

**SG01 – Makair**

*Gaborit Paul*

**Dossier technique du projet – partie individuelle**

<b>I - SITUATION DANS LE PROJET .....</b>	<b>2</b>
INTRODUCTION .....	2
SYNOPTIQUE DE LA REALISATION.....	5
<b>II - REALISATION DU CAS D'UTILISATION UCC1.....</b>	<b>6</b>
CONCEPTION DETAILLEE.....	6
TESTS UNITAIRES .....	11
I.1.1 - Test unitaire du module logiciel .....	11
I.1.2 - Problèmes rencontrés.....	12
<b>II - REALISATION DU CAS D'UTILISATION UCC6.....</b>	<b>13</b>
CONCEPTION DETAILLEE.....	13
.....	16
TESTS UNITAIRES .....	17
II.1.1 - Test unitaire du module matériel/logiciel xxx .....	17
II.1.2 - Problèmes rencontrés.....	19
<b>III - BILAN DE LA REALISATION PERSONNELLE .....</b>	<b>20</b>

Ce document doit comporter entre 20 et 30 pages.

## I - Situation dans le projet

### Introduction

MakAir est utilisé dans un contexte hospitalier. Dans ce contexte, plusieurs patients peuvent être sous assistance respiratoire.

Il se pose alors un problème de surveillance de plusieurs patients simultanément et de la gestion des alarmes associées aux patients.

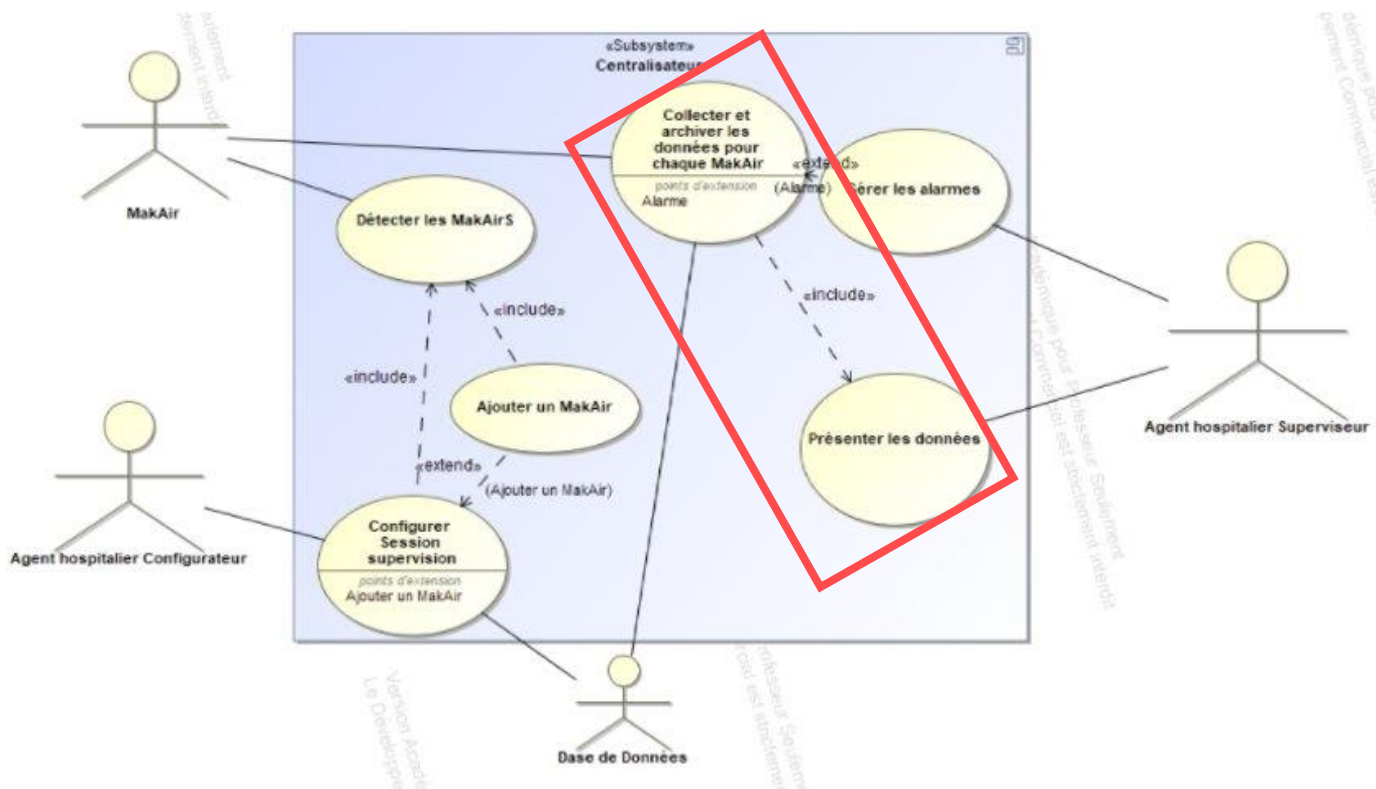
**Permettre la supervision simultanée et à distance d'un certain nombre (non limité et défini par l'utilisateur) de MakAirs.**

	Fonctions à développer et tâches à effectuer
<b>Étudiant 4 (GABORIT)</b> XX EC <input type="checkbox"/> IR <input checked="" type="checkbox"/>	<div> Liste des fonctions / UCs assurés par l'étudiant : </div> <div> <b>Centralisateur</b> <ul style="list-style-type: none"> <li>UCC1 – Collecter et archiver les données pour chaque MakAir</li> <li>UCC6 – Présenter les données</li> </ul> </div> <div> <b>Installation :</b> <ul style="list-style-type: none"> <li>✓ Installation du serveur web.</li> <li>✓ Installation de la bibliothèque graphique</li> </ul> </div> <div> <b>Mise en œuvre :</b> <ul style="list-style-type: none"> <li>✓ Avec l'étudiant 1 définir le format des données à recevoir.</li> <li>✓ Avec l'étudiant 3, définir le format d'une session (nombre de MakAirs, données à présenter etc...).</li> <li>✓ Librairie graphique HighCharts.</li> </ul> </div> <div> <b>Configuration :</b> <ul style="list-style-type: none"> <li>✓ Configurer le serveur web.</li> </ul> </div> <div> <b>Réalisation :</b> <ul style="list-style-type: none"> <li>✓ Définir la/les IHMs associées aux Use Cases</li> <li>✓ Codage des classes et des IHMs associées aux Use Cases.</li> </ul> </div> <div> <b>Documentation :</b> <ul style="list-style-type: none"> <li>✓ Procédure de configuration du serveur web</li> <li>✓ Dictionnaire de classes</li> </ul> </div>

