping ProFTPD...ok.
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
XAMPP: Starting ProFTPD...ok.
yaba@yaba-HP-Pavilion-g4-Notebook-PC:~/Documents/Livrables Serveurs/Programmes/Api_rest$ python essai.py
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

Figure 8 - Exécution de l'API REST

Conclusion

A travers ce chapitre, nous avons présenté les étapes de développement de notre API REST partant de son architecture à ses différentes interfaces et ce à partir des technologies, langages et outils de développement choisis. Il nous a par ailleurs fallu une conduite de projet. Nous allons, dans la dernière partie de ce rapport, exposer la

démarche de conduite de projet qui nous a permis de mener à bien ce travail et faire le bilan.

PARTIE 3

GESTION DU PROJET ET BILAN

Gestion de Projet et Bilan

I. Conduite de projet

Un projet est un processus unique qui consiste en un ensemble d'activités coordonnées et maîtrisées, comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences du cahier des charges.

De ce fait d'après Wikipédia, la gestion d'un projet devient alors une démarche visant à organiser de bout en bout le bon déroulement du projet.

Plusieurs outils aident à la bonne conduite d'un projet, ainsi dans la cadre de ce projet j'ai travaillé sur Trello.

Vous trouverez en annexe, quelques captures d'écrans.

II. Bilan

Bilan technique

Enregistrer les données issues d'un fichier csv dans une base de données était la préoccupation majeure d'AXELIFE. Notre travail a abouti à la réalisation d'une application mobile Android, le client, et d'un API REST, le serveur. Les deux entités communiquent à travers un même réseau en local. Les données sont enregistrées dans une base de données locale. L'accès au serveur se fait à partir de son adresse IP dans le réseau local.

Pour aborder le développement de notre projet, nous avons étudié les concepts liés au domaine d'étude, avons fait une analyse et conception du sujet, choisi des technologies utiles pour notre projet et enfin procéder au développement. L'analyse et la conception a été faite en 2TUP et UML.

Bilan personnel

Cette expérience de stage nous a été bénéfique et enrichissante tant du point de vue social que professionnel. Nous noterons :

- ☞ Une connaissance du monde professionnel.
- ☞ L'utilisation de nouvelles technologies comme Flask.
- ☞ Une bonne connaissance du concept Client-Serveur.

- ☞ Une adaptation à un nouvel environnement de travail et technologique.

III. Difficultés

Toute œuvre est sujette aux obstacles, dans notre cas, voici les difficultés rencontrées lors du stage :

- ☞ L'apprentissage de nouvelles technologies.
- ☞ la familiarisation avec les concepts du domaine comme API, API REST.
- ☞ Rendre l'API accessible dans un réseau local.

IV. Perspectives

En termes de perspective, nous souhaitons arriver à l'aboutissement de ce projet en le rendant accessible sur le web.

CONCLUSION GENERALE

En définitive, ce stage a été très enrichissant pour moi. En effet, il m'a permis d'approfondir par la pratique, mes connaissances théoriques acquises durant mes années de formation. Il m'a permis aussi de savoir comment mener un vrai projet.

Fort de cette expérience, j'aimerais beaucoup essayer de m'orienter vers le domaine de l'administration Système.

