

ISEN - 2022/2023 – AP4 – Projet transversal WEB
RAPPORT D'ACTIVITÉ

Supervision Raspberry PI

Groupe 08

NOM Prénom apprenant 1	CLEDELIN Loïc
NOM Prénom apprenant 2	MATANGA Allan
NOM Prénom apprenant 3	ZAJAC Clément
Date de la soutenance	05/05/2023 - 9h30

Sommaire:

1. Introduction.....	3
2. Etat de l'art et problématique.....	4
3. Solution envisagée.....	4
3.1. Présentation de la solution.....	4
3.1.1. Exposition de la proposition.....	4
3.1.2. Périmètre fonctionnel.....	5
3.1.3. Choix techniques (matériels et logiciels).....	6
3.1.4. Pilotage du projet.....	7
3.1.4.1. Estimation des charges.....	7
3.1.4.2. Planning prévisionnel.....	8
3.1.4.3. Communication.....	8
3.1.5. Coûts de développement.....	8
3.2. Analyse fonctionnelle.....	9
3.2.1. Détail des fonctionnalités.....	9
3.2.2. Maquettes.....	10
Vous trouverez joint à ce document notre document maquette pour le projet "Maquettes pages.jpg" rendant compte des maquettes travaillées lors du cours de UX/UI qui définissent la disposition de nos pages.....	10
3.3. Analyse technique.....	10
3.3.1. Architecture technique (interfaçage entre les composants, flux...).....	10
3.3.2. Modélisation de la data (données, flux...).....	11
3.4. Réalisation.....	12
3.5. Documentation.....	13
4. Résultats obtenus.....	17
4.1. Difficultés rencontrées et solutions.....	17
4.2. Respect des délais.....	17
4.3. Respect du périmètre fonctionnel.....	17
4.4. Mise en production.....	17
5. Conclusion.....	18
5.1. Montée en compétences de l'équipe.....	18
5.2. Valeur ajoutée pour le client (Junia ou autre).....	18
5.3. Axes d'amélioration.....	18
5.4. Et si c'était à refaire (prise de recul) ?.....	18

1. Introduction

Dans le cadre du module "WEB" de notre quatrième année du cycle d'ingénieur à JUNIA en apprentissage, nous avons dû réaliser une application de supervision sur le domaine de notre choix.

En premier lieu, il nous fallait choisir un sujet, notre groupe étant composé de "tech", nous avons choisi de nous orienter vers un périmètre informatique. Une application de supervision (en informatique) fait référence à une technique de suivi et/ou de pilotage d'un produit. Nous avons voulu nous rapprocher le plus d'un sujet concret, afin d'être confronté à des problématiques que nous pourrions rencontrer dans le milieu professionnel, et de mettre plus facilement en œuvre nos compétences techniques et organisationnelles.

Notre choix s'est porté sur la supervision d'un Raspberry PI. En effet, un des projet d'étude de cette année (mis en place par Marc BAYART, César DUVAL, Etienne EL GUEDER, Allan MATANGA LOVET et Clément ZAJAC) est le "développement d'un programme de simulation des interactions électriques entre des charges isolées", nommé également "SIMEC". Supervisé par monsieur Lambert, ce projet d'étude requiert un raspberry PI. Créer une application de supervision pour un Raspberry est donc un projet concret, avec monsieur Lambert comme client.

La seconde partie du projet (imposé) est de créer un CRUD afin d'alimenter une base de données de test, pour créer des scénarios particuliers (par exemple une surchauffe).

2. Etat de l'art et problématique

Comme dit ci-dessus, un Raspberry est mis à disposition de notre client afin qu'il puisse utiliser un logiciel de simulation. Ce projet est nécessaire pour plusieurs raisons. Premièrement pour prévenir des pannes, en surveillant les performances du Raspberry Pi, il est possible de détecter certains problèmes et ainsi de prendre des mesures pour les éviter. Deuxièmement, pour optimiser les performances de l'ordinateur, en effet, nous pouvons identifier les processus qui ralentissent la machine et traitée cela afin d'améliorer l'efficacité du système. Ensuite, afin d'augmenter la sécurité, il est possible de détecter des comportements suspects ou des tentatives d'intrusion et de prendre des mesures pour renforcer le système. Enfin, pour économiser de l'énergie, en surveillant les performances du raspberry Pi, les applications qui consomment beaucoup d'énergie seront identifiées, en fonction de leur utilité nous pourrons adapter la consommation du raspberry. En somme, ce projet est important afin de garantir la haute disponibilité et la durabilité de la station.

Nous avons conçu la solution dans son intégralité. L'enjeux de cette application de supervision est d'avoir un produit fonctionnel et accessible pour notre client, afin qu'il puisse avoir des indicateurs sur son matériel informatique, sans pour autant avoir de compétences dans ce domaine.

3. Solution envisagée

3.1. Présentation de la solution

3.1.1. Exposition de la proposition

Cette application de supervision permettra à l'utilisateur d'accéder facilement à différentes informations sur l'état de son Raspberry Pi. En surveillant des points critiques tels que la température, le pourcentage de RAM utilisé et le niveau de version de l'OS, l'utilisateur pourra prendre des mesures préventives pour éviter les pannes et les dysfonctionnements.

L'interface utilisateur sera conçue de manière à offrir un accès facile à ces informations. Un tableau de bord clair et concis permettra à l'utilisateur d'obtenir un aperçu rapide de l'état général de son Raspberry Pi. Des pages synthétiques fourniront également des détails plus spécifiques sur les différentes constantes de l'appareil.

Ainsi, l'utilisateur disposera de toutes les informations nécessaires pour surveiller et gérer efficacement son Raspberry Pi, afin de maintenir un fonctionnement optimal et éviter les problèmes potentiels.

