

P2022 : Sunshare

Dossier technique du projet - partie commune

Table des matières

| | |
|--|----------|
| 1 - INTRODUCTION..... | 3 |
| 1.1 - SITUATION DU PROJET DANS SON CONTEXTE INDUSTRIEL..... | 3 |
| 1.1.1 - Synoptique général du système..... | 3 |
| 1.1.2 - Missions du système..... | 4 |
| 1.1.3 - Diagramme de déploiement d'exploitation..... | 5 |
| 1.2 - CONTRAINTES DIVERSES EXPRIMÉES PAR LE DEMANDEUR..... | 6 |
| 1.2.1 - Exigences fonctionnelles..... | 6 |
| 1.2.2 - Contraintes matérielles..... | 6 |
| 1.2.3 - Contraintes logicielles..... | 7 |
| 1.2.4 - Exigences économiques..... | 7 |
| 2 - SPÉCIFICATIONS FONCTIONNELLES..... | 7 |
| 2.1 - CATALOGUE DES ACTEURS..... | 7 |
| 2.2 - DIAGRAMME DES CAS D'UTILISATION..... | 8 |
| 2.3 - CAS D'UTILISATION « RÉCUPÉRER LES INDICATEURS »..... | 8 |
| 2.3.1 - Description du cas d'utilisation..... | 8 |
| 2.3.2 - Diagramme de séquence du scénario nominal..... | 9 |
| 2.4 - CAS D'UTILISATION « VISUALISER LES INDICATEURS LOCAUX » INSTANTANÉE PAR INTERFACE WEB..... | 10 |
| 2.4.1 - Description du cas d'utilisation..... | 10 |
| 2.4.2 - Diagramme de séquence du scénario nominal..... | 11 |
| 2.4.3 - IHM associée au cas d'utilisation..... | 11 |
| 2.4.4 - Diagramme de navigation entre pages de l'application web : page d'accueil, l'affichage des données instantanées..... | 12 |
| 2.5 - CAS D'UTILISATION « VISUALISER LES INDICATEURS LOCAUX » SOUS FORME D'HISTORIQUE PAR INTERFACE WEB..... | 13 |
| 2.5.1 - Description du cas d'utilisation..... | 13 |
| 2.5.2 - Diagramme de séquence..... | 14 |
| 2.5.3 - IHM associée au cas d'utilisation..... | 14 |
| 2.5.4 - Diagramme de navigation entre pages de l'application : page historique..... | 15 |
| 2.6 - CAS D'UTILISATION « SUIVRE ÉVOLUTION SOUS FORME GRAPHIQUE » VIA INTERFACE WEB..... | 15 |
| 2.6.1 - Description du cas d'utilisation..... | 16 |
| 2.6.2 - Diagramme de séquence du scénario nominal..... | 16 |
| 2.6.3 - IHM associée au cas d'utilisation..... | 17 |
| 2.6.4 - Diagramme de navigation entre pages de l'application web : page graphique..... | 17 |
| 2.7 - CAS D'UTILISATION « VISUALISER LES INDICATEURS LOCAUX » SOUS FORME GRAPHIQUE SUR SMARTPHONE..... | 18 |
| 2.7.1 - Description du cas d'utilisation..... | 18 |
| 2.7.2 - Diagramme de séquence du scénario nominal..... | 19 |
| 2.7.3 - IHM associée au cas d'utilisation..... | 19 |

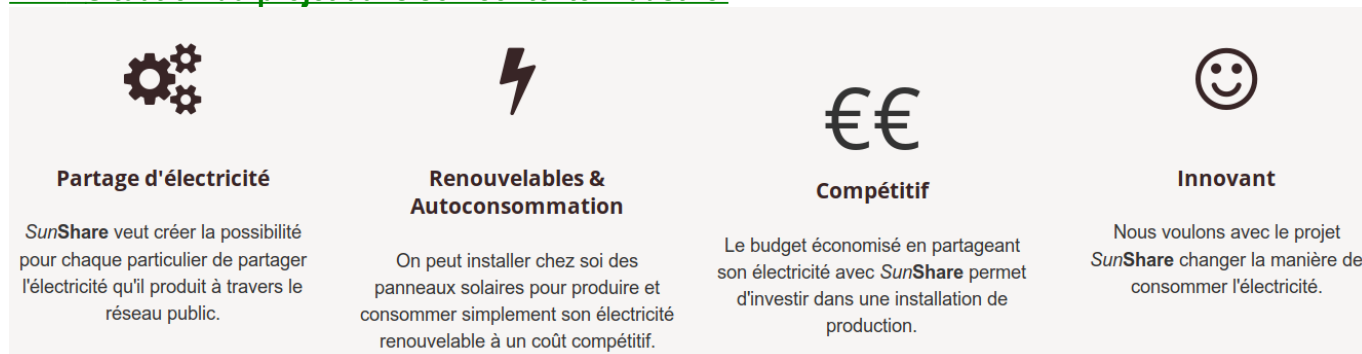
| | |
|---|-----------|
| 2.8 - CAS D'UTILISATION « VISUALISER LES INDICATEURS LOCAUX » EN TEMPS RÉEL SUR SMARTPHONE..... | 20 |
| 2.8.1 - Description du cas d'utilisation..... | 20 |
| 2.8.2 - IHM associé au cas d'utilisation..... | 21 |
| 2.9 - CAS D'UTILISATION « VISUALISER LES INDICATEURS VIA UN HISTORIQUE » SUR SMARTPHONE..... | 21 |
| 2.9.1 - Description du cas d'utilisation..... | 22 |
| 2.9.2 - IHM associé au cas d'utilisation..... | 22 |
| 3 - ÉTUDE PRÉLIMINAIRE..... | 23 |
| 3.1 - SCHEMA DE NOTRE BASE DE DONNÉES «SUNSHAREBdD»..... | 23 |
| 3.2 - DIAGRAMME DE CLASSE DE L'APPLICATION Qt..... | 24 |
| 3.3 - OUTILS UTILISÉS..... | 24 |
| 4 - RECETTE..... | 27 |
| 5 - CONCLUSION..... | 28 |
| 5.1 - TÂCHES..... | 28 |
| 5.1.1 - En cours..... | 28 |
| 5.1.2 - À venir..... | 28 |
| 5.2 - AMÉLIORATIONS ENVISAGEABLES..... | 28 |

1 - Introduction

Dans le cadre de la formation de technicien supérieur : BTS Systèmes Numériques Informatique et Réseaux, le lycée Nicolas Appert basé à Orvault (44) et la société SunShare entrent en relation pour un partenariat autour du projet de partage d'énergie solaire.

La société SunShare offre la possibilité aux étudiants de formation continue du lycée Nicolas Appert de développer des outils permettant la mise en place de tout un nouvel écosystème dans le domaine de la fourniture de courant électrique.

1.1 - Situation du projet dans son contexte industriel



1.1.1 - Synoptique général du système

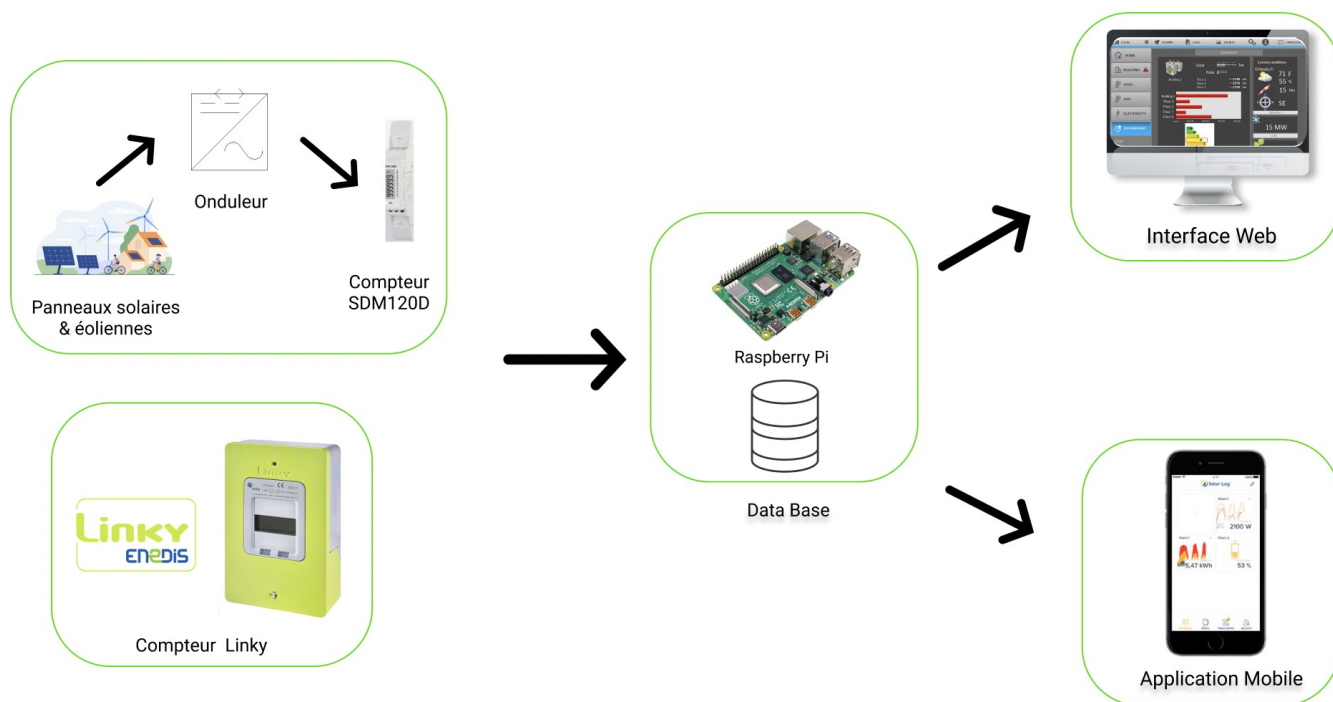


Figure 1: Synoptique général du système

Dans un premier temps, on retrouve ENEDIS, gestionnaire du réseau électrique, responsable de la distribution d'énergie électrique en France.

Il y a ensuite le compteur Linky qui mesure les indicateurs de consommation des foyers pour l'envoyer à ENEDIS et permet aux particuliers d'accéder aux données grâce à la TIC (protocole de Télé-Information Client).

Les indicateurs sont les suivants :

- Soutirage : énergie fournie par ENEDIS et consommée dans le foyer
- Production : énergie produite localement (solaire, éolien) et consommée dans le foyer
- Injection : surplus d'énergie produite localement donnée à ENEDIS

La TIC du Linky permet aux particuliers de collecter les informations relatives à la consommation, comme le soutirage et l'injection. La TIC se récupère par l'intermédiaire d'une interface unidirectionnelle (disponible uniquement en lecture) en liaison série RS485.

Un autre compteur (le SDM120D) sert uniquement à contrôler et certifier l'énergie produite localement (panneaux solaire, éoliennes) et met à disposition un signal en sortie sous forme d'impulsions correspondant aux mesures réalisées par celui-ci.

Vient ensuite le Raspberry Pi, récupérant les indicateurs grâce à la TIC pour les stocker dans une base de donnée et permettre aux particuliers d'accéder aux données par l'intermédiaire d'une interface web (pour PC) ou d'une application (pour mobile).

1.1.2 - Missions du système

Notre section SNIR a été sollicitée par l'entreprise SunShare pour explorer les solutions qui permettent de récupérer sur un Raspberry Pi la mesure de la production et la consommation électrique, en vue d'une solution utilisable par le grand public, pour moins de 100 €.

