

18 janvier 2019

Rapport Iot Santé

Antoine Laudré – Gatien Poirrier

Table des matières

[Enumération des User-Stories choisies en début de projet 2](#_Toc535526553)

[Spécifications fonctionnelles et techniques 3](#_Toc535526554)

[Architecture technique mise en place : 6](#_Toc535526555)

[Fonctionnalités réalisées 8](#_Toc535526556)

[Améliorations du projet + fonctionnalités non réalisées 8](#_Toc535526557)

[Apports personnels du projet et répartition des tâches 8](#_Toc535526558)

# Enumération des User-Stories choisies en début de projet

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| [US](https://docs.google.com/drawings/d/1sIZ2N1toCjw7KRv7frqe1Bh1N-s_AYpFElnAlCThUz8/edit) | US-01 | US-02 | US-03 | US-04 | US-05 | US-06 |
| **En tant que** | **Patient** | **Médecin** | **Médecin** | **Patient** | **Patient/Médecin** | **Patient/Médecin** |
| **Je souhaite** | Accéder aux données en temps réel | Faire un diagnostic | Visualiser l'historique des données | Visualiser l'historique des données | Changer l'échelle temps des graphiques | M'inscrire |
| **Afin de** | Afficher le tableau de bord des données reçues par les capteurs sous forme de graphiques et courbes | Aider mon patient | Voir l'évolution de mes patients | Voir mon évolution | - Avoir une vue d'ensemble de mes résultats  - Visualiser mes données sur la période temps souhaitée | Bénéficier des services proposés par la plateforme |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [US](https://docs.google.com/drawings/d/1sIZ2N1toCjw7KRv7frqe1Bh1N-s_AYpFElnAlCThUz8/edit) | US-07 | | US-08 | US-09 | US-10 | US-11 | US-12 | US-13 |
| **En tant que** | | **Patient/Médecin** | **Médecin** | **Médecin** | **Médecin** | **Patient** | **Patient** | **Médecin** |
| **Je souhaite** | | Me connecter | Être alerté en cas d’anomalies dans les résultats de mes patients | Accéder à la liste de mes patients | Que personne d'autre que moi n'accède aux données de mes patients | Que seul mon médecin ait accès à mes données | Echanger avec mon médecin | Echanger avec mon patient |
| **Afin de** | | Accéder à mon compte | Pouvoir intervenir le plus rapidement possible auprès de mon patient touché | Obtenir des informations sur le patient souhaité | Garder le secret médical et ne pas violer le règlement sur la protection des données personnelles | Garder le secret médical et ne pas violer le règlement sur la protection des données personnelles | De personnaliser mon suivi | De personnaliser le suivi de chacun de mes patients |