Le site sera hébergé sur un serveur Junia, le site sera donc accessible sur le réseau de l'établissement.

3.1.2. Périmètre fonctionnel

Notre objectif étant de surveiller l'état des indicateurs de fonctionnement du matériel. Nous avons divisé notre périmètre en quatre catégories: sécurité, logiciel, système et réseau. Chaque catégorie fait l'état de plusieurs indicateurs liés au fonctionnement, à l'intégrité ou à la sécurité d'un raspberry.

Nous avons effectué cette catégorisation comme suit:





 Sécurité <ul style="list-style-type: none">• Connection BDD:<ul style="list-style-type: none">◦ Nombre de connexion◦ <u>Liste des utilisateurs connectés</u>• Connection au site:<ul style="list-style-type: none">◦ Nombre de d'utilisateur(s) connectés◦ <u>Liste des utilisateurs connectés</u>• Liste des personnes connectées pour:<ul style="list-style-type: none">◦ Physique◦ SSH◦ VNC• Une mise à jour du système est elle disponible	 Logiciel <ul style="list-style-type: none">• Accès à la BDD• <u>Accès au site</u>• Logs d'erreurs• Nombre de simulations• Une mise à jour du logiciel est elle disponible
 Système <ul style="list-style-type: none">• <u>Température</u>• <u>CPU</u>• RAM• <u>Stockage disponible</u>• Alimentation	 Réseau <ul style="list-style-type: none">• Accès à la Raspberry Pi est il possible pour:<ul style="list-style-type: none">◦ VNC◦ SSH◦ <u>Ping</u>• <u>Bande passante</u>

Figure 1 : Catégorisation de nos indicateurs

Nous prévoyons également un onglet "paramètre" pour gérer les utilisateurs, en ajouter, supprimer, ou administrer les parties de l'application auquel chaque utilisateur à accès.

Dans le cadre de ce projet nous avons rencontré certaines contraintes qui nous ont conduit à définir des priorités.

certain éléments ne seront donc pas disponibles lors de la présentation de notre solution, soit par manque de temps ou parce qu'ils dépendent de l'avancé d'un autre projet pour notre client (SIMEC):

Manque de temps:

- Gestion des utilisateurs et de leurs droits (pour le moment il n'y a qu'un seul utilisateur qui a accès à l'ensemble de l'application).
- Liste des personnes connectés de manière physique, SSH et VNC.
- Alimentation : liste des erreurs d'alimentation.

Nécessite que le projet SIMEC soit terminé:

- Nombre de personnes connectées à la base de données.
- Liste des personnes connectés à la base de données.
- Nombre de personnes connectées au site du projet SIMEC.
- Liste des personnes connectés au site du projet SIMEC.
- Accès au site du projet SIMEC
- La liste des différents logs d'erreurs du projet SIMEC.

3.1.3. Choix techniques (matériels et logiciels)

Pour le frontend : React est utilisé (comme imposé) et fonctionne donc avec Node JS. Bootstrap et Canva JS sont deux librairies qui ont été ajoutées, elles permettent respectivement de faire de la mise en forme de contenu (CSS) et de l'affichage de graphiques.

Les libraires moins importantes utilisées:

- "moment" pour ma gestion du "timestamp".
- "dotenv" pour l'utilisation des fichiers .env

Pour le backend : Un serveur Node JS est aussi utilisé, avec le framework Express.

Les libraires suivantes ont été utilisées pour répondre à des problématiques techniques:

- axios : requêtes HTTP pour récupérer les informations sur un site externe.
- bcryptjs : chiffrement des mots de passe.
- body-parser : récupération de
- cheerio:requêtes HTTP pour récupérer les informations sur un site externe.
- cors : résolution du problème 'CORS' impactant me requête API d'une application à une autre.
- dotenv : pour l'utilisation des fichiers .env
- jsonwebtoken : génération de token pour l'authentification et
- mysql : requêtes SQL
- net-snmp : requêtes SNMP pour récupérer les informations du Raspberry Pi
- ping: requêtes IP

Pour la base de données : nous avons choisi MySQL, c'est une base de données relationnelle performante, fiable et facile. De plus, MySQL est une base de données open source populaire et largement utilisée, ce qui facilitera la maintenance de l'application et l'implémentation de nouvelles fonctionnalités à l'avenir

3.1.4. Pilotage du projet

3.1.4.1. Estimation des charges

Selon le plan de charge que nous avons réalisé, la durée nécessaire à ce projet est de 39 heures par personne. Cela dépasse largement les 15h de cours mis à disposition pour ce projet. Un investissement personnel a été nécessaire pour avoir une application en état de marche.

Vous pouvez retrouver ci dessous le détail de notre plan de charge :

Intitulé	Temps en h
Fonctionnel	
Rapport d'activité	4
Document de soutenance (PPT)	4
Documentation technique	2
Documentation fonctionnel	2
Pilotage projet	2
Technique	
Simulation de l'application de supervision	
CRUD	10
Base de données	4
Application de supervision	
Authentification	4
Page "sécurité"	6
Page "logiciel"	6
Page "système"	6
Page "réseau"	6
Page "administrateur"	6
Service de récupération régulier de la MIB	2
Base de données	2
Réserver & configurer une machine JUNIA pour le projet	2
Mise en place du serveur backend (Node JS & MySQL)	4
Mise en place du serveur Frontend	4
Mise en place du GIT	1
Recherche pour récupérer les infos de la MIB	8
Revue de code	1
Commentaire du code	1
Somme de la partie fonctionnel	14
Somme de la partie technique	73
Tests	10,95
Somme brut	97,95
Contingence	20%
Somme totale	117,54
Nombre d'intervenants	3
Temps par intervenant	39,18

3.1.4.2. Planning prévisionnel

La courte durée des séances attribués au projet, l'envergure de ce dernier et les différentes tâches pouvant presque toutes se faire indépendamment les unes des autres, font qu'un diagramme de GanTT n'était pas pertinent à produire.

