



Rapport Projet Tuteuré

Projet CART :

**Comment la technologie peut-elle prévenir les
catastrophes sur des personnes âgées ?**

Lucas Abitbol
Colin Gay
Diogo Neiva Abreu
FIE5 – Promotion 2023

Sommaire

<i>Introduction.....</i>	<i>3</i>
<i>Enjeux et objectifs.....</i>	<i>4</i>
<i>Analyse et couverture du besoin.....</i>	<i>6</i>
<i>Spécifications de la solution</i>	<i>7</i>
<i>Choix d’environnement technique et développements</i>	<i>9</i>
1) Application côté client.....	9
2) Application côté serveur.....	12
<i>Résultats obtenus</i>	<i>14</i>
<i>Gestion de projet</i>	<i>17</i>
<i>Conclusions et perspectives</i>	<i>19</i>
<i>Bibliographie</i>	<i>20</i>
<i>Annexes.....</i>	<i>21</i>

Introduction

En France, la gestion de la vieillesse est un sujet qui nous anime. Prendre soin de nos anciens comme ils ont pris soin de nous dans notre enfance est en effet une volonté commune à tous les Français.

Mais alors que l'espérance de vie augmente d'années en années, les technologies pour suivre cette population vieillissante ne sont pas encore au point. Les chutes, la malnutrition ou encore l'isolation sont des sujets que beaucoup de personnes âgées connaissent et ils n'ont pas de solution pour remédier à cela.

Des moyens sont mis en place pour accompagner ces personnes comme des aides à domicile mais elles permettent seulement de limiter le risque sans réelles améliorations notables.

C'est dans ces conditions que le CHU de Toulouse a initié un projet de recherche d'envergure sur les capteurs ambiants et portés au domicile des patients âgés pour la prédiction d'évènements : CART France.

Ce projet étant étalé sur les 10 prochaines années a pour ambition de mettre en œuvre la médecine de précision du futur : des soins préventifs, personnalisés et proactifs.

Le projet CART France, mené en partenariat avec le centre de recherche ORCATECH aux Etats-Unis qui possède déjà un système mis en place et fonctionnel, consiste à corréler des données issues de capteurs recueillies par un Raspberry Pi (nano-ordinateur) au domicile des patients à des données cliniques collectées par questionnaire mail, via une application mobile ou lors de visites présentiels.

Les capteurs déployés sont pour l'instant des capteurs de mouvement, de contact de porte et de vitesse de marche au domicile de 4 participants, ils remontent des données en temps réel vers ORCATECH.

L'objectif serait d'équiper jusqu'à 100 maisons en France. Actuellement, c'est ORCATECH qui gère la réception et le traitement des données reçues par les Raspberry Pi.

Notre équipe a donc été contactée dans ce contexte afin de faire évoluer ce projet.

L'objectif de notre travail étant d'arriver à récupérer ces données brutes reçues sur les Raspberry Pi et de les intégrer dans une base de données en France en temps réel. L'équipe souhaite en effet pouvoir être souveraine sur les données des patients français et se détacher au plus possible d'ORCATECH sur les sujets de la gestion des données.

Ce rapport va décrire et détailler les travaux réalisés dans le cadre du projet tuteuré de dernière année menés par ABITBOL Lucas, GAY Colin et NEIVA ABREU Diogo.

Enjeux et objectifs

Les commanditaires de ce projet sont le Docteur Antoine Piau et Jean Marc Alliot du CHU de Toulouse.

Le Dr Antoine Piau est un médecin gériatre mais aussi le pilote de l'Unité Médicale Transverse de Télésanté au CHU de Toulouse. Il a une importante activité de recherche en France et aux Etats-Unis, dans le domaine du médicament mais surtout des nouvelles technologies pour le « bien vieillir » (communication, objets ou environnement connectés, intelligence artificielle). Son expertise est régulièrement sollicitée par les autorités de santé ou encore la commission européenne sur les sujets de gériatrie.

Jean Marc Alliot est actuellement directeur scientifique en Intelligence artificielle au CHU de Toulouse. Il est ingénieur, haut fonctionnaire et chercheur. Il est connu pour son travail de recherche qui porte sur les applications des techniques d'optimisation globales (algorithmes de type génétiques, recuit simulé, branch and bound, etc...) aux problèmes du trafic aérien.

En initiant ce projet, les enjeux principaux étaient de prévenir différents accidents pouvant survenir chez des patients et ainsi agir avant que ceux-ci ne se produisent. En utilisant les données de capteurs placés dans des maisons de patients et en les corrélant à des données cliniques, à travers divers algorithmes d'intelligence artificielle, il est possible de prévoir quand certaines actions ou mouvements sont anormaux.

En effet, les accidents survenant le plus souvent sont les chutes. Les conséquences de ces chutes, lorsqu'elles ne sont pas mortelles, sont lourdes en termes de santé : séquelles physiques, perte d'autonomie et fragilisation de l'état de santé initial. La diminution de ces accidents domestiques chez les personnes âgées repose principalement, pour le moment, sur la sensibilisation, la vigilance, l'aménagement des lieux de vie et la pratique d'une activité physique régulière.

