



Remerciement

En premier lieu, je tiens à exprimer toute ma reconnaissance envers **Saïd MAHMOUDI**, mon tuteur de stage et professeur d'informatique au service ILIA de la Faculté Polytechnique de Mons, pour m'avoir accordé ce sujet de stage et pour la confiance sans faille qu'il m'a accordée.

Par la même occasion, je remercie **Olivier DEBAUCHE**, assistant doctorant du même service, pour m'avoir guidé tout au long de ce stage, ainsi que pour son soutien et sa bienveillance.

J'exprime toute ma gratitude envers **le service ILIA** pour son accueil chaleureux et qui a contribué au bon déroulement de ce stage.

Je tiens également à remercier **Thierry NOUNDOU NJIKE**, étudiant en M1¹ à Polytech Mons et spécialiste en Flutter, pour m'avoir transmis ses connaissances et pour m'avoir accompagné dans la réalisation de mes missions.

Je saisis l'occasion pour remercier toute l'équipe enseignante de Polytech Paris-Saclay ainsi que ma tutrice et professeur **Emmanuelle FRENOUX** pour la qualité de leur enseignement.

Sans oublier de remercier **les relations internationales** de Polytech Paris-Saclay et de l'Université de Mons qui m'ont permis de réaliser ce séjour en Belgique.

Enfin, je tiens à remercier toutes les personnes m'ayant apporté leur soutien, ainsi que leur aide pour la confection et la relecture de ce rapport.

¹ Master 1

Résumé

Ayant pour objectif de m'orienter dans le **développement logiciel** et afin de valider ma période de mobilité obligatoire, j'ai décidé de réaliser mon stage assistant ingénieur à l'étranger. J'ai ainsi intégré, durant 3 mois, le service **ILIA** (Informatique Logiciel et Intelligence Artificielle) de la **Faculté Polytechnique de Mons**, en Belgique.

Avec l'augmentation de la population mondiale et son vieillissement, le modèle médical traditionnel a atteint ses limites. En effet, il devient difficile d'obtenir des rendez-vous médicaux, le prix des consultations et des traitements sont chers.

En parallèle, le développement de **l'Internet des objets de la santé** (HIoT) a permis de proposer une solution pour réaliser des diagnostics à distance, développer la consultation de télémédecine, suivre les malades chroniques et les personnes âgées. Ces nouveaux services sont basés sur **l'Internet des objets médicaux** (IoMT).

Dans ce contexte, le service ILIA travaille sur une nouvelle architecture en temps réel de l'IoMT pour le suivi des patients âgés : l'architecture **RAMi**. Afin de visualiser et d'analyser toutes ces données en temps réel, j'ai développé une application multiplateforme qui servira de tableau de bord.

Abstract

In order to evolve in the fields of **Software Development**, and to validate my mobility period, I decided to realize my assistant engineer internship abroad. Therefore, I integrated, for 3 months, the **ILIA** (Informatics Software and Artificial Intelligence) department of the **Polytechnic Faculty of Mons**, in Belgium.

With the increase of the world population and its aging, the traditional medical model has reached its limits. Indeed, it is becoming difficult to get medical appointments, the price of consultations and treatments are expensive.

At the same time, the development of the **Internet of Health Objects** (IoHT) has made it possible to offer a solution for making remote diagnoses, developing telemedicine consultations, and monitoring the chronically ill and the elderly. These new services are based on the **Internet of Medical Things** (IoMT).

In this context, the ILIA department is working on a new real-time IoMT architecture for the monitoring of elderly patients: the **RAMi** architecture. To visualize and analyse all these real time data, I developed a cross-platform application (Web/Desktop/Mobile) that will serve as a Dashboard.

Sommaire

Introduction	1
I. Présentation de l'organisme d'accueil.....	2
A. Présentation de l'Université de Mons (UMONS).....	2
1. Activité d'enseignement.....	2
2. Activité de recherche.....	2
B. Présentation de la Faculté Polytechnique de Mons	3
C. Présentation du service ILIA (Institut Infortech).....	3
1. Activités d'enseignement.....	4
2. Activités de recherches.....	4
II. L'objectif de la mission : problèmes et enjeux globaux	5
A. Présentation du projet.....	5
B. L'objectif du stage.....	7
C. Les limites et enjeux.....	7
III. Réalisation de la mission.....	8
A. Récupération des données via le protocole MQTT	9
1. Description du protocole MQTT	9
2. Système acquisition des données côté publication	10
3. Système acquisition des données côté souscription.....	11
B. Développement côté serveur : traitement des données	12
1. Choix de la base de données.....	12
2. Architecture de traitement des données.....	13
3. Conception	14
4. Authentification des utilisateurs.....	15
C. Développement côté client : interface utilisateur	15
1. Choix technologiques.....	15
2. Conception UI/UX	16
D. Gestion de projet	17
E. Bilan et perspectives.....	17
1. Les fonctionnalités réalisées.....	18
2. Apports pour le service	18
3. Limites et perspectives	19
IV. Retour d'expérience	20

A.	Les apports personnels du stage	20
1.	Apports sur le plan humain	20
2.	Apports sur le plan professionnel	20
B.	Difficultés rencontrées et solutions apportées	21
C.	Ouverture vers ma future carrière.....	22
Conclusion.....		24
Références		26
Annexes		27
Annexe 1 : CV.....		28
Annexe 2 : Diagramme de Gantt.....		29
Annexe 3 : Liste des figures et tableaux.....		30
Annexe 4 : Tableau comparatif des technologies.....		31
Annexe 5 : Manuel d'utilisation.....		33

Introduction

Actuellement étudiante en 4^e année à Polytech Paris-Saclay, je prépare un diplôme d'ingénieur en informatique jusqu'à l'été 2023. Dans le cadre d'un échange **ERASMUS+**, j'ai eu la chance de réaliser un stage à **l'Université de Mons** pour une durée de 3 mois, du 02/05/2022 au 31/07/2022. J'ai réalisé ce stage plus particulièrement à la Faculté Polytechnique de Mons, au sein du service **ILIA** (Informatique Logiciel et Intelligence Artificielle).

Durant cette période, j'ai eu l'occasion de découvrir la ville de Mons, ainsi que la Belgique en général. J'ai beaucoup apprécié cette immersion à l'étranger et je garderai en mémoire cette belle expérience.

Afin de faire face aux difficultés des systèmes médicaux actuel, dû à l'accroissement de la population et son vieillissement, **l'Internet des objets de la santé** (HIoT) se développe afin de proposer des solutions pour des télédiagnostics, les consultations de télémedecine, le suivi des malades chroniques et des personnes âgées.

Dans ce cadre, le service ILIA travaille sur une nouvelle architecture en temps réel de l'IoMT pour le suivi des patients âgés : **l'architecture RAMi**. Ainsi l'objectif de mon stage était de développer une application multiplateforme (Web/Desktop/Mobile) qui servira de tableau de bord pour visualiser et d'analyser ces données en temps réel.

Ce rapport sera composé d'une présentation de l'organisme d'accueil, puis d'une explication du contexte de ma mission. Ensuite, les détails sur la réalisation de ma mission ainsi que mon planning seront présentés. Enfin, un retour d'expérience sera énoncé pour clore ce rapport.

I. Présentation de l'organisme d'accueil

A. Présentation de l'Université de Mons (UMONS)



L'université de Mons (UMONS) est une université wallonne² de Belgique située dans la ville de Mons, près de la frontière franco-belge, dans la province de Hainaut. Elle dispose également d'un site à Charleroi depuis 1978. On peut voir son campus « Plaine de Nimy » sur la figure 1.

Figure 1 - Campus Plaine de Nimy, web.umons.ac.be

Elle est née en 2009 de la fusion entre **l'Université de Mons-Hainaut** et de la **Faculté Polytechnique de Mons**, sa plus ancienne Faculté créée en 1837.

1. Activité d'enseignement

L'UMONS est composé de près de **10 000 étudiants** et comprend **7 Facultés**, dont la **Faculté Polytechnique de Mons**, **3 écoles** (droit, formation des enseignants, sciences humaines et sociales) et l'**InstitutsIRSTL** (Institut de Recherche en Sciences et Technologie du Langage). Elle délivre les grades de **Bachelier**, de **Master** et de **Doctorat** dans des domaines très diversifiés.

2. Activité de recherche

Avec près de **900 chercheurs**, la recherche de l'UMONS poursuit ses ambitions tant au niveau régional qu'international, avec l'objectif commun de développer une expertise largement reconnue et bénéfique pour la société, notamment pour sa région. Au niveau **international**, elle s'intègre dans des réseaux où ses chercheurs disposent d'une réelle expertise, tels que la Chimie et les Matériaux. L'UMONS est également titulaire du label européen « Research in Excellence ».

Depuis 2012, l'Université de Mons a réorganisé ses activités de à l'aide d'une structure reposant sur 10 **instituts de recherche** (Biosciences, Complexys, Énergie, **InforTech**, langage

² L'UMons se situe en Wallonie, région francophone du sud de la Belgique.

matériaux, Numediart, Risques, Santé, Soci&ter), avec une taille d'ordre de 50 à 100 chercheurs.

B. Présentation de la Faculté Polytechnique de Mons

La **Faculté Polytechnique de Mons**, autrement appelé **Polytech Mons**, est la première école d'ingénieur belge. Elle comporte **1000 étudiants** en Sciences de l'Ingénieur et **250** académiques, scientifiques, chercheurs, techniciens et administratifs.



Figure 2 - Entrée principale de la Faculté Polytechnique de Mons

Fondée à Mons par deux Ingénieurs diplômés de l'École Centrale Paris, visible sur la figure 2, elle forme des **Ingénieurs civils depuis 1837** dans divers domaines. Elle délivre le diplôme de **Bachelier** Ingénieur Civil, le diplôme de **Master** Ingénieur civil dans **6 spécialités** (science des matériaux, électricité, architecte, informatique et gestion, mécanique et mines et Géologie).

La poursuite en **Doctorat** est également possible dans les spécialités : sciences de l'ingénieur et technologie, art de bâtir et urbanisme, art et sciences de l'art.

Elle comporte **4 centres de recherche ou pôles d'excellences** : le CETIC, Multitel, Materia Nova, Inisma, NUMEDIART.

C. Présentation du service ILIA (Institut Infortech)

Le début du XXI^e siècle a vu se développer considérablement les sciences et technologies de l'information et de la communication (**TICs**) associant l'informatique, l'internet et les télécommunications, représentés schématiquement sur la figure 3. Les chercheurs de l'UMONS actifs dans ce domaine se sont associés dans l'institut de recherche **Infortech**.

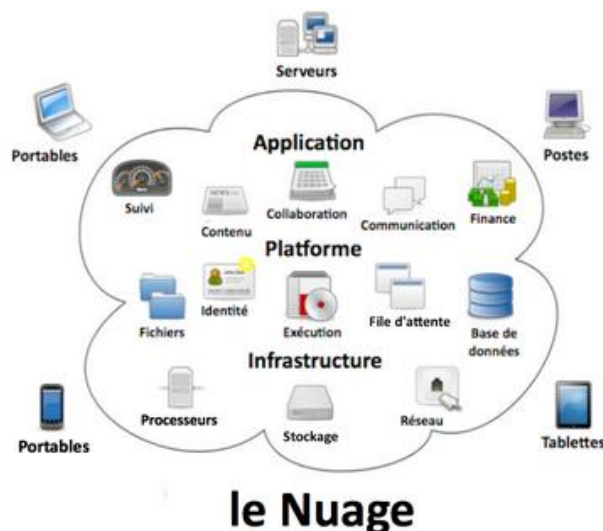


Figure 3 - Aperçu schématique des grandes composantes des TICs, wikipedia.fr

L'institut **Infortech** est dédié aux technologies de l'information et sciences de l'informatique et plus spécifiquement aux technologies permettant de détecter, formater, transmettre, stocker, analyser et exploiter les données et signaux, et cela, quel que soit leur volume ou leur mode de collecte.

