
Analysis Report

Ethan GAUTHIER - Sterenn LE HIR
Alexandre LECOMTE - Axel MARTIN
Kechiedou MEDA - Ivann VYSLANKO

Sommaire

1	Rappel du besoin et critères de succès	2
1.1	Objectifs	2
1.2	User stories	2
2	Modèle du domaine métier	3
2.1	Lexique	3
2.2	Logique métier	4
3	Description de l'écosystème	5
3.1	Personnes et Partenaires	5
3.2	Technologies et Outils	5
3.3	Processus et Méthodologie	6
3.4	Données	6
3.5	Communauté et Support	6
4	Principe de solution	6

1 Rappel du besoin et critères de succès

1.1 Objectifs

Le projet a pour objectif de concevoir, développer et mettre en œuvre un système de tableau de bord interactif et une application mobile permettant le contrôle à distance d'un chauffage canette.

Ce chauffage canette, placé en extérieur, emmagasine la chaleur issue du soleil, à la manière d'un panneau solaire. Il se constitue d'une caisse en bois rectangulaire, munie sur sa façade d'une vitre en plexiglass, et dans laquelle des canettes peintes en noir ont été percées et empilées pour former des tubes. Ces tubes récupèrent l'énergie thermique des rayons du soleil et la stocke. Deux trappes, en haut et en bas du système permettent à l'air de circuler dans les tubes et ainsi être réchauffé, de bas en haut (l'air froid entre par en bas et remonte en se réchauffant). On connecte ces trappes à la pièce qu'on souhaite chauffer.

Le système proposé doit offrir aux utilisateurs la possibilité de surveiller en temps réel la température, de régler les paramètres de chauffage à distance, et de recevoir des notifications liées à l'état de fonctionnement. L'intégration du système avec le chauffage canette existant doit être réalisée de manière sécurisée, en garantissant la confidentialité des données et en respectant les normes de sécurité applicables. L'objectif ultime est d'optimiser l'efficacité du chauffage canette, en permettant aux utilisateurs de maximiser l'utilisation de la chaleur stockée tout en facilitant la gestion de l'appareil via une interface mobile ergonomique.

1.2 User stories

On propose une liste d'histoires utilisateurs classée par ordre de priorité :

- En tant qu'utilisateur, je souhaite allumer et éteindre le chauffage manuellement.
- En tant qu'utilisateur, je souhaite régler la température du chauffage manuellement.
- En tant qu'utilisateur, je souhaite voir la température de la pièce et du chauffage.
- En tant que responsable, je souhaite allumer et éteindre le chauffage depuis une application.
- En tant que responsable, je souhaite régler la température du chauffage depuis une application.
- En tant que responsable, je souhaite voir l'énergie produite par le chauffage.

- En tant que responsable, je souhaite prédire la puissance produite par le chauffage.
- En tant que responsable, je souhaite programmer des routines d’allumage. Par exemple, je veux que tous les jours, à 18h, le chauffage s’allume.
- En tant que responsable, je souhaite que les entrées et sorties d’air soient fermées la nuit pour ne pas perdre l’air chaud de la pièce.
- En tant qu’utilisateur, je souhaite pouvoir accéder à l’application pour visualiser les données.

Notre critère de succès pour le projet est de remplir les sept premières user stories, le reste étant facultatif.

2 Modèle du domaine métier

2.1 Lexique

On détaille dans cette partie tout le vocabulaire nécessaire à la compréhension du projet.

Chauffage canette : système physique comprenant le boîtier de chauffage et le système embarqué

Système embarqué : Système composé du microcontrôleur, de son logiciel embarqué, et de ses capteurs/boutons/périphériques

Logiciel embarqué : programme s’exécutant sur notre microcontrôleur

ESP-32 : Carte microcontrôleur intégrant la gestion du Wi-Fi et du Bluetooth

RPC : Remote procedure call ou procédure d’appel à distance

Puissance : Quantité d’énergie par unité de temps fournie par un système à un autre

Cloud : Désigne tous nos serveurs distants communicants avec le système embarqué ou l’application

API : Interface logicielle pour se connecter à un de nos service

Application Mobile : notre application, s’exécutant sur smartphone Android ou IOS

Flutter : kit de développement logiciel d’interface utilisateur open-source créé par Google, utilisé pour développer notre application

Dart : langage de programmation optimisé pour les applications, créé par Google.