La somme totale des heures prévisionnelles est de 118, en comptant 8h une journée de travail, le projet s'étale sur 15 jours.

Avec 3 intervenants, 5 jours complets d'intervention seraient nécessaires pour mettre en place la solution dans un environnement professionnel.

Les tâches ont été attribuées au fur et à mesure de l'avancement du projet.

3.1.4.3. Communication

Au début de chaque séance, une petite réunion de 5-10 min était réalisée afin de faire le point sur où en est chaque tâche, et organiser le travail de la séance.

Pour communiquer hors cours, un groupe Discord a été créé afin d'échanger de manière instantanée.

L'outil trello était également utilisé afin de suivre l'avancé des tâches.

3.1.5. Coûts de développement

Dans notre estimation des coûts du projet l'hébergement n'a pas été pris en compte, cependant, nous avons estimé le coût de développement de la solution comme suit :

Nb de jours requis pour le projet		14,6925	
Profil	Nombre	Prix/jour	Tot/profil
Junior	2	250	2 448,75 €
Intermediaire	1	350	1 714,13 €
		Total	4 162,88 €

La phase de développement a été estimée à 4163 € pour 15 jours de projet.

pour cette estimation nous avons utilisé les paramètres suivants :

Allan et Loïc ont été qualifiés comme des profils junior et Clément comme un profil intermédiaire du fait de leurs expériences respectives dans le domaine du développement informatique.

Les prix associés sont en adéquation avec les prix moyens du marché pour une prestation de ce type.

3.2. Analyse fonctionnelle

3.2.1. Détail des fonctionnalités

L'application se découpe en plusieurs parties:

- Sécurité:
 - Nombre d'utilisateurs connectés à la base de donnée MySQL (depuis l'application MySQLWorkbench du serveur)
 - Liste de tous les utilisateurs connectés à la base de donnée MySQL (depuis l'application MySQLWorkbench du serveur)
 - Nombre d'utilisateurs connectés au site du projet SIMEC. Il faut que le backend du projet SIMEC puisse donner par API, le nombre d'utilisateurs connectés.
 - Liste des utilisateurs connectés au site du projet SIMEC. Il faut que le backend du projet SIMEC puisse donner par API, la liste des utilisateurs connectés, dans le but de contrôler à quel moment les utilisateurs utilisent l'application.
 - Liste des utilisateurs connectés physiquement sur le Raspberry PI (un utilisateur est identifié par sa session).
 - Liste des utilisateurs connectés en SSH sur le Raspberry PI (un utilisateur est identifié par sa session).
 - Liste des utilisateurs connectés en VNC sur le Raspberry PI (un utilisateur est identifié par sa session).
- Logiciel:
 - Accès à la base de données, est elle allumée ou éteinte ?
 - Accès au site, le site du projet SIMEC est-il démarré ?
 - Logs d'erreur : Liste des logs d'erreur liés au projet SIMEC.
 - Nombre de simulations : nombre de simulations total du projet SIMEC.
 - Une mise à jour logiciel est-elle disponible ?
- Système:
 - Température : graphique pour afficher la courbe de la température.
 - CPU : graphique pour afficher la courbe d'utilisation du CPU.
 - RAM : graphique pour afficher la courbe d'utilisation de la RAM.
 - Stockage : graphique pour afficher le stockage disponible.
 - Alimentation : Erreurs liées à l'alimentation du Raspberry (Voltage insuffisant ...)
- Réseau:
 - SSH : L'accès au Raspberry par SSH est-il disponible ?
 - VNC : L'accès au Raspberry par VNC est-il disponible ?
 - Ping : Le Raspberry est-il disponible sur le réseau de JUNIA ?
 - Bande passante : valeurs de la bande passante (En Wifi et ethernet)
- Gestion des utilisateurs CRUD:
 - Créer un utilisateur
 - Modifier son mot de passe
 - Supprimer des utilisateurs
 - Gérer les rôles des utilisateurs

3.2.2. Maquettes

Vous trouverez joint à ce document notre document maquette pour le projet "Maquettes pages.jpg" rendant compte des maquettes travaillées lors du cours de UX/UI qui définissent la disposition de nos pages.

3.3. Analyse technique

3.3.1. Architecture technique (interfaçage entre les composants, flux...)

Notre architecture technique est composée de trois éléments clefs :

- Le frontend (interface vu par les utilisateurs)
- Le backend (intermédiaire entre le frontend et la base de données)
- La base de données (système de stockage des données)