Le matériel dépend de la solution trouvée (voir diagramme de déploiement), voici tout le matériel utilisé :



Linky : compteur
Prix : fourni par ENEDIS



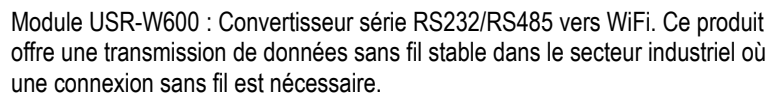
Module D2L : premier ERL (Émetteur Radio Local) Wi-Fi pour Linky, il rend les données accessibles localement ou à distance par d'autres produits ou via une simple application sur smartphone.
Prix : 59,80€



Compteur SDM120D : compteur d'énergie numérique écran LCD (5 chiffres) pour montage sur rail.
Prix : 15€



Raspberry Pi 3 (Rpi) : c'est un microordinateur qui peut exécuter des programmes, très apprécié dans les systèmes embarqués.
Prix : 40€



µTéléInfo : Dongle USB permettant de collecter la TIC

Shield BoxEnergie : Carte libre pour l'acquisition des données du compteur linky et de production d'énergie sur le raspberry pi (GPIO) pour le projet de boxenergie.

Les principales missions du système sont :

- Récupérer les indicateurs (TIC du Linky + impulsions du sous-compteur SDM120D)
- Stocker les indicateurs dans une base de données
- Permettre au client de visualiser les indicateurs en temps réel

Figure 2: Diagramme de déploiement

L'ordinateur accède à l'application web hébergé sur le Raspberry Pi par l'intermédiaire d'un navigateur (Chrome, Firefox...).

Le smartphone utilise une application mobile, qui pour communiquer avec la base de données, doit passer par une API convertissant les requêtes HTTP en fichier json.

La récupération des indexes se fera en quatre temps :

- La première, en **jaune** consiste en récupérer la TIC filaire avec le μ TéléInfo et les impulsions grâce au GPIO de du Rpi. Cette solution est intéressante si le Raspberry est proche du Linky.
- La seconde, en **rose**, collecte la TIC à l'aide du module sans-fil USR-W600. Cette solution est utile si le compteur Linky n'est pas facile d'accès par câble.
- La troisième, en **blanc**, utilise le shield BoxEnergie conçu spécialement pour ce projet par les élèves de Polytech Nantes et financé par financement participatif (crowdfunding). Ce composant peut à la fois récupérer la TIC et les impulsions.
- Enfin la dernière, en **cyan**, utilise le module D2L, qui est un émetteur radio conçu spécialement pour le Linky (un emplacement est prévu sur le compteur pour l'y insérer). L'inconvénient de cette solution est qu'elle nécessite un partenariat avec l'entreprise eeSmart pour pouvoir utiliser l'API pour exploiter les données recueillies.

1.2 - Contraintes diverses exprimées par le demandeur

1.2.1 - Exigences fonctionnelles

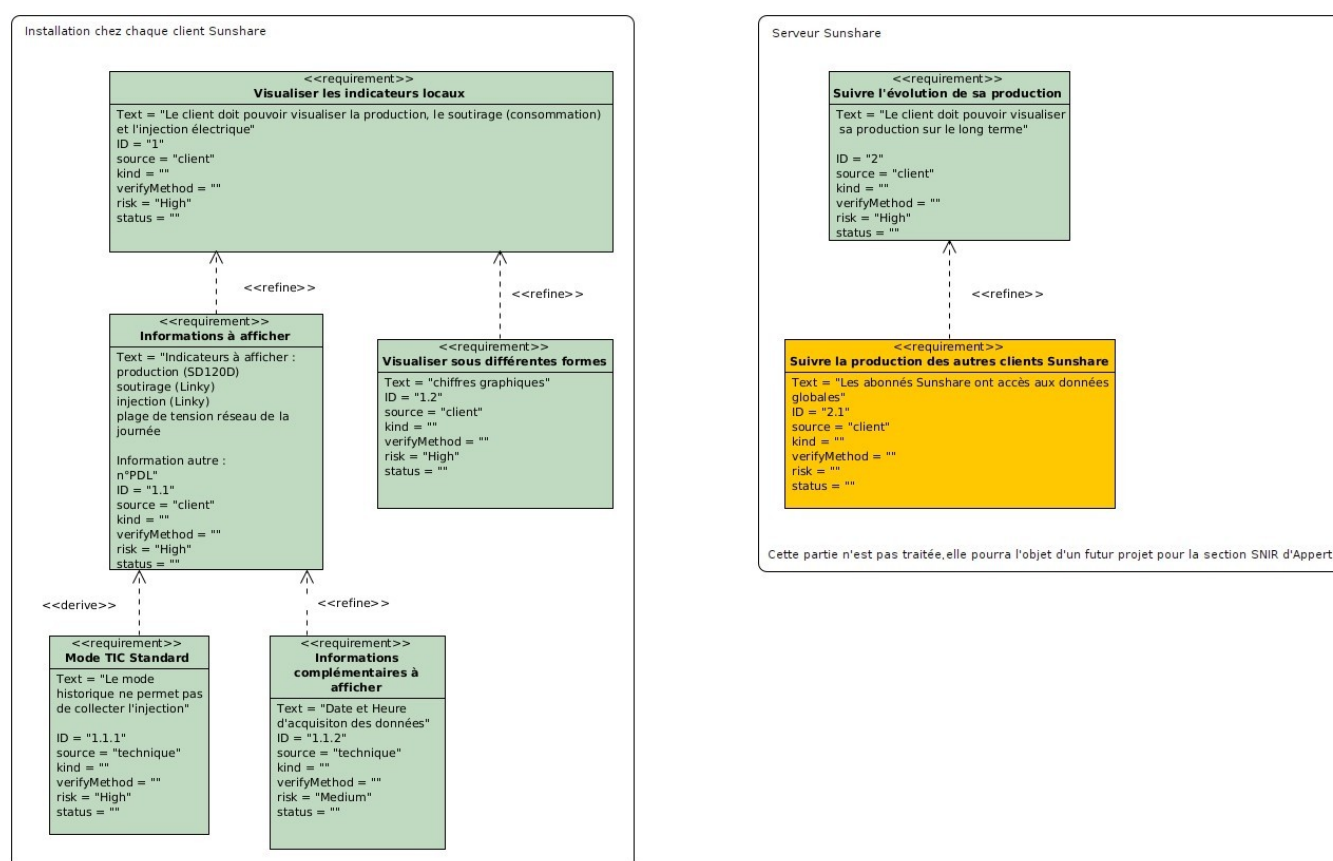


Figure 3: Diagramme d'exigences

Le numéro point de livraison (PDL) permet d'identifier le compteur Linky. Il s'appelle désormais le PRM (Point de Référence Mesure).

1.2.2 - Contraintes matérielles

Tout le matériel est fourni :

- Compteur Linky + module D2L
- Compteur SD120D : compteur servant à mesurer la production locale
- Module USR-W600 : convertisseur wifi/série

- Raspberry Pi 3 : micro-ordinateur faisant le traitement des indicateurs et gère la base de données
- µTéléInfo : c'est un dongle USB qui permet de récupérer la TIC
- Shield BoxEnergie : carte d'extension du RPi qui se place sur le GPIO permettant de récupérer la TIC + impulsions du compteur SD120D
- Maquette (rhéostats, alternostat) : sert à simuler la consommation locale

1.2.3 - Contraintes logicielles

Pour récupérer les indicateurs :

- Langage C++, langage de programmation orienté objet
- Qt 6.2 5.12.5 => version retenue pour la cross-compilation

La cross-compilation (compilation croisée) permet de compiler un programme pour une architecture différente de processeur. Dans ce projet, les programmes sont compilés depuis un ordinateur de bureau Intel d'architecture x86 pour être utilisés sur un Rpi d'architecture ARM.

Cependant, mettre en place la cross-compilation a été une tâche complexe : les tutoriels pour l'installer avec Qt 6 n'existent qu'avec le Rpi 4, non compatible avec le Rpi 3, il a fallu changer pour la version 5 de Qt.

- Bibliothèque Qt Serial Port, permettant le contrôle du port série
- Bibliothèque Piggpio, permettant de contrôler le GPIO (General Purpose Input Output) du RPi

Pour l'application mobile :

- Framework React Native, pour développer l'application mobile

Documentation :

- Libre Office, la suite bureautique libre et gratuite.
- Doxygen, générateur de documentation pour le code C++

Gestion du projet :

- Visual Paradigm, outil permet de créer des schémas SysML
- Trello, outil de gestion de projet en ligne
- Gitlab, logiciel libre de forge basé sur git permettant de stocker les fichiers du projet en Cloud

1.2.4 - Exigences économiques

Le budget total du projet est de 100€.

2 - Spécifications fonctionnelles

2.1 - Catalogue des acteurs

| Acteur | Rôle |
|------------------|--|
| Client normal | Utilisateur pouvant accéder à ses indicateurs de consommation. |
| Client Sunshare | Abonné des services de SunShare désireux de partager son électricité et de réduire ses frais de fourniture d'énergie. |
| Serveur Sunshare | Serveur de l'entreprise sur lequel seront stockées toutes les données de production et de consommation des abonnés, rendues accessibles aux clients SunShare ayant payé un abonnement. |

2.2 - Diagramme des cas d'utilisation

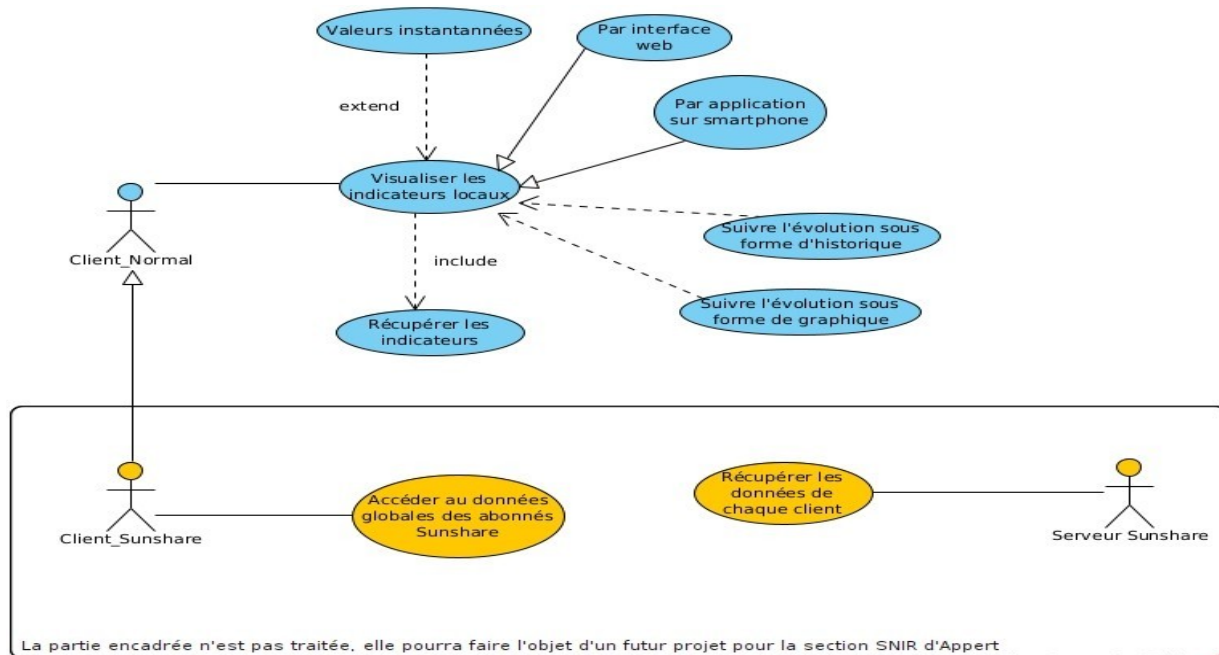


Figure 4: Diagramme des cas d'utilisation du système

2.3 - Cas d'utilisation « Récupérer les indicateurs »