De plus, grâce à ce projet, les personnes âgées qui sont encore indépendantes et souhaitent rester dans leurs maisons pourraient le faire car elles seraient en quelque sorte surveillées en temps réel. Ceci limiterait les déplacements inutiles et préviendrait les comportements à risques. Sachant que l'environnement influe sur la santé mentale, il est préférable de laisser certains patients dans un environnement positif pour leur bien-être.

Actuellement, toutes les données cliniques sont stockées sur la base de données de recherche clinique INSPIRE mais pas les données recueillies par les capteurs. Les données brutes obtenues par les capteurs sont envoyées aux USA pour pouvoir faire un prétraitement et les stocker par la suite. Parallèlement à l'envoi des données de capteur collectées par le Raspberry-Pi à ORCATECH, l'objectif de ce projet est d'avoir une base de données propre au CHU de Toulouse pour conserver les données brutes pour d'autres analyses et pour ne pas dilapider la production.

Ce sont des données de santé pseudonymisées soumises à un stockage HDS. Il faut donc commencer à réfléchir sur le type de stockage HDS à utiliser qui correspond le mieux à ces données pour le CHU.

Le but est que CART soit opérationnel rapidement grâce au système d'ORCATECH qui existe et est fonctionnel depuis 10 ans. En s'appuyant sur ce que l'équipe placée aux Etats-Unis a déjà produit, ce qui serait intéressant pour le projet CART FRANCE serait de

se détacher et de produire leur propre API avec leur propre base de données, et par conséquent il faudrait réussir à savoir comment sont programmés les Raspberry-Pi des différents capteurs utilisés dans les multiples maisons. De cette manière, l'équipe française serait indépendante et les données de santé des différents patients participants à ce projet ne seraient pas en circulation à travers le monde et seraient donc moins sujettes à des attaques malveillantes.

De plus, dans l'avenir, l'objectif est que CART puisse avoir un plan de communication qui permet un partage de données et de ressources facilité entre les utilisateurs, pour garantir un impact et une diffusion plus large.

Analyse et couverture du besoin

Dans le but de commencer à développer ce projet qui possède un grand potentiel, il nous a été demandé de fournir une première idée et une base de travail pour lancer le projet. Dans nos premières réflexions, nous avons immédiatement pensé que la première étape était de trouver un hébergeur de données de santé (HDS) qui saurait répondre à nos attentes car nous étions soumis à ces normes en utilisant des données de patients pour ce projet.

Nous sommes donc partis sur cette piste après la première réunion en recensant notamment tous les HDS de France. Nous manquions cependant toujours d'informations concernant le volume de données recueillies qui nous permettrait d'adapter notre choix d'HDS.

Cependant, après un entretien avec M. Alliot plus d'un mois après le début du projet, nous avons convenu que, pour le moment, cet impératif n'en était pas forcément un car premièrement les patients étaient tous soignés au CHU de Toulouse mais aussi car toutes ces données issues de capteurs sont des données brutes et ne correspondent donc pas à des données sensibles.

Grâce à ces réflexions, nous avons conclu que l'on peut héberger les données au sein du CHU, même si pour l'instant il n'y a pas encore de base de données certifiée HDS (celle-ci sera probablement effective dans un an). Nous pouvions donc sauter cette étape et commencer directement à réfléchir à comment mettre ces fameuses données en base.

Dans le but de répondre à l'objectif d'avoir une base de données propre au CHU de Toulouse, nous avons proposé trois scénarios possibles pour débiter le projet du plus au moins souhaitable, on pourrait alors :

- Récupérer les données des Raspberry Pi directement dans un fichier, puis les structurer nous-même dans une base de données propre au CHU.
- Dupliquer les données dans une base au CHU avant de les envoyer aussi vers les USA.
- Faire des requêtes à l'aide de leur API pour pouvoir créer notre base de données non relationnelle avec les informations extraites des requêtes.

L'idée principale reste de continuer à travailler avec leur structure et leur modèle de données vu qu'il est fonctionnel depuis une dizaine d'années.

Nous verrons plus tard que les contraintes temporelles et organisationnelles ont fait que l'on a dû choisir une de ces trois possibilités même si elle n'est pas la meilleure, mais à terme, il faudrait pouvoir réussir à récupérer les données directement du Raspberry Pi et ainsi éviter que les données aient à faire l'aller-retour vers les Etats-Unis. Ceci permettrait notamment d'avoir des suivis en temps réel au lieu de mensuel comme c'est le cas actuellement.

Spécifications de la solution

Après nous être entretenus avec le Dr Antoine Piau, Jean Marc Alliot ainsi que Éric Campo, nous avons convenu qu'il était temps de demander plus d'informations de la part d'ORCATECH.

Nous avons en effet réfléchi à trois scénarios possibles pour récupérer les données et il nous a fallu rédiger un mail qui nous permettrait de leur soumettre nos idées et de commencer à réellement travailler en fonction de leur réponse.

Bonjour,

Suite à notre échange de la semaine dernière, vous trouverez ci-après le mail que nous avons écrit et qu'il faudrait envoyer à Orcatech.
N'hésitez pas à revenir vers nous s'il y a des choses à modifier ou si nous avons omis certaines informations.