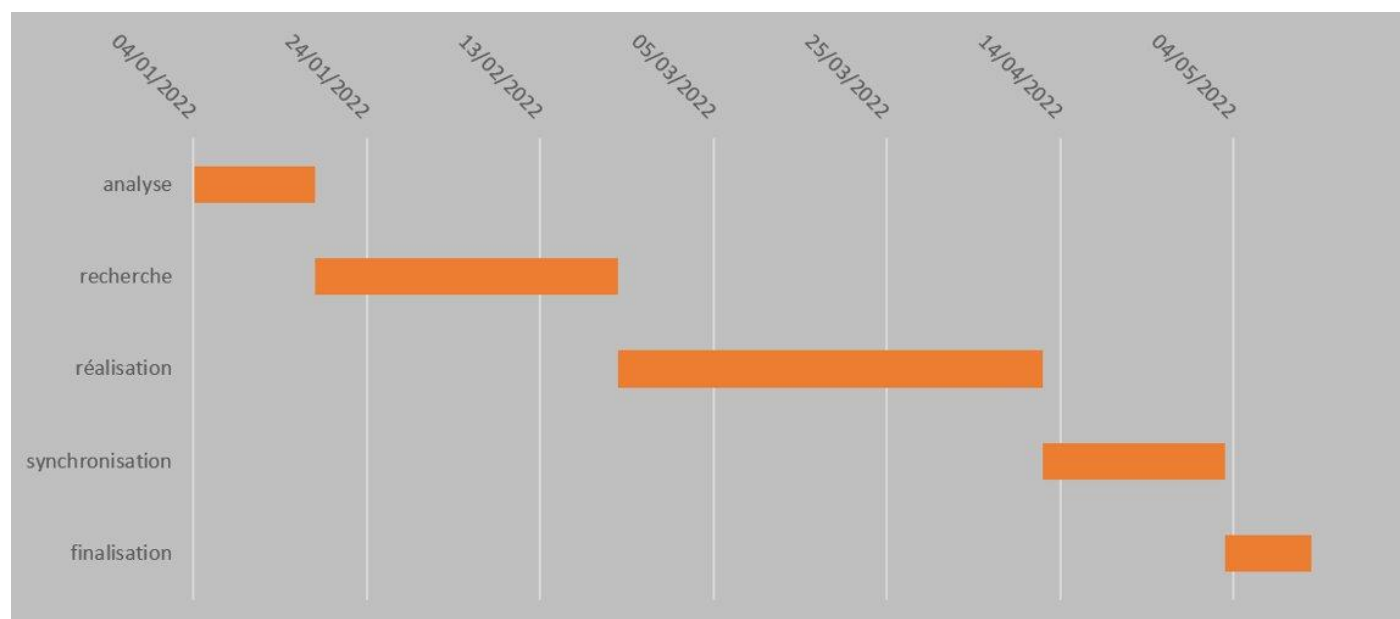
<b>Nom du cas d'utilisation</b>	<b>UCC1 – Collecter et archiver les données pour chaque MakAir</b>
<b>Pré-condition(s)</b>	Les différents MakAirs ont été détectés et une session de collecte est définie.
<b>Scénario nominal</b>	Le Use Case collecte et archive les données de tous les MakAirs faisant partie de la session.
<b>Séquencement</b>	En boucle (à la période T) - Pour tous les MakAirs de la session, collecter et archiver les données
<b>Post-condition</b>	Les données sont correctement collectées et enregistrées
<b>Exigences</b>	La collecte de toutes les données est effectuée à la période T

<b>Nom du cas d'utilisation</b>	<b>UCC6 – Présenter les données</b>
<b>Pré-condition(s)</b>	Les données sont disponibles.
<b>Scénario nominal</b>	Le serveur web assure la présentation des données de façon à ce qu'elles puissent être visualisées sur n'importe quel navigateur web. Les données peuvent être présentées sous forme graphique ou tabulaire.
<b>Séquencement</b>	Les données sont remises à jour à la période T.
<b>Post-condition</b>	
<b>Exigences</b>	

On peut voir dans le diagramme des cas d'utilisations les deux cas qui me concernent.



Pour réaliser mes deux cas d'utilisations, il a fallu s'organiser de la façon suivante :



étapes	Date de début	Durée	Date de fin
analyse	04/01/2022	14	18/01/2022
recherche	18/01/2022	35	22/02/2022
réalisation	22/02/2022	58	21/04/2022
synchronisation	21/04/2022	21	12/05/2022
finalisation	12/05/2022	10	22/05/2022

Avant de commencer le travail, il y a eu une phase d'analyse de deux semaines environ. Elle m'a permis de comprendre dans les détails ce qu'on demandait, pour le groupe, et pour moi. Ensuite, les phases de recherche et de réalisation se croisaient. Car pour les recherches que je faisais, il fallait tester ce que je trouvais, donc dans la phase réalisation.