# Spécifications fonctionnelles et techniques

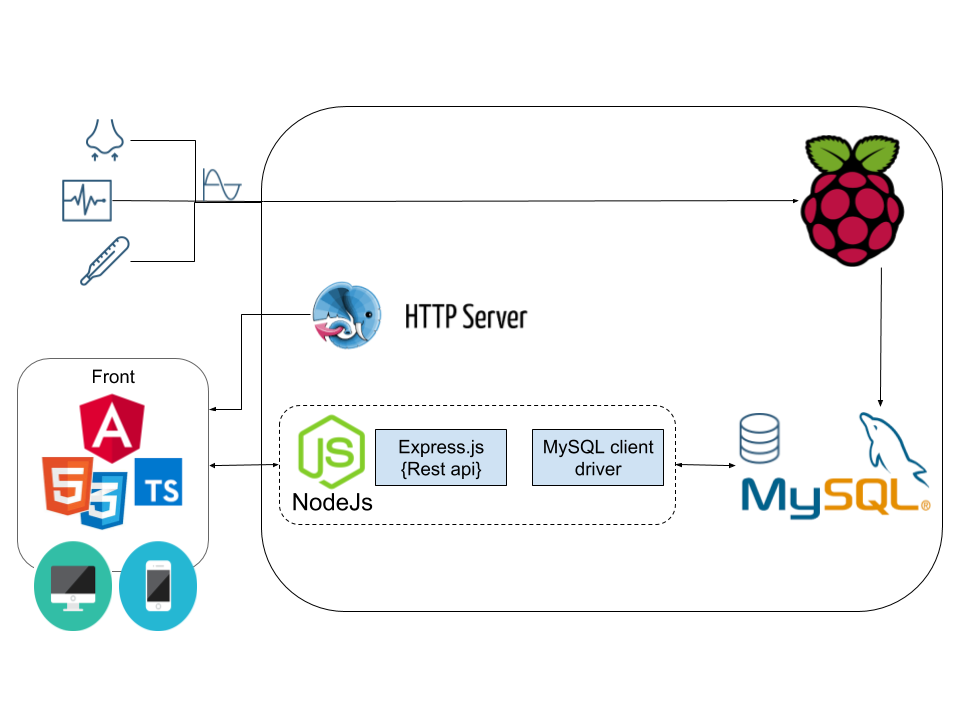
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Spécification fonctionnelle (à condition que) | | Les données des capteurs soient reçues par la plateforme pour les traiter | Les données soient disponibles sur différentes échelles de temps et comparées à des valeurs moyennes | Les données stockées soient accessibles par le médecin | Les données stockées soient accessibles par le patient |
| Spécifications techniques | **Front End** | Créer des graphiques à partir des données traitées provenant des capteurs | - Ajouter des valeurs moyennes en comparaison avec les données du patient  - Mettre en place des filtres et les appliquer aux graphiques | Afficher les données de la BDD sous forme de graphiques compréhensibles et montrant l'évolution des résultats | Afficher les données de la BDD sous forme de graphiques compréhensibles et montrant l'évolution des résultats |
| **Back End** | Les données des capteurs (respiratoire, température, électrocardiogramme) sont :  - envoyées à la Raspberry PI 3  - traitées  - envoyées directement à la partie front-end pour assurer le visionnage en temps réel des données | Enregistrer le diagnostic dans la BDD et notifier le patient | Récupérer la demande du front end et traiter la demande et renvoyer les données requises de la BDD | Récupérer la demande du front end et traiter la demande et renvoyer les données requises de la BDD |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Spécification fonctionnelle (à condition que) | | Des filtres adaptés soient fonctionnels | Les champs obligatoires du formulaire d'inscription soient remplis par des données valides | Se soit déjà inscris et que je renseigne le bon identifiant et le mot de passe | - La plateforme puisse envoyer un mail au médecin  - Les seuils d'alertes aient été remplis au préalable |
| Spécifications techniques | **Front End** | Afficher les données transmises par la BDD | Formulaire disponible :  L’utilisateur remplit les champs identifiant, mot de passe, numéro de téléphone et email | Formulaire disponible :  L’utilisateur remplit les champs identifiant et mot de passe | Template de mail de notification |
| **Back End** | Récupérer la demande du front end et traiter la demande et renvoyer les données requises de la BDD | On vérifie que l'identifiant est unique  Si oui, on enregistre le nouvel utilisateur dans la BDD et on renvoie un message de succès et on redirige vers la page de connexion  Si non, on renvoie un message d'échec | Comparer les données rentrées avec celles de la BDD  Si les données comparées sont identiques, on affiche un message de succès et on redirige vers la page d'accueil du site web  Sinon on affiche un message d'erreur et on affiche de nouveau la page de connexion | Remplir le Template et envoyer le mail au médecin |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Spécification fonctionnelle (à condition que) | | Le médecin ait l'autorisation d'accéder aux données de ses patients | Mes données soient sécurisées, qu'il faille un module de connexion pour accéder aux données | Mes données soient sécurisées, qu'il faille un module de connexion pour accéder aux données | Le numéro/email du médecin soit affiché  Le contact se fait en dehors du site web | Le numéro/email du patient soit affiché |
| Spécifications techniques | **Front End** | - | - | - | Affichage le moyen de communication du médecin | Affichage le moyen de communication du patient |
| **Back End** | - Donner les bonnes autorisations d'accès  - Hashage des données en transit | - Une BDD sécurisée  - Hashage des données en transit | - Une BDD sécurisée  - Hashage des données en transit | Récupérer les moyens de communication de la BDD | Récupérer les moyens de communication de la BDD |

# Architecture technique mise en place :

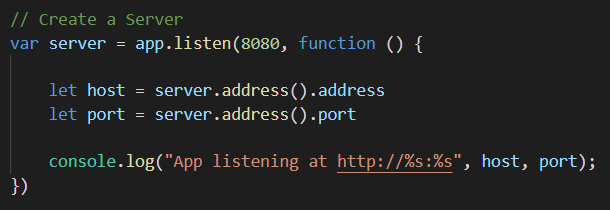
Notre application santé récupère les données de trois capteurs qui sont : le capteur de température, le capteur de position et le capteur de respiration. Nous envoyons ces données depuis la Raspberry Pi 3 sur notre base de données MySQL. Et nous lisons ensuite notre base de données MySQL à l’aide d’une API qui transmet les informations nécessaires à la partie front-end pour dessiner les graphiques.