« Hello,

I'm sending you this email regarding the CART project.
Lately, a group of French students specialized in connected health have been working on the subject to advance the project and try to make concrete the exploitation of the data collected on the sensors we have already installed in France.

To support their research, they would need access to the same data that we collect.
They would indeed like to be able to access this data on a regular basis and have proposed several solutions in order to achieve this.
We would then like to see with you to what extent their requests could be realized.

In the most satisfactory case, they would like to know if it is possible to receive the data directly from the Raspberry Pi.
If it goes through a broker via MQTT, is it possible to connect to it?
If this is not possible, do you think it is possible to duplicate the data you store in the United States in a database in France (in this case at the hospital in Toulouse where we were working)? I suppose indeed that you structure the data received from the Raspberry, if at the time of your storage you could store it in our hospital that would help them a lot.

In case the above two proposals are not possible, we want to know if we can access your API on a more regular basis (maybe weekly) so that they can move forward quickly.

Thank you in advance for your response, I remain at your disposal for any further information you may require. »

Bonne fin de journée

GAY Colin, étudiant ISIS promo 2023

FIGURE 1 – MAIL ORCATECH

Le Dr Piau s'est alors chargé d'envoyer le mail à ORCATECH qui nous a rapidement répondu qu'il serait très difficile de récupérer les données directement depuis les Raspberry Pi car il faudrait être en mesure de se connecter aux réseaux Wifi des patients équipés. Or il est trop compliqué pour nous d'installer une machine par maison qui sert uniquement à récupérer les données des capteurs pour les envoyer à un serveur.

ORCATECH nous a cependant expliqué comment accéder à leur API qui comprenait toutes les données qui pourraient nous intéresser. Vous pourrez trouver la documentation de cette API en Annexe 1.

De cette base, nous nous sommes dit que la solution la plus simple dans le peu de temps qu'il nous restait était de construire un petit programme qui nous permettait de récupérer

les données de l'API et de les intégrer dans une base de données NoSQL en local comme MongoDB.

En allant plus loin dans cette idée, nous avons également pensé qu'il serait intéressant de faire une application visuelle qui permet uniquement de voir les nouvelles données qui ne sont pas encore dans la base et de les y mettre avec un seul clic. Cela évite en effet de tout faire par ligne de commande et permet d'avoir un aperçu visuel de ce qui va entrer dans la base.

Ce frontend effectuera alors un appel à l'API d'ORCATECH pour récupérer les données des capteurs, et enverra le même JSON que l'on reçoit à notre serveur qui se chargera de le mettre en base de données.

Choix d'environnement technique et développements

Pour construire cette application, nous avons décidé d'utiliser une méthode serveur/client où le client envoie des informations au serveur (ex : son identifiant et son mot de passe lors de la connexion) et le serveur traite l'information et produit une réponse en conséquence.

Pour réaliser cela, une des méthodes les plus utilisées est la création d'une API REST. Le but de cette méthode est de créer des requêtes de différents types que le client doit appeler dans le frontend.

1) Application côté client

Nous nous sommes servis de la bibliothèque JavaScript React.js afin de développer le côté client de notre application.

L'implémentation de React nous a permis de créer facilement l'interface utilisateur de notre application.

En effet, le but de cette bibliothèque est de concevoir du code réutilisable, pouvant s'adapter aux différents composants de notre projet. L'architecture de notre solution regroupe donc des composants HTML/CSS et JS.

Ce ne sont cependant pas les seuls avantages de React.

Il s'agit d'une bibliothèque très populaire, qui possède une documentation riche, alimentée et régulièrement mise à jour par une communauté active. Nous nous sommes d'ailleurs aidés de nombreuses chaînes YouTube, newsletters, blogs... afin de concevoir l'interface client de cette application ou bien de résoudre un problème au niveau du code. Effectivement, il est quasiment impossible que personne n'ait déjà rencontré le même problème.

De plus, il nous a paru intéressant de mettre en pratique les procédés de code actuels qui sont enseignés à ISIS.

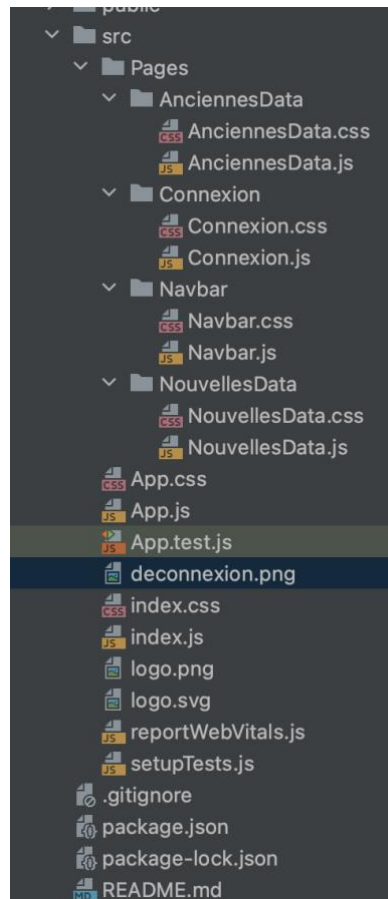


FIGURE 2 – STRUCTURE DU FRONTEND

Dans la figure 2, on voit qu'un dossier pages contient lui-même des sous-dossiers qui comprennent un fichier Javascript où toute la logique du code est décrite ainsi que HTML permettant de construire le visuel de la page. Ces sous-dossiers se composent également d'un fichier CSS permettant de mettre en forme la page.