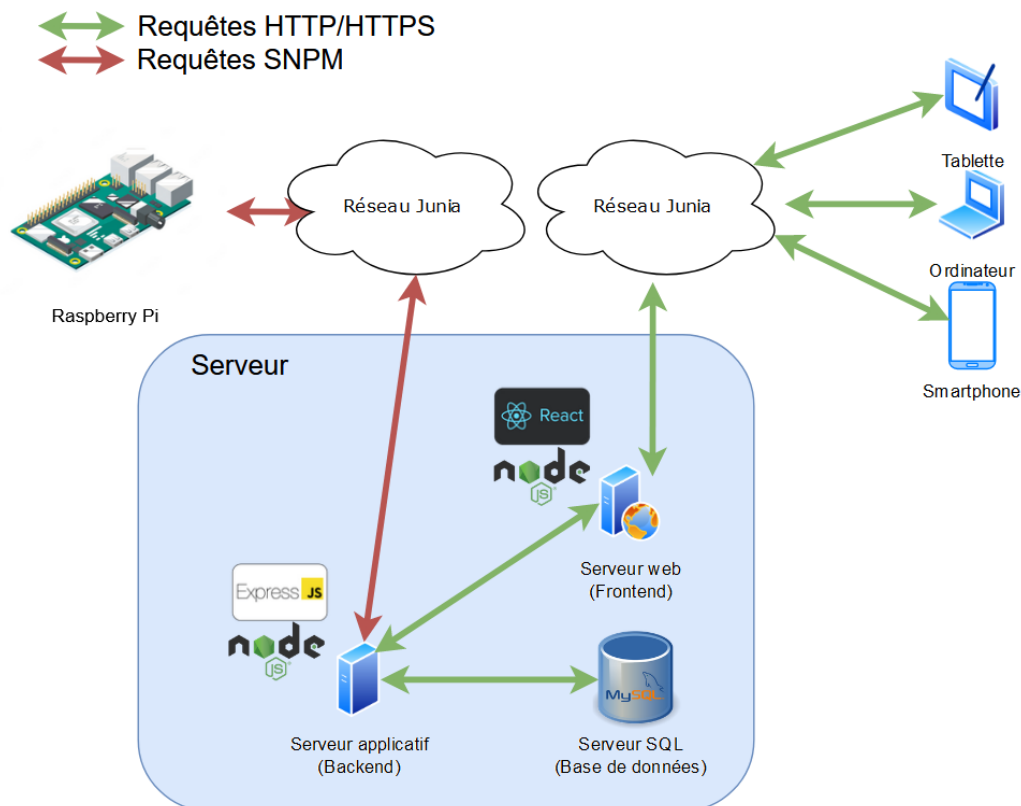
Les interactions entre ces trois éléments se passe de la façon suivante :

Le frontend communiquer par API avec le backend.

Le backend récupère les données de la base de données pour les envoyer au frontend.

La base de données est alimentée par un service hébergé sur le backend, qui fait régulièrement des requêtes SNMP (protocole dédié à la supervision vastement utilisé dans les SI) au Raspberry.

Schéma de l'architecture du projet:



3.3.2. Modélisation de la data (données, flux...)

L'authentification se fait via ta table "users".

Les données sont tirées de la table "raspberry" si le mode sélectionné est "production" c'est-à-dire si les données affichées proviennent du Raspberry.

Les données sont tirées de la table "raspberrysimulation" si le mode sélectionné est "simulé" c'est-à-dire si les données affichées sont fictives, et donc générées par le CRUD.

raspberry users
id : int(11)
username : varchar(255)
password : varchar(255)

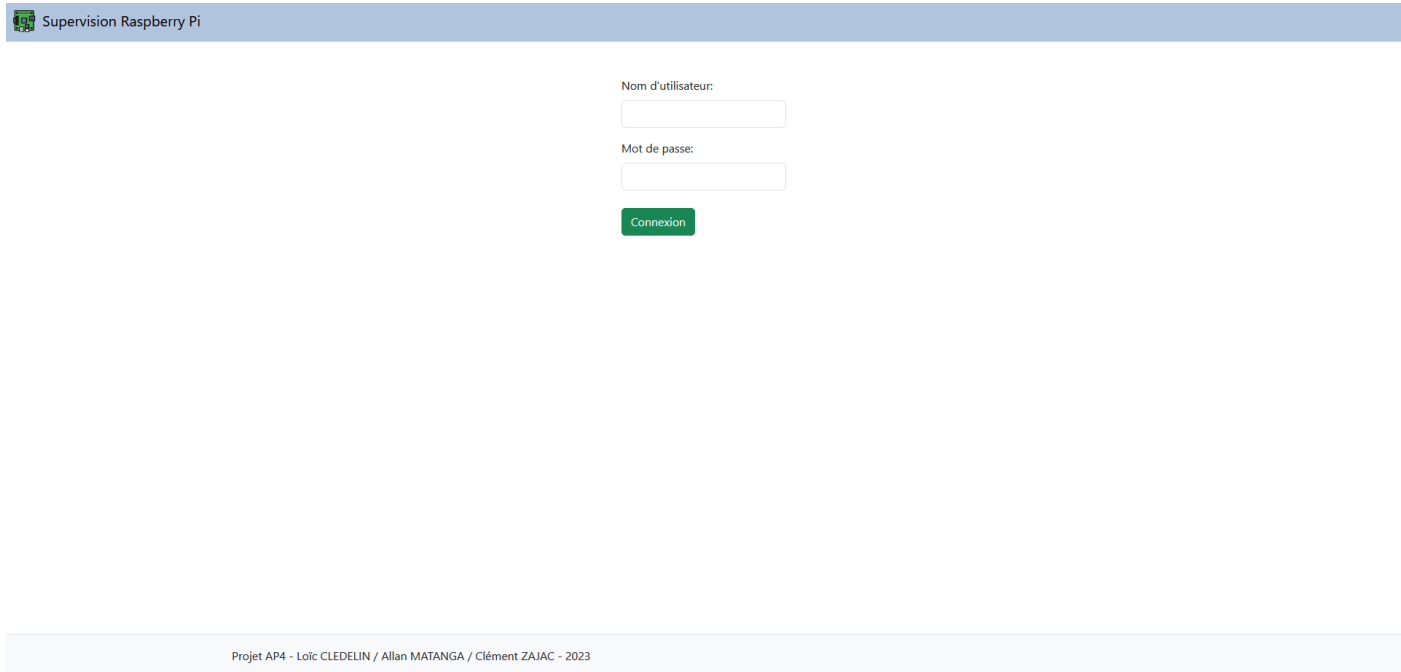
raspberry raspberry
id : int(11)
timestamp : timestamp
userdatabase : int(11)
userwebsite : int(11)
physicallogin : tinyint(1)
sshlogin : tinyint(1)
vnclogin : tinyint(1)
databaselogin : tinyint(1)
websitelogin : tinyint(1)
temperature : double
usagecpu : double
usageram : double
usagestorage : double
maxstorage : double
powersupply : tinyint(1)
ping : int(11)
bandwidthWifi : double
bandwidthFilaire : double
MAJ : int(11)