Firebase : Service sur lequel sera hébergé notre système de base de données

2.2 Logique métier

Pour déterminer la puissance de notre chauffage, on effectue des mesures de la température et du débit d'air. Pour ce faire, on utilise la formule fondamentale de transfert de chaleur. La puissance thermique (Q) peut être calculée en utilisant l'équation suivante :

$$Q = m * c * \Delta T$$

- Q : la puissance thermique (W)
- m : le débit massique d'air (kg/s)
- c : la capacité thermique de l'air à pression constante = 1005 J/(kg.°C).
- ΔT : la différence de température en entrée et sortie du chauffage (°C)

$$\text{avec } m = \rho * A * V$$

- ρ : la densité de l'air (kg/m³)
- A : la section transversale du flux d'air (m²)
- V : la vitesse de l'air (m/s)

On a trois méthodes possibles pour calculer m :

- Utiliser un débitmètre d'air qui nous donnera directement la valeur m . Avantage : mesures précises de m . Inconvénients : prix ($\approx 20+\text{€}$), réduire l'interface d'entrée et/ou sortie d'air pour qu'elle corresponde au diamètre du capteur
- Utiliser un anémomètre pour connaître la vitesse de l'air V et déduire m par la formule. Avantage : mesures précises de V . Inconvénients : prix ($\approx 20+\text{€}$), semble peu ergonomique, prend de la place
- Utiliser la vitesse des ventilateurs pour connaître la vitesse de l'air V et déduire m par la formule. Avantage : pas de matériel en plus. Inconvénient : moins précis.

On souhaite ensuite pouvoir prédire cette puissance en fonction de plusieurs facteurs :

Récupérés par une api météo :

- l'ensoleillement
- la température extérieure
- l'humidité (optionnel)

Récupéré par notre capteur :

- la température de la pièce

Pour cela, on réalisera une régression linéaire multivariée sur les données obtenues. Cette dernière sera d'autant plus précise qu'on aura récolté des données. Ces calculs s'effectueront dans le cloud sur un serveur de calcul.

3 Description de l'écosystème

On énumère dans cette partie tous les éléments qui composent notre projet, matériels comme immatériels.

3.1 Personnes et Partenaires

La personne à l'origine du projet est notre product owner, Johann Bourcier. Ce dernier surveille l'avancée de l'équipe projet composée d'Ethan Gauthier, Axel Martin, Sterenn Le Hir, Alexandre Lecomte, Kechiedou Meda, Ivann Vyslanko. On compte aussi sur l'aide de la responsable du FabLab, Camille Bisson. Les utilisateurs finaux ne sont pas encore déterminés, on imagine que le chauffage canette et son système seront utilisés par l'université mais rien n'est sûr.

3.2 Technologies et Outils

Cette partie traite des logiciels, frameworks, langages de programmation, bases de données et autres outils utilisés dans le cadre du projet. On utilisera Arduino IDE pour le développement du logiciel embarqué dans l'ESP-32 et Flutter pour l'application mobile. Leurs langages respectifs sont Arduino et Dart. Le MQTT broker s'exécutera sur un serveur Mosquitto. Le système de contrôle à distance et de prédictions s'exécuteront sur un serveur Vercel. Notre prédiction sera développée en Python en utilisant le framework FastAPI. Notre système de base d'historisation des données s'exécutera sur Firebase. On gèrera la qualité du code par SonarQube PMD PITest et l'intégration continue sera assurée par Jenkins.

On aura besoin d'acheter différents matériels et capteurs, on propose alors la *Bill of materials* suivante :

- 1 ESP-32 : 10€
- 2 capteurs de température : 5€
- 1 capteur d'humidité : 4€
- 1 débitmètre : 20€
- 2 servomoteurs : 10€
- fils
- 2 trappes

3.3 Processus et Méthodologie

On utilise la méthode Agile pour découper notre projet sous forme de sprint successifs. D'autres outils de gestion de projet sont mis en place, Clickup pour gérer notre avancement dans les tâches et le diagramme de Gantt pour la planification globale.

3.4 Données

Différents types de données sont échangés : des requêtes RPC (allumer/éteindre à distance) des données chiffrées (température, puissance...). La source de ces données est diverse, elles proviennent de capteurs, d'API internes ou externes à notre système ou de notre Application. Ces données sont échangées par le protocole MQTT et HTTPs. Le stockage de ces données se fait en base de données dans Firebase.

3.5 Communauté et Support

Notre projet est open source et disponible sur Github au lien suivant <https://github.com/chauffapagnan/documentation>. La communauté GitHub peut contribuer au projet par pull request.

4 Principe de solution

Notre solution se compose donc du chauffage canette et de son système embarqué, d'une application mobile pour piloter le chauffage et consulter des données à distance et de plusieurs API dans le cloud où l'on gère les RPC et stocke la logique métier et les données.