On voit également le fichier App.js qui comporte toute la logique des routes de notre application et permet de créer des url spécifiques à chaque page de notre application. L'application a également une page de connexion afin d'être sûr que c'est bien un administrateur qui a la main sur ces données.

On peut voir que le client se compose de deux dossiers principaux :

- Un dossier AnciennesData, ce dernier vient récupérer toutes les anciennes données déjà existantes dans notre base de données locale. On délimite la date en fonction d'une variable incluse dans le backend et qui nous indique la dernière fois qu'un appel à l'API a été effectué afin de ne pas dupliquer les informations en base de données.
- Un dossier NouvellesData qui nous indique les nouvelles données pas encore enregistrées en base de données en fonction de la même variable venant du backend. De plus, dans cette page, on a un bouton qui, lorsqu'on clique dessus, permet d'envoyer à notre backend ces mêmes données afin de les intégrer en base de données.

```

function NouvellesData() {

  const url="https://juno.orcatech.org/apis/orcatech/v0.9/items/"
  const [data, setData] = useState( initialState: []);

  useEffect( effect: () => {
    const asyncFn = async () => {
      try {
        let result = await fetch(url, init: {
          method: 'GET',
          mode: 'no-cors',
          headers: {
            'Authorization': 'bearer lysV6g0tgyMbyZVLac99PwbHjnt0VL9FA3K3P1jFD+a19VfIDp8i2Q==',
            'Content-Type': 'application/json',
            'Accept': 'application/json'
          }
        });
        result = await result.json();
        setData(result);
        console.log(result);
      } catch {
        console.log("Error")
      }
    };
    asyncFn();
  }, deps: []);
}

```

FIGURE 3 –FICHER NOUVELLESDATA.JS

Dans la figure 3, on voit tout le code nous permettant de récupérer les informations des capteurs depuis l'API. On y indique l'URL qui nous intéresse et on fait un appel à ce dernier via une méthode GET de la fonction fetch.

On stocke ensuite les données dans une variable data qui nous permettra ensuite de les afficher dans notre application.

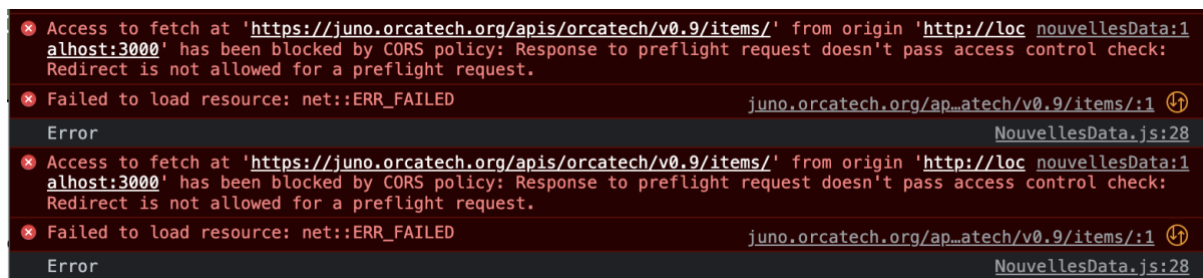


FIGURE 4 – ERREUR CORS

Cependant, cette méthode ne fonctionne pas dans notre cas à cause d'un problème d'autorisation. Il semble que dans le serveur fournissant l'API d'ORCATECH il manque un package appelé CORS qui permet à des utilisateurs d'accéder aux données quel que soit la machine et le port utilisé.

On peut voir l'erreur dans la figure 4.

2) Application côté serveur

Pour le back end de ce projet, nous avons décidé d'utiliser Node.js qui permet d'utiliser le langage de programmation Javascript pour développer un serveur notamment via son package Express.

Nous avons trouvé pertinent d'utiliser Node.js car c'est un logiciel libre, qui possède une grande communauté et qui permet de créer des applications très efficaces.

De plus, le front end étant développé en React, qui est également une librairie Javascript, nous pouvions nous aider dans la programmation.

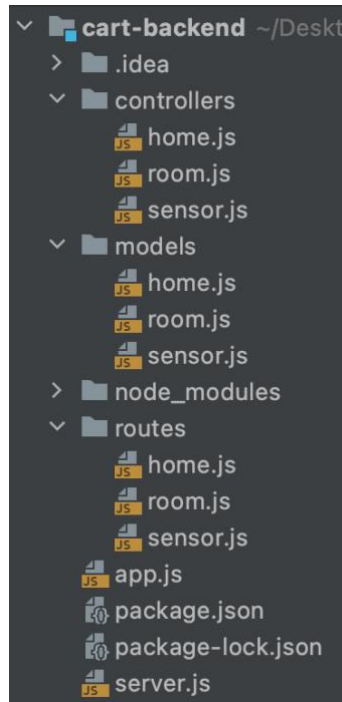


FIGURE 5 – STRUCTURE BACKEND

Nous avons décidé de structurer notre back end comme cela :

- Les controllers servent à définir les requêtes de l'API REST. Nous y écrivons le code qui explique ce qu'il doit se passer lors de l'ajout d'un capteur, d'une pièce ou de la modification d'une maison par exemple.
- Les modèles sont les classes qui permettent de créer nos objets Javascript. Ce sont ces objets que l'on appelle ensuite dans les requêtes REST (home, room, sensor). Nous avons créé ces modèles avec le package mongoose qui permet d'utiliser MongoDB.
- Enfin, les routes permettent d'indiquer au serveur la structure de l'API, on y indique la route et la méthode. On appelle alors les fonctions créées dans les controllers.