raspberry raspberrysimulation
id : int(11)
timestamp : timestamp
userdatabase : int(11)
userwebsite : int(11)
physicallogin : tinyint(1)
sshlogin : tinyint(1)
vnclogin : tinyint(1)
databaselogin : tinyint(1)
websitelogin : tinyint(1)
temperature : double
usagecpu : double
usageram : double
usagestorage : double
maxstorage : double
powersupply : tinyint(1)
ping : int(11)
bandwidthWifi : double
bandwidthFilaire : double
MAJ : int(11)

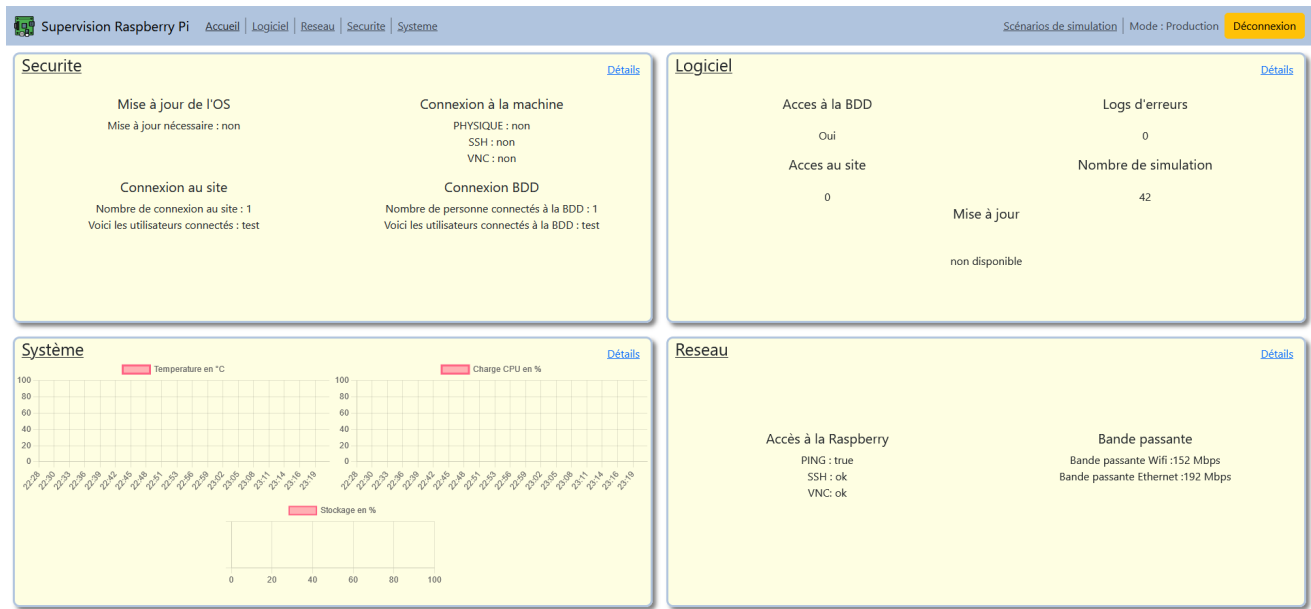
3.4. Réalisation

Notre phase de réalisation s'est correctement déroulée, les objectifs fonctionnels que nous avons définis précédemment sont remplis. Vous pourrez retrouver les captures d'écran de cette partie dans le dossier "Réalisation Captures" joint à ce dossier, afin de pouvoir mieux visualiser les pages si besoin.

L'utilisateur est tout d'abord dirigé vers une page d'authentification.



Sur la page d'accueil nous retrouvons tous nos indicateurs affichés de façon claire et organisée.



Il est possible pour chaque catégorie de passer sur une page plus détaillée qui fournit un peu plus d'information je prends ici l'exemple de la page système.



Nous retrouvons également la page des scénarios de simulation, où nous pouvons changer de mode (Production pour avoir les données récoltées du Raspberry, et Simulé pour charger dans l'application les données générées avec le CRUD).

Le bouton "reset" permet de remettre à zéro toutes les données simulées.

Supervision Raspberry Pi

Accueil | Logiciel | Réseau | Sécurité | Système

Scénarios de simulation | Mode : Production | Déconnexion

Scénarios de simulation

Sur cette page vous aller pouvoir:

- Changer le mode de fonctionnement de l'application : Production (données du Raspberry Pi) ou Simulé (données fictives que vous pouvez créer)
- Créer, modifier, mettre à jour et supprimer le scénario de simulation

Mode actuellement utilisé : Production

Passer en mode Simulé

Mes scénarios de simulation:

Ajouter une ligne

Reset du jeu de test :