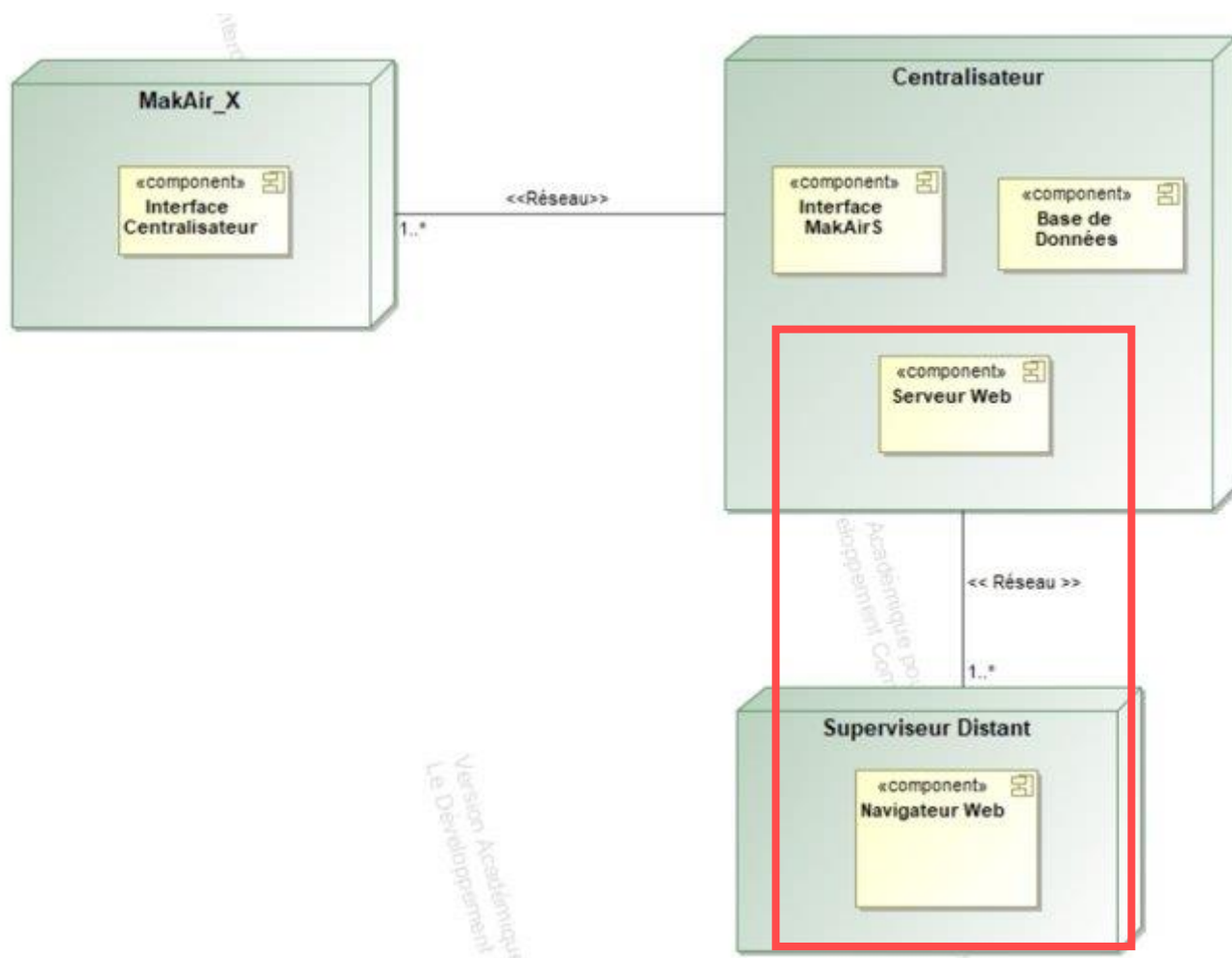
La synchronisation a duré moins que trois semaines car tout était sur le même serveur donc facile à mettre en place.

Et il n'y a pas de finalisation car le projet n'est pas encore terminé et je vais vous expliquer pourquoi par la suite.

## Synoptique de la réalisation

Dans ce projet, le but est de récupérer des données d'un Makair puis de les stocker dans le centralisateur. Ensuite, avec les données stockées, le but est de les afficher sur un navigateur Web depuis un superviseur distant.

Mon but dans le projet est donc dans la deuxième partie, en rouge. Pour réaliser mes UCC1 et UCC6, je me place sur le serveur pour collecter et archiver les données et le navigateur Web pour présenter les données.