Pour ce qui est de la base de données, nous avons décidé d'utiliser MongoDB, qui est NoSQL et est très utilisée en Node.js.

La création et la connexion à la base de données a été cependant été un gros problème. Nous n'avons en effet pas réussi à résoudre les problèmes dans le peu de temps qu'il nous restait et nous n'avons donc pas pu tester notre serveur.

```

var mongoose = require("mongoose");

// Declare schema
var sensorSchema = new mongoose.Schema({
  sensor_id: {type: String, required: true, index: {unique: true}},
  home_write_key: {type: String, required: true},
  sensor_type: {type: String, required: true},
  sensor_location: {type: String, required: true}
});

// Export schema ("sensor_info" - is name of DB)
module.exports = mongoose.model("Sensor", sensorSchema);

```

FIGURE 6 – MODELE DE CAPTEUR

Pour utiliser MongoDB dans Node.js, on doit d'abord créer un modèle. Dans la figure 6, on peut voir le modèle créé pour un capteur. N'ayant pas pu tester notre serveur à cause du frontend qui ne fonctionne pas et des problèmes liés à la base, nous ne savons pas si ce schéma est le plus optimal mais nous avons pensé qu'il serait intéressant d'avoir un ID, une clé pour relier ce capteur à une maison, le type du capteur et la pièce dans laquelle il est situé. Nous aurions pu en faire un adapté si nos appels à l'API fonctionnaient et que l'on pouvait voir la structure du JSON reçu.

Résultats obtenus

Malgré les contraintes de temps, nous avons réalisé le squelette d'une petite application web très simple d'utilisation.

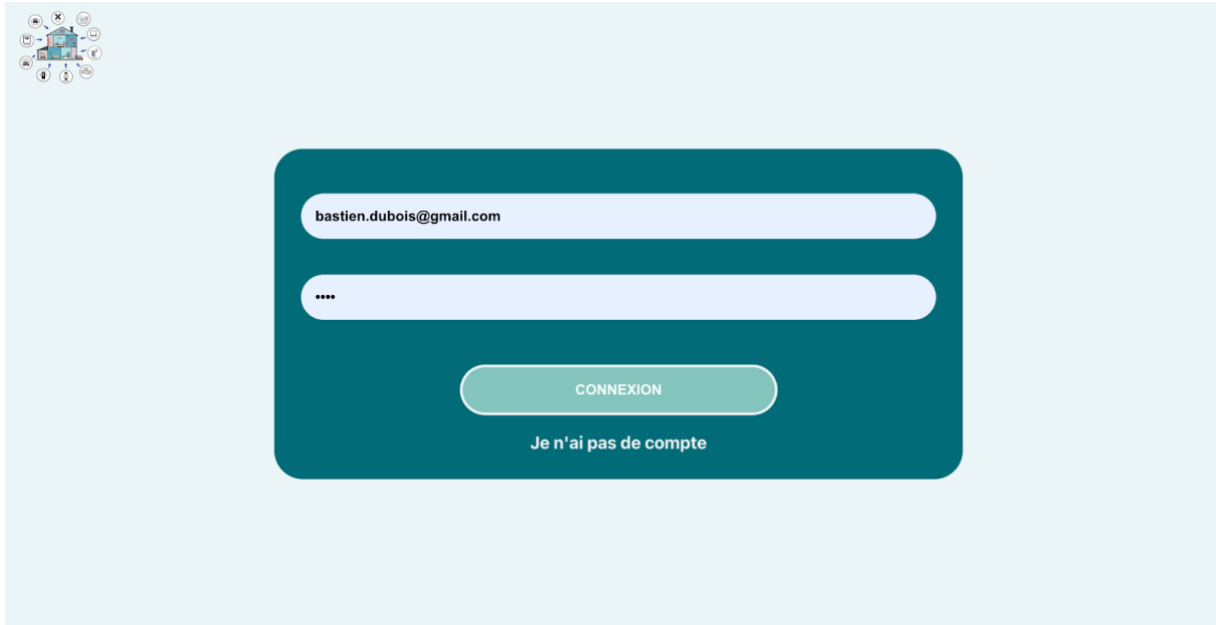


FIGURE 7 – PAGE D'ACCUEIL

Pour y accéder, il faudra d'abord s'authentifier ou créer son propre compte comme nous pouvons le voir dans la figure 7 ci-dessus.

Nous avons en effet pensé que seul le personnel de l'équipe CART France pourrait y accéder au vu des données sensibles que la base contient. De plus, il faudrait même mettre en place une vérification des demandes de nouvelle création de compte pour ne pas laisser une personne avec des intentions malveillantes accéder à cette plateforme.