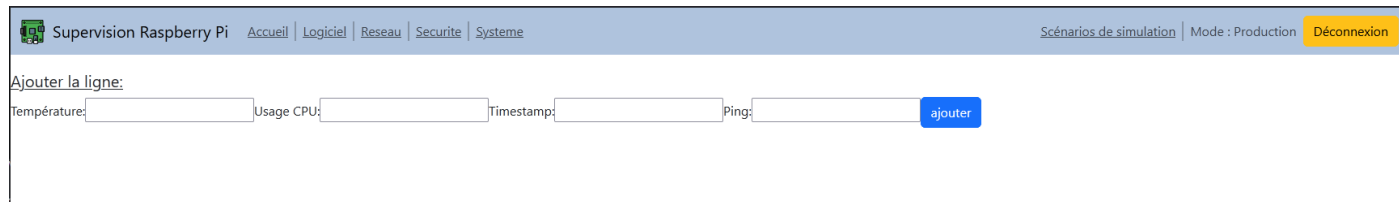
Reset

Données de simulation:

id	timestamp	temperature	usagecpu	ping	Actions
174	01/05/2023 18:56:00	56	56	0	Editer Supprimer
173	01/05/2023 18:55:00	55	55	0	Editer Supprimer
171	01/05/2023 18:53:00	53	53	0	Editer Supprimer
170	01/05/2023 18:52:00	52	52	0	Editer Supprimer
169	01/05/2023 18:51:00	51	51	0	Editer Supprimer
168	01/05/2023 18:50:00	50	50	0	Editer Supprimer
167	01/05/2023 18:49:00	49	49	0	Editer Supprimer

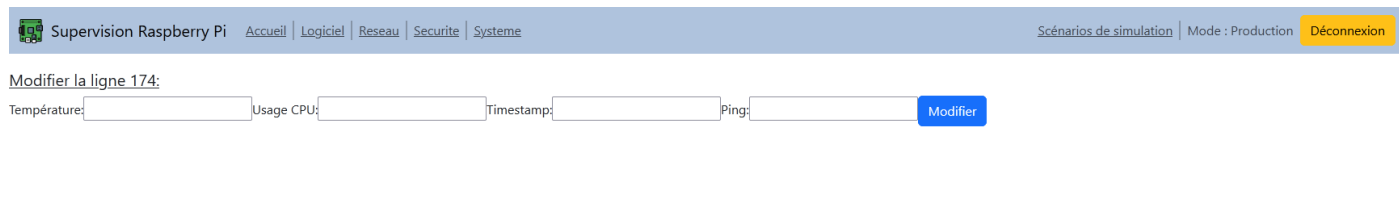
Trois autres actions sont possibles : Ajouter, Éditer, et Supprimer. Ces boutons redirigent l'utilisateur vers d'autres pages, que nous allons décrire ci-dessous.

La page "Ajouter" permet de créer un enregistrement:



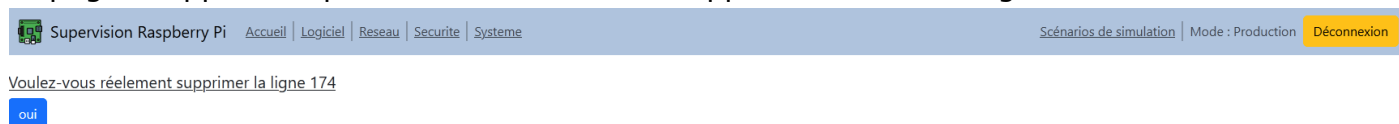
The screenshot shows the 'Ajouter' page of the Supervision Raspberry Pi interface. The header bar includes the title 'Supervision Raspberry Pi' and navigation links: 'Accueil', 'Logiciel', 'Reseau', 'Securite', and 'Systeme'. On the right, it shows 'Scénarios de simulation', 'Mode : Production', and a 'Déconnexion' button. The main content area is titled 'Ajouter la ligne:' and contains four input fields: 'Température:', 'Usage CPU:', 'Timestamp:', and 'Ping:'. A blue 'ajouter' button is positioned to the right of the 'Ping:' field.

La page "Modifier" permet de mettre à jour un enregistrement en particulier:



The screenshot shows the 'Modifier' page of the Supervision Raspberry Pi interface. The header bar is identical to the previous page. The main content area is titled 'Modifier la ligne 174:' and contains four input fields: 'Température:', 'Usage CPU:', 'Timestamp:', and 'Ping:'. A blue 'Modifier' button is positioned to the right of the 'Ping:' field.

La page "Supprimer" permet de confirmer la suppression d'un enregistrement:



The screenshot shows the 'Supprimer' page of the Supervision Raspberry Pi interface. The header bar is identical to the previous pages. The main content area displays the text 'Voulez-vous réellement supprimer la ligne 174' followed by a blue 'oui' button.

3.5. Documentation



Guide d'installation pour la phase de DEV

Table des matières

Table des matières

Installer WAMP & la BDD

Mise en place du 'front'

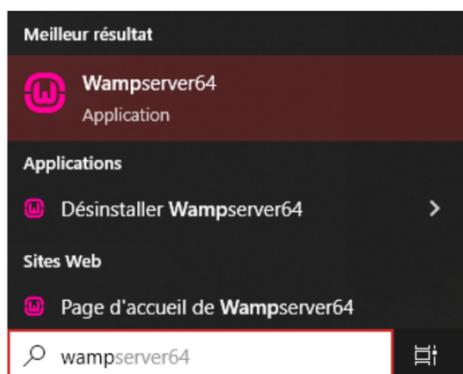
Mise en place du 'back'

Installer WAMP & la BDD



WAMP est une plateforme de développement qui permet de mettre en place un serveur WEB et une BDD en local. Pour ce projet nous utiliseront seulement la partie BDD de WAMP.

1. Cliquez sur le lien suivant, et attendre que le téléchargement se lance: [https://sourceforge.net/projects/wampserver/files/WampServer 3/WampServer 3.0.0/wampserver3.3.0_x64.exe/download](https://sourceforge.net/projects/wampserver/files/WampServer%203/WampServer%203.0.0/wampserver3.3.0_x64.exe/download)
2. Installez WAMP, laissez tout les champs par défaut.
3. Lancez WAMP :



4. Rendez vous sur <http://localhost>
5. En bas à gauche, dans la catégorie 'Outils', cliquer sur 'phpmyadmin'. Ou sur le lien suivant : <http://localhost/phpmyadmin/>

Bienvenu sur PHPMyAdmin ! C'est un outil de gestion de de base de données (SGBD).