Il est composé de près de **120 chercheurs** dont **35 doctorants**. Il maintient des relations étroites avec les centres de recherche agréés de la Région wallonne actifs dans le domaine des technologies de l'information, qui sont MULTITEL et le CETIC.

1. Activités d'enseignement

D'une part, le service assure des activités d'enseignements dans des disciplines en informatique telles que **l'algorithmique**, l'apprentissage des **langages informatiques** comme le C++, Java, Python ainsi que des langages et Framework du développement web et mobile.

D'autre part, il assure l'enseignement de disciplines plus spécifiques comme l'IA, le Machine Learning, la gestion de bases de données, la cybersécurité, le génie logiciel ainsi que la gestion de projets. Le service est également impliqué dans la formation et certificat IA « **Hands on AI** » à l'UMONS.

2. Activités de recherches

Le service **ILIA** (Informatique Logiciel et Intelligence Artificielle) est actif dans le domaine de la **Data Intelligence** et du **traitement automatique** des données. Il détient une expertise dans les domaines de **l'IA**, du **Cloud** et **Edge Computing**, du **Big Data**, de la **vision par ordinateur**, de la gestion des données dans **l'Internet des objets** (IoT) et du **traitement d'images**.

Les membres du service ILIA sont impliqués dans de nombreuses recherches théoriques et appliqués dans leur domaine d'expertises. Ils entretiennent plusieurs collaborations nationales et internationales, et leurs travaux sont reconnus internationalement.

II. L'objectif de la mission : problèmes et enjeux globaux

Alors que la population mondiale augmente et vieillit, le modèle médical traditionnel a atteint ses limites. En effet, obtenir des rendez-vous médicaux devient difficile, et le coût des consultations et des traitements est élevé.

Pour faire face à ces problématiques, le développement de **l'Internet des Objets de Santé** (HIoT) a permis de proposer une solution dans le domaine du télédiagnostic, du développement des consultations de télémédecine et du suivi des malades chroniques et des personnes âgées. Ces nouveaux services sont basés sur **l'Internet des objets médicaux** (IoMT).

A. Présentation du projet

Le service ILIA travaille sur une **nouvelle architecture** en temps réel de l'IoMT pour le suivi des patients âgés. Cette nouvelle architecture s'intitule **"RAMi: a new Real-time internet of medical things Architecture for elderly patients Monitoring"**.

Durant ce projet, les membres ont rédigé une **publication** concernant cette nouvelle architecture en temps réel de l'IoMT.

Elle contient un système d'alerte précoce utilisant l'analyse en temps réel des données au moyen d'algorithmes de Machine Learning (ML) et un système d'annotation pour marquer spécifiquement l'alerte individuelle du patient afin d'éviter les fausses alertes positives.

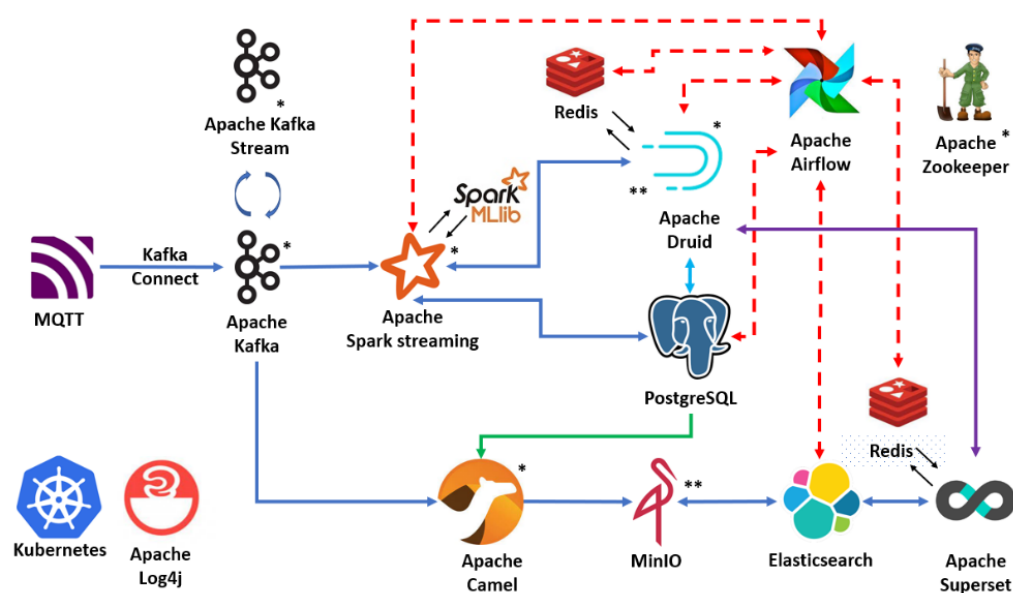


Figure 4 - Partie Cloud de l'architecture RAMi

Sur la figure 4, on peut voir que cette architecture est construite autour de plusieurs composants logiciels :

- Apache **Kafka**, une file d'attente de messages largement utilisée pour stocker temporairement des données avant ou après leur traitement
- Apache **Spark Streaming** est mis en œuvre pour traiter les flux de données, et la bibliothèque **MLlib** réalise l'analyse par ML
- Apache **Camel** est un cadre d'intégration qui permet d'intégrer divers systèmes consommant ou produisant des données
- **MinIO** est un système de stockage d'objets haute performance
- **ElasticSearch** est un logiciel utilisant Apache **Lucene** pour indexer et rechercher des données
- Apache **Airflow** est une plateforme permettant de créer, de planifier et de surveiller des flux de travail
- Apache **Super-set** est une plateforme d'exploration et de visualisation de données
- **PostgreSQL** est un système de gestion de bases de données (SGBD) objet-relationnel puissant et open-source
- Apache **Druid** est une base de données en temps réel destinée à alimenter les applications analytiques modernes
- **Redis** (redis.io) est un SGBD de stockage clé-valeur en mémoire open source
- Apache **Zookeeper** est un service centralisé pour maintenir les informations de configuration, le nommage, la synchronisation distribuée et les services de groupe
- Apache **Log4j** est un utilitaire de journalisation pour surveiller les incidents aux points clé de l'architecture
- **Kubernetes** utilisé pour les déploiements des composants des architectures

On a pu voir un aperçu de l'architecture dans laquelle certains liens avec Apache Zookeeper et MinIO sont respectivement remplacés par des astérisques simples et doubles pour améliorer la lisibilité.

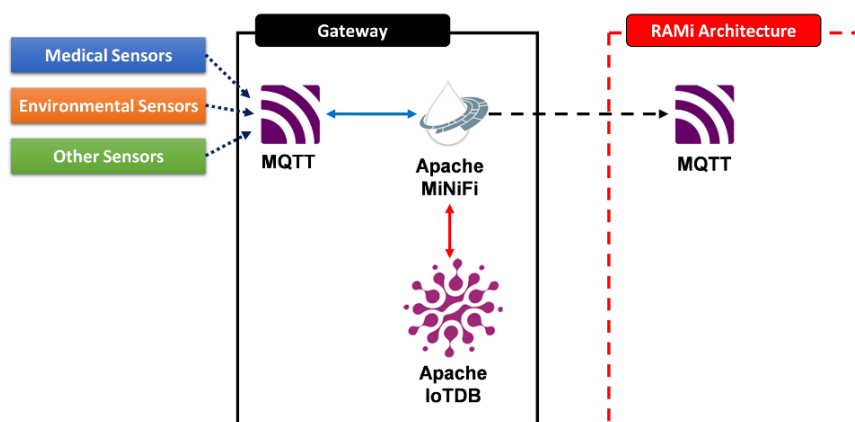


Figure 5 - Partie Fog de l'architecture RAMi

Du côté Fog, au niveau de la Gateway, les dispositifs envoient leurs données au serveur MQTT de la Gateway où les données sont traitées par Apache **Minifi** (une version légère d'Apache **Nifi** dédiée aux dispositifs contraints). Ils ont aussi implémenté Apache **IoTDB** pour stocker les données temporaires et locales, visible sur la figure 5.

B. L'objectif du stage

Afin de pouvoir visualiser et analyser ces résultats, l'objectif de ce stage était de réaliser une application **responsive** présentant un **tableau de bord médical**. Il est destiné à consulter et analyser les données provenant de différents objets connectés, notamment médicaux (IoTM), et cela, en utilisant différentes plateformes de collecte de données. On pourra par exemple visualiser le rythme cardiaque d'un patient, à l'aide des données issues des capteurs.

L'application devait être **multiplateforme, c'est-à-dire** qu'elle soit compatible avec le format Mobile (Android et iOS), Web et Desktop. Le choix de cette caractéristique justifie pour que les utilisateurs puissent y avoir accès sur n'importe quel appareil. De plus, avec une seule source de code, on peut créer une application compatible pour tout type de plateforme. Cela fait donc gagner beaucoup de temps et c'est moins coûteux.

L'interface devait aussi être **user-friendly**, autrement dit ergonomique et simple d'utilisation, pour pouvoir être manipulé par un large public n'ayant pas de connaissance en informatique.

C. Les limites et enjeux.

Une des limites du projet étant que les applications multiplateformes, autrement appelées **hybrides**, ont une contrainte. Elles ne sont pas aussi performantes que les **applications natives**. Les applications Android sont développées majoritairement avec Java ou Kotlin, tandis que pour iOS, on utilise Swift et Obj-C. Toutes ces applications natives sont conçues dans l'objectif d'avoir une performance très élevée. Pour remédier à cela, les communautés de développement tentent de mettre au point des outils qui suppriment la distinction entre les ces deux types d'applications.

Puis, une réflexion devait également être menée quant à la gestion et au **stockage des données** afin de ne pas nuire au caractère mobile de l'application d'une part, et conserver des temps d'analyse acceptables, d'autre part. La problématique qui s'est posé était de réfléchir à quelles données devaient être stockées en local, et lesquels devaient être stockés de manière distante, sur un cloud, ou serveur, afin de ne pas surcharger l'application.

Un autre enjeu à respecter était l'aspect **temps réel** des données. Dans le médical, il est important d'obtenir des informations à la seconde. Il faut donc tenir compte que le nombre de requêtes réalisé sur la base de données, ainsi que la quantité de stockage sera assez importante.

III. Réalisation de la mission

En 2021, le nombre d'objets connectés était de **12,3 milliards** en France, selon le cabinet d'études de marché américain IoT Analytics. L'internet des objets (IoT) est un secteur technologique innovant qui sera de plus en plus sollicité à l'avenir, notamment avec le développement de la **5G**. Il désigne tous les objets ou lieux qui nous entourent et qui ont la capacité de communiquer entre eux en temps réel (Wifi, Bluetooth...). Il peut s'agir par exemple de véhicules autonomes, de montre connectée, villes intelligentes...