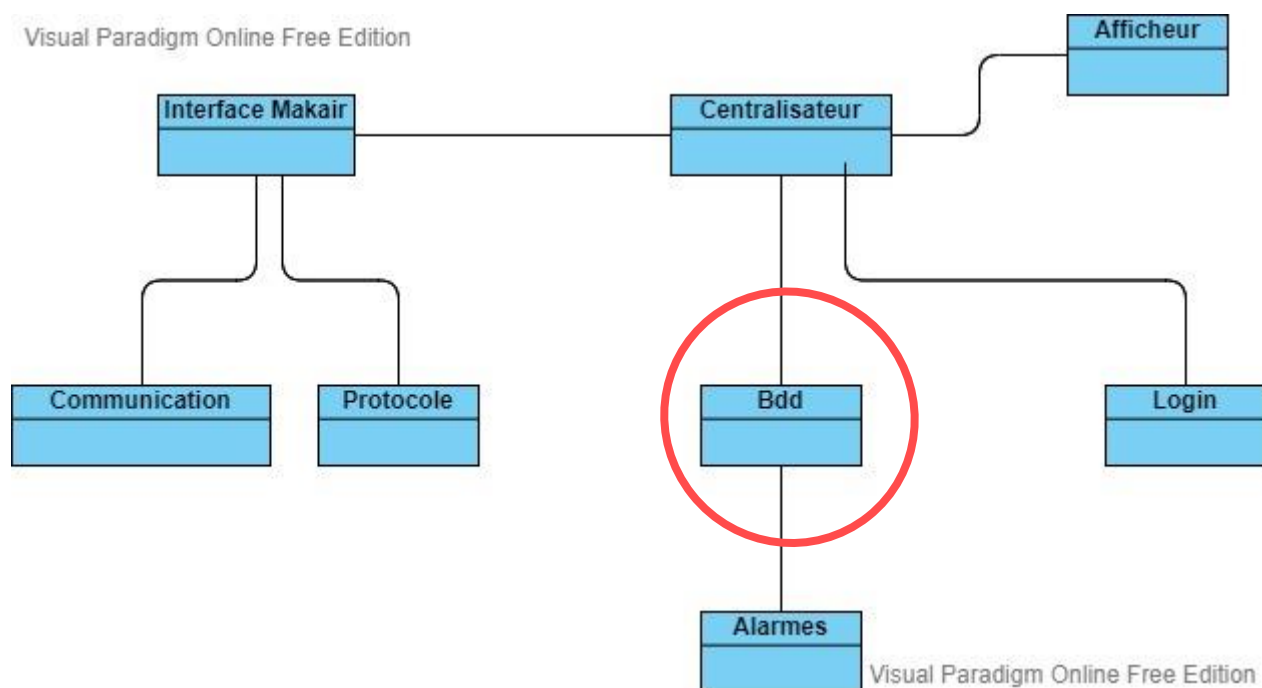


## II - Réalisation du cas d'utilisation UCC1

### Conception détaillée

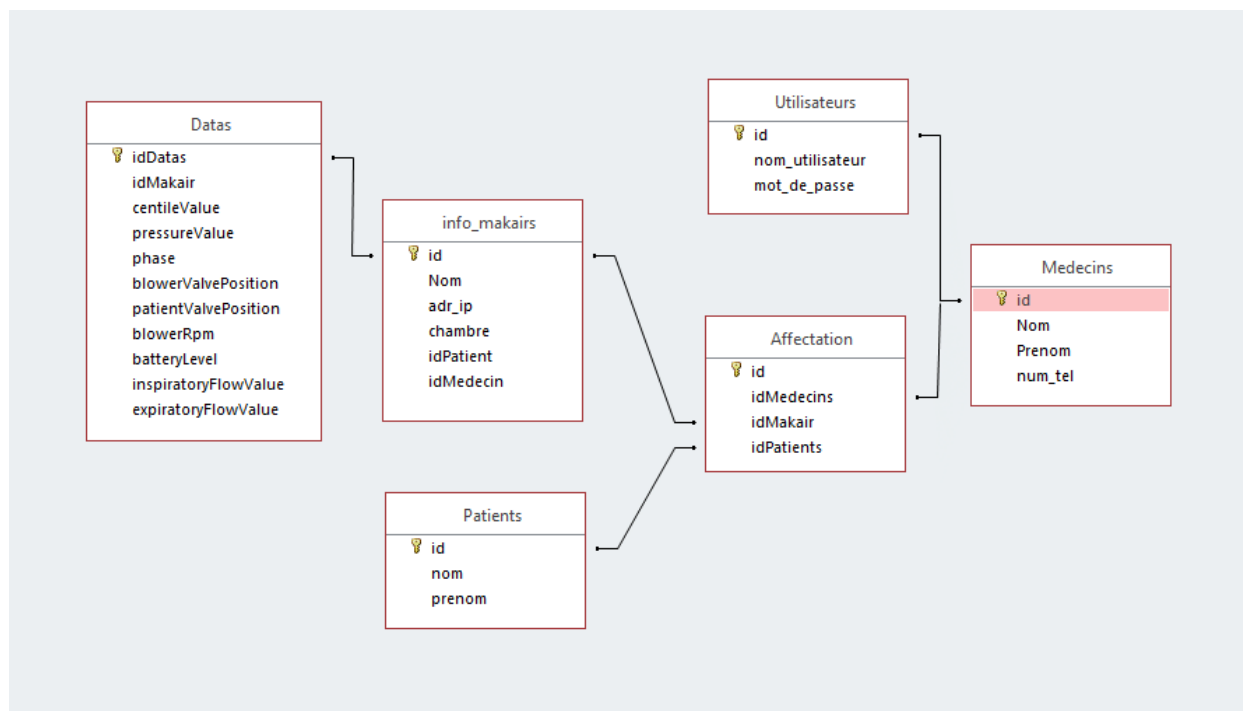
<b>Nom du cas d'utilisation</b>	<b>UCC1 – Collecter et archiver les données pour chaque MakAir</b>
<b>Pré-condition(s)</b>	Les différents MakAirs ont été détectés et une session de collecte est définie.
<b>Scénario nominal</b>	Le Use Case collecte et archive les données de tous les MakAirs faisant partie de la session.
<b>Séquencement</b>	En boucle (à la période T) - Pour tous les MakAirs de la session, collecter et archiver les données
<b>Post-condition</b>	Les données sont correctement collectées et enregistrées
<b>Exigences</b>	La collecte de toutes les données est effectuée à la période T

Pour le projet, on a réalisé un diagramme de classe comme celui-ci-dessous.



Pour collecter et archiver, la classe la plus importante et celle de la Base de Données « BDD ». Elle va me permettre de me connecter à la base de données et de récupérer les valeurs médicales du MakAir.

Voici la structure de la base de données réalisé par l'étudiant 3.



La table « Datas » est la table qui va nous intéresser. C'est dans cette table que les données seront archivées pour chaque MakAir.

Ce qui va nous intéresser ici c'est la table « Datas » et les données qui sont dedans.  
Voici ce qu'il y a dans la table Datas et que je vais pouvoir utiliser.

	idDatas	idMakair	centileValue	pressureValue	phase	blowerValvePosition	patientValvePosition	blowerRpm	batteryLevel	inspiratoryFlowValue	expiratoryFlowValue
<input type="checkbox"/> Editer Copier Supprimer	1	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	2	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	3	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	4	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	5	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	6	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	7	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	8	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	9	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	10	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	11	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	12	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	13	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	14	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	15	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	16	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	17	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	18	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	19	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	20	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	21	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	22	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	23	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	24	0	0	8192	50	52	80	140	255	-4608	8448
<input type="checkbox"/> Editer Copier Supprimer	25	0	0	8192	50	52	80	140	255	-4608	8448