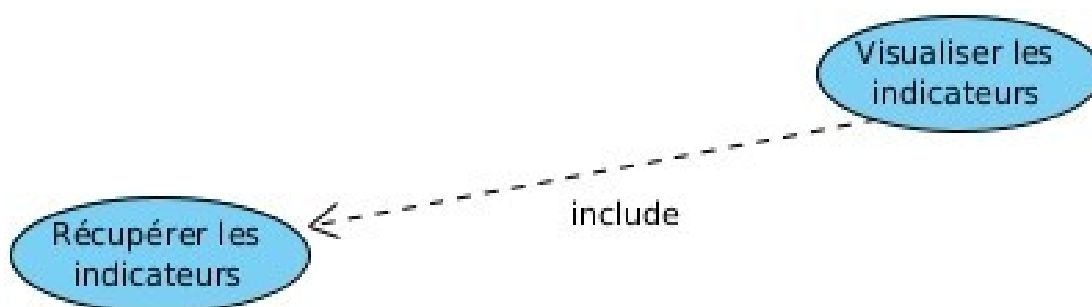


Figure 5: Cas d'utilisation : Récupérer les indicateurs

2.3.1 - Description du cas d'utilisation

| Nom CU: Récupérer les indicateurs | Référence : CU1 | Nom : COURJAUD Melvin |
|---|--|-----------------------|
| Pré-condition(s) (Liste l'(es) état(s) dans le(s)quel(s) le système peut être avant que ce cas d'utilisation débute) | 1. Matériel correctement câblé (compteurs, Raspberry Pi, matériel d'acquisition) 2. Raspberry Pi en état de fonctionnement (BdD + programme d'acquisition) | |
| Scénario nominal (Décrit le déroulement "normal", sans accroc, du processus) | 1. Un délai d'acquisition a été choisi au préalable 2. Les indicateurs (soutirage, injection, production) sont prélevés 3. Les indicateurs sont envoyés à la base de données | |
| Scénario alternatif A (Décrit un cas variant du déroulement du processus) | Condition: Matériel mal câblé A1. Les indicateurs ne peuvent pas être collectés | |
| Scénario alternatif B (Décrit un cas variant du déroulement du processus) | Condition: Base de données indisponible B1. Les données ne peuvent pas être envoyés à la base de données | |

| | | |
|---|---|-----------------------------|
| Nom CU: Récupérer les indicateurs | Référence : CU1 | Nom :COURJAUD Melvin |
| Post-condition(s) (Listez l'(es) état(s) dans le(s)quel(s) le système peut être quand le cas d'utilisation se termine) | 1. La base de données a bien reçu les données | |

2.3.2 - Diagramme de séquence du scénario nominal

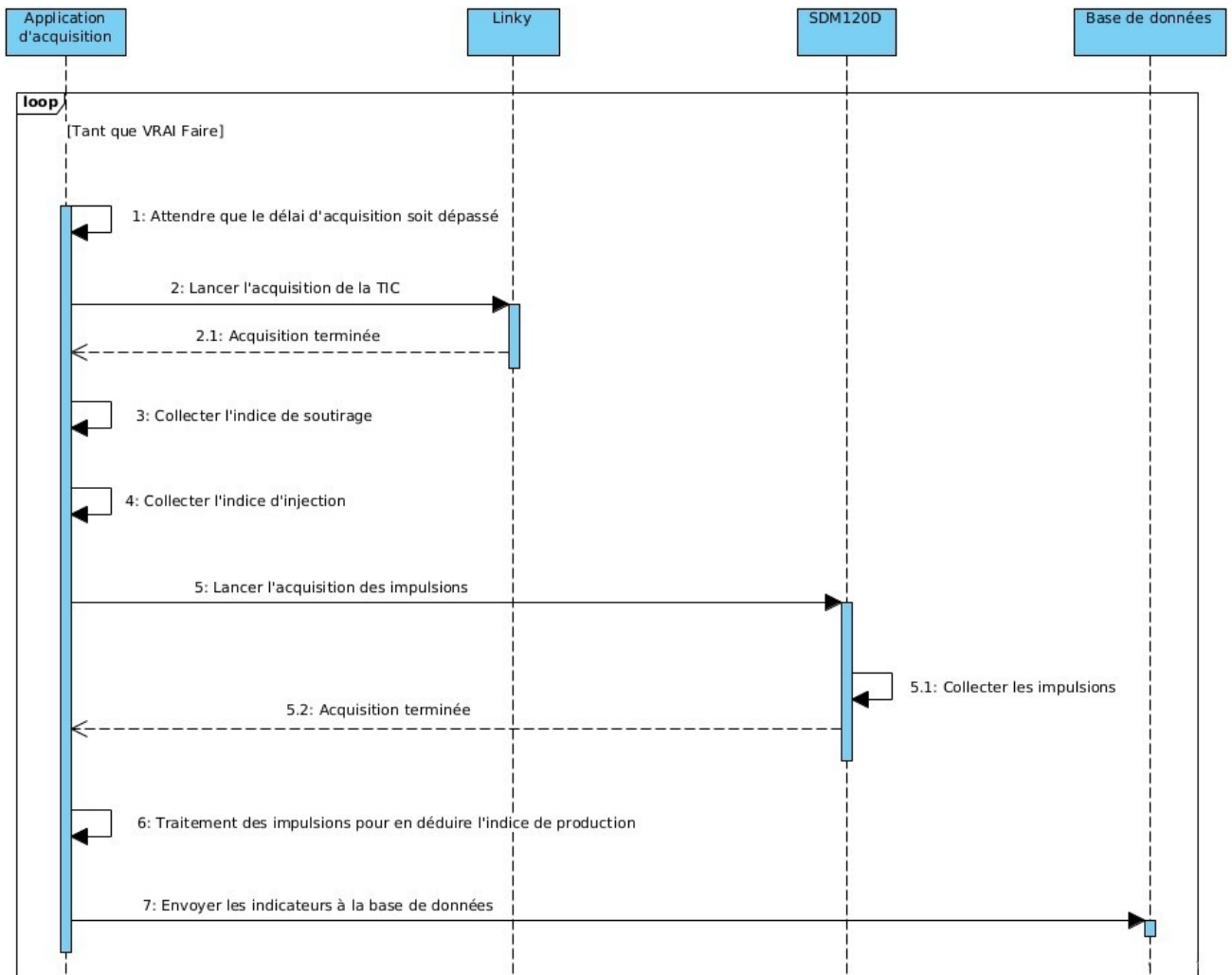


Figure 6: Diagramme de séquence : Récupérer les indicateurs

2.4 - Cas d'utilisation « Visualiser les indicateurs locaux » instantanée par interface web

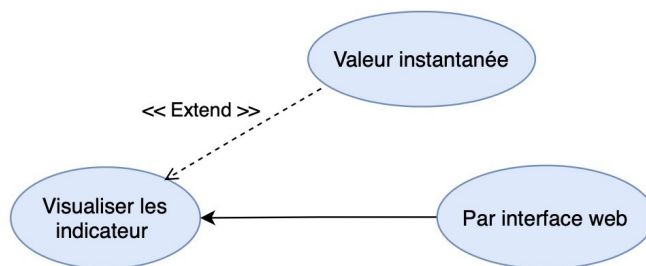


Figure 7: Cas d'utilisation : Visualiser les indicateurs via interface web

2.4.1 - Description du cas d'utilisation

| Nom CU: Visualiser les indicateurs | Référence : CU2 | Nom : AHAMADA Ayaad |
|---|--|---------------------|
| Pré-condition(s) <i>(Liste l'(es) état(s) dans le(s)quel(s) le système peut être avant que ce cas d'utilisation débute)</i> | 1. Raspberry Pi en état de fonctionnement 2. PC en état de fonctionnement (Connecter au même réseau que la Rpi) | |
| Scénario nominal <i>(Décrit le déroulement "normal", sans accroc, du processus)</i> | 1. Le client allume son pc et ouvre la page web 2. La page d'accueil s'ouvre 2. Le serveur web sur la Rpi envoie les données les plus récentes au navigateur web. 3. Le client consulte les données reçues sur la page web local. | |
| Scénario alternatif A <i>(Décrit un cas variant du déroulement du processus)</i> | Condition: Pas de données sur la base A1. L'interface demande des valeurs à BDD A2. Aucune donnée n'est disponible sur le serveur RPI A3. Affichage de « pas de données disponibles » | |
| Scénario alternatif B <i>(Décrit un cas variant du déroulement du processus)</i> | Condition: veille de données B1. Temporisation de réception de données | |
| Post-condition(s) <i>(Listez l'(es) état(s) dans le(s)quel(s) le système peut être quand le cas d'utilisation se termine)</i> | 1. L'interface affiche les données «production», «consommation» et «injection» instantanées. 2. Pas de données à affiche | |

2.4.2 - Diagramme de séquence du scénario nominal

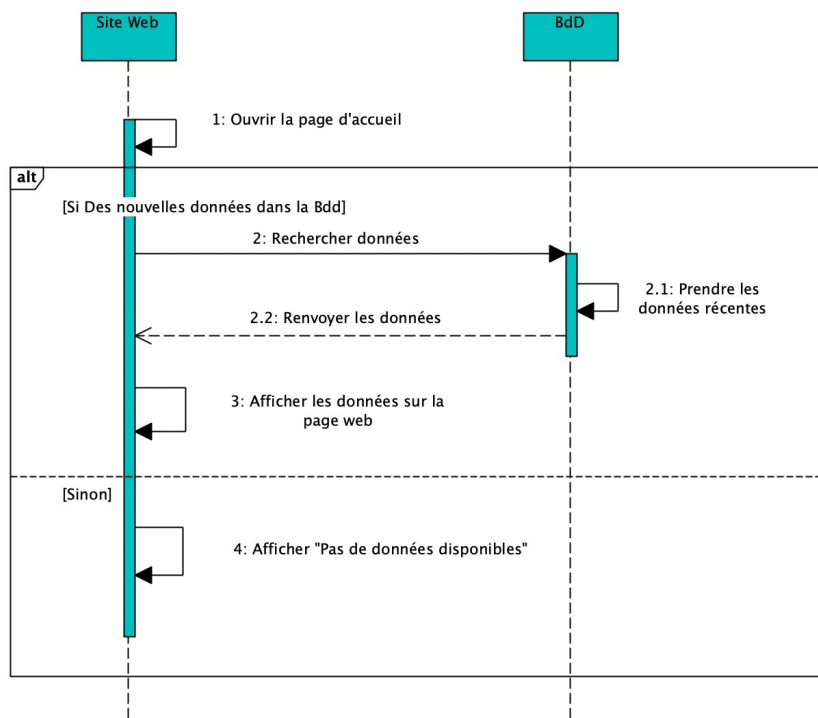


Figure 8: Diagramme de séquence visualisation indicateurs en temps réel sur interface web