L'objectif de ma mission était de réaliser une application de visualisation et d'analyse de données IoT sous forme de tableau de bord. Puisque le projet n'avait pas encore été commencé, je suis parti d'un code « from scratch » ³, en allant du système d'acquisition des données jusqu'à l'interface utilisateur.

Afin d'avoir une vue d'ensemble de la chaîne d'acquisition des données, j'ai pu me familiariser aux architectures logicielles de traitement de données IoT et au protocole de transmission de données **LoRaWan**, à l'aide d'un TP sur **Node-RED**.

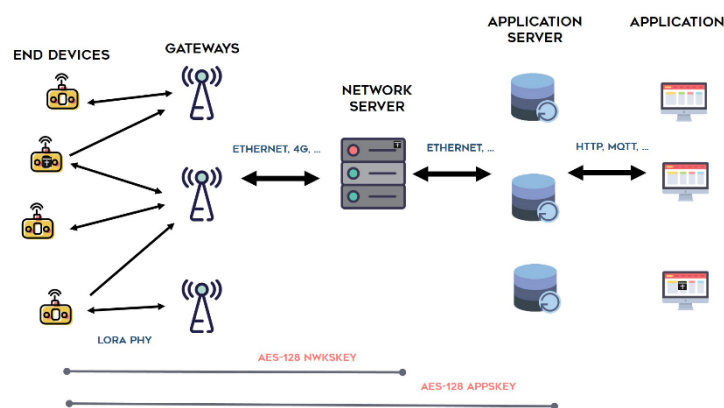


Figure 6 – Architecture d'un réseau LoraWAN, frugalprototype.com

Comme on peut le voir sur la figure 6, le réseau LoRaWan est utilisé pour transmettre des données de faibles quantités, mais avec une grande portée (10 à 30km). Ces données sont réceptionnées par les **Gateway** (passerelles) LoraWan, et ensuite transmises aux serveurs applicatifs qui en font l'exploitation.

Node-RED est un outil de développement basé sur le flux pour la programmation visuelle, pour connecter ensemble des dispositifs matériels, des API et des services en ligne.

À travers ce TP, j'ai pu voir comment on pouvait traiter les données provenant d'un appareil programmable : **Lopy**, muni d'une carte d'extension **pysense**, qui concentre plusieurs capteurs, et réaliser un tableau de bord à partir de ces données.

³ Créer quelque chose en partant de rien

Pour mener à bien le développement de l'application, il a fallu réaliser une analyse des briques logicielles nécessaires et les plus adaptées à la confection de ce tableau de bord. D'une part, le choix et la mise en place de l'architecture de stockage IoT, autrement dit la gestion et le stockage des données IoT, d'autres part, la réalisation de l'interface utilisateur.

A. Récupération des données via le protocole MQTT

Dans notre cas, le protocole de communication est le protocole MQTT.

1. Description du protocole MQTT



MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie basé sur TCP/IP, développé par IBM en 1999. En raison de sa légèreté et sa simplicité, il est devenu un des piliers de l'IoT. Ces caractéristiques représentent des atouts puisque les équipements IoT sont souvent limités en matière de puissance, de consommation d'énergie et de bande passante.

MQTT un protocole de type publish/subscribe composé de 2 entités :

- Le **broker** qui reçoit et distribue les messages
- Les **clients** qui émettent et/ou reçoivent les messages

Les clients peuvent publier les informations sur un canal appelé **topic** : on les appelle les **publieurs**. Ils peuvent également souscrire à des topics afin de lire les messages : on parle de **souscripteurs** ou d'**abonnés**.

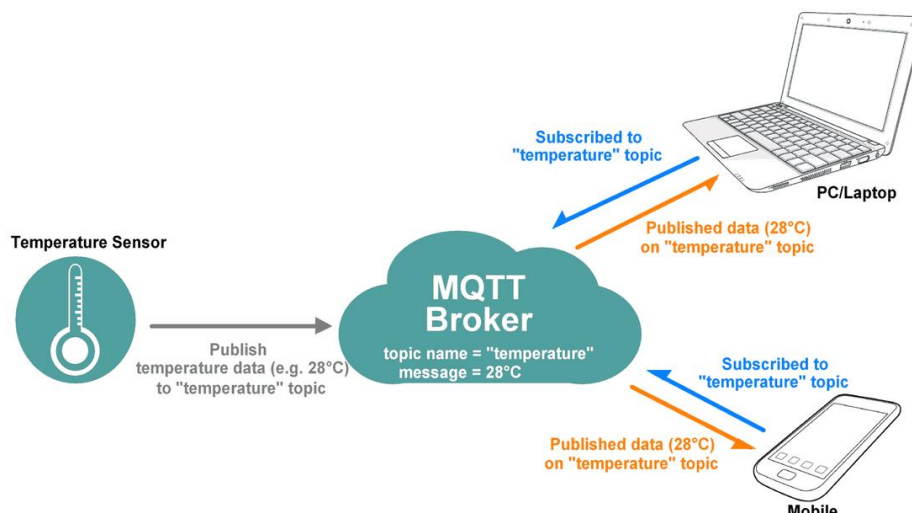


Figure 7 - Schéma présentant le protocole MQTT, arduino.blaisepascal.fr

Sur la figure 7, on peut voir un exemple de protocole MQTT. On dispose d'un capteur de température qui publie ses données sur un broker, via un topic nommé « temperature ». Le

téléphone mobile, ainsi que le PC se souscrivent au topic « temperature » du broker. Ce dernier va alors publier les messages de ce topic sur ces deux appareils.

La qualité de service (QoS) est également un point particulièrement important. Il se compose en 3 niveaux :

- Niveau 0 (At most once) : le message sera délivré tout au plus une fois, c'est-à-dire sans garantie de réception
- Niveau 1 (At least once) : le message sera envoyé au moins une fois jusqu'à que le broker lui confirme l'envoi du message sur le réseau
- Niveau 2 (Exactly once) : le message sera obligatoirement sauvegardé jusqu'à sa réception par le souscripteur.

La connexion MQTT peut être sécurisée au niveau transport en utilisant SSL/TLS, au niveau authentification avec le certificat SSL/TLS ou par couple utilisateur/mot de passe.

2. Système acquisition des données côté publication

Dans notre cas, certains membres du projet ont mis à disposition un broker avec quatre topics. Les publieurs sont les différents capteurs tels que les capteurs de température, d'humidité relative, l'indice de chaleur (heat index) et le rythme cardiaque.

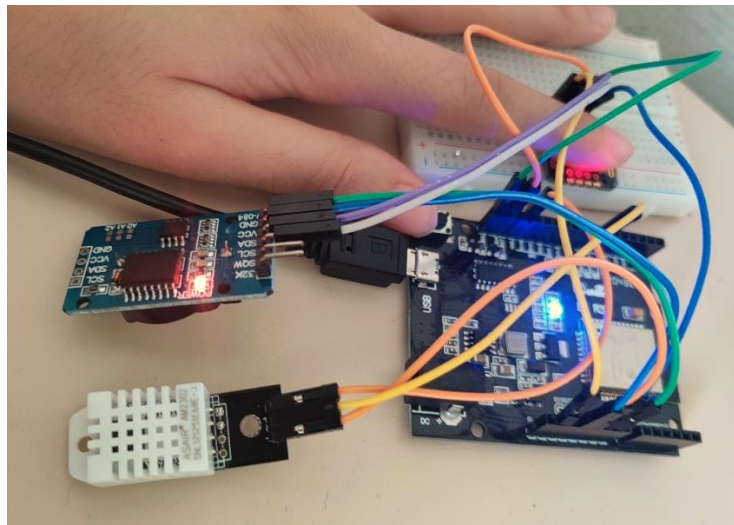


Figure 8 - Connexion des capteurs avec le microcontrôleur Wemos D1 R3

Sur la figure 8, on peut voir la disposition des capteurs avec le microcontrôleur. Le capteur blanc, **DHT22**, est un capteur de température (°C) et humidité relative (%). Le Heat Index Coefficient (°C) est calculé à partir de ces deux valeurs. Il s'agit de l'indice de chaleur ou encore la température ressentie. Le capteur bleu est un module de mémoire d'horloge, **RTC DS3231**, en temps réel. Enfin, le capteur sur lequel je pose le doigt est un **MAX30102**, qui est un module capteur d'oxygène et de rythme cardiaque.

Les messages envoyés vers le broker sont de la forme : « **{timestamp : value}** », où timestamp est le temps linux à laquelle la donnée est envoyée, et value, la valeur de la donnée à cet instant. Pour chaque grandeur, on envoie les messages sur un topic spécifié. Pour la température, on envoie sur le topic « rami/tmp », pour l'humidité relative sur « rami/hr », pour l'indice de chaleur sur « rami/hic » et enfin pour le rythme cardiaque sur « rami/beat ».

Les messages des topics « rami/hr », « rami/tmp », « rami/hic » sont envoyés à des intervalles de temps assez court, qui tourne autour de 2 secondes. Alors que les messages du topic « rami/beat » ne s'envoient que lors de la pose d'un doigt sur le capteur. De plus, ses messages sont envoyés dans un intervalle encore plus court.

La qualité de service utilisée dans l'application est de **niveau 0** puisqu'on ne nécessite pas d'accusé de réception et les messages doivent circuler en temps réel. De plus, les données sont transmises par Wifi à la Gateway avec le protocole MQTT avec le protocole **TLS**. Il s'agit d'un protocole de sécurité qui assure la confidentialité et l'intégrité des données pour les communications Internet.

3. Système acquisition des données côté souscription

On peut se connecter au broker souhaité et souscrire à différents topics, en saisissant un nom de domaine (alias), l'identifiant du broker (host), les topics (sujets). Comme on peut le voir sur la figure 9, on peut rentrer ces diverses informations depuis l'application développée. L'appareil sur lequel on a lancé l'application devient alors un souscripteur.

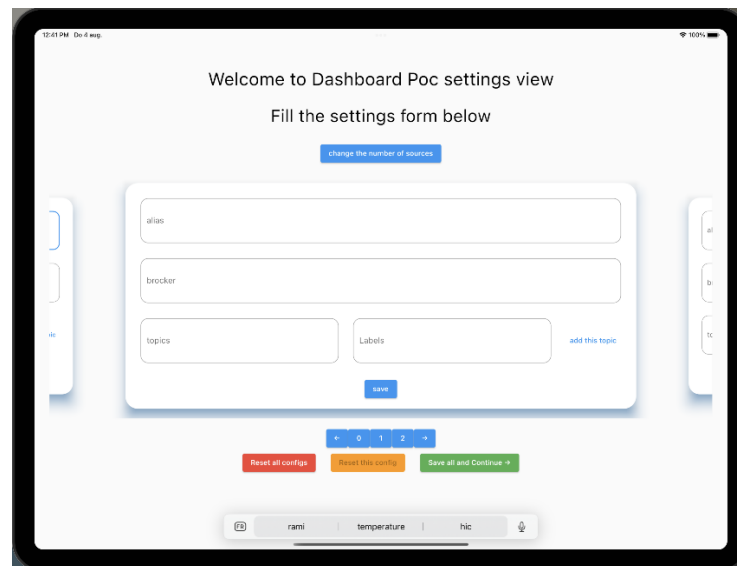


Figure 9 - Formulaire de saisie d'une source MQTT

Pour le champ topics, on a ajouté un champ nommé « labels » qui permet d'identifier la grandeur des données. Par exemple, s'il s'agit de « rami/hr », on indiquera pour label « humidité relative ». Cela permettra de bien identifier la nature des données lors de l'utilisation de l'application.