L’application a été réalisé avec Angular6 et Angular-cli. Nous avons créé nos vues et composent web avec des éléments HTML et SCSS. La partie script, a été réalisée en TypeScript.

Une fois notre application finie et Bluidé, nous l’avons placé sur la Raspberry afin de la faire tourner grâce à un server « http server ». L’application est donc directement hébergée sur la carte RaspberryPi.

Afin de connecter le front (Angular6) et la BDD (MySQL), nous avons décidé de faire tourner un server Node couplé à Express afin de requêter notre MySQL et de répondre aux requêtes venant de notre Client http.



#### Description des capteurs :

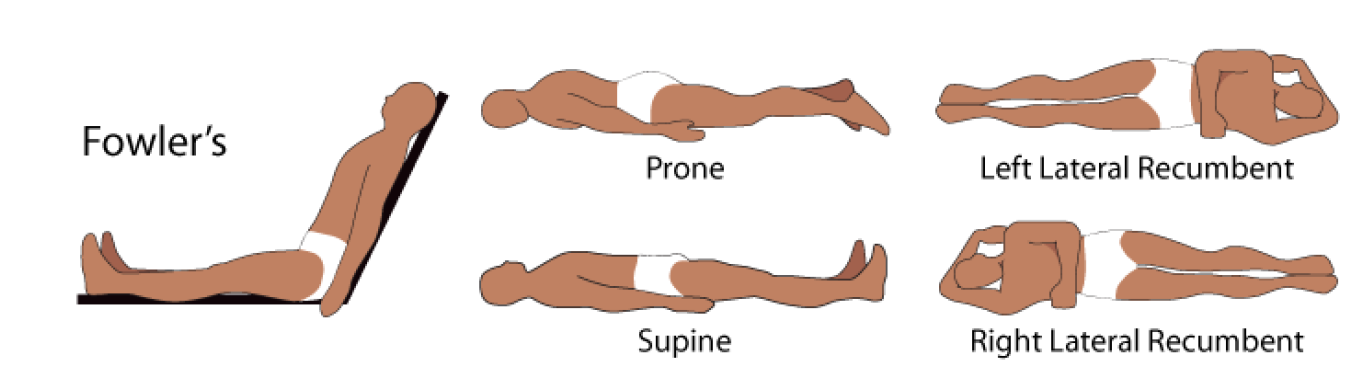
Nous avons choisi d’afficher 3 types de données à l’aide de 3 capteurs qui sont le capteur de température, le capteur de respiration ainsi que le capteur de position.

##### Capteur de température :

Ce premier capteur permet de mesurer la température du patient tout au long de l’effort.

##### Capteur de position :

Ce deuxième capteur permet de suivre les positions exécutées par le patient durant les exercices. Par conséquent, il est possible de vérifier si les exercices à faire sont bien fait. Il y a 5 positions différentes comme vous pouvez le constater sur la figure ci-dessous :