Cette interface de connexion n'est cependant pas encore opérationnelle car nous n'avons pas eu le temps d'adapter le serveur pour gérer les connexions et les inscriptions. Un identifiant et un mot de passe sont donc inscrits en brut directement dans le frontend.

Pour rappel, nous trouverons dans cette application des données qui proviennent de capteurs installés dans les différentes maisons de patients qui participent à ce projet de recherche et la sécurisation de la connexion est donc indispensable.

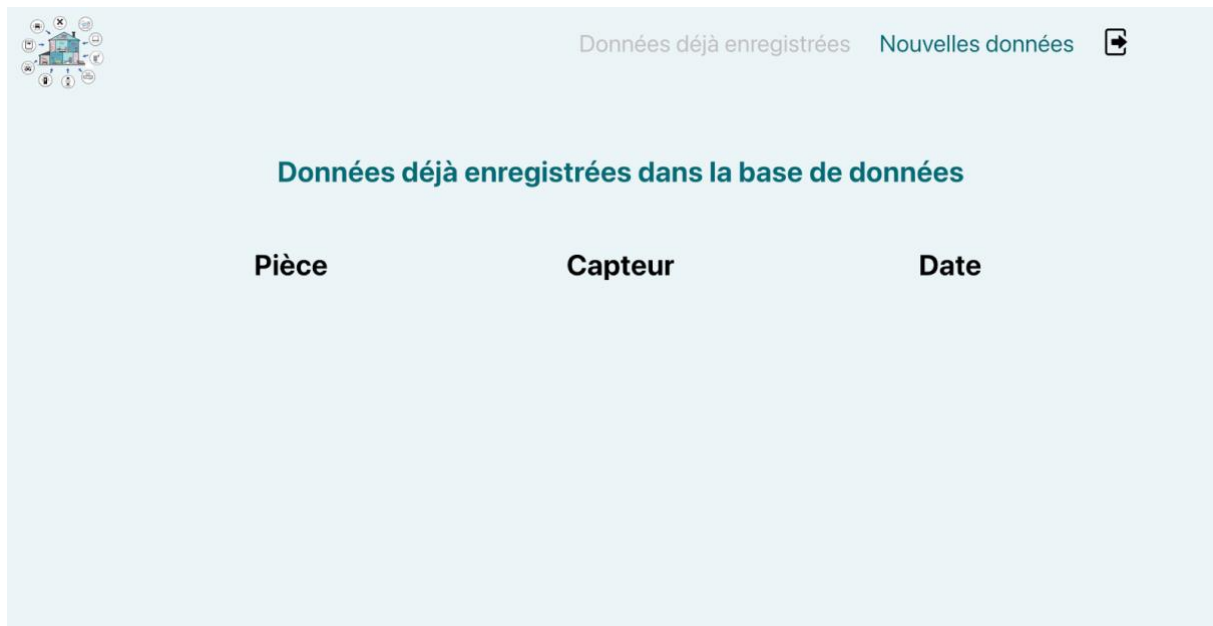


FIGURE 8 – PAGE DONNEES DEJA ENREGISTREES

Par suite de la connexion, nous arrivons sur une page que l'on peut voir en figure 8, qui nous permet de consulter un tableau avec les données déjà enregistrées dans la base de données.

Ce tableau permet notamment de voir la pièce dans laquelle se trouve le capteur et la date exacte de la mesure. Mais également les informations concernant la mesure (la valeur que celui-ci renvoi à cet instant T. Par exemple, un capteur de mouvement présent dans cette pièce enverrait "1" quand il détecterait une personne ou alors "0" dans le cas contraire)

Dans les données brutes qui nous ont été fourni pour qu'on puisse se baser sur quelque chose, nous avons pu observer que le retour du capteur n'était pas fait tout le temps quand il ne détecte rien. Nous aurions pu avoir une succession de 0 pendant le temps où le capteur ne détecte rien mais celui-ci ne renvoie des valeurs que de temps en temps sans forcément avoir un intervalle régulier.

Le but de cette page est plus généralement d'avoir un retour concernant les dernières dates d'ajout en base et pourrait être amélioré en ne filtrant que sur un seul capteur ou une seule pièce afin de savoir à quelle date on a enregistré dans la base pour ce capteur ou cette pièce.



FIGURE 9 – NOUVELLES DONNEES A ENREGISTRER

Enfin, en sélectionnant l'onglet "Nouvelles données", nous pouvons comme le nom l'indique accéder aux données qui se trouve sur l'API d'ORCATECH mais qui ne sont pas encore présentes dans notre base en local. Nous retrouvons cette page en figure 9. Cette page se compose principalement d'un tableau de la même forme que sur la précédente page.

La seule différence sur cette page est qu'en cliquant sur le bouton vert que nous pouvons voir, les nouvelles données seront envoyées au backend et enregistrées dans la base de données. Le tableau de cette page serait donc vidé au moment de l'enregistrement. De plus, pour vérifier que les données sont bien enregistrées ou simplement pour les consulter, il suffit de retourner sur la page précédente en cliquant sur l'onglet "Données déjà enregistrées". Si tout s'est bien passé, on devrait logiquement retrouver toutes les informations qui sont sur l'API à cet instant sur notre base de données.