Côté sécurité, les souscripteurs ne peuvent envoyer des messages sur le topic, ils ne peuvent que recevoir. On évite ainsi des attaques de sécurité par injection de code.

B. Développement côté serveur : traitement des données

1. Choix de la base de données

Puisque l'intervalle de réception des données est d'environ 2 secondes, il était primordial de stocker toutes ces données dans un cloud, afin de ne pas surcharger l'application. Étant donnée de l'aspect temps réel, se tourner vers une base de données **NoSQL** semblait plus judicieux.

Ainsi, la plateforme **Backend-as-a-Service** (BaaS), **Firebase** (Cloud Firestore) s'est avérée être un moyen très pertinent et simple à mettre en place. Elle fournit des services backend hébergés tels qu'une base de données en temps réel, un stockage dans le cloud, une authentification, une configuration à distance, un hébergement pour les fichiers statiques...

Sur Cloud Firestore, les données sont stockées dans des collections. Pour chaque collection, on peut avoir plusieurs documents. Chaque document stocke les données sous forme de map.

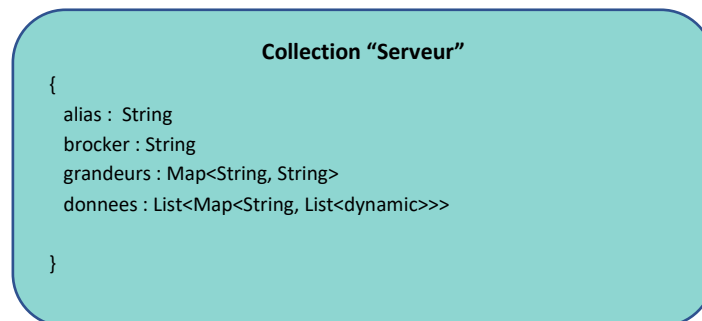


Figure 10 - Schéma de la collection serveur

Sur la figure 10, on peut voir la structure de la collection « Serveur ». On y stocke les champs rentrés dans le formulaire, soit l'alias, le broker, les grandeurs et les données.

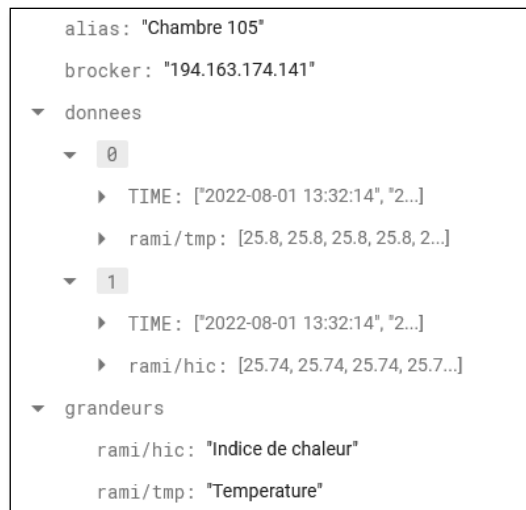


Figure 11 - Exemple d'une instance de la collection "Serveur"

Sur la figure 11, on peut visualiser une instance de la collection « Serveur ». Le choix de la structure de donnée a été réalisé pour différencier facilement les données pour chaque topic (temps et valeur). Tout a été regroupé dans un seul document pour qu'on puisse économiser les requêtes de lectures et écritures. En effet, sur Cloud Firestore, il y a une limitation en termes de requêtes et de stockage par jour. Au-delà de ce quota, le service devient payant.

2. Architecture de traitement des données

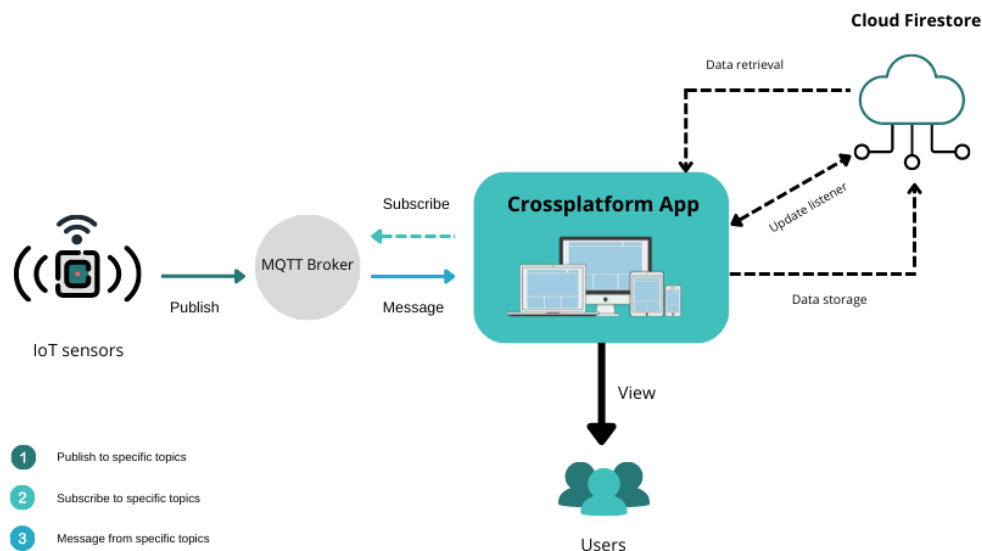


Figure 12 - Architecture de l'application côté serveur

La figure 12 résume l'architecture de l'application côté serveur. Tout d'abord, comme expliqué précédemment les données issues des capteurs IoT sont publiées via un protocole

3. Conception



14


Pour le côté serveur, la classe MQTTManager se charge d'établir la connexion, la déconnexion et la réception des messages d'un broker MQTT. La classe MQTTAppState contient l'état actuel de la connexion MQTT. De plus, les classes Serveur, Users modélisent respectivement les serveurs MQTT et les utilisateurs. Enfin, les classes DatabaseService et AuthService contiennent toutes les requêtes vers Cloud Firestore et Firebase Authentication.

4. Authentification des utilisateurs

Étant donné du choix de Firebase, et ayant les configurations déjà faite pour cette plateforme, la gestion des utilisateurs a pu être réalisé avec le service **Firebase Authentication**. Il fournit des services backend et des bibliothèques d'interface utilisateur prêtes à l'emploi pour authentifier les utilisateurs d'une application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros de téléphone, de fournisseurs d'identité fédérés populaires comme Google, Facebook et Twitter, et plus encore.

Nous avons pu réaliser l'inscription, l'authentification, et la déconnexion des utilisateurs d'une manière très simple. Une vérification de l'adresse mail a aussi été mise en place lors de l'inscription et donc l'authentification est bloquée pour les utilisateurs ne l'ayant pas faite.

Sur la figure 14, on peut voir la manière dont sont stockés les utilisateurs sur Firebase Authentication.



The screenshot shows the Firebase Authentication console interface. At the top, there is a search bar with the placeholder text 'Recherchez par adresse e-mail, numéro de téléphone ou ID utilisateur' and a button 'Ajouter un utilisateur'. Below the search bar is a table with the following columns: 'Identifiant', 'Fournisseurs', 'Date de création', 'Dernière connexion', and 'UID utilisateur'. The table contains four rows of user data. At the bottom right of the table, there is a pagination control showing 'Lignes par page : 50' and '1 - 4 of 4'.

Identifiant	Fournisseurs	Date de création	Dernière connexion	UID utilisateur
thierry.noundou.isbm@gmail...	📧	28 juil. 2022	28 juil. 2022	5F4ZFkZBYFPwOohCuxCt6pqlTw23
dounia913@gmail.com	📧	27 juil. 2022	27 juil. 2022	f855fT3QrpX0MVG3Rrkjs89YTx22
dounia.messaoudi9@gmail...	📧	27 juil. 2022	5 août 2022	bCqPFb4TMSMc3NWfo1NeYle...
thierrynjike14@gmail.com	📧	27 juil. 2022	5 août 2022	a1m8diS7FkUq017RSp5KdFJzq1

Figure 14 - Utilisateurs stockés sur Firebase Authentication

C. Développement côté client : interface utilisateur

1. Choix technologiques

Parmi les choix qui se présentaient à moi, il y avait diverses possibilités pour réaliser une application multiplateforme : les PWA, React Native, Flutter ou encore Xamarin ou Ionic.

Une étape de comparaison de ces technologies a donc été nécessaire. Pour résumer cette étude, un tableau comparatif a été réalisé (cf. annexe 4). En parallèle, j'ai commencé une

formation Flutter sur UDEMY « Flutter & Dart - The Complete Guide ⁴ », pour explorer aux mieux ses avantages et inconvénients. J'ai également suivi les TP du cours de Bac+3 « **Applications mobiles d'exploitation de contenus** » qui concernait les PWA, React Native et Flutter.

Le choix final s'est porté sur le **SDK Flutter** (langage Dart) pour sa performance, rapidité et pour son interface utilisateur très personnalisable.

2. Conception UI/UX

L'interface finale de l'application a été réfléchi pour viser un public large et s'adapte à toute taille d'écran. Le choix du design a été réalisé de manière simple et épuré, afin de rendre l'application plus ergonomique.

Sur la figure 15, on peut voir le menu Dashboard de l'application qui contient un tableau de bord avec des graphes en temps réel présentant les données issues de différents capteurs. On retrouve des **diagrammes linéaires** qui permettent de visualiser l'historique des données, et des **diagrammes de jauge** pour indiquer la valeur des capteurs en temps réel. On peut aussi observer la valeur moyenne des capteurs.

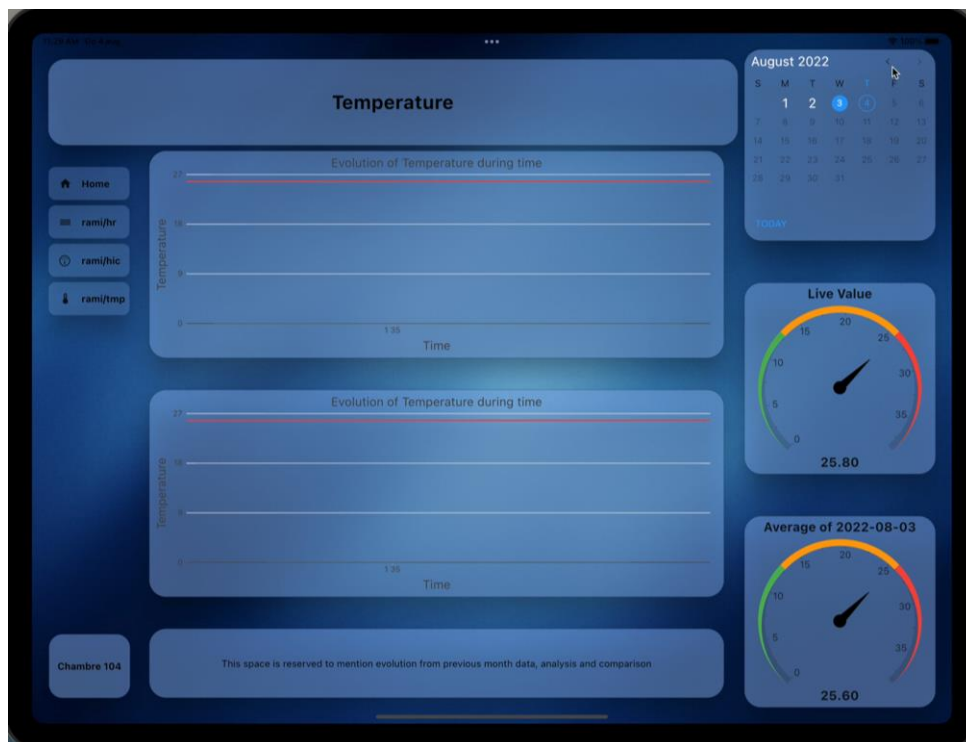


Figure 15 - Capture d'écran du menu Dashboard réalisé

⁴ (Flutter & Dart - The Complete Guide [2022 Edition], s.d.)