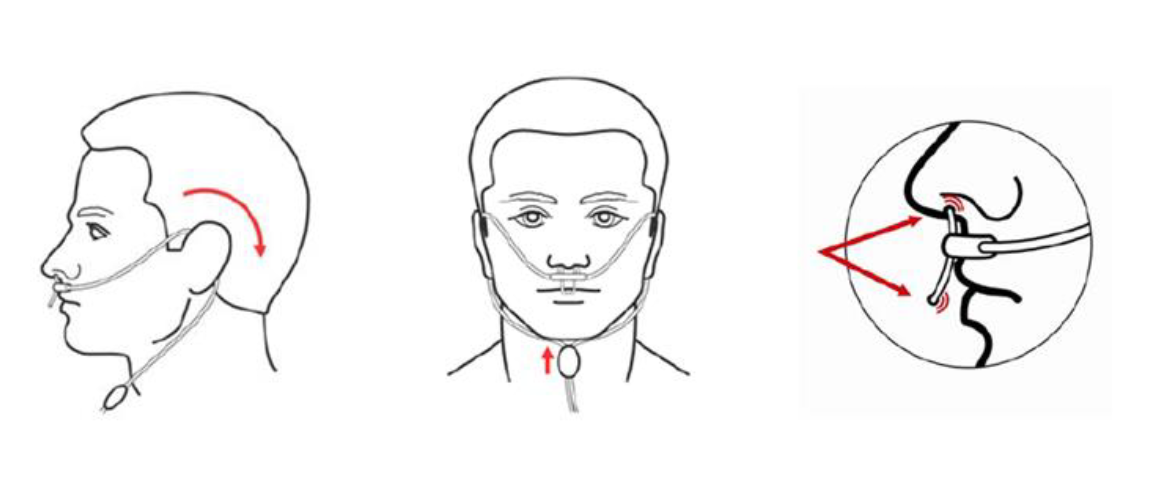


Ces positions sont détectées à l’aide d’une ceinture pectorale qui reconnaît la position de l’utilisateur. Il y a un délai de 3 secondes entre chaque position donnée par le capteur pour ne pas surcharger les informations reçues et rendre inutilisable le graphique construit à partir de ces données.



##### Capteur de respiration :

Pour compléter les informations données par nos deux précédents capteurs, nous avons choisi de mettre en place un capteur de respiration qui permettra de suivre en temps réel la respiration du patient. On pourra mesurer sa fréquence de respiration et son intensité au cours de l’exercice effectué par le patient.

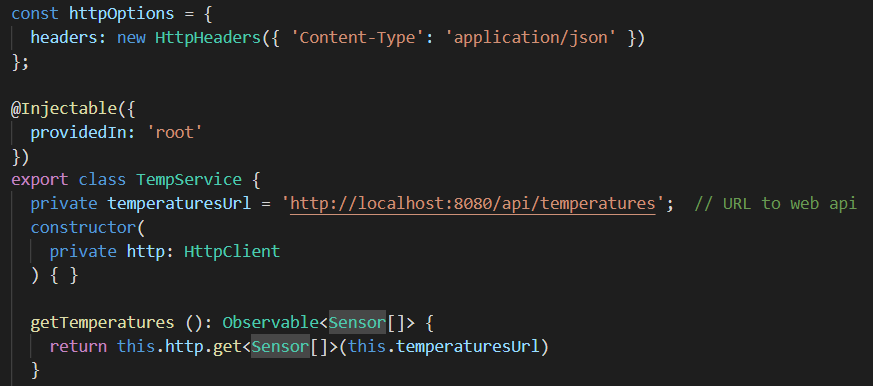


Grâce à ses trois capteurs nous avons des informations précises, complètes et interprétables du patient. En effet, le médecin traitant du patient peut traiter le cardio du patient, ses positions effectuées pendant les différents exercices ainsi que sa température corporelle.