On peut voir que plusieurs valeurs sont disponibles pour être utiliser par la suite. Les valeurs de la table se justifient avec le code ci-dessous :

```
void sendDataSnapshot(uint16_t centileValue,
int16_t pressureValue,
CyclePhases phase,
uint8_t blowerValvePosition,
uint8_t patientValvePosition,
uint8_t blowerRpm,
uint8_t batteryLevel,
int16_t inspiratoryFlowValue,
int16_t expiratoryFlowValue) {
    uint8_t phaseValue;
    if (phase == CyclePhases::INHALATION) {
        phaseValue = 17u; // 00010001
    } else if (phase == CyclePhases::EXHALATION) {
        phaseValue = 68u; // 01000100
    } else {
        phaseValue = 0u;
    }
}
```

Pour mieux comprendre ce que les valeurs signifient :

- centileValue (valeur en pourcentage)
- pressureValue (valeur de la pression)
- phase
- blowerValvePosition (position de la vanne de soufflage)
- patientValvePosition (position de la vanne du patient)
- blowerRpm (souffleur en rythme par minute)
- batteryLevel (niveau de la batterie)
- inspiratoryFlowValue (Valeur du débit inspiratoire)
- expiratoryFlowValue (Valeur du débit expiratoire)

Les valeurs sont ajoutées par l'étudiant 1 avec une trame prise sur le MakAir.



Chaque champ correspond à une valeur, repéré au préalable dans une trame sur le MakAir. On retrouve ici les valeurs du champ « pressureValue », les mêmes que les valeurs sorties dans le format vu auparavant.

Les graphiques Highcharts utilisent les données en format JSON. Comme on peut le voir ci-dessus, les données que je ressors sont dans le format demandé. Ce qui va me permettre de les afficher par la suite.

Avec une requête, je ressors les enregistrements du champ « pressureValue » sur une page Web. De façon comme ci-dessous :

[8192,8192,8192,8192,8192,8192,8192,!

On peut voir que le tableau ci-dessus est de même type que le format de données qu'utilise les graphiques Highcharts. Maintenant, il faut utiliser ce tableau pour faire un graphique avec les données.

Le code ci-dessus montre la classe Bdd. Elle permet de se connecter et de sortir toutes les valeurs d'un champ. Pour se connecter et faire une requête, on instancie dans le main.

```
class Bdd
{

    public function CoBdd($server, $user, $pwd, $Bdd_name, $requete)
    // public function CoBdd()
    {

        $db = mysqli_connect("$server", "$user", "$pwd", "$Bdd_name"); ///  


        if($db->connect_error)
        {
            die('Erreur : ' . $db->connect_error); ///  

        }
        echo 'Connexion réussie';          ///  


        $stmt=mysqli_query($db,$requete);
        echo mysqli_num_rows($stmt)." ce qu'il y a dans la table";
        echo "<hr />";
        $json = [];
        while($row=mysqli_fetch_assoc($stmt))
        {
            //echo $row["centileValue"];
            //echo $row["pressureValue"];
            //echo "<br />";

            extract($row);
            $json []= (int)$phase;
            //(int)$centileValue
        }

        $stmt->free(); ///  

        $db->close();    ///  

        return $json;
    }
}
```

Dans le code ci-dessus, il y a une phase de connexion avec la variable « \$db » avec le nom du serveur, le nom d'utilisateur, le mot de passe de la Base de Données et la requête. Ensuite on vérifie si on s'est bien connecté. Le but ensuite est de créer un tableau avec les valeurs de la Base de Données.

La requête permet de sortir les valeurs d'un champ. On peut voir comme au-dessus, que la fonction sort les données.

## Tests unitaires

### I.1.1 - Test unitaire du module logiciel

Élément testé :	Connexion à la base de données			
Objectif du test :	Réussir la connexion à la Bdd			
Nom du testeur :	Paul Gaborit		Date :	02/03/2022
Moyens mis en œuvre :	Logiciel : VS Code	Matériel : Html/ Php		Outil de développement : VS Code
Procédure du test : Se connecter à la Bdd puis sortir des valeurs				
Id	Description du vecteur de test	Résultat attendu	Résultat obtenu	Validation (O/N)
1	Connexion à la Bdd	« Connexion réussie »	« Connexion réussie »	O
Conclusion du test :		La connexion à la Bdd est réussie.		

Ce test est la première partie du projet. C'est-à-dire de se connecter à la base de données.

Élément testé :	Connexion à la base de données			
Objectif du test :	Sortir des valeurs			
Nom du testeur :	Paul Gaborit		Date :	09/03/2022
Moyens mis en œuvre :	Logiciel : VS Code	Matériel : Html/ Php		Outil de développement : VS Code
Procédure du test : Se connecter à la Bdd puis sortir des valeurs				
Id	Description du vecteur de test	Résultat attendu	Résultat obtenu	Validation (O/N)
1	Sortir Valeurs	Valeurs de la Bdd	Valeurs de la Bdd	O
Conclusion du test :		La connexion à la Bdd est réussie et on sort les valeurs.		

Le deuxième test permet de récupérer les données.

### I.1.2 - Problèmes rencontrés

Avoir les noms d'utilisateurs, mot de passe, nom du serveur et nom de la Base de Données dans la classe. Le but était d'avoir tous les noms dans le 'main' pour ensuite pouvoir réutiliser la classe pour n'importe quelle autre projet.