L'utilisateur bénéficie aussi d'un agenda interactif qui lui permet de visualiser l'historique des capteurs selon la date qu'il choisit de sélectionner. Tous ces graphes ont été réalisés à l'aide des packages `charts_flutter` et `syncfusion_flutter`.

D. Gestion de projet

Durant ce stage, j'ai travaillé de manière autonome et organisé. En fonction des priorités des tâches et leur complexité, j'organisais mes journées avec un planning Excel et l'outil de gestion de projet, **Trello**. À partir de ce planning, un diagramme de **Gantt** a été réalisé (cf. annexe 2). Une partie du stage a été réalisée à l'aide de la méthode agile dénommée **Scrum**, qui est la plus populaire à ce jour. Elle est centrée sur le produit, le besoin du client et des utilisateurs. Elle permet d'optimiser la productivité en privilégiant des livraisons rapides.

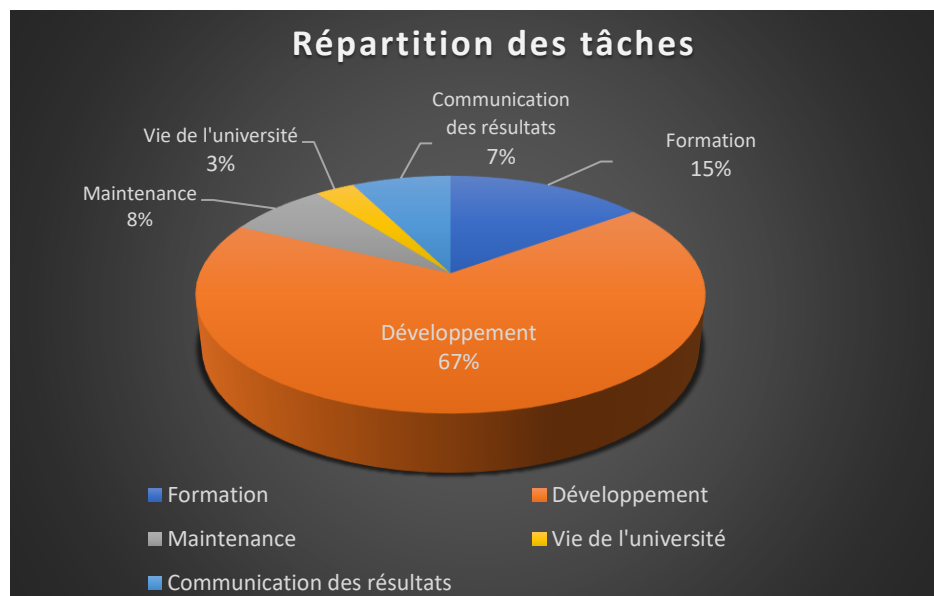


Figure 16 - Répartitions des tâches durant le stage

Comme on peut le voir sur la figure 16, la majorité de mon temps a été consacré au développement de l'application. Mais avant cela, il y a d'abord eu une étape de formation et choix technologiques. Il y a aussi une phase de tests, d'analyse et de maintenance qui a été faite, dans le but de faire progresser l'application. J'ai aussi eu l'opportunité de découvrir le monde universitaire en Belgique, en assistant à deux thèses, à des séminaires, des TFE (Travaux de Fin d'Études) et les présentations du cours d'applications mobiles. De plus, la communication des résultats, à savoir les réunions et la rédaction du rapport de stage ont aussi pris une part non négligeable.

E. Bilan et perspectives

Le développement de l'application a bien été achevé avec des améliorations réalisées, tel que la portabilité et l'extensibilité, dû à une bonne gestion de temps du projet.

1. Les fonctionnalités réalisées

Concernant les **fonctionnalités** réalisées, depuis l'application, on est capable de se connecter à une source MQTT en connaissant le broker et le/les topics voulu. On peut s'inscrire, s'authentifier, se déconnecter de l'application. Une architecture de traitement des données a été mise en place pour récupérer les données depuis le système d'acquisition pour les stocker dans un cloud.

L'application dispose d'une interface utilisateur **responsive** et simple d'utilisation, destinée pour un public large. Elle dispose d'un tableau de bord qui affiche les données issues du cloud en temps réel, sous forme de graphiques. L'application est compatible avec le format Android et iOS.

Le code a été organisé en fichiers et documenté dans un but de **réutilisabilité** du code. La portabilité et extensibilité de l'application sont garanties pour tout MQTT disposant de message de type (timestamps : value). Un processus de gestion de version a été réalisé à l'aide de git.

2. Apports pour le service

L'application développée durant mon stage servira, d'une part, de tableau de bord médical, dans le cadre du projet RAMi, et d'autre part, de démonstrateur pour les différents projets mis en œuvre au sein de service ILIA. Elle sera ensuite reprise et réadaptée par les membres du service, en fonction de leurs besoins. D'où l'importance accordée à la maintenance de l'application et à la communication des résultats.

Dans cette optique, j'ai rédigé un manuel d'installation et d'utilisation, afin de laisser la possibilité aux membres du service de reprendre mon travail et de l'étendre à des nouveaux besoins (cf. annexe 5). La documentation du code et l'ajout de commentaires étaient tout aussi important. Pour la maintenance de l'application, la garantie de la qualité logicielle a été assurée par le logiciel de qualimétrie **SonarQube**, dont les résultats sont présentés sur la figure 17.

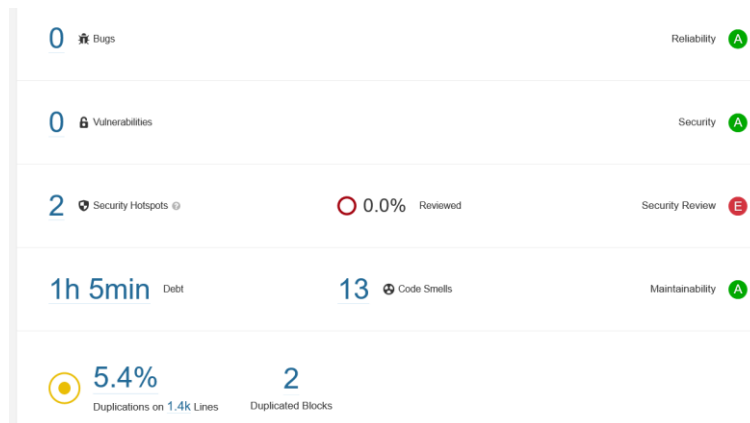


Figure 17 - Résultats de l'analyse de la qualité logiciel avec SonarQube

3. Limites et perspectives

Cependant, comme toute application, elle dispose de certaines limites et de futures améliorations possibles. En effet, **la solution Firebase** devient payante lorsque l'on dépasse une certaine quantité de stockage et un certain quota de requêtes par jour. Avoir plusieurs utilisateurs connectés sur l'application ne ferait qu'accroître le problème. Pour remédier à cela, on pourrait réfléchir à un changement de la structure de la collection sur le cloud, ou réfléchir à une alternative à Firebase, non payante.

Afin d'améliorer l'application, d'autres **perspectives** sont envisageables. Tout d'abord, on pourrait configurer le projet pour permettre la compatibilité Web et Desktop. L'ajout d'un mode hors-ligne serait également intéressant, ainsi qu'une réflexion sur la suppression des données obsolètes. En effet, il n'est pas nécessaire de garder en stock toutes les données dans le cloud. Par exemple, on pourrait supprimer les données qui fluctuent très peu ou alors celles qui n'ont plus d'importance à compter d'une certaine date. Afin d'avoir une meilleure visibilité, on pourrait aussi permettre de zoomer sur les graphiques.

IV. Retour d'expérience

Tout d'abord, ce stage, réalisé à l'étranger, a été une expérience très enrichissante sur le plan humain comme sur le plan professionnel et m'a permis de développer mon savoir-faire et mon savoir-être.

A. Les apports personnels du stage

1. Apports sur le plan humain

Ce stage, durant lequel j'ai été amené à vivre à l'étranger, durant trois mois, m'a permis de devenir plus responsable, d'apprendre à gérer mes différentes dépenses et à m'adapter à un nouvel environnement. J'ai également eu l'occasion de découvrir **Mons** et la **Belgique** en général, plus précisément la Wallonie.

Au cours de ce stage réalisé en Belgique, au plein cœur du centre-ville de Mons, j'ai pu découvrir la ville et ses environs. Mons est une ville et commune de Wallonie, et la capitale de la province du Hainaut. Sur la figure 18, on peut voir sa localisation précise en Belgique.

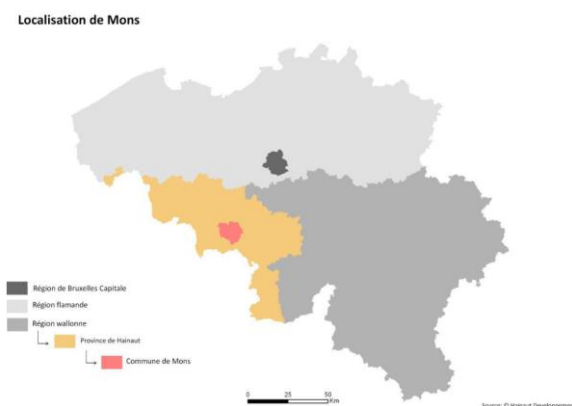


Figure 18 - Localisation de Mons, Hainaut Développement

2. Apports sur le plan professionnel

Au-delà de l'aspect personnel, j'ai aussi enrichi mes connaissances techniques et acquis de nouvelles compétences professionnelles.

Au cours de ma formation à Polytech Paris-Saclay, j'ai été amené à programmer avec divers langages, tels que le C++, Java, Python, JavaScript... Durant les projets de groupe, nous avons appris à concevoir et à développer des applications de bureau, des logiciels, des sites web...

Cependant, il me manquait une compétence clé, qui s'inscrit dans la continuité de ma formation : le **développement mobile**.

J'ai ainsi appris à développer des applications mobiles, en **Dart** et **Flutter**, ce qui a été quelque chose que j'ai vraiment apprécié. Les cours de complément objet, de Java, d'IHM, d'objets et services répartis ont été d'une très grande aide.

En complément de nos cours de SQL, j'ai pu me familiariser avec les bases de données **NoSQL**, en particulier **FireBase** (Cloud Firestore).