BIBLIOGRAPHIE / WEBOGRAPHIE

Bibliographie

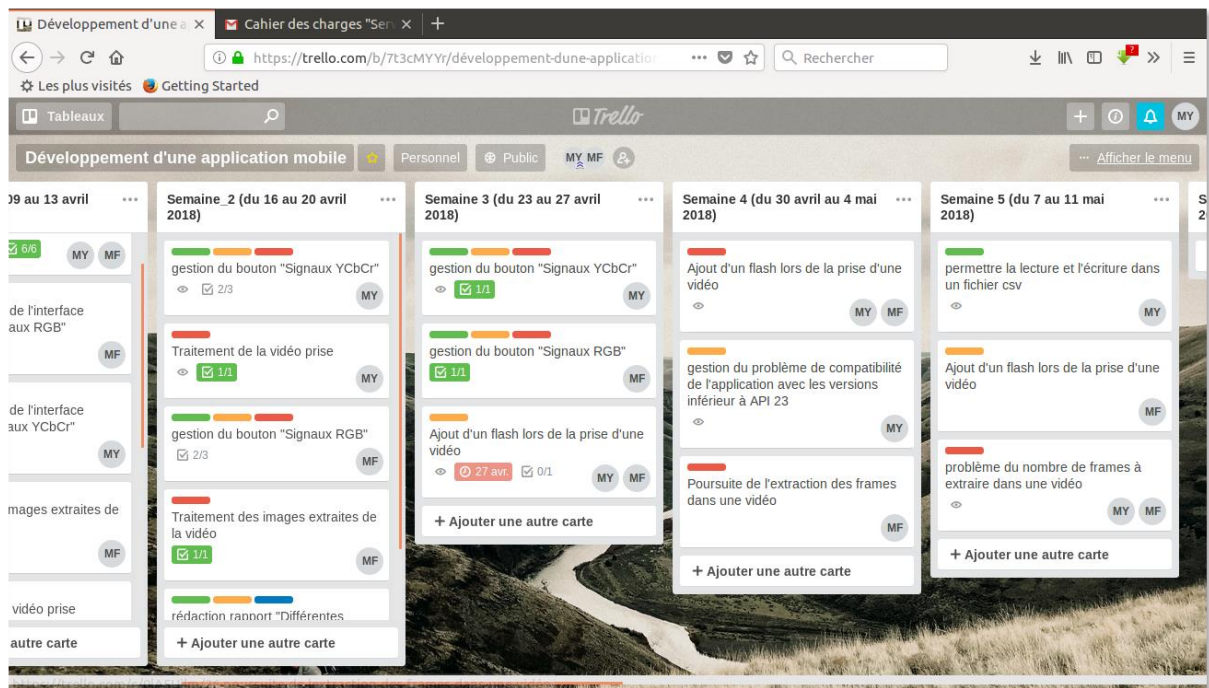
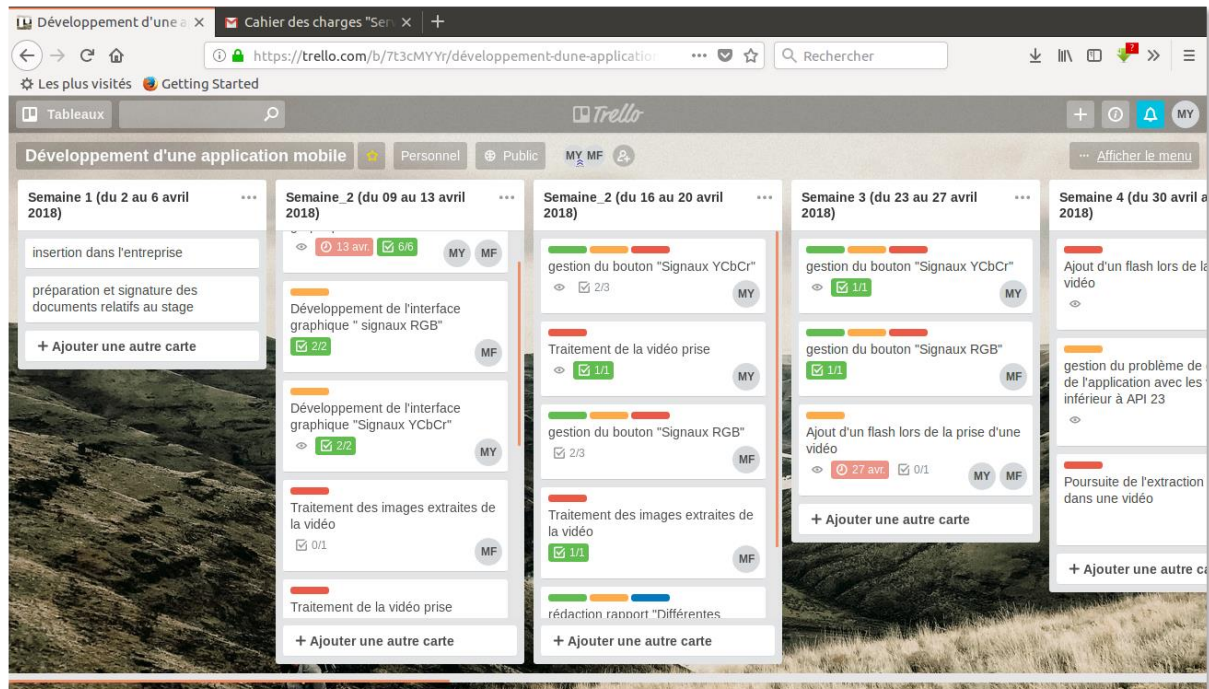
FireZéro – Créez vos applications web avec Flask – Openclassrooms, 2013. 51 p.

Webographie

- ☞ Emily Reese - Utiliser des API REST dans vos projets – Openclassrooms - Consulté le 1 juin 2018 - <https://openclassrooms.com/courses/utilisez-des-api-rest-dans-vos-projets-web/qu-est-ce-qu-une-api>
- ☞ Le Journal du Net – Interface de Programmation : API – Consulté le 8 mai 2018 - <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203559-api-application-programming-interface-definition-traduction/>
- ☞ Winner Concilie DEDJI – C'est quoi une API ? – SupInfo – consulté le 9 mai 2018 - <https://www.supinfo.com/articles/single/1345-c-est-quoi-une-api>
- ☞ Python Programming – Web Development Tutorial – consulté du 10 au 18 mai 2018 - <https://pythonprogramming.net/>
- ☞ Site officiel de Flask – consulté du 10 au 18 mai 2018 - <http://flask.pocoo.org/docs/0.12/>
- ☞ Site officiel MySQL – MySQL Connector / Python Developer Guide – consulté du 15 au 18 mai 2018 - <https://dev.mysql.com/doc/>
- ☞ Bruno Krebs – Developing RESTful APIs with Python and Flask – Autho blog. Publié le 28 Septembre 2017 - <https://autho.com/blog/developing-restful-apis-with-python-and-flask/>

ANNEXE

Captures d'écran de la Gestion de Projet sur Trello



Développement d'une Application Android et d'une API REST dans le cadre du Projet iHeartWatch