2.4.3 - IHM associée au cas d'utilisation

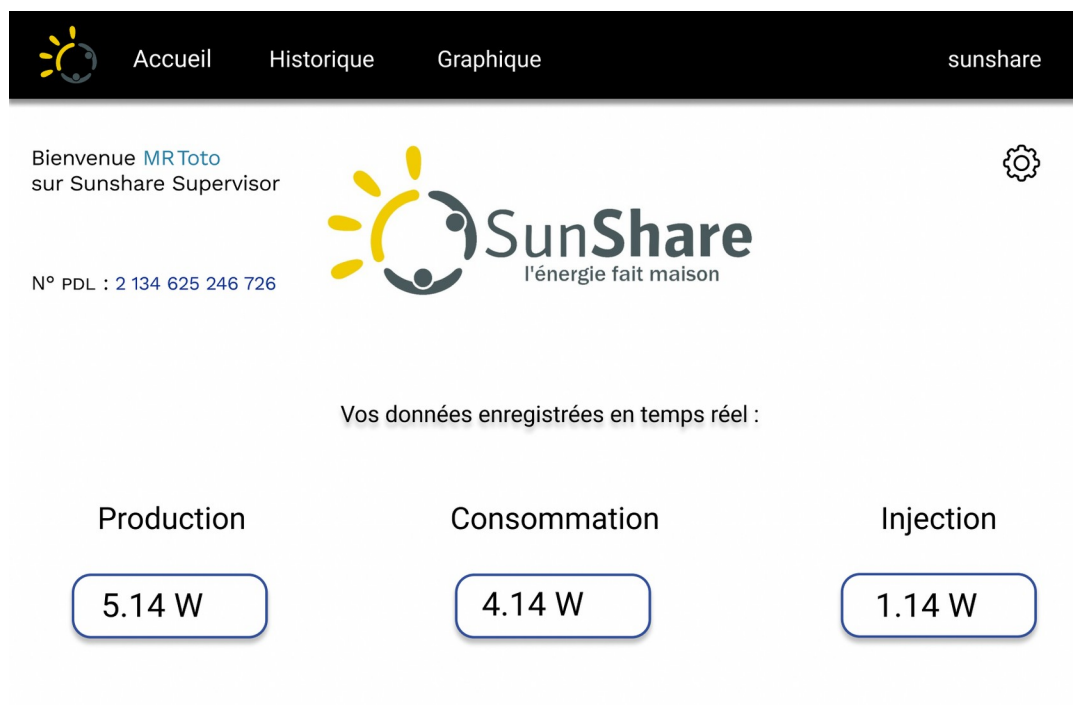


Figure 9: Page d'accueil, un prototype réalisé avec Figma

Le nom et le prénom et les autres données (Nom, Prénom, n° pdl) seront des données qui seront mise dans un fichier de configuration et qu'on affichera par la suite.

Ces données ne seront pas dans la base de données, car c'est un système qui sera utilisé seulement par un client. C'est la raison pour lequel on s'est dit qu'il n'était pas nécessaire de les faire figurer dans la base de données.

Ensuite les données, production, consommation et injection, sont des données qui seront affichées en fonction de la date d'enregistrement dans la base de données, l'interface web va afficher les indicateurs les plus récents qui ont été mesurés et enregistré dans la base de données.

Par exemple, quand le client a ouvert la page web, une consommation de 4.14 W et une injection de 1.14 W sur le réseau Enedis a été enregistrée. Et si des nouvelles données ont été enregistrée dans la base et qu'il actualise la page des nouvelles valeurs s'afficheront.

2.4.4 - Diagramme de navigation entre pages de l'application web : page d'accueil, l'affichage des données instantanées

Pour ce diagramme il y aura trois page web, qui réaliseront notre principale fonction de navigation. C'est aussi parce que, avec ces trois pages (Accueil, historique et graphique) la barre de menus est différente en fonction de la page où vous êtes .

- **Bleu** c'est ce qu'on fait si on veut sortir de la page
- **Orange** c'est ce qui se passe quand on est sur la page concernée. Voir les prototype IHM pour la compréhension.

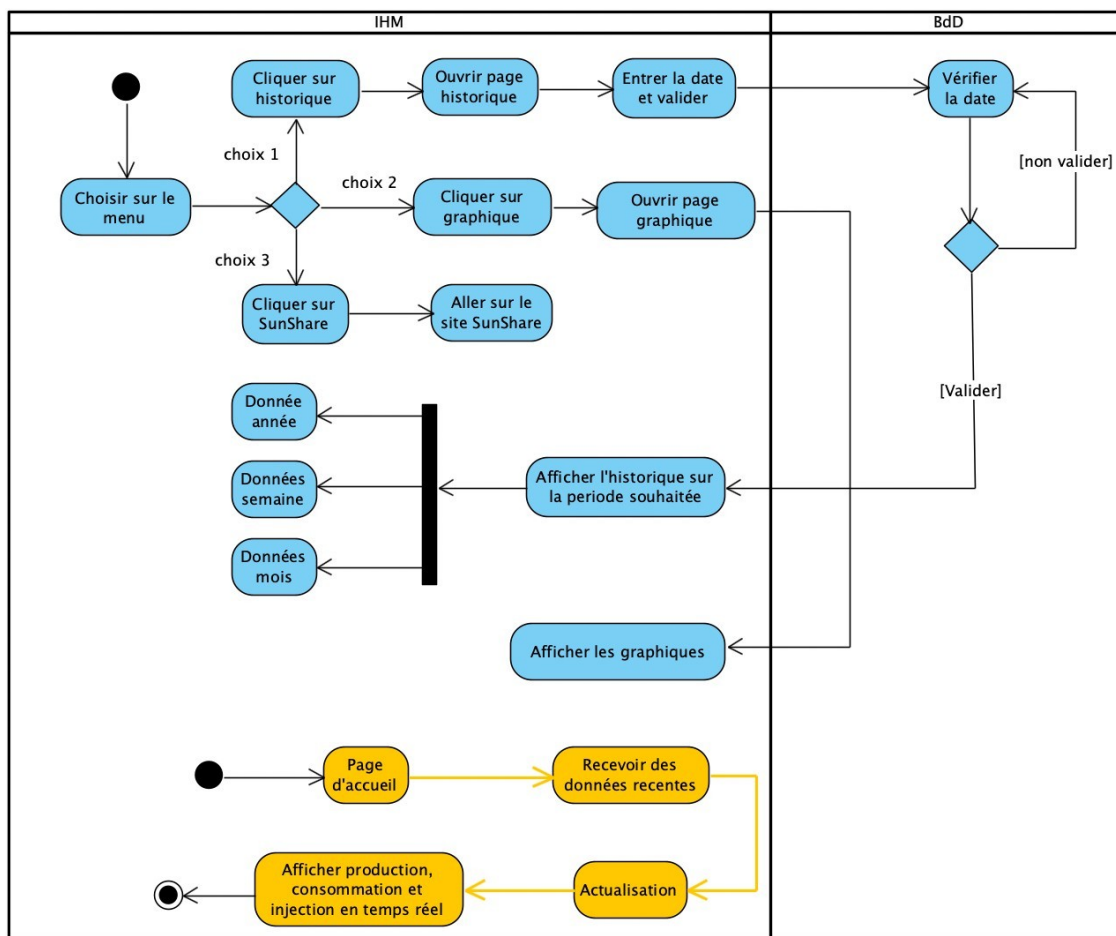


Figure 10: Diagramme activité : la page d'accueil

2.5 - Cas d'utilisation « Visualiser les indicateurs locaux » sous forme d'historique par interface web

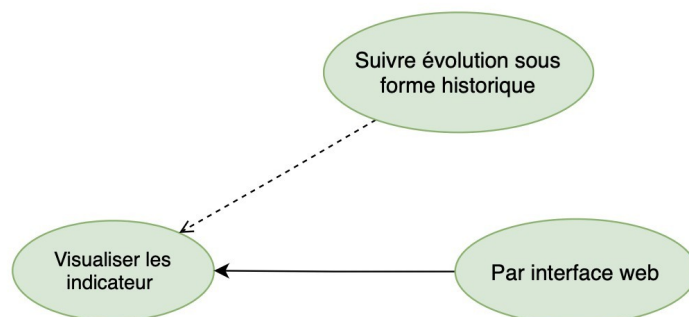


Figure 11: Cas d'utilisation : Visualiser les indicateurs via interface web

2.5.1 - Description du cas d'utilisation

| Nom CU: Visualiser les indicateurs | Référence : CU3 | Nom : AHAMADA Ayaad |
|--|---|---------------------|
| Pré-condition(s) <i>(Liste l'état(s) dans le(s)quel(s) le système peut être avant que ce cas d'utilisation débute)</i> | 1. Raspberry Pi en état de fonctionnement (Base de données). 2. PC en état de fonctionnement (Connecter au même réseau que la Rpi). | |
| Scénario nominal <i>(Décrit le déroulement "normal", sans accroc, du processus)</i> | 1. Sur la barre de menu, le client choisit « Historique » afin de voir les indicateurs sous forme de tableau. 2. Une page est ouverte, le client renseignera une période pour laquelle il souhaite voir l'historique de sa consommation, production ou encore de son injection sur le réseau Enedis. 3. Il confirme en appuyant sur « valider ». 4. Le serveur recherche en fonction de la date renseignée et envoie les données mesurer à la page web local. 5. Le client consulte les données reçues sur la page web. | |
| Scénario alternatif A <i>(Décrit un cas variant du déroulement du processus)</i> | Condition: A1. Le client n'a pas renseigné la période d'affichage A2. La date renseignée n'est pas dans la base de données A3. Aucune donnée n'est disponible sur le serveur RPI | |

2.5.2 - Diagramme de séquence

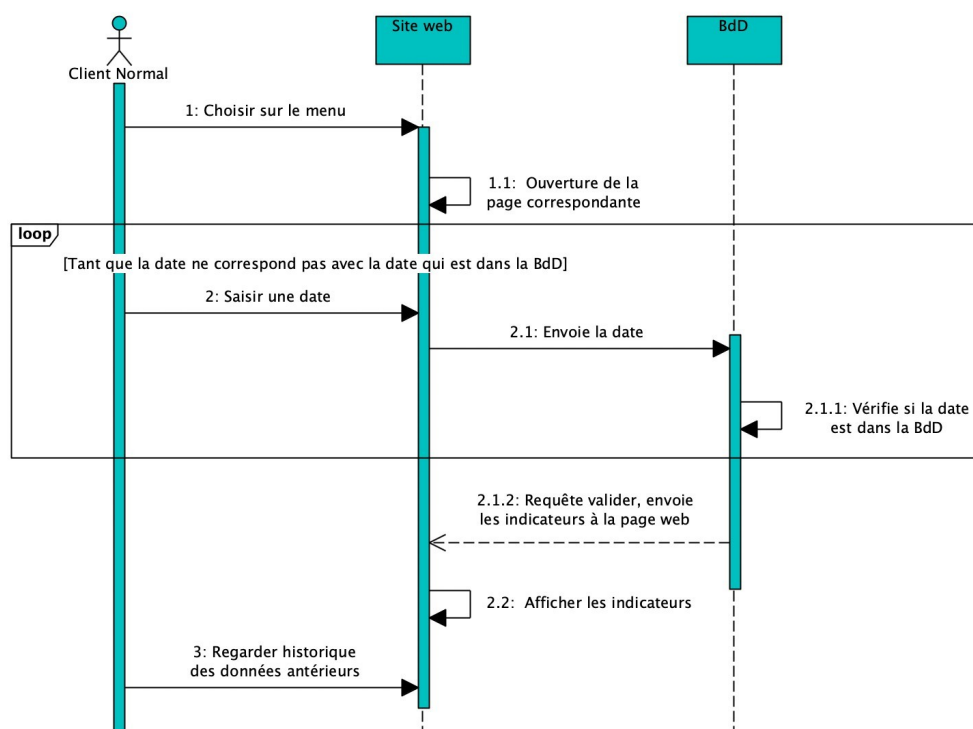


Figure 12: Diagramme séquence : Scénario nominal de la visualisation d'indicateurs (historique)