En plus des compétences techniques, j'ai acquis des **compétences organisationnelles**. En effet, durant mon stage, j'ai tenu un planning Excel quotidien, pour optimiser mon temps, et utilisé Trello pour la gestion de projet. J'ai également mis en place un diagramme de Gantt et pu expérimenter la méthode agile SCRUM. J'ai ainsi eu l'occasion de mettre en pratique les cours de gestion de projet et de management d'équipe.

Enfin, une prise de conscience de mes capacités a été très bénéfique puisque c'est ma seconde expérience professionnelle technique. Le fait de voir que l'on peut se rendre utile et apporter des solutions pour une réelle mission est vraiment gratifiant.

B. Difficultés rencontrées et solutions apportées

Tout d'abord, un des obstacles auquel j'ai dû faire face était l'apprentissage du langage « **Dart** » et du Framework « **Flutter** ». De plus, le développement mobile était quelque chose de nouveau pour moi. Pour accélérer le processus d'apprentissage, j'ai bénéficié de la formation UDEMY. Comme dit précédemment, j'ai également eu la chance d'assister à des TP de Polytech Mons, ce qui m'a permis de pratiquer et d'éclaircir certains points sur Dart et Flutter.

Par ailleurs, certaines nouveautés/particularités de **Dart** et **Flutter** n'ont pas été simple à gérer. En mars 2021, la version 2.0 de Flutter est sortie, accompagnée de la version 2.12 de Dart. Cela a apporté l'arrivée du "**null safety**", la plus grande mise à niveau de Dart depuis que la version 2.0 a introduit un système de type solide. Le principe de base est d'éliminer les erreurs de référence nulle au moment de l'exécution, et de les attraper directement au moment de la compilation. Cependant, la plupart des tutoriels disponibles sur internet, qui ont été réalisés à une période supérieure à un an, sont alors devenus obsolètes. Ainsi, la recherche de documentation n'était pas évidente.

De plus, les fonctionnalités ou packages de Flutter n'étaient pas compatibles avec toutes les plateformes comme le web et desktop. Cela restreignait donc le choix des extensions Flutter et allongeait la phase de tests sur les différents types de plateformes.

Une difficulté majeure concernait le choix et la mise en place de la **base de données**. En effet, n'ayant jamais implémenté de base de données pour un projet, c'était une première

expérience pour moi. Tout d'abord, il a fallu explorer les différentes bases de données disponibles, puis faire un choix selon le processus de stockage et de gestion des données qu'on voulait pour l'application. À la suite de ce choix, il fallait configurer le projet pour qu'il soit compatible avec Cloud Firestore.

Différents problèmes de compréhension ont été rencontrés lors de cette étape puisque les tutoriels disponibles étaient soit orientés Android, soit iOS, soit Web. Les tutoriels concernant Flutter n'étaient pas assez explicites. Ensuite, la mise en place des requêtes de lectures et d'écriture était assez complexe puisqu'il fallait gérer des types asynchrones, Future et Stream, que je n'avais pas l'habitude d'utiliser. Par ailleurs, la recherche de la structure de donnée la plus optimale a été un réel défi à relever.

Enfin, l'utilisation de l'IDE **Android Studio** était très lourde à supporter pour mon ordinateur personnel, étant donné qu'il prend énormément d'espace sur le disque. Ainsi, lors de la compilation, il y avait pas mal de latence et cela prenait du temps à s'exécuter. Pour faire face à cela, j'ai souvent compilé mon code, en physique, sur mon smartphone personnel.

C. Ouverture vers ma future carrière

Le domaine du **développement logiciel** est la branche de l'informatique qui me plaît le plus. Ce stage sera un atout pour ma carrière puisque j'aimerais orienter mon projet professionnel spécifiquement vers ce domaine-là. De plus, la maîtrise de **Flutter et Dart**, que j'ai pu développer, sera un réel avantage puisque c'est une compétence qui sera de plus en plus demandée par les entreprises, dans les années à venir.

À l'occasion de ma collaboration sur le projet RAMi, j'ai aussi appris davantage sur un domaine qui est **l'IoT**, et particulièrement **l'IoMT**, qui sont des branches de l'informatique qui pourrait potentiellement m'intéresser à l'avenir.

J'ai aussi découvert le secteur de la **recherche** et de **l'enseignement** en général et particulièrement en Belgique. J'ai assisté à deux thèses et à des soutenances d'étudiants de TFE (Travaux de fin d'études), ce qui a été très enrichissant et inspirant. Être en contact avec des chercheurs a été une source de motivation, puisque j'ai pu apprendre à leur côté, à innover et à concevoir de nouvelles solutions pour le monde de demain.

Au cours de ce stage, j'ai bénéficié de nombreux enseignements pour ma future carrière d'ingénieur.

Un développeur doit savoir s'adapter à n'importe quelle technologie. En effet, l'informatique est partout et son importance est de plus en plus présente. Les technologies évoluent rapidement, il faut savoir être **flexible** et faire de la vieille technologique. De plus, le fait de vouloir programmer avec un langage ou Framework, selon son expérience, ne sera pas forcément le meilleur choix pour accroître les performances d'une entreprise.

En informatique, il est rare de trouver la solution du premier coup. Il est nécessaire de persévérer et de chercher divers moyens pour arriver à l'objectif souhaité. Les compétences comme la recherche de solutions sont très importantes pour progresser dans ce domaine.

Enfin, j'ai aussi eu l'occasion de développer mon **réseau** à l'international, ce qui a été très enrichissant d'un point de vue personnel et professionnel.

Conclusion

J'ai réalisé mon stage de 4e année d'école d'ingénieur en spécialité informatique, au sein du service ILIA de la **Faculté Polytechnique de Mons**. Cette expérience réalisée à l'étranger, a été très enrichissante du côté professionnel mais aussi personnel. Elle m'a permis de passer de la théorie au concret, de développer mes compétences techniques et d'en ressortir plus compétente.

Tout d'abord, l'objectif de ma mission était de développer une **application** responsive et multiplateforme, de visualisation et d'analyse de données issues de différents objets connectés, sous forme de tableau de bord.

Le projet a été réalisé dans son entièreté, depuis le système d'acquisition des données jusqu'à l'interface utilisateur. D'une part, cette application servira de tableau de bord médical, dans le cadre du projet **RAMi**. D'autre part, elle pourra être utilisée en tant que démonstrateur des différents projets réalisés au sein du service. Quelques perspectives d'amélioration sont envisageables et le code a été réalisé de manière à être réutilisable.

Cette expérience m'en a appris sur la **flexibilité** que doit posséder un développeur. Il doit toujours être au courant des nouvelles technologies et pouvoir s'adapter rapidement. Les différents cours que j'ai eus au sein de Polytech m'ont permis de m'adapter au changement assez rapidement.

Aimant beaucoup découvrir les nouvelles avancées technologiques et relever de nouveaux défis, cela m'a réellement conforté dans le fait d'avoir choisi l'ingénierie informatique afin de pouvoir m'épanouir pleinement dans ma vie professionnelle.

Enfin, j'ai pu développer mes compétences en techniques, humaines et organisationnelles. Par la même occasion, j'ai pu mettre en perspective mes ambitions et les compétences techniques que je dois encore développer.

Lexique

H

HIoT	
L'Internet des objets de la santé	4, 1

I

ILIA	
Informatique Logiciel et Intelligence Artificielle	4
IoMT	
l'Internet des objets médicaux	4, 1
IoT	
Internet des objets	5

T

TICs	
Technologies de l'Information et de la Communication	4

U

UMONS	
Université de Mons	2

V. Références

(2022, Juillet 4). Récupéré sur Syncfusion documentation:

<https://help.syncfusion.com/flutter/introduction/overview>

charts_flutter 0.12.0. (2022, Mai 10). Récupéré sur pub.dev:

https://pub.dev/packages/charts_flutter

Développement mobile : Microsoft met à la retraite Xamarin. (2022, Mai 10). Récupéré sur

Le Monde Informatique: <https://www.lemondeinformatique.fr/actualites/lire-developpement-mobile-microsoft-met-a-la-retraite-xamarin-83681.html>

Firebase documentation. (2022, Juin 3). Récupéré sur Firebase:

<https://firebase.google.com/docs/guides>

Flutter & Dart - The Complete Guide [2022 Edition]. (s.d.). Récupéré sur Udemy:

<https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/>

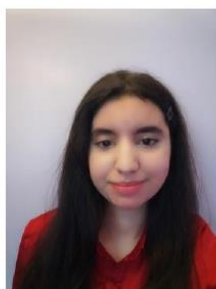
Overview of Progressive Web Apps (PWAs). (2022, Juin 6). Récupéré sur Microsoft:

<https://docs.microsoft.com/en-us/microsoft-edge/progressive-web-apps-chromium/>

Annexes

Annexe 1 : CV	28
Annexe 2 : Diagramme de Gantt.....	29
Annexe 3 : Liste des figures et tableaux	30
Annexe 4 : Tableau comparatif des technologies.....	31
Annexe 5 : Manuel d'utilisation.....	33

Annexe 1 : CV



Dounia MESSAOUDI

CONTACT



06 51 91 28 56



Massy, 91300



dounia.messaoudi@u-psud.fr



[Linkedin.com/in/dounia-messaoudi/](https://www.linkedin.com/in/dounia-messaoudi/)

CENTRES D'INTERET

- Membre du club d'algorithmique et du club DDRS de Polytech Paris-Saclay
- Pratique du basket-ball et du volley-ball

LANGUES

Français	Projet Voltaire : 839/1000
Anglais	B1
Espagnol	A2
Arabe	A1

Etudiante ingénieure en 4^e année d'informatique

Recherche d'un stage en informatique à partir du 2 mai (3 mois)

PROJETS

Application « OBIS3D »

[Github.com/gdescomps/OBIS3D](https://github.com/gdescomps/OBIS3D)

- Modéliser une application de visualisation 3D en UML
- Développer la partie applicative et l'interface utilisateur avec Java et JavaFX
- Equipe de 3

Projet « PolyHost »

- Développer une interface logicielle pour un robot d'accueil avec Qt
- Échanger régulièrement avec le client
- Equipe de 6



EXPERIENCE PROFESSIONNELLE

2021 (1 mois) | Stage ouvrier – technicienne informatique | Gendarmerie des Transports Aériens de Paris-Orly, Athis Mons

- Développer et tester des formulaires de saisies et de recherche pour le wiki
- Former un gendarme à la création de formulaires sur MediaWiki

2020 (4 mois) | Organisatrice | Forum Entreprises BAC+5, Orsay

- Démarcher des entreprises
- Créer des publications sur les réseaux sociaux

2019 (7 semaines) | Stage ouvrier - archiviste | CEA Paris-Saclay, Gif-sur-Yvette

- Préparer des notices bibliographiques
- Enrichir la base des thèses et la base INIS : <https://inis.iaea.org/search/>

FORMATION SCOLAIRE



2020 – Ingénierie informatique – Polytech Paris-Saclay, Orsay

- Enseignement de spécialité : base de données, réseaux, C++ et Java, algorithmique, IHM, POO, compilation
- Culture de l'ingénieur : mathématiques, physique, communication, management, marketing, économie