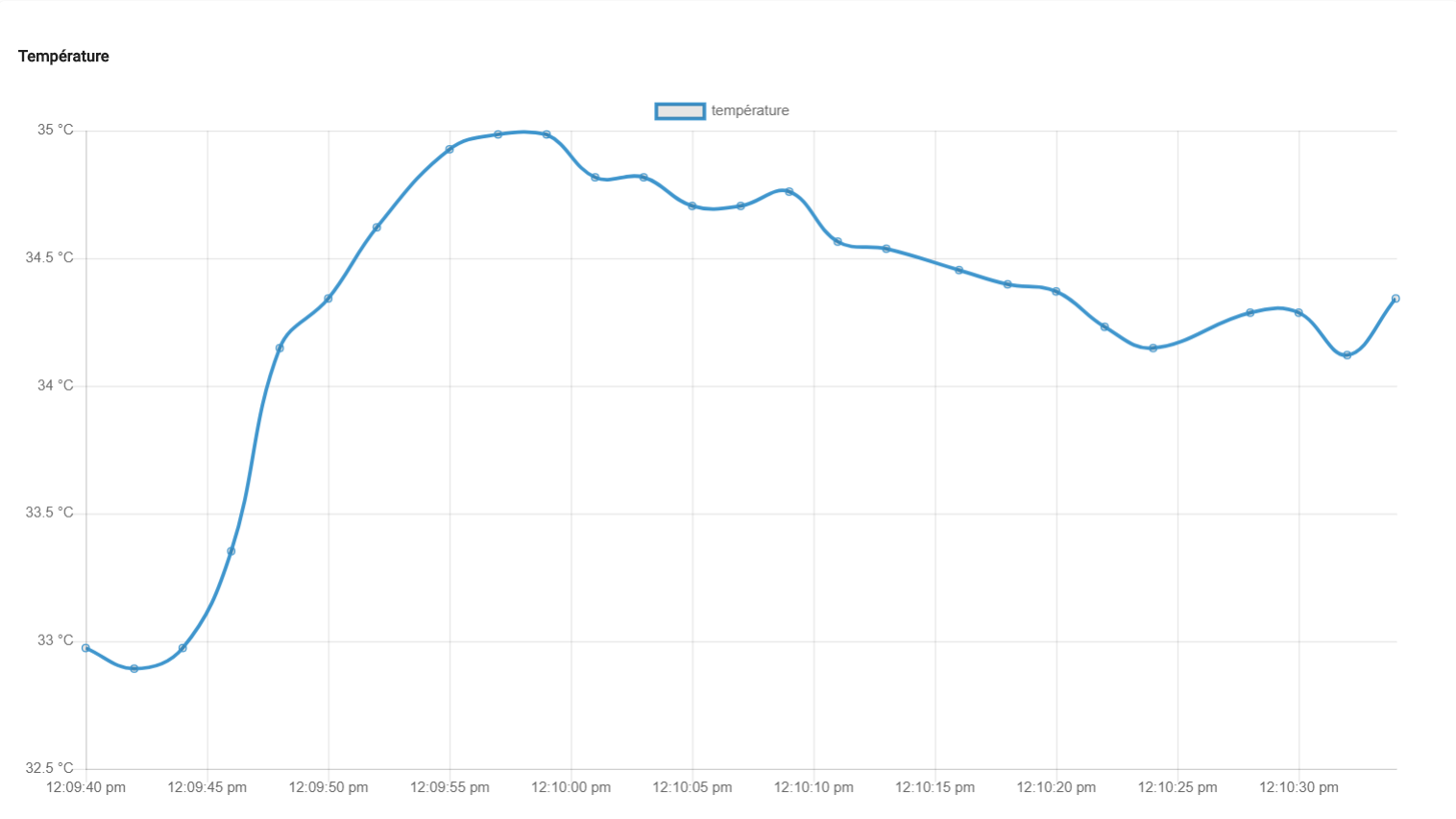
# Fonctionnalités réalisées

Notre site web permet l’affichage des différents graphiques souhaités par le patient et le médecin.

Le site fait des requêtes Get auprès de notre api, le client MySQL récupère ainsi toutes les valeurs enregistrées et nous les transmet.



Ceci nous permet d’obtenir toutes les valeurs passées et de les exploiter sous forme de tableaux représentant les données dans le temps.



Nous avons mis en place un système « d’intervalle » qui nous permet de relancer les requêtes à l’API toutes les 3 secondes. Cela nous permet d’obtenir les valeurs des capteurs en casi-temps réel. Les graphes sont alors mis à jour, affichant ainsi les nouvelles données.



# Améliorations du projet + fonctionnalités non réalisées

En termes d’amélioration, nous pourrions demander aux utilisateurs de se connecter et de créer des sessions pour que chaque utilisateur ait un compte personnel. Nous pourrions différencier l’interface web d’un utilisateur médecin qui pourrait accéder aux dossiers de tous ses patients, d’une interface patiente où il verrait seulement ses résultats et ses communications avec son médecin traitant. Nous pourrions également créer un système d’alerte pour le médecin en cas de problème de santé du côté du patient ou seulement une alerte pour signaler une nouvelle activité d’un patient ou encore une notification pour un nouveau message reçu sur la plateforme web.

# Apports personnels du projet et répartition des tâches

Ce projet IoT est un projet très complet. En effet, il nous a permis de manipuler une Raspberry Pi 3, des outils pour créer le front-end et le back-end d’un site web. Nous avons construit une architecture 3-tier en entier. Nous avons appris à planifier notre projet à l’aide de user-stories détaillées, de se répartir le travail en binôme suivant les tâches à faire et de s’améliorer dans les concepts et la technique de l’implémentation d’une architecture web pour des capteurs IoT.

En ce qui concerne la répartition du travail, Antoine a été responsable de la partie front-end de l’application, Gatien a été responsable de récupérer les données des différents capteurs et nous avons travaillé conjointement pour créer la partie back-end.