2.5.3 - IHM associée au cas d'utilisation

Sunshare Supervisor

Accueil Graphique

SunShare
l'énergie fait maison

Historique

- > journalière
- hebdomadaire
- mensuelle
- annuelle

Définir une période, pour consulter votre historique :

Du 01/01/2022 à 13 : 00 Au 07/01/2022 à 13: 00 Valider Format CSV

| Date | Énergie consommer | Date | Énergie produite | Date | Énergie injectée |
|------------|-------------------|------------|------------------|------------|------------------|
| 01/01/2022 | 03.12 W | 01/01/2022 | 04.12 W | 01/01/2022 | 01.12 W |
| 05/01/2022 | 05.21 W | 05/01/2022 | 06.25 W | 05/01/2022 | 01.10 W |
| 07/01/2022 | 04.02 W | 07/01/2022 | 06.12 W | 07/01/2022 | 02.12 W |

Figure 13: La page historique, un prototype réalisé avec Figma

2.5.4 - Diagramme de navigation entre pages de l'application : page historique

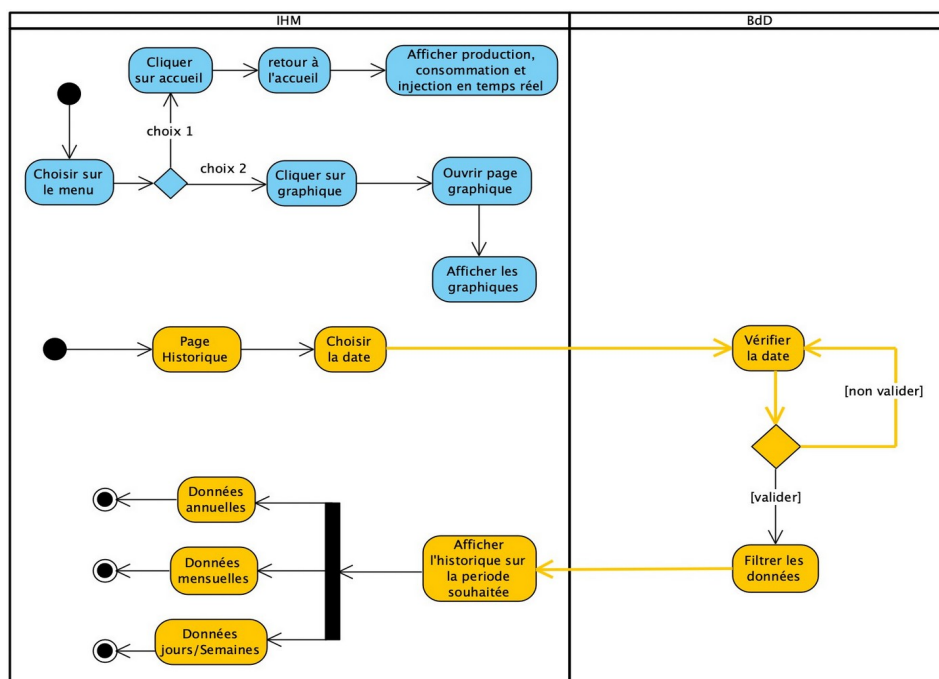


Figure 14: Diagramme activité : la page historique

2.6 - Cas d'utilisation « suivre évolution sous forme graphique » via interface web

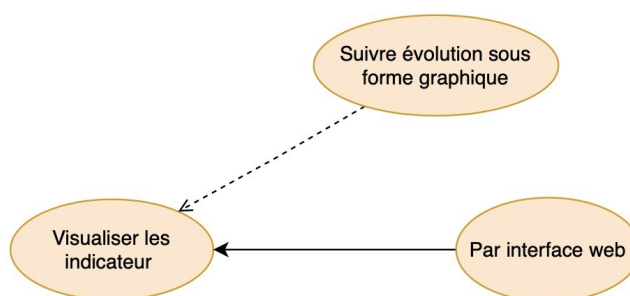


Figure 15: Cas d'utilisation : Consulter l'évolution de sa production via interface web

2.6.1 - Description du cas d'utilisation

| | | |
|--|--|----------------------------|
| Nom CU: Suivre évolution sous forme graphique | Référence : CU4 | Nom : AHAMADA Ayaad |
| Pré-condition(s) | 1. Raspberry Pi en état de fonctionnement 2. PC en état de fonctionnement (Connecter au même réseau que la Rpi) | |
| Scénario nominal | 1. Sur la barre de menu, le client choisit « graphique » afin de voir l'évolution en graphique. 2. Le serveur envoie les données calculer sous forme graphique au navigateur du client normal. 3. Le client consulte les graphiques sur la page web. | |
| Scénario alternatif A | Condition: Pas de donnée dans la base de données. A1. Le client se rend sur la page dédiée à l'affichage des courbes A2. L'interface demande des valeurs à BDD A3. Aucune donnée n'est disponible sur le serveur RPI pour cette période | |
| Post-condition(s) | 1. L'interface affiche un graphique sur la période sélectionnée 2. Pas de données à afficher | |

2.6.2 - Diagramme de séquence du scénario nominal

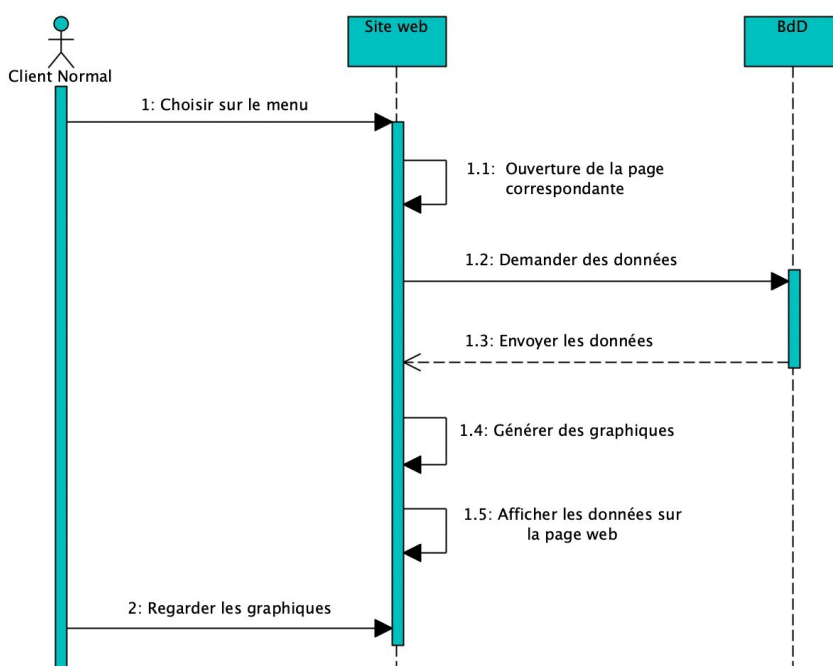


Figure 16: Diagramme de séquence représentant l'affichage graphique par interface web

2.6.3 - IHM associée au cas d'utilisation

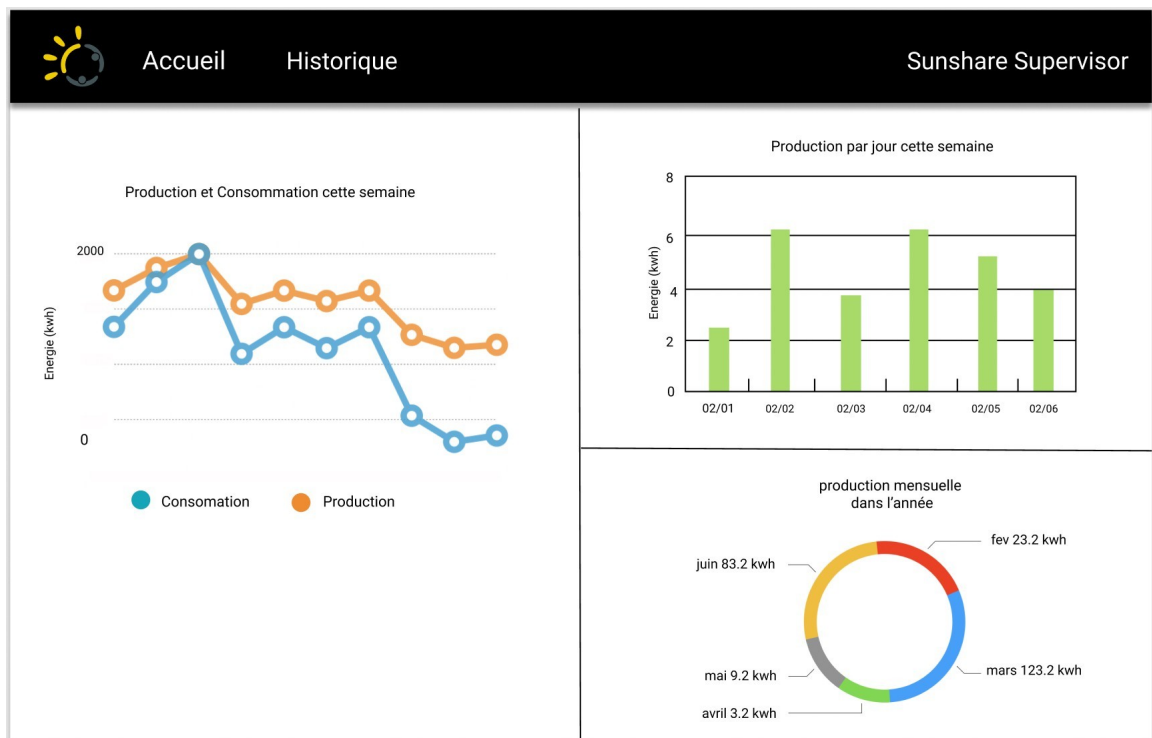


Figure 17: Page représentant des graphiques, un prototype réalisé avec Figma

2.6.4 - Diagramme de navigation entre pages de l'application web : page graphique