2018 - 2020 – Classe préparatoire intégrée – Polytech Paris-Saclay, Orsay

- Formation scientifique fondamentale : mathématiques, physique, informatique

2018 – Baccalauréat scientifique avec mention – Lycée Fustel de Coulanges, Massy

COMPETENCES

C++



Java



Python



JavaScript



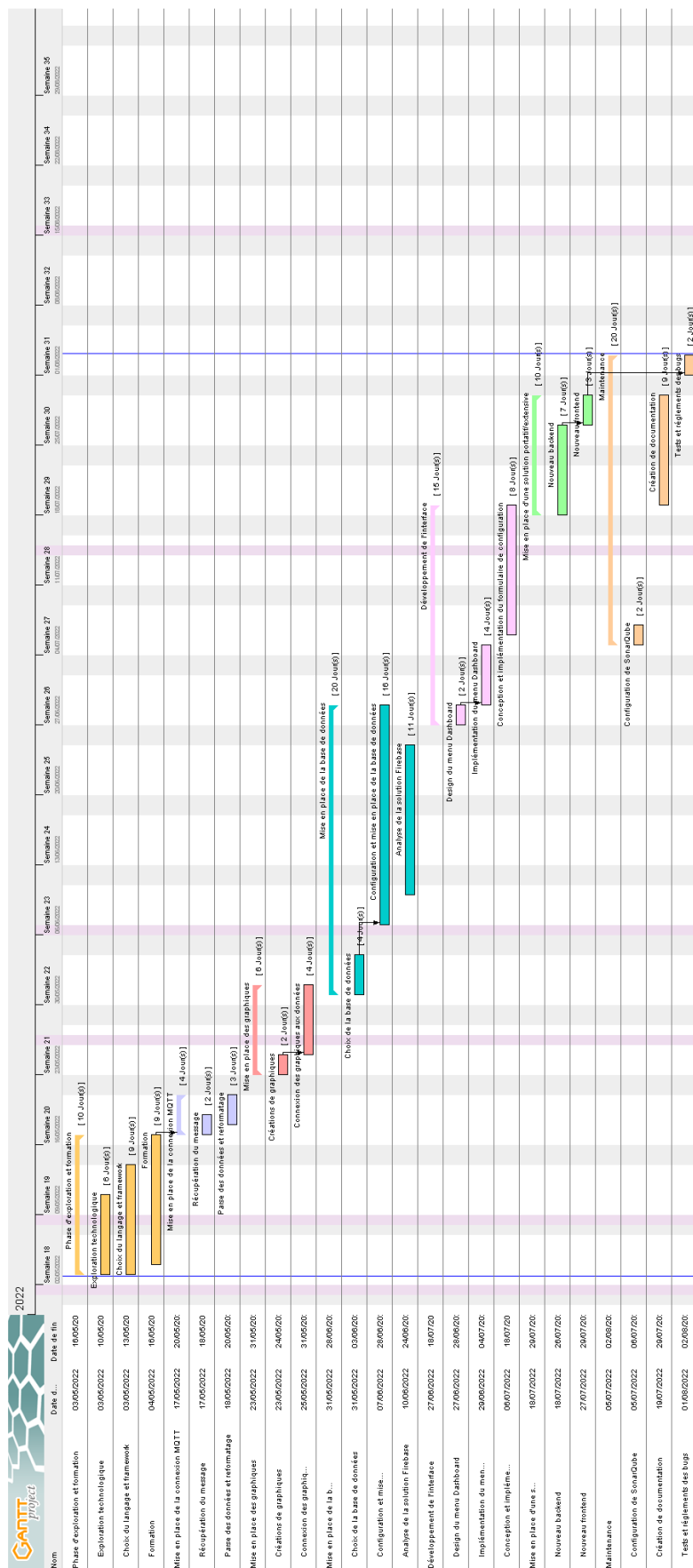
PostgreSQL



Github



Annexe 2 : Diagramme de Gantt



Annexe 3 : Liste des figures et tableaux

Figure 1 - Campus Plaine de Nimy, web.umons.ac.be	2
Figure 2 - Entrée principale de la Faculté Polytechnique de Mons.....	3
Figure 3 - Aperçu schématique des grandes composantes des TICs, wikipedia.fr	4
Figure 4 - Partie Cloud de l'architecture RAMi	5
Figure 5 - Partie Fog de l'architecture RAMi	6
Figure 6 – Architecture d'un réseau LoraWAN, frugalprototype.com	8
Figure 7 - Schéma présentant le protocole MQTT, arduino.blaisepascal.fr	9
Figure 8 - Connexion des capteurs avec le microcontrôleur Wemos D1 R3	10
Figure 9 - Formulaire de saisie d'une source MQTT	11
Figure 10 - Schéma de la collection serveur	12
Figure 11 - Exemple d'une instance de la collection "Serveur".....	13
Figure 12 - Architecture de l'application côté serveur.....	13
Figure 13 - Diagramme de classe de l'application.....	14
Figure 14 - Utilisateurs stockés sur Firebase Authentication	15
Figure 15 - Capture d'écran du menu Dashboard réalisé.....	16
Figure 16 - Répartitions des tâches durant le stage.....	17
Figure 17 - Résultats de l'analyse de la qualité logiciel avec SonarQube	19
Figure 18 - Localisation de Mons, Hainaut Développement.....	20
 Tableau 1 - Comparaison des différentes technologies pour le développement de l'application multiplateforme	 31
Tableau 2 - Comparaison de React Native et Flutter	32

Annexe 4 : Tableau comparatif détaillés des technologies

PWA		React Native	Flutter	Xamarin	Ionic
Description	Les PWA sont des applications web universelles qui offrent l'expérience proche d'une application native (notifications push, la disponibilité hors ligne...)	React Native est un Framework open-source orienté Frontend permettant de développer de véritables applications natives multiplateformes	Flutter est un SDK/Framework open-source orienté Frontend. Il prend en charge la création d'applications multiplateformes	Xamarin est une plateforme open-source qui permet de générer des applications modernes et performantes	Ionic est un Framework open-source qui permet de créer des applications multiplateformes en utilisant les technologies web
Date de création	2015	Mars 2015	Mai 2017	Mai 2011	2013
Langages de développement	JavaScript, HTML, CSS	JavaScript	Dart	C#	JavaScript, TypeScript, HTML, CSS
Interface utilisateur	HTML, CSS	Utilise des contrôleurs d'interface utilisateur natifs	Utilise des widgets propriétaires et offre une interface utilisateur étonnante	Utilise des contrôleurs d'interface utilisateur natifs	HTML, CSS
App mobile (iOS, Android)	Oui	Oui	Oui	Oui	Oui
App web (site web)	Oui	Oui	Oui (Chrome, Edge)	Oui	Oui
App Desktop (de bureau)	Oui ⁵	Oui (Windows, MacOS)	Oui (Linux, MacOS, Windows, Google Fuchsia)	Oui (Windows, MacOS)	Oui
Performance⁶	Chargement quasi instantané	Proche du Native	Excellente rapidité	Proche du Native	Modéré
Mise à jour	Oui	Régulièrement	Régulièrement	Progressivement mis à la retraite ⁷	Oui

Tableau 1 - Comparaison des différentes technologies pour le développement de l'application multiplateforme

Un comparatif détaillé des technologies hybrides a été réalisé sur le tableau 1. Tout d'abord, Ionic n'est pas intéressant pour notre application puisque ses performances sont modérées. Xamarin n'a pas été sélectionné puisqu'il est progressivement mis à la retraite. De plus, sa popularité n'était pas très importante. Concernant les trois technologies restantes, il y en a deux qui ont des caractéristiques excellentes : **Flutter** et **React Native**. Du fait de leurs performances, leur popularité et leur interface utilisateur très personnalisable.

⁵ (Overview of Progressive Web Apps (PWAs), 2022)

⁶ Source spécifiée non valide.

⁷ (Développement mobile : Microsoft met à la retraite Xamarin, 2022)

	React Native	Flutter
<i>Vendeur</i>	Facebook	Google
<i>Performance</i>	Bon	Meilleur
<i>Langage de programmation</i>	JavaScript	Dart
<i>Coût en production</i>	Excellent	Meilleur
<i>Documentation</i>	Pas de documentation officielle pour le déploiement automatisé ni la configuration CI/CD	Conviviale et facile à lire
<i>Taille de la communauté</i>	Grande communauté, active et expérimentée	Large communauté mais moins expérimentée en Dart
<i>Interface utilisateur</i>	Utiliser des contrôleurs d'interface utilisateur natifs	Utiliser des widgets propriétaires et offrir une interface utilisateur étonnante
<i>Applications connues</i>	Instagram, Netflix, Amazon	Google Ads, Alibaba, Ebay

Tableau 2 - Comparaison de React Native et Flutter

Un comparatif détaillé de React Native et Flutter a été réalisé sur le tableau 2. Tout d'abord, Flutter dispose de performance plus élevée dû à son utilisation du langage Dart. De plus, son coût de production est meilleur, sa documentation est facile à lire, et elle propose des interfaces utilisateur très impressionnantes. Le seul point où React Native est avantageux est sur sa grande communauté, très active et expérimentée. Cependant, Flutter commence à prendre de l'ampleur et finira par dépasser la communauté React Native dans les années à venir. C'est pourquoi le choix final s'est porté sur le **SDK Flutter**.

Le choix de version est 2.0 puisque cette version dispose des compatibilités **Windows, Linux et Web**.

Annexe 5 : Manuel d'utilisation

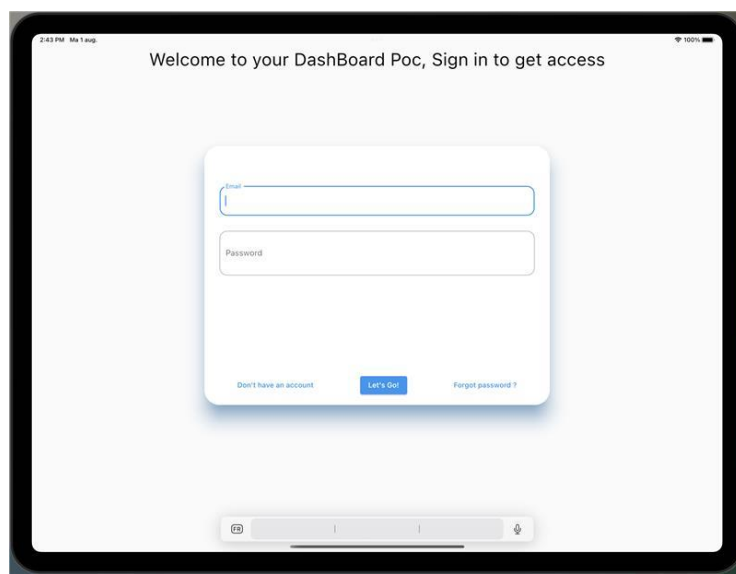
Ce manuel expliquera comment utiliser l'application en version 1.0. Elle a été réalisée pour consulter et analyser les données provenant de différents objets connectés, et cela, en utilisant différentes sources MQTT.

La version 1.0 est disponible en APK (lib > build > outputs > apk > debug > app-debug.apk. Cette version est disponible sur Android et iOS.