J'ai donc instancié dans le 'main' comme ci-dessous :

```
<?php
include "Bdd.php";

$ConnectedBdd = new Bdd;

$server = "localhost";
$user = "makair";
$pwd = "makair";
$Bdd_name = "makair";
$requete = "SELECT * FROM Datas";

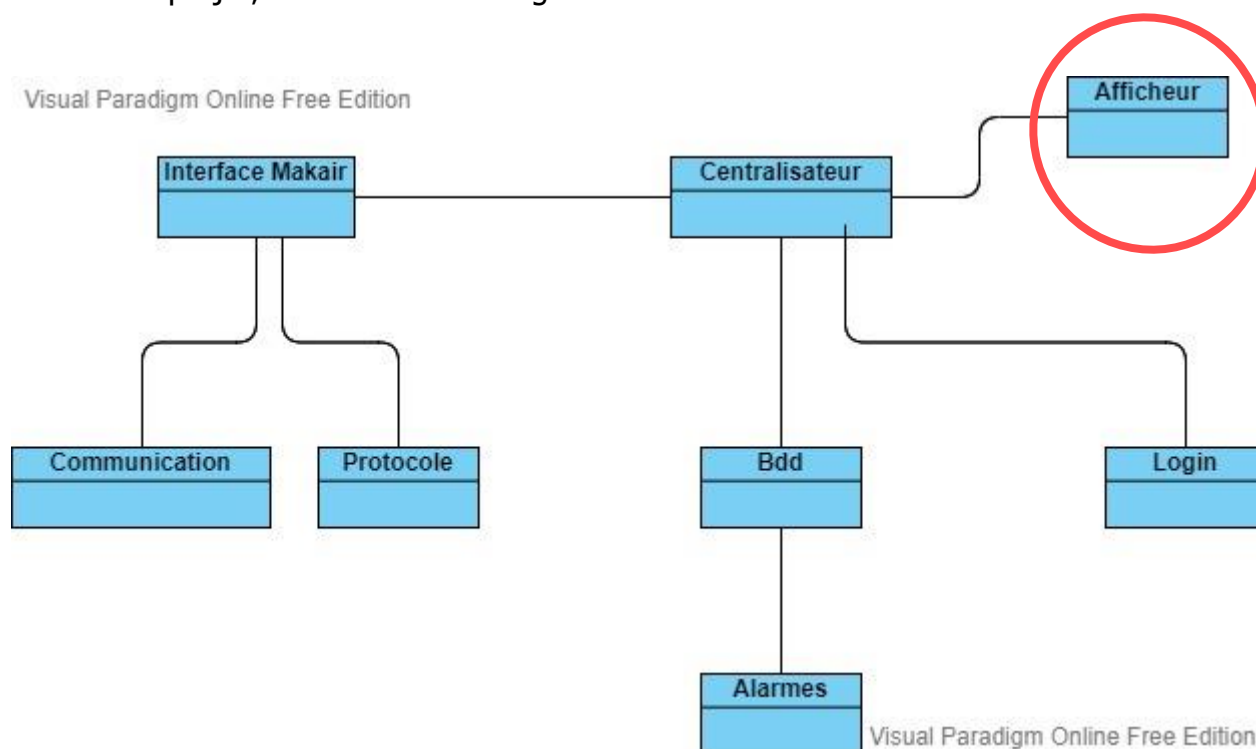
$values = $ConnectedBdd->CoBdd(
    $server,
    $user,
    $pwd,
    $Bdd_name,
    $requete
);
?>
```

## II - Réalisation du cas d'utilisation UCC6

<b>Nom du cas d'utilisation</b>	<b>UCC6 – Présenter les données</b>
<b>Pré-condition(s)</b>	Les données sont disponibles.
<b>Scénario nominal</b>	Le serveur web assure la présentation des données de façon à ce qu'elles puissent être visualisées sur n'importe quel navigateur web. Les données peuvent être présentées sous forme graphique ou tabulaire.
<b>Séquencement</b>	Les données sont remises à jour à la période T.
<b>Post-condition</b>	
<b>Exigences</b>	

### Conception détaillée

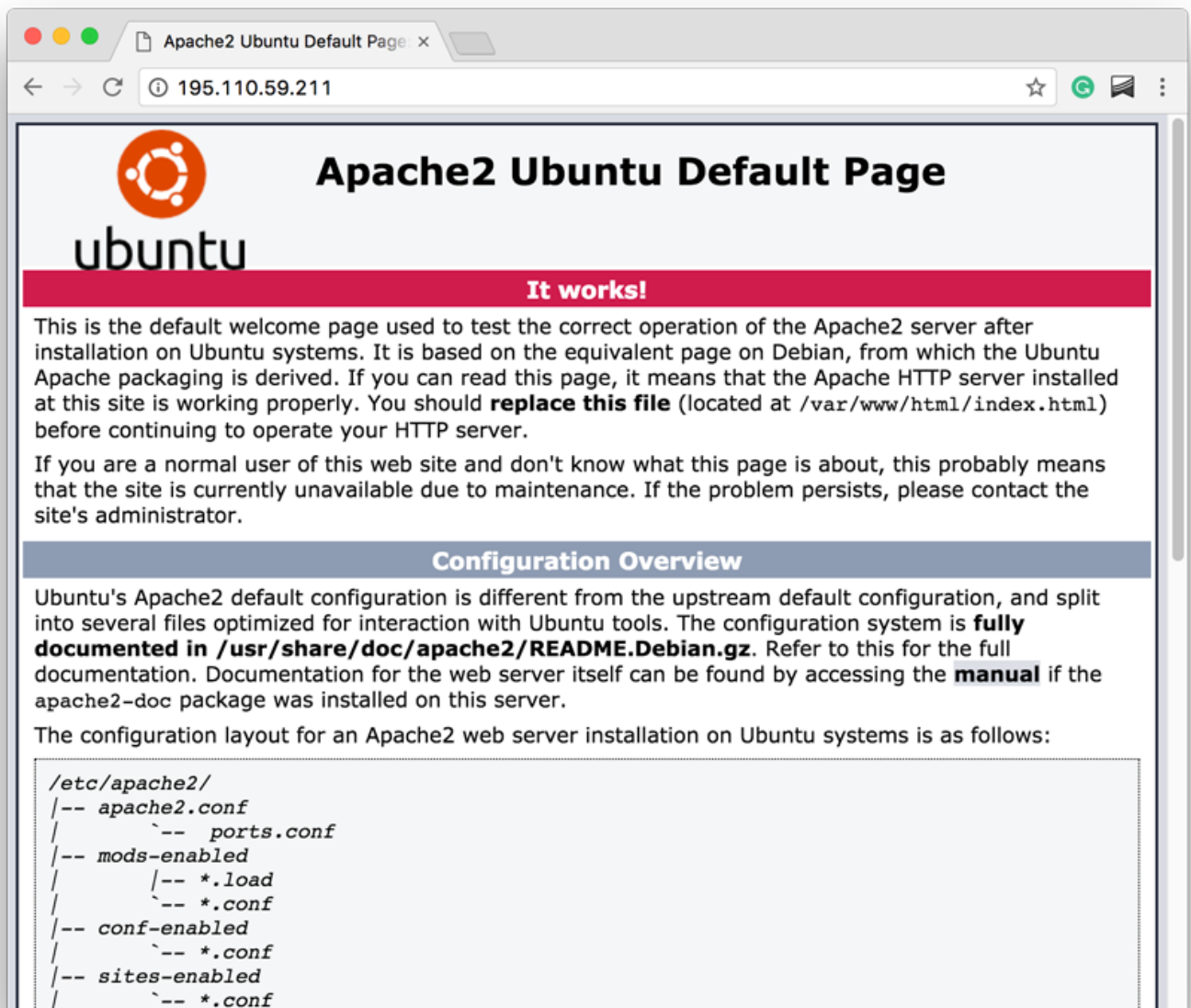
Pour le projet, on a réalisé un diagramme de classe comme celui-ci-dessous.