Gestion de projet

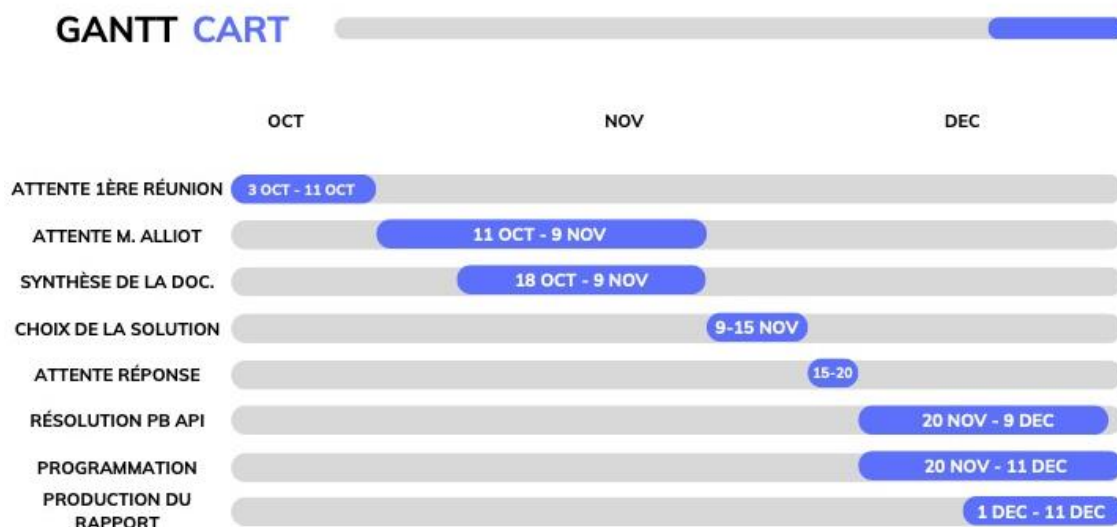


FIGURE 10 – GANTT GESTION DE PROJET

Au niveau de la gestion de projet, nous avons eu énormément de difficultés à avancer efficacement à cause de nombreux facteurs.

En effet, dès le début du projet, le client avait des impératifs vis-à-vis de son travail et ne pouvait pas être disponible régulièrement, ce qui a provoqué du retard dans nos prévisions d'avancement. Nous avons pu avoir notre première réunion une semaine après le début du projet et nous n'avons pas pu avancer avant cette dernière car nous ne connaissions pas les détails du projet qui nous étaient nécessaires. Cela nous a fait perdre un temps considérable mais nous n'avions pas le choix car nous ne pouvions pas commencer à travailler sans ces informations. De plus, cette réunion était juste une prise de contact avec le Dr Piau et ce dernier nous a seulement expliqué le concept du projet ainsi que les différents objectifs de celui-ci. Nous n'avons obtenu aucun renseignement technique de sa part car il n'avait pas les compétences pour nous les donner, il nous a juste indiqué notre rôle dans ce projet de grande envergure.

Nous avons donc dû programmer une réunion avec M. Alliot afin d'obtenir des informations techniques qui nous permettraient d'avancer dans notre travail mais ce dernier, en congés à ce moment-là, n'était pas disponible avant la première semaine de novembre ce qui correspondait à un mois d'attente et donc pratiquement à un tiers du temps dédié à notre projet. Nous leur avons donc demandé un document technique que nous pourrions utiliser pour avancer avant la réunion afin de pouvoir poser des questions précises lors de contact avec M. Alliot. Après une semaine, nous avons obtenu un document technique explicatif en anglais de 90 pages avec les informations complètes du projet CART. Vous pourrez trouver ce document en Annexe 2.

Ainsi, nous nous sommes reparti le document pour que chacun fasse un résumé des informations importantes de sa partie afin de synthétiser le document le plus rapidement possible. Vous pourrez trouver ce résumé en Annexe 3.

Grâce à ce document, nous avons pu commencer à réfléchir aux différentes options pour satisfaire au mieux le client afin d'avoir la réunion la plus efficace possible étant donné que nous avions déjà énormément de retard sur notre projet.

Nous avons donc réellement commencé à avancer uniquement après la réunion avec M. Alliot et suivant la synthétisation des documents. Durant cette dernière nous avons pu échanger avec des personnes étant plus dans le domaine technique du projet, afin d'obtenir des informations qui ne sont pas dans le domaine d'expertise du Dr Piau, essentielles pour la réalisation de nos objectifs.

À l'aide de ces informations, nous avons pu envisager les différentes solutions qui permettraient de récupérer et traiter les données de l'API et les classer selon l'efficacité de ces dernières. Mais nous avons remarqué un nouveau problème dans notre projet grâce à cette documentation : nous devons récupérer les différentes données de santé dans un centre de recherche américain mais nous ne pouvions pas les contacter directement, nous devons passer par notre commanditaire.

Ainsi nous avons encore une fois dû attendre le retour de ce dernier afin de pouvoir avancer dans l'application que nous voulions coder. Mais, après le retour du Dr Piau qui nous a envoyé des identifiants d'un membre de l'équipe, nous avons remarqué que nous n'avions pas les autorisations nécessaires pour accéder aux données CART de l'API.