Par défaut ⇒ Utilisateur : 'root', mot de passe vide, serveur 'MySQL'



6. Cliquez sur "SQL" en haut, ou sur le lien suivant : http://localhost/phpmyadmin/server_sql.php
7. Copiez le code contenu dans le fichier `bddSupervisionRaspberry.sql` dans l'éditeur SQL.

Exécutez la requête avec le bouton en bas à droite

La base de données est créée ! Si elle ne s'affiche pas à gauche, 'F5' fera de la magie.

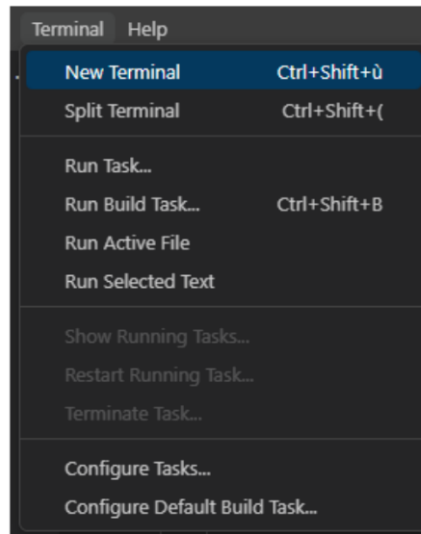
Mise en place du 'front'

1. Clonez le repo suivant:

<https://github.com/ClementZajac/Supervision-Raspberry-Pi-front>

```
modules by path ./node_modules/react-dom/ 1000 KiB 3 modules
modules by path ./node_modules/react-refresh/ 20.2 KiB 2 modules
modules by path ./node_modules/scheduler/ 17.3 KiB
  ./node_modules/scheduler/index.js 198 bytes [built] [code generated]
  ./node_modules/scheduler/cjs/scheduler.development.js 17.1 KiB [built]
+ 7 modules
```


2. Sur VS Code, ouvrir le dossier ainsi qu'un terminal



3. Lancez les commandes suivantes :

```
npm i
```

Puis :

```
npm run start
```

Le serveur front est lancé ! Ici sur <http://localhost:8000/>

```
> Supervision-Raspberry-Pi@1.0.0 start
> webpack serve --hot --mode=development

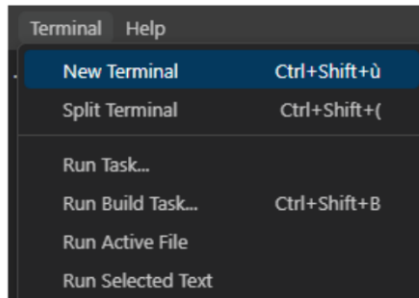
<i> [webpack-dev-server] Project is running at:
<i> [webpack-dev-server] Loopback: http://localhost:8000/
<i> [webpack-dev-server] On Your Network (IPv4): http://192.168.1.171:8000/
<i> [webpack-dev-server] On Your Network (IPv6): http://[fe80::be54:d636:923d:c589]:8000/
<i> [webpack-dev-server] Content not from webpack is served from './' directory
asset app.bundle.js 1.88 MiB [emitted] (name: main) 1 related asset
runtime modules 28.8 KiB 14 modules
modules by path ./node_modules/core-js-pure/ 122 KiB 119 modules
modules by path ./node_modules/@pmmwh/react-refresh-webpack-plugin/ 52.7 KiB 23 modules
modules by path ./node_modules/webpack-dev-server/client/ 55.8 KiB 12 modules
modules by path ./src/ 8.95 KiB 7 modules
modules by path ./node_modules/webpack/hot/*.js 4.59 KiB 4 modules
modules by path ./node_modules/react/ 127 KiB 4 modules
modules by path ./node_modules/html-entities/lib/*.js 81.3 KiB 4 modules
modules by path ./node_modules/react-dom/ 1000 KiB 3 modules
modules by path ./node_modules/react-refresh/ 20.2 KiB 2 modules
modules by path ./node_modules/scheduler/ 17.3 KiB
  ./node_modules/scheduler/index.js 198 bytes [built] [code generated]
  ./node_modules/scheduler/cjs/scheduler.development.js 17.1 KiB [built] [code generated]
+ 7 modules
webpack 5.79.0 compiled successfully in 2625 ms
```

Mise en place du 'back'

1. Clonez le repo suivant:

```
https://github.com/ClementZajac/Supervision-Raspberry-Pi-back
```

2. Sur VS Code, ouvrir le dossier ainsi qu'un terminal



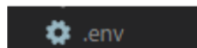
3. Lancez la commande suivante :

```
npm i
```

4. Modifier le fichier `.env.exemple` avec les informations correspondante à la configuration de la base de données de WAMP

Par défaut, il n'y a pas besoin de modifier le fichier.

5. Renommer le fichier `.env.exemple` par `.env`



Tips: les fichiers `.env` ne sont pas partagés sur Git (lorsque le `.gitignore` est correctement configuré). Cela permet d'avoir des variables qui ne sont pas en dur dans le code, et qui peuvent être différent entre les collaborateurs ! Pratique pour garder un mot de passe secret !

6. Lancez la commande suivante :

```
npm run start
```

Le serveur back est lancé ! Ici sur <http://localhost:5000/>

```
> client@0.1.0 start
> node server.js

http://localhost:5000/
Server is running on port 5000
```

4. Résultats obtenus

4.1. Difficultés rencontrées et solutions

Le manque de formation sur la partie backend et sur l'architecture web en règle générale a été un frein pour l'avancée du projet. Pour pallier cela, nous avons organisé des séances de formation animées par Clément, et également travaillé à donner les tâches petit à petit, avec des exemples et explications au préalable.