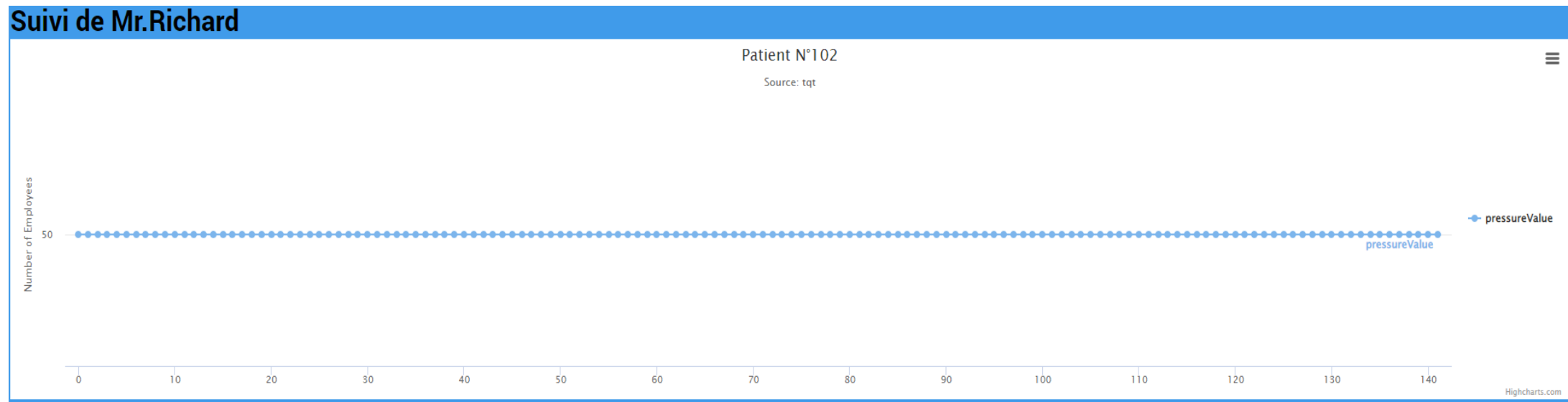
Pour présenter les données, la classe la plus importante et celle du navigateur web « afficheur ». Cette classe permet de récupérer les données et de les afficher.

La page de base du serveur LAMP est celle ci-dessous.

Le but est de la changer pour avoir une page web qui permet d'afficher les valeurs du MakAir quand on se connecte au serveur.



Voici la nouvelle page Web du serveur lorsqu'on se connecte. Elle permet bien d'afficher la valeur d'un patient.



L'IHM montre ce que l'on veut voir. C'est-à-dire, les valeurs de la base de données dans un graphique. Ce qui permet de visualiser les constantes du patient.

Voici quelques extraits de code de la classe « afficheur » :

```
const values = document.currentScript.getAttribute('arrayValues').split(",").map(Number)
Highcharts.chart('container', {
  chart: {
    type: 'line'
  },
  -
```

La première ligne du code est la ligne qui permet de récupérer les valeurs de la classe « Bdd ».

```
series: [{
  name: 'pressureValue',
  data: (function () {
    // generate an array of data
    var datas = values
    // time = (new Date()).getTime(),
    // i;
```

Ce bout de code permet de générer un tableau des données de la Base de Données que la bibliothèque Highcharts utilisera pour faire un graphique.