I. Profil utilisateur (admin inclut)

1. Menu de connexion

Lors de l'ouverture de l'application, on peut s'inscrire ou se connecter avec une adresse mail et un mot de passe. A l'inscription, un message est envoyé dans la boîte mail pour vérifier l'adresse mail avec un lien pour confirmer l'inscription. Une fois cela fait, on peut accéder à l'application. Il y a aussi une option si l'utilisateur a oublié son mot de passe. Si on ferme l'application et qu'on revient dessus, le statut de connexion est enregistré.



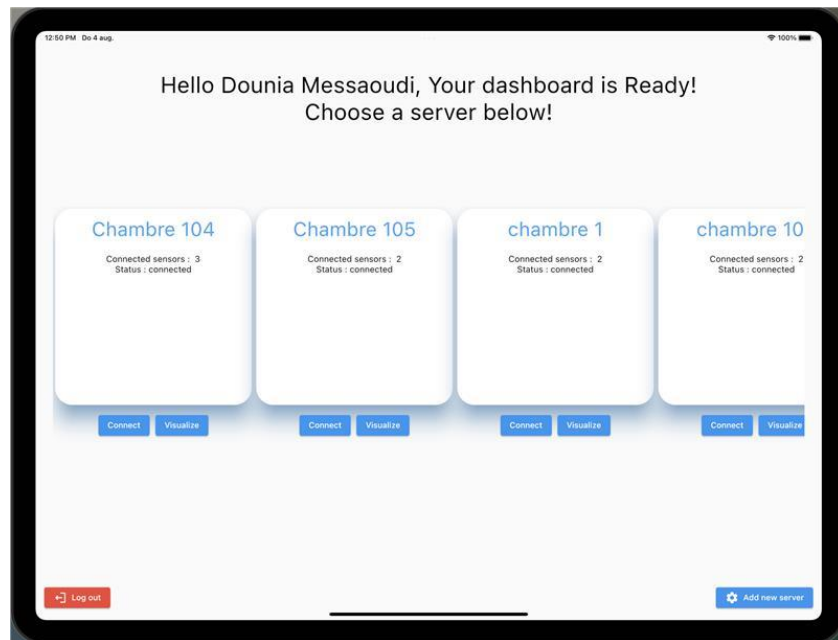
2. Menu principal

Une fois la configuration faite, les utilisateurs pourront avoir accès au menu principal, qui affiche le nombre de topic et le statut (connecté/déconnecté) pour chaque source MQTT.

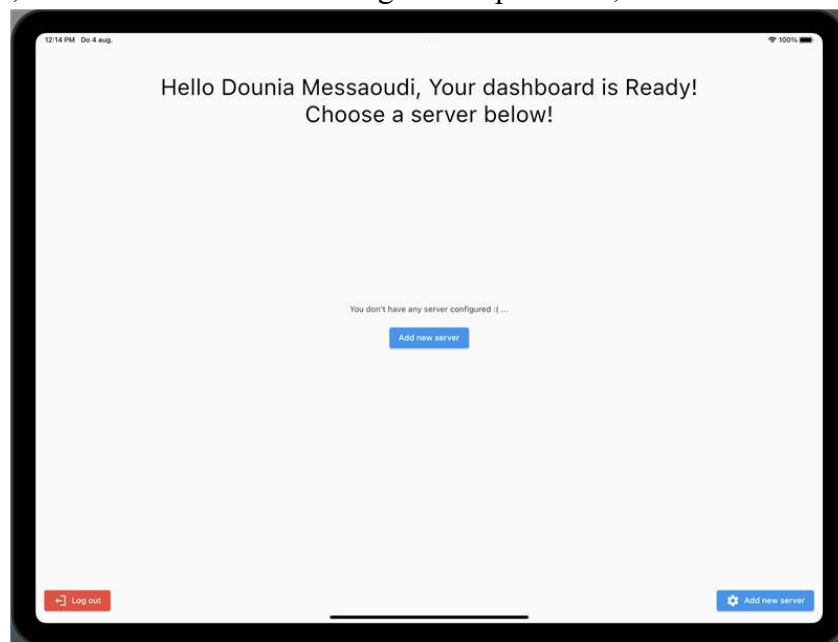
Les fonctionnalités possibles sont :

- Se connecter/déconnecter à une source MQTT en cliquant sur le bouton connexion/déconnexion
- Visualiser la source de données en cliquant sur le bouton « Visualize »
- Se déconnecter de l'application avec le bouton « Log out »

- Accéder à la configuration des données avec le bouton « Add new server » (PS : ce bouton est réservé à l'administrateur. Un utilisateur lambda ne pourra accéder aux configurations)



Si en revanche, aucune source n'a été configurée au préalable, on aura cet affichage.

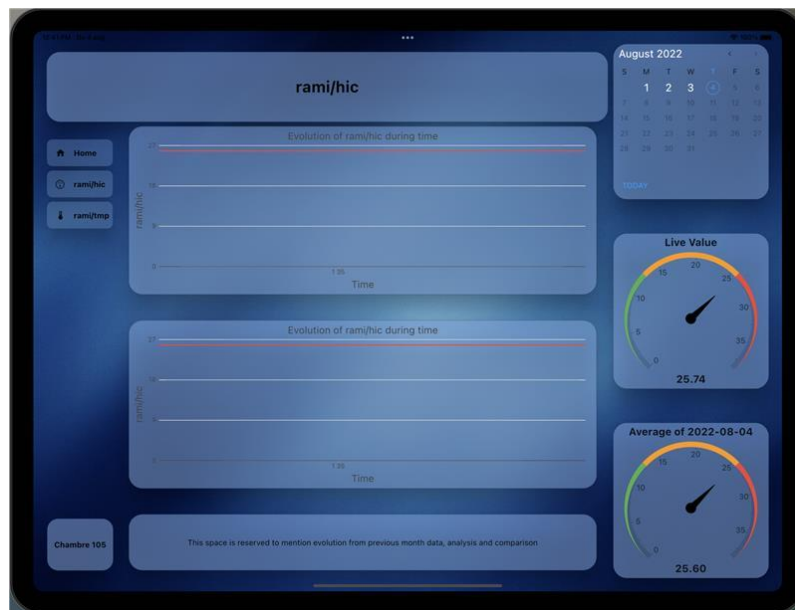


3. Menu Dashboard

Ce menu permet de visualiser en détail l'évolution des données des capteurs en temps réels pour une source MQTT donnée.

Les fonctionnalités possibles sont :

- Choisir le capteur que l'on veut en cliquant sur les boutons du menu à gauche.
- Visualiser l'historique des données du capteur sélectionné, à l'aide d'un diagramme linéaire
- Visualiser la valeur en temps réels du capteur choisi, sur un diagramme en jauge.
- Sélectionner la date pour laquelle on veut visualiser les graphes, à l'aide d'un calendrier. Si à cette date aucune donnée n'a été relevé, un message s'affiche « Aucune donnée relevée à la date sélectionnée ».
- Visualiser la valeur moyenne du capteur pour la date sélectionnée, sur un diagramme en jauge
- Être au courant d'une potentiel insécurité ou anomalie à l'aide du cadrant en bas de l'application (si une donnée est trop forte ou trop faible, un message l'indiquera).

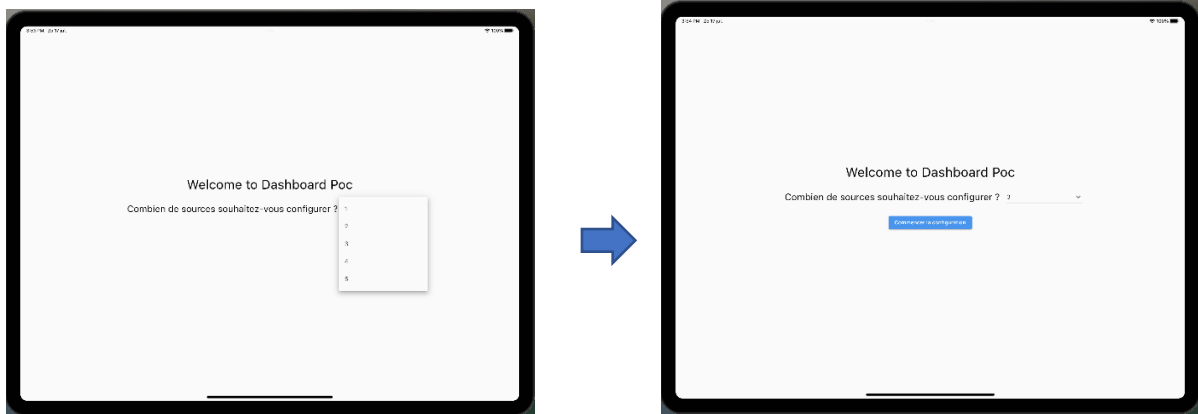


II. Profil administrateur – Configuration des sources

En cliquant sur « Add new server » dans le menu principal, l'administrateur peut accéder à un menu de configuration.

1. Choix du nombre de sources

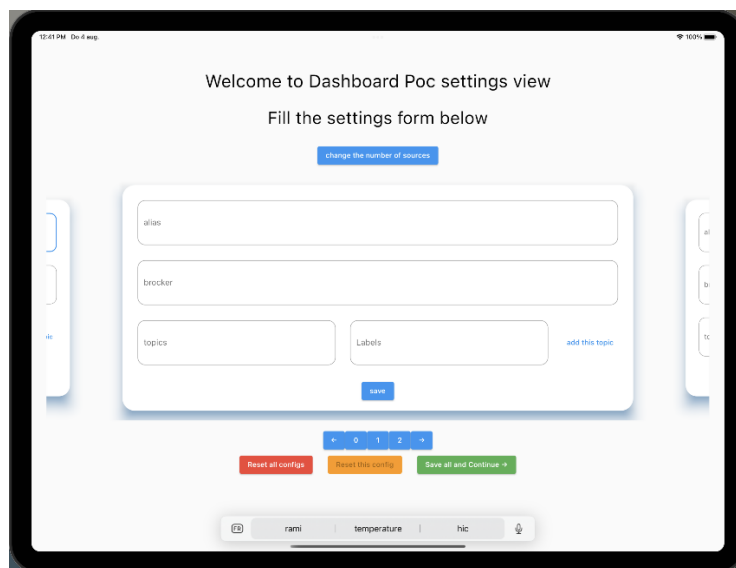
L'administrateur peut configurer le nombre de sources, en fonction des besoins de son organisation, à l'aide d'un menu déroulant. Une fois ce nombre choisi, un bouton « commencer à configuration » apparaît, afin de passer à la page suivante.



2. Remplissage des caractéristiques des sources

Ce menu affiche autant de formulaires que de sources MQTT, afin que l'administrateur puisse remplir les informations concernant toutes les sources. Les informations concernent :

- L'alias (nom de domaine)
- Le broker (host)
- Les topics (sujets)



Plus précisément, les fonctionnalités qui se présentent sont :

- Naviguer entre les formulaires en swipant sur les côtés ou en cliquant sur les boutons bleus (<- 1 2 3 ... ->)
- Revenir à la page précédente en cliquant sur le bouton « modifier le nombre de sources »
- Sauvegarder ses modifications actuelles avec le bouton « save »
- Réinitialiser toute la configuration (bouton rouge), ou le formulaire actuel (bouton orange)
- Sauvegarder tout et continuer (bouton vert)

Si la source configurée ne dispose pas de donnée, on aura un message d'erreur qui indiquera de contacter l'admin.