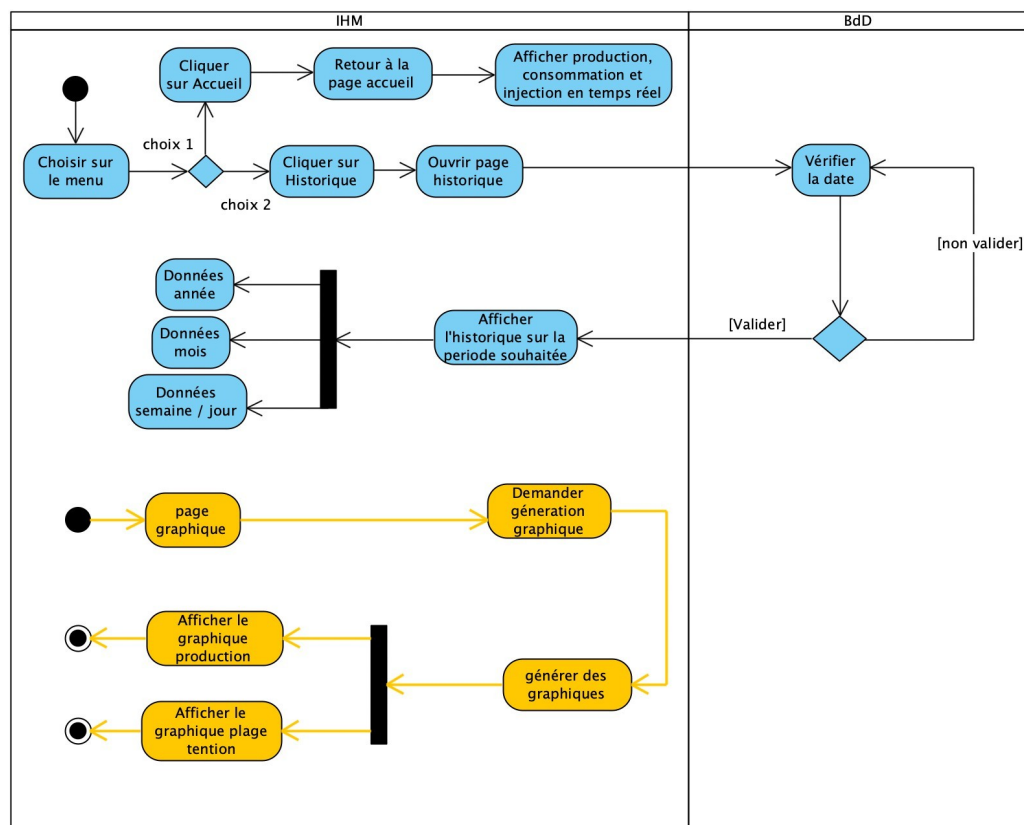


Figure 18: Diagramme activité : la page graphique

2.7 - Cas d'utilisation « Visualiser les indicateurs locaux » sous forme graphique sur smartphone

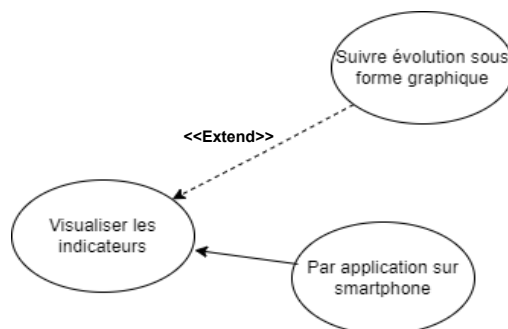


Figure 19: Cas d'utilisation : Visualiser les indicateurs sur smartphone

2.7.1 - Description du cas d'utilisation

| Nom CU: Visualisation indicateurs sur smartphone | Référence : CU5 | Nom : LHOMMEAU Nicolas |
|---|--|-------------------------------|
| Pré-condition(s) (Liste l'(es) état(s) dans le(s)quel(s) le système peut être avant que ce cas d'utilisation débute) | 1. Smartphone est en marche 2. La base de données est fonctionnelle | |
| Scénario nominal (Décrit le déroulement "normal", sans accroc, du processus) | 1. L'application demande à recevoir des données 2. Une requête est faite 3. Les données sont converties en JSON 4. L'application récupère et affiche les données | |
| Scénario alternatif A (Décrit un cas variant du déroulement du processus) | Condition : Smartphone non connecté au même réseau que box SunShare A1. L'application affiche une erreur indiquant un problème de connexion A2. Elle affiche également un message demandant à l'utilisateur de vérifier sa connexion | |
| Scénario alternatif B (Décrit un cas variant du déroulement du processus) | Condition: Pas de données dans la base de données B1. L'application n'affiche rien même si le smartphone se trouve sur le bon réseau | |
| Post-condition(s) (Listez l'(es) état(s) dans le(s)quel(s) le système peut être quand le cas d'utilisation se termine) | 1. Les données ont été reçues 2. Les données n'ont pas été reçues | |

2.7.2 - Diagramme de séquence du scénario nominal

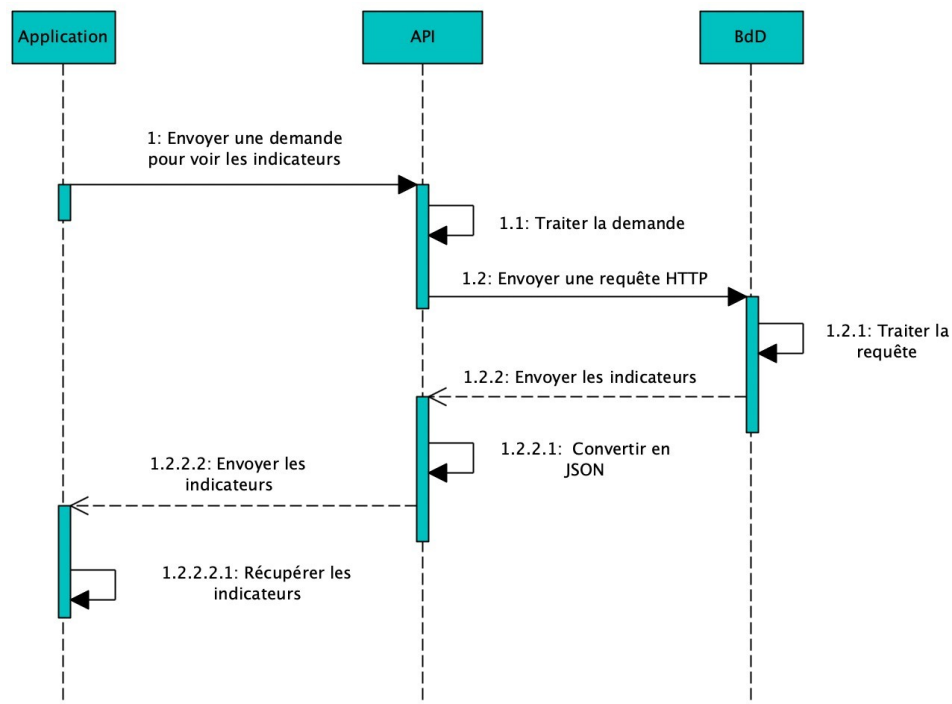


Figure 20: Diagramme de séquence : Scénario nominal de la visualisation d'indicateurs sur smartphone

2.7.3 - IHM associée au cas d'utilisation

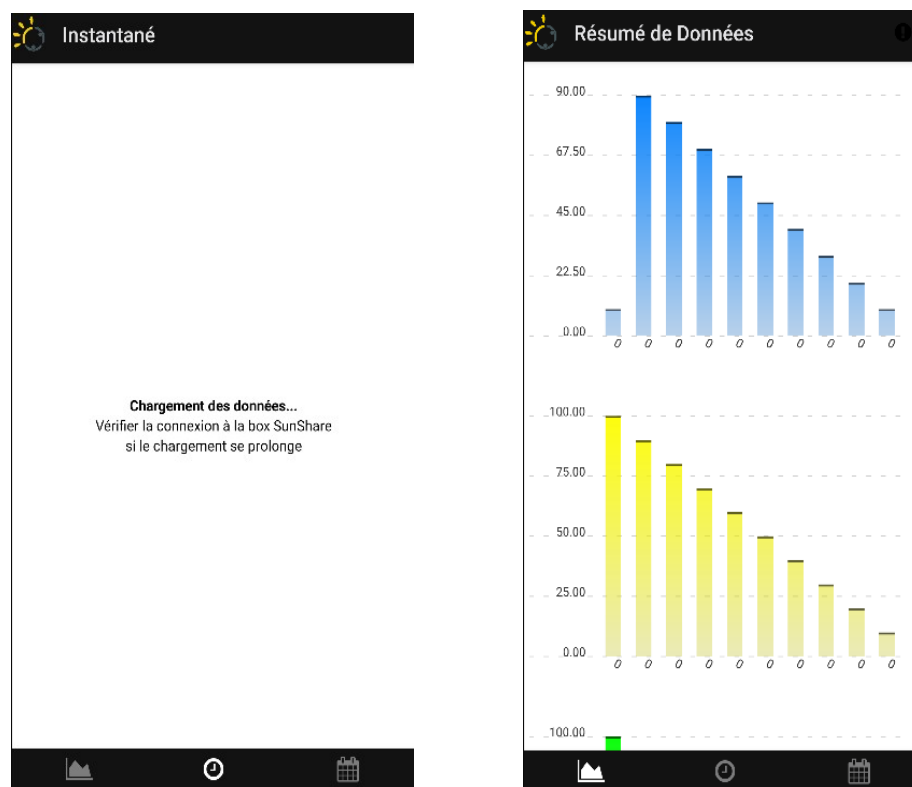


Figure 21: Page de résumé de données, à gauche lorsque l'utilisateur n'est pas sur le même réseau que la box SunShare, à droite quand tout est fonctionnel

2.8 - Cas d'utilisation « Visualiser les indicateurs locaux » en temps réel sur smartphone

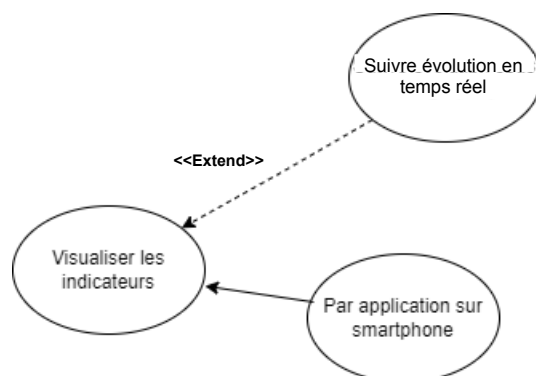


Figure 22: Cas d'utilisation : Visualiser les indicateurs en temps réel sur smartphone

2.8.1 - Description du cas d'utilisation

| | | |
|---|--|-------------------------------|
| Nom CU: Visualisation indicateurs en temps réel sur smartphone | Référence : CU6 | Nom : LHOMMEAU Nicolas |
| Pré-condition(s) <i>(Liste l'(es) état(s) dans le(s)quel(s) le système peut être avant que ce cas d'utilisation débute)</i> | 1. Smartphone est en marche 2. La base de donnée est fonctionnelle | |
| Scénario nominal <i>(Décrit le déroulement "normal", sans accroc, du processus)</i> | 1. L'application demande à recevoir des données 2. Une requête est faite 3. Les données sont converties en JSON 4. L'application récupère et affiche les dernières données enregistrées | |
| Scénario alternatif A <i>(Décrit un cas variant du déroulement du processus)</i> | Condition : Smartphone non connecté au même réseau que box SunShare A1. L'application affiche une erreur indiquant un problème de connexion A2. Elle affiche également un message demandant à l'utilisateur de vérifier sa connexion | |
| Scénario alternatif B <i>(Décrit un cas variant du déroulement du processus)</i> | Condition: Pas de données dans la base de données B1. L'application n'affiche rien même si le smartphone se trouve sur le bon réseau | |
| Scénario alternatif C <i>(Décrit un cas variant du déroulement du processus)</i> | Condition: La base de données ne reçoit plus les données mais n'est pas déconnectée C1. L'application affiche les dernières données enregistrées et non plus les données en temps réel | |
| Post-condition(s) <i>(Listez l'(es) état(s) dans le(s)quel(s) le système peut être quand le cas d'utilisation se termine)</i> | 1. Les données les plus récentes ont été reçues 2. Les données ont été reçues mais ne sont pas les plus récentes 3. Les données les plus récentes n'ont pas été reçues | |

2.8.2 - IHM associé au cas d'utilisation

En bas se situent les 3 boutons des 3 différentes pages accessibles. L'écran affiché ici correspond à la page de temps réel.

Celle-ci permet de visualiser les données en temps réelles, celles-ci sont misent à jour toutes les secondes pour avoir des valeurs les plus précises possibles.