Les spécificités techniques du React (useState, useEffect ...) n'ayant pas été évoqués en cours, ont fait sujet de recherches et tests afin de bien comprendre le comportement de ces structures de code, pour les implémenter au mieux dans le projet.

4.2. Respect des délais

Le projet ayant une durée limitée, nous avons dû faire face à plusieurs défis pour rendre une première version fonctionnelle et utilisable dans les délais impartis. Cela nous a obligés à travailler sous pression, notamment lors du "rush final". Cela a entraîné des compromis sur certaines fonctionnalités qui n'ont pas été implémentées dans cette première version.

4.3. Respect du périmètre fonctionnel

Les fonctionnalités développées permettent l'utilisation d'une V1 de l'application, avec les composants essentiels mis en place. Il y a d'un côté le projet pour notre client, avec des requêtes sur la MIB du raspberry qui sont ensuite affichées sur le site web. et de l'autre côté le CRUD qui permet une simulation de certaines valeurs du raspberry.

4.4. Mise en production

La solution pourra être poussée en production à la demande de Monsieur Lambert, et si JUNIA peut allouer des ressources afin d'héberger cette application. La solution pourra être un outil pour monsieur Lambert, qui viendra compléter le projet SIMEC, et pourquoi pas servir de base pour le projet à d'autres étudiants.

5. Conclusion

5.1. Montée en compétences de l'équipe

Loïc : durant ce projet que nous avons voulu, significatif, j'ai acquis une expérience concrète en développement web. Que ce soit sur les théories (architectures), que sur les implémentations techniques qui permettent le fonctionnement de ces solutions. Le sujet nécessitant des outils de supervision, j'ai pu perfectionner mes compétences et mes connaissances sur les systèmes permettant ces services comme le SNEP. La prise en main des outils de développement (GIT, visualstudio) hors de mon périmètre de compétence en entreprise a été très instructive et me servira. Ce projet a, dans l'ensemble, été très bénéfique pour moi, je remercie Clément pour sa pédagogie.

Allan: Ce projet m'a permis de découvrir l'ensemble du processus de création d'un projet web, depuis l'idée initiale jusqu'à la mise en production. Sur le plan technique, j'ai acquis de solides compétences dans la construction d'une architecture web, avec une partie front, une partie back et une base de données. J'ai également renforcé mes connaissances en programmation en travaillant avec des langages tels que React, JavaScript et CSS. J'ai conçu une base de données, mis en place des transactions entre celle-ci et le backend, ainsi que des échanges entre le backend et le frontend. De plus, j'ai eu l'occasion de consolider mes compétences en réseau en installant une MIB sur un équipement pour le monitorer.

Travailler en équipe avec deux camarades a été une expérience humaine enrichissante, où nous avons fait preuve d'organisation et de détermination pour mener à bien le projet, même si la solution reste pour le moment en version 1. Je suis confiant dans notre capacité à continuer et à finaliser ce projet.

Clément: J'ai pu monter en compétence sur la gestion de projet et l'encadrement de collaborateur, en me positionnant comme leader sur ce projet. En équipe, j'ai pu me confronter à la différence de niveau sur le plan technique, ce qui m'a amené à vulgariser et former mes camarades. Accompagner Allan et Loïc sur des sujets que je maîtrisais plus, les voir assimiler, et utiliser de plus en plus aisément les technologies que nous avons utilisées tout au long de ce projet, à fait apparaître une forme de satisfaction en moi.

Techniquement, l'apprentissage était plus pauvre pour moi, j'ai pu mettre en application des technologies sur lesquelles j'avais déjà travaillé.

5.2. Valeur ajoutée pour le client (Junia ou autre)

La solution que nous avons développée va permettre à Monsieur Lambert de surveiller à distance l'état de son Raspberry Pi, grâce à une interface web accessible sur le réseau Junia. Il pourra ainsi avoir des indicateurs en temps réel sur des éléments critiques tels que la température, la RAM utilisée et la version de l'OS. Cette solution va lui permettre de prendre des décisions éclairées quant à la maintenance de son équipement et ainsi éviter des pannes. De plus, cette application pourra également être utilisée par d'autres étudiants en tant que base de travail pour leurs projets futurs.

5.3. Axes d'amélioration

Plusieurs axes d'améliorations sont possible pour ce projet:

- Réaliser la partie de gestion des utilisateurs.
- Avoir un paramètre, qui permet de sélectionner la plage horaire que doit nous afficher les graphiques.
- Améliorer l'esthétique.
- Améliorer le rafraîchissement des données.
- Pouvoir consulter de manière intuitive l'historique de certains indicateurs.
- paramétrer un serveur de mail pour les alertes.

5.4. Et si c'était à refaire (prise de recul) ?

Si ce projet était à refaire, avec plus de temps de réalisation, nous aurions pu améliorer notre architecture, réfléchir plus en détail aux fonctionnalités à implémenter et à la manière de les intégrer de manière optimale dans notre application

L'utilisation d'un framework pour structurer l'application aurait permis de gagner du temps et d'optimiser le code.

Nous sommes tous frustrés de ne pas avoir eu l'occasion d'avoir plus de temps pour livrer une solution complètement aboutie, et ne pas avoir pu monter plus en compétence. Néanmoins, ce projet a été une excellente occasion pour nous de mettre en pratique nos connaissances techniques et de travailler en équipe sur un projet concret. Nous avons également pu découvrir de nouvelles technologies et améliorer nos compétences en matière de développement web et de gestion de projet.