Nous avons donc contacté M. DEROUICHE afin d'obtenir un compte qui pourrait accéder aux données.

Mais, même avec les informations qu'il nous a données lors de notre première réunion avec lui, nous n'avons pas réussi à récupérer les informations de l'API. En revanche, il nous a envoyé un lien qui permettait de visualiser la forme des données.

Nous nous sommes donc reparti le travail, un devait travailler sur le rapport, un sur la programmation de l'application et le dernier devait travailler sur l'analyse de la forme des données afin de produire notre modèle de données. Malgré cela, ne pas pouvoir récupérer les données de l'API ne nous permettait pas de coder l'application correctement.

Nous avons donc demandé une nouvelle réunion à M. DEROUICHE afin de réessayer de se connecter à l'API, lors de cette dernière nous avons réussi à récupérer les données via PostMan mais il ne nous restait qu'une semaine pour produire l'application avant la soutenance. De plus, nous devons également rédiger le rapport avant le 12 décembre, nous avons donc privilégié ce dernier même si nous avons essayé d'avancer le code pendant ce temps. En effet, nous n'avons eu le premier retour de l'API que le 8 décembre pour un rendu le 11, ce qui laisse un laps de temps extrêmement court pour développer une application.

De plus, de nouvelles erreurs sont apparues pendant le développement et ont fait que nous n'arrivions toujours pas à appeler l'API depuis notre application

La dernière semaine sera cependant consacrée à la production de l'application afin d'avoir un résultat correct pour la soutenance.

Pour résumer, les différents blocages que nous avons rencontré pendant toute la durée de ce projet ne nous ont pas permis de travailler correctement. En effet, la multiplicité des acteurs présents dans ce projet et notre incapacité à contacter directement les bonnes personnes aux bons moments a fait que nous étions bloqués à chaque étape de notre avancée.

Conclusions et perspectives

Ce projet nous a fait passer par de nombreux états.

Tout d'abord l'excitation, de s'imaginer apporter sa pierre à l'édifice d'un projet d'une telle envergure, mais également d'espérer voir les résultats dans quelques années sur certains de nos proches ou sur nous-mêmes.

Ensuite est venue l'appréhension, nous avons en effet rapidement compris que c'était un projet d'une envergure énorme qui s'étendrait sur une dizaine d'années et que nous étions la fondation qui se voudrait solide d'un tel projet. Nous avons voulu faire de notre mieux et nous nous sommes directement imaginés comment faire pour répondre aux attentes qui nous étaient destinées.

Après ce temps est venue l'incompréhension. Les informations étaient multiples et, il est vrai, nous avons eu du mal à comprendre quels étaient réellement les attendus de notre travail. Entre gestion de la base, récupération des données, choix d'un hébergeur, compréhension de l'existant aux US, développement d'une application, nous ne savions plus par où commencer.

Et enfin est venue la frustration. Après le deuxième entretien et plus d'un mois après le début du projet, nous pouvions enfin réellement commencer. Mais même avec les informations qui nous étaient manquantes jusque-là, nous avons fait faces à de nombreux problèmes. Nous avons en effet été bloqués à quasiment toutes les étapes du projet.

Mais malgré tout ça, ce projet a été enrichissant car il nous a permis de nous imprégner d'un milieu hospitalier où le temps est précieux et où les acteurs sont surchargés de travail. De plus, nous croyons réellement en ce projet et pensons qu'il peut vraiment faire avancer le sujet des accidents de vieillesse en France. Nous remercions donc nos commanditaires pour cela.

De plus, nous espérons que nos réflexions aideront les futurs groupes de travail à avancer dans le futur.

La solution optimale serait que le CHU de Toulouse possède ses propres Raspberry Pi implantées dans les maisons des patients volontaires.

Ainsi l'alimentation d'une base de données locale serait plus simple car on se séparerait de la contrainte liée à la communication avec l'API d'ORCATECH et les données seraient plus facilement accessibles. Cela favoriserait un accès beaucoup plus rapide et un suivi en temps réel des patients.

Il serait ensuite intéressant de développer une application pour les professionnels de santé pour qu'ils aient des alertes quand les données sont anormales. Ainsi, les patients quel qu'ils soient pourraient avoir un suivi personnalisé et seraient surveillés pour qu'ils continuent à être en bonne santé.

Bibliographie

<https://www.ohsu.edu/oregon-center-for-aging-and-technology>

<https://www.ohsu.edu/collaborative-aging-research-using-technology/about-us>

<https://fr.reactjs.org/docs/getting-started.html>

[https://developer.mozilla.org/fr/docs/Learn/Tools and testing/Clientside JavaScript frameworks/React getting started](https://developer.mozilla.org/fr/docs/Learn/Tools_and_testing/Clientside_JavaScript_frameworks/React_getting_started)

<https://openclassrooms.com/fr/courses/6390246-passez-au-full-stack-avec-node-js-express-et-mongodb>

Annexes

Annexe 1 :

Documentation API ORCATECH

<https://api.orcatech.org/orcatech/v0.9/docs/index.html>

Annexe 2 :

Documentation CART

Annexe 3 :

Document récapitulatif de la documentation CART