En haut à droite il y a également un bouton permettant l'affichage d'astuces et conseils proposés par SunShare.

| Instantané | |
|-------------------|--------|
| Energie Produite | 125 kW |
| Energie Consommée | 90 VA |
| Energie Injectée | 254 VA |

Figure 23: Page d'affichage des données en temps réel



2.9 - Cas d'utilisation « Visualiser les indicateurs via un historique » sur smartphone

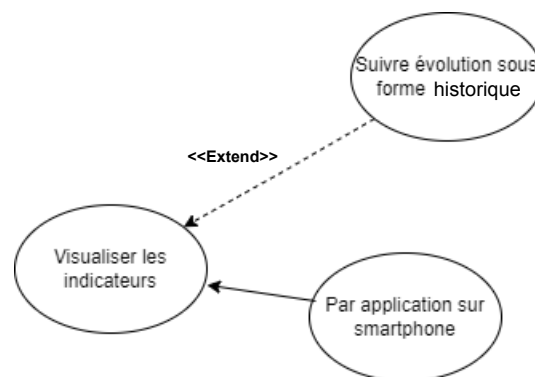
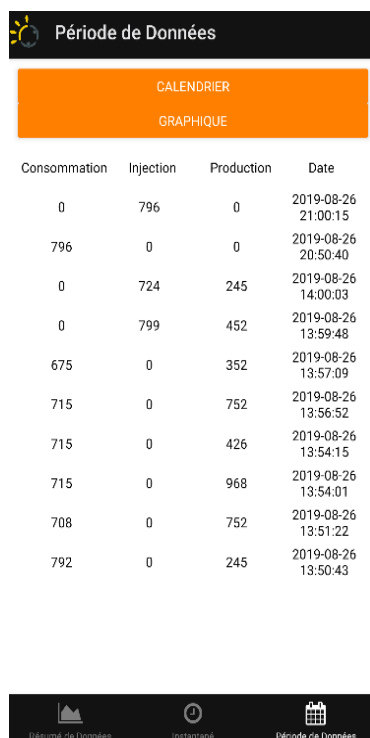


Figure 24: Cas d'utilisation : Visualiser les indicateurs sous forme d'historique sur smartphone

2.9.1 - Description du cas d'utilisation

| | | |
|--|--|-------------------------------|
| Nom CU: Visualisation indicateurs sous forme d'historique sur smartphone | Référence : CU7 | Nom : LHOMMEAU Nicolas |
| Pré-condition(s) (Liste l'(es) état(s) dans le(s)quel(s) le système peut être avant que ce cas d'utilisation débute) | 1. Smartphone est en marche 2. La base de donnée est fonctionnelle | |
| Scénario nominal (Décrit le déroulement "normal", sans accroc, du processus) | 1. L'application demande à recevoir des données 2. Une requête est faite 3. Les données sont converties en JSON 4. L'application récupère et affiche les dernières données enregistrées | |
| Scénario alternatif A (Décrit un cas variant du déroulement du processus) | Condition : Smartphone non connecté au même réseau que box SunShare A1. L'application affiche une erreur indiquant un problème de connexion A2. Elle affiche également un message demandant à l'utilisateur de vérifier sa connexion | |
| Scénario alternatif B (Décrit un cas variant du déroulement du processus) | Condition: Pas de données dans la base de données B1. L'application n'affiche rien même si le smartphone se trouve sur le bon réseau | |
| Scénario alternatif C (Décrit un cas variant du déroulement du processus) | Condition: La base de données n'affiche aucunes données après avoir sélectionner la période C1. L'application ne trouve pas de données dans la période indiquée si la période est trop ancienne ou si le système SunShare n'était pas en fonctionnement lors de cette période | |
| Post-condition(s) (Listez l'(es) état(s) dans le(s)quel(s) le système peut être quand le cas d'utilisation se termine) | 1. Les données les plus récentes ont été reçues 2. Les données les plus récentes n'ont pas été reçues | |

2.9.2 - IHM associé au cas d'utilisation



En bas se situent les 3 boutons des 3 différentes pages accessibles. L'écran affiché ici correspond à la page d'historique en fonction d'une période sélectionnée par l'utilisateur via le bouton "Calendrier".

Celle-ci permet de visualiser les données moyennées, celles-ci s'affichent en fonction de la période définie au préalable.

Figure 25: Page d'affichage des données sous forme d'historique

3 - Étude préliminaire

3.1 - Schéma de notre base de données «SunshareBdD»

Ce schéma sera implémenté dans la Raspberry pi, avec ce schéma les données seront stockées le plus simplement possibles pour que le client puisse consulter le plus facilement et rapidement ses données.

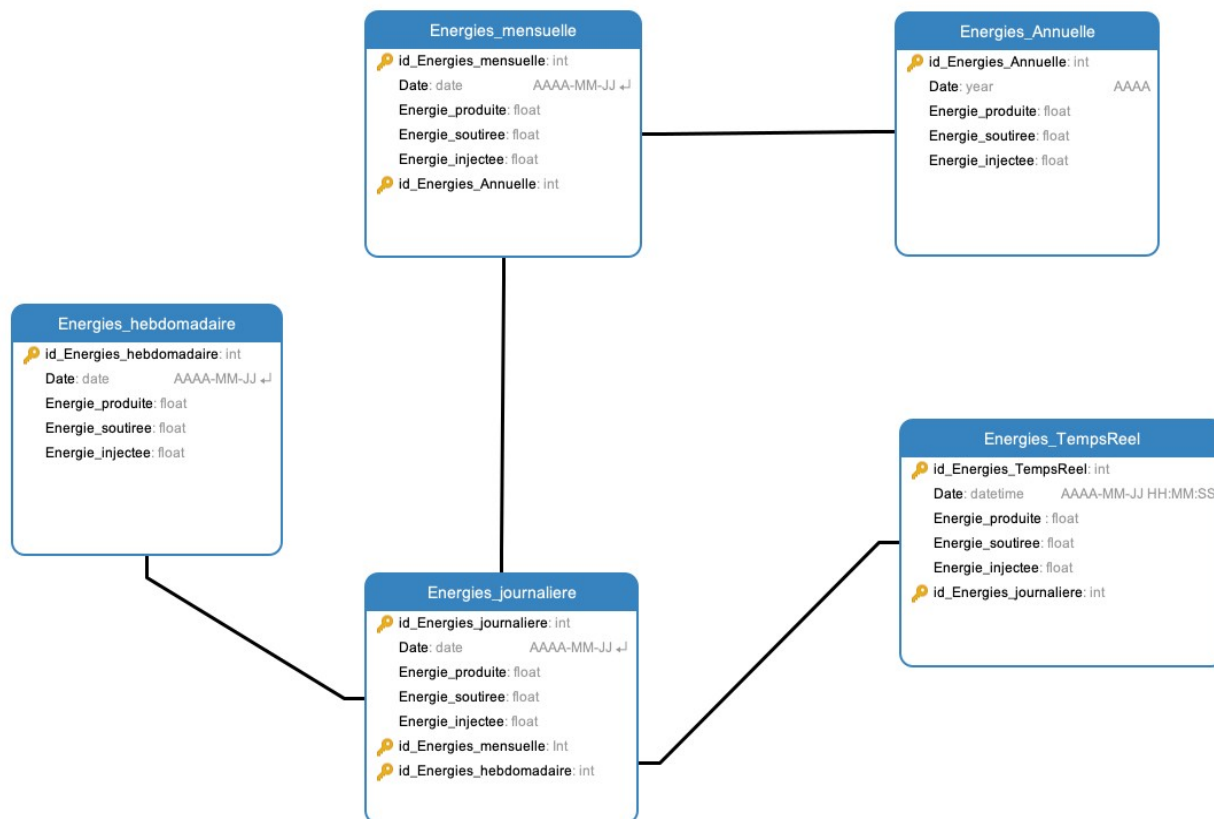


Figure 26: Schéma de base de données

Le système utilise une base de données composé de 5 tables :

- La table **Energies_TempsReel** qui contiendra les énergies que les compteurs (Compteur Linky et Compteur SD120D) vont envoyer à la base de données, mesurées en temps réel ainsi que la date (Date) à laquelle les mesures ont été faites. Ces mesures comprennent donc l'énergie produite (energie_produite), l'énergie consommée (energie_soutiree), l'énergie injectée (energie_injectee).
- La table **Energies_journaliere** qui contiendra les énergies (Energie_produite, Energie_soutiree et Energie_injectee) moyennes d'une journée. Ces valeurs seront calculées chaque jour et permettent de gagner du temps pour la consultation journalière.
- La table **Energies_hebdomadaire** va contenir les énergies (Energie_produite, Energie_soutiree et Energie_injectee) moyennes d'une semaine. Les données de chaque jour durant 7 jours, calculées en valeur moyenne.
- La table **Energies_mensuelle** va contenir les énergies (Energie_produite, Energie_soutiree et Energie_injectee) moyennes de chaque mois
- La table **Energies_annuelle** contiendra les énergies (Energie_produite, Energie_soutiree et Energie_injectee) moyennes d'une année

3.2 - Diagramme de classe de l'application Qt

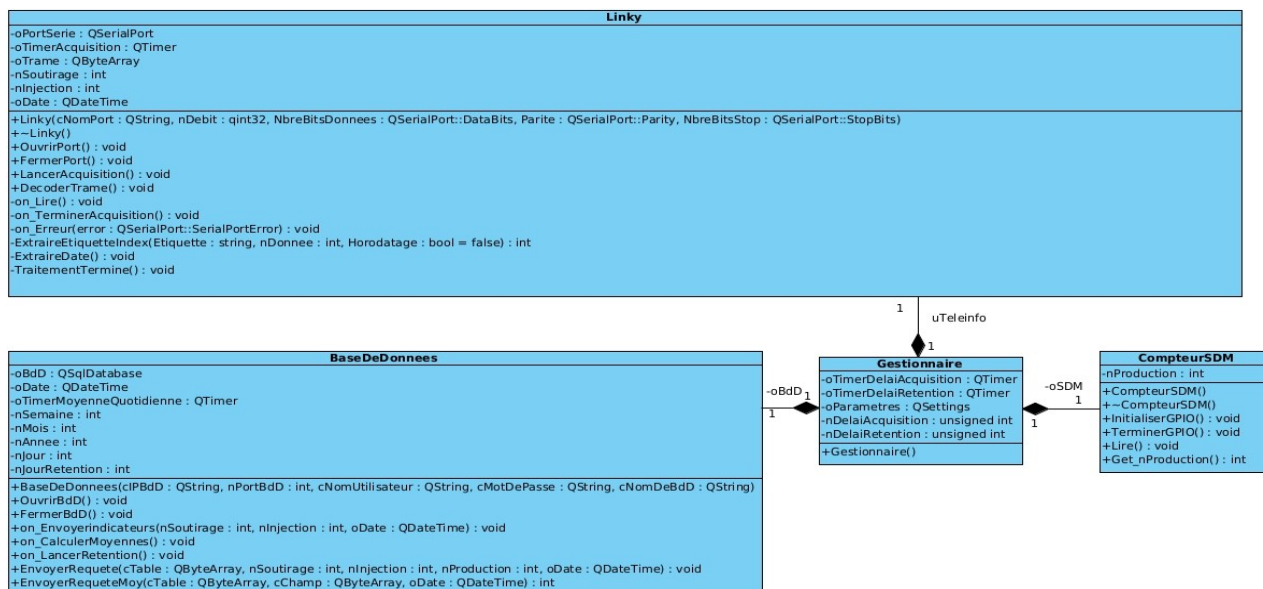


Figure 27: Diagramme de classe

- La classe **Gestionnaire** est la classe « cerveau » du projet, c'est elle qui donne les ordres.
- La classe **Linky** sert à récupérer la TIC et d'en extraire les indexes de soutirage et d'injection.
- **CompteurSDM** sert à collecter les indicateurs de production du compteur SDM120D
- La classe **BaseDeDonnees** permet d'envoyer des requêtes à la base de données du Raspberry Pi.