## Tests unitaires

### II.1.1 - Test unitaire du module matériel/logiciel xxx

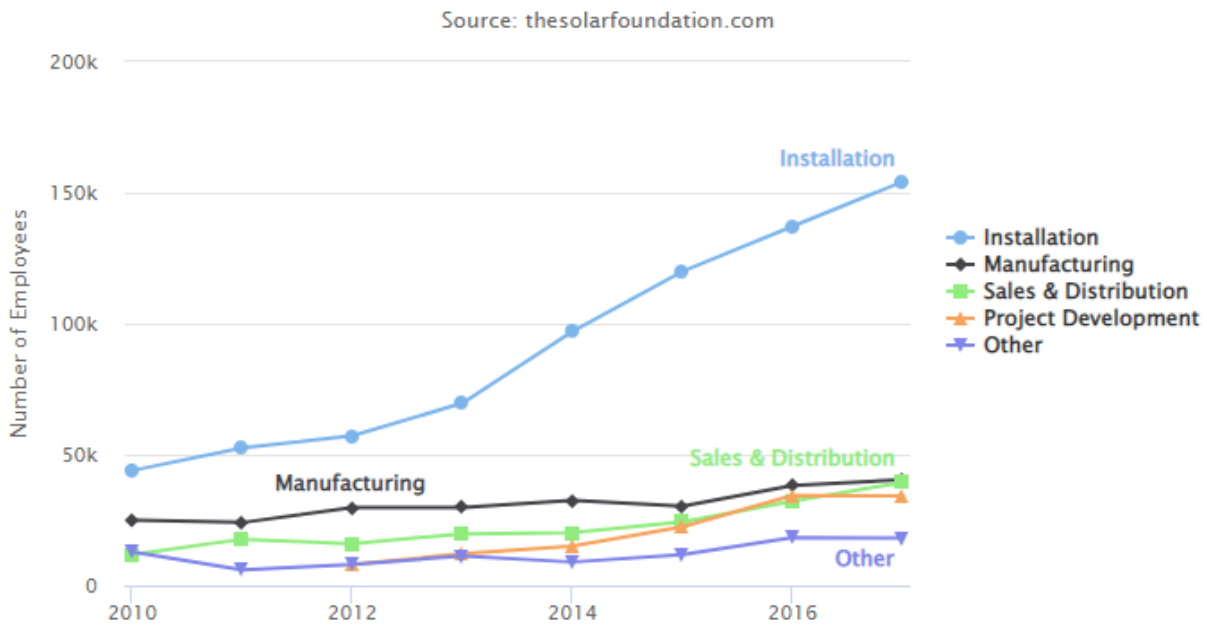
Élément testé :	Diagramme dynamique			
Objectif du test :	Réaliser un diagramme dynamique avec n'importe quelle valeur.			
Nom du testeur :	Paul Gaborit		Date :	28/02/2022
Moyens mis en œuvre :	Logiciel : VS Code	Matériel : Html/Js	Outil de développement : VS Code	
Procédure du test : tester diagramme par diagramme pour finir par le dynamique				
Id	Description du vecteur de test	Résultat attendu	Résultat obtenu	Validation (O/N)
1	Diagramme en ligne stable avec valeur connu	Diagramme en ligne sur page html	Diagramme en ligne sur page html	O
2	Diagramme en bar stable avec valeur connu	Diagramme en bar stable sur page html	Diagramme en bar stable sur page html	O
3	Diagramme en ligne dynamique	Diagramme dynamique sur page html	Diagramme dynamique sur page html	O
Conclusion du test :	Tous les diagrammes ce sont afficher avec les valeurs demandés. Le diagramme dynamique est fonctionnel.			

Avant de réaliser un tableau avec les données, je voulais faire un graphique avec des valeurs connus qui sont dans le code.

Le dernier test est un test avec des valeurs aléatoires en temps réel (toutes les secondes). C'est ce qui m'intéresse car les données du MakAir arriveront toutes les secondes également. A la fin, il me faudra donc un graphique dynamique.

Voici un bout de code avec des données connus et ce que cela donne sur une page Web.

```
series: [{
  name: 'Installation',
  data: [43934, 52503, 57177, 69658, 97031, 119931, 137133, 154175]
}, {
  name: 'Manufacturing',
  data: [24916, 24064, 29742, 29851, 32490, 30282, 38121, 40434]
}, {
  name: 'Sales & Distribution',
  data: [11744, 17722, 16005, 19771, 20185, 24377, 32147, 39387]
}, {
  name: 'Project Development',
  data: [null, null, 7988, 12169, 15112, 22452, 34400, 34227]
}, {
  name: 'Other',
  data: [12908, 5948, 8105, 11248, 8989, 11816, 18274, 18111]
}],
```



On peut voir pour chaque ligne que les données du graphique sont les mêmes que les données du code.

Élément testé :	Graphique avec les données de la Bdd			
Objectif du test :	Faire un graphique avec les données stable de la Bdd			
Nom du testeur :	Paul Gaborit		Date :	06/03/2022
Moyens mis en œuvre :	Logiciel : VS Code	Matériel : Html/ Php		Outil de développement : VS Code
Procédure du test : Se connecter à la Bdd puis sortir des valeurs				
Id	Description du vecteur de test	Résultat attendu	Résultat obtenu	Validation (O/N)
1	Afficher graphique avec valeurs de la Bdd	Graphique avec données	Pas de graphique	N
2	Afficher graphique avec valeurs de la Bdd	Graphique avec données	Affichage du graphique	O
3	Rajouter une valeur dans la Bdd puis afficher	Graphique avec la nouvelle donnée	Affichage réussie	O
Conclusion du test :		Je peux afficher les valeurs de la Bdd sur un graphique		

Maintenant que je peux faire des graphiques avec des valeurs connus et inconnus, le but est de réaliser un graphique avec les données d'une base de données.

J'ai donc essayé d'avoir un graphique avec les datas, mais le premier essaie n'a pas marché car le format des données n'était pas le bon.

Après avoir réctifié le problème, le graphique est apparu avec les données de la Base de Données.

Je voulais m'assureré que ce que j'avais fait fonctionnait, donc j'ai ajouter une valeur à la base de données moi-même. Et quand j'ai rafraichi la page, la valeur s'était ajouter.

### II.1.2 - Problèmes rencontrés

Pour les graphiques, le début n'était pas simple, ils ne voulaient s'afficher malgré un « bon » code.

Pour réglé ce problème, j'ai télécharger la bibliothèque Highcharts et effectué la mise en place de celle-ci dans le code.

J'ai eu également un problème pour afficher le graphique avec les données de la Bdd.

J'ai donc inclus la classe 'Bdd.php' (qui contient les valeurs de la Bdd) dans la classe 'afficheur.js' (qui affiche les données dans un graphique). Cela restait quand même compliqué et m'a pris beaucoup de temps.

### **III - Bilan de la réalisation personnelle**

Ce qui a été validés est que je devais afficher les données d'un patient sur un PC distant pour faciliter le travail des médecins.

Il me reste à afficher les données en temps réel du patient. J'attends donc que l'étudiant 1 arrive à insérer les datas du MakAir en temps réel (toutes les secondes) dans la Base de Données pour ensuite les utiliser dans un graphique.

En termes d'améliorations possibles, il serait bien d'avoir une meilleure ergonomie sur la page Web.

Du point de vue du projet, la partie est presque terminée. Le but étant d'afficher des valeurs d'une base de données sur un graphique afin de suivre un patient de n'importe quel poste.

D'un point de vue personnel, ce projet ma vraiment confirmé mon orientation et le fait de vouloir faire développeur Web.