3.3 - Outils utilisés

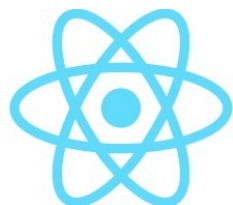
Qt :



Version utilisée : 5.12.5
API orienté objet

Qt est multi-plateforme, ce qui signifie que ses bibliothèques peuvent aussi bien être utilisées sur des ordinateurs de bureau (Windows, Mac, Linux) mais aussi mobile (Android, iOS). Cela pourrait par exemple, permettre de créer une application sur PC afin d'éviter de passer par un navigateur Web.

React Native :



Créé par Facebook
Framework Open-source
Permet de développer des applications pour Android, IOS et UWP (Universal Windows Platform)

Bootstrap :



Framework Open-Source, créé en 2011
Contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option

Nous avons choisi Bootstrap pour sa simplicité et son énorme gain de temps. En effet, on n'a pas besoin de coder toute la partie CSS, il suffit seulement de sélectionner les classes que l'on veut en codant notre page HTML. De plus, ça nous permet d'avoir un site **responsive** design (adapté à tous les écrans : ordinateur, mobile, tablettes).

MariaDB :



SGBD (Système de gestion de base de données) Libre et open source,
Fork de MySQL

Facilité de déploiement et de prise en main et très bonne intégration dans l'environnement Apache/PHP

PHP :



Langage de script et programmation open-source, principalement utilisé pour le développement web

Utile pour la communication et la Bdd, interface web et smartphone

Figma :



Plateforme collaborative qui permet de faire du prototypage
Figma nous aide dans le développement d'interface graphique, on a d'abord fait des prototypes sur Figma avant de se lancer sur la création de notre site web et l'application mobile

Trello :



Un outil de gestion de projet gratuite, qui nous permet d'organiser notre projet sous forme de tableaux, eux-mêmes composés de listes en colonnes, qui répertorient des tâches sous formes de cartes.

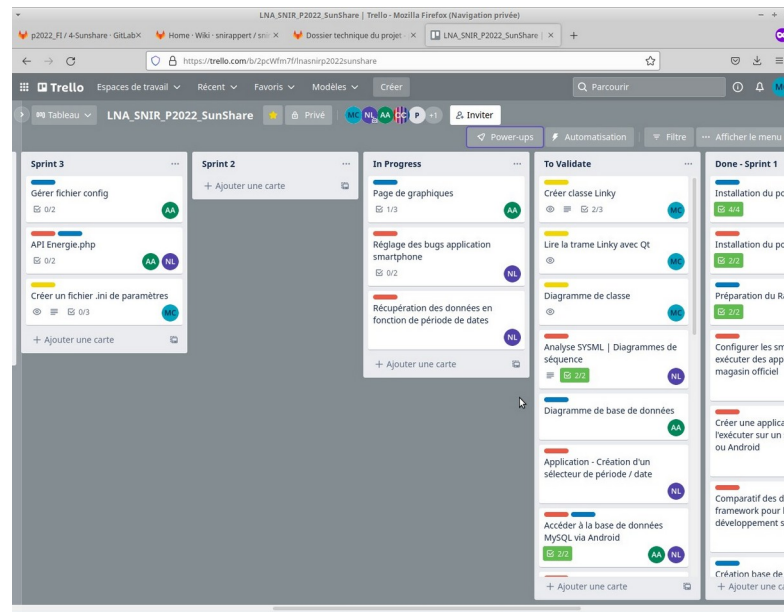


Figure 28: Capture d'écran du planning sur Trello

Gitlab :



Plateforme de développement collaborative
Open-source

Permet de piloter des dépôts de code source et d'en gérer les différentes versions.

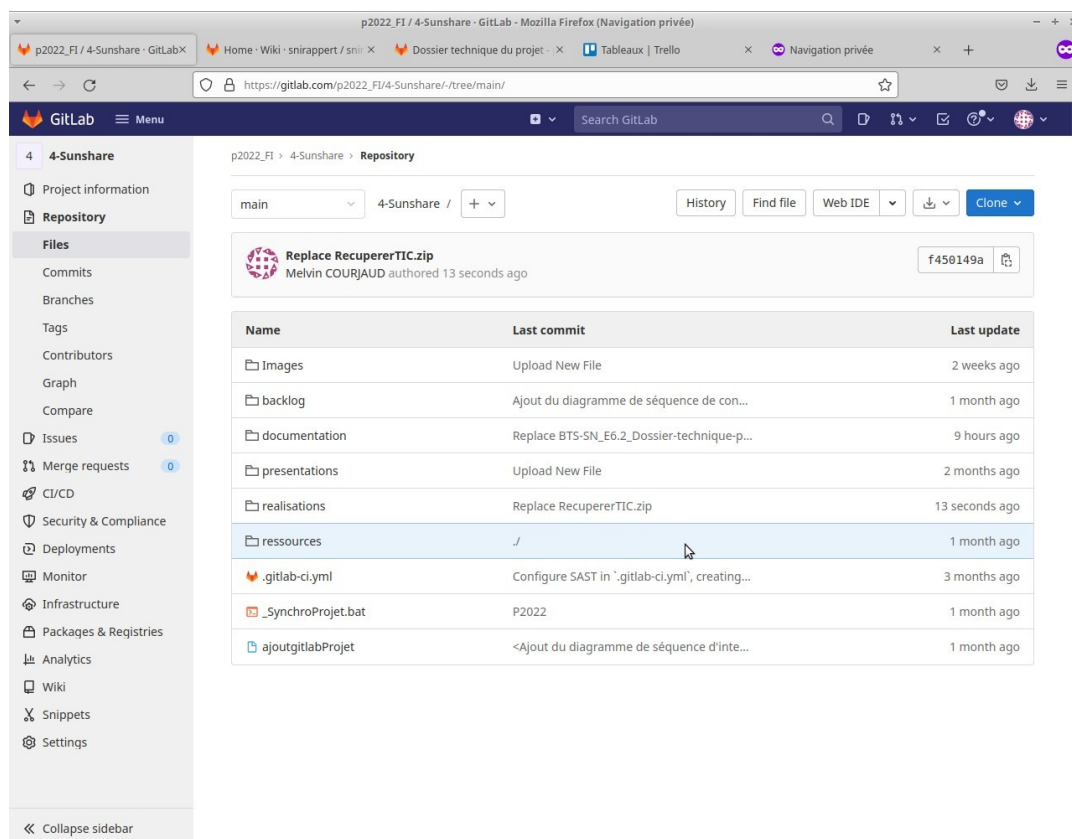


Figure 29: Capture d'écran du projet sur Gitlab

4 - Recette

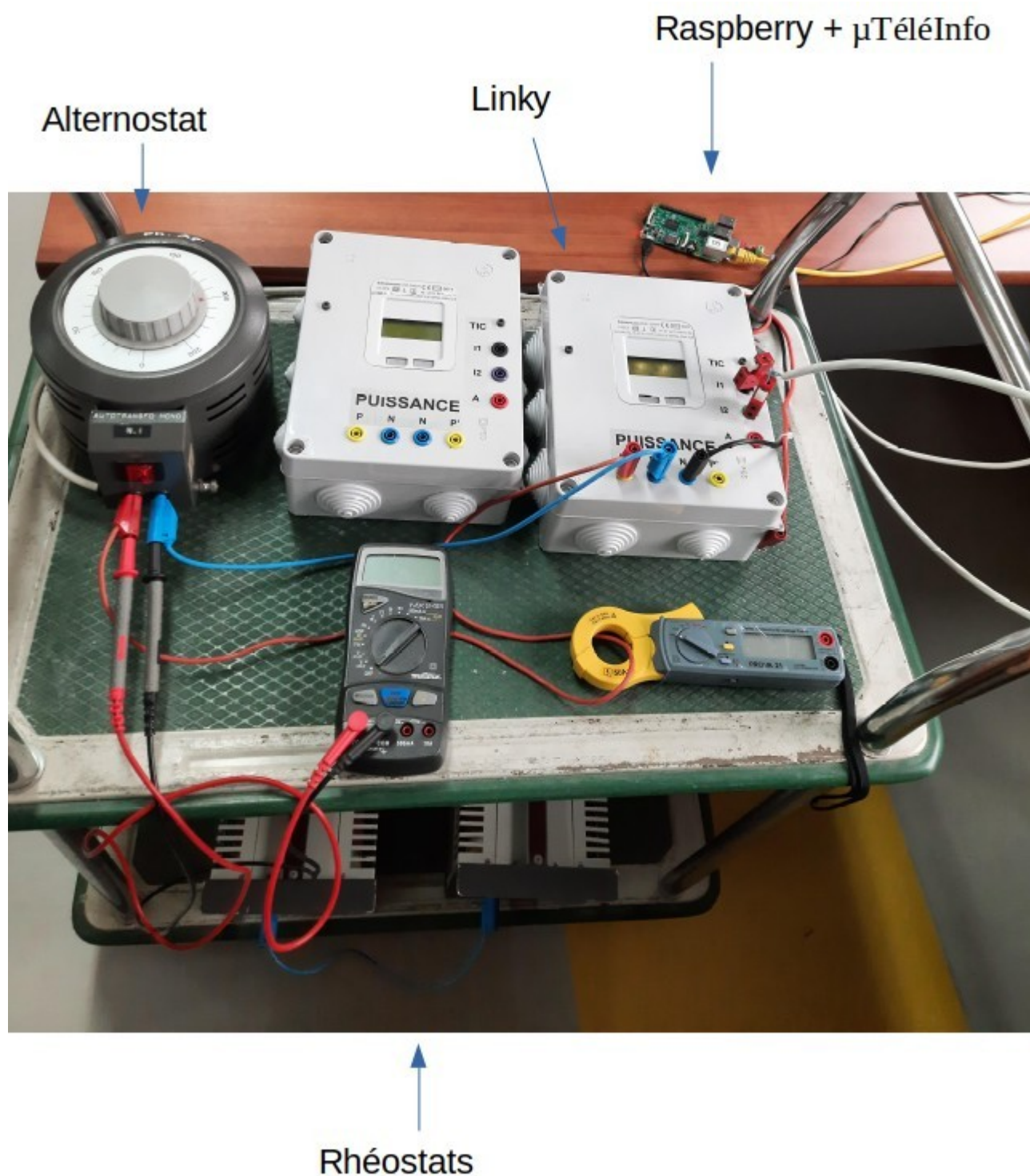


Figure 30: Maquette pour le déploiement

5 - Conclusion

5.1 - Tâches

5.1.1 - En cours

- Récupération des données du Linky fonctionnelle
- Site web fonctionnel et création d'une API pour gérer les données
- Application et affichage des données sous forme de graphiques fonctionnels

5.1.2 - À venir

- Envoi des données récupérées à la base de données
- Utilisation de l'API pour stocker les données sur le long terme
- Récupération des données et ajout de détails graphiques

5.2 - Améliorations envisageables

- Application PC
- Mieux exploiter la TIC (couleur du lendemain, état du Linky)
- Plus d'informations disponibles pour l'utilisateur (graphiques, données résumées de la veille...)