

RAPPORT DE STAGE D'ÉTÉ

Présenté en vue de l'obtention de la

LICENCE EN SCIENCE DE L'INFORMATIQUE

Parcours : Génie Logiciel et Systeme d'Information

Par

INSAF BEN TALOUBA

Développement d'une application Web : Gestion des Utilisateurs et Suivi en Temps Réel

Réalisé au sein de Orbit Engineering Solutions



Président :
Rapporteur :

M.Wassim ABBESSI
M.Hazem YAHYAOUÏ

Année Universitaire : 2024-2025

Remerciements

Je tiens à exprimer ma profonde gratitude à toute l'équipe de "Orbit Engineering Solutions" pour m'avoir accueilli et accompagné durant cette première expérience professionnelle enrichissante.

Je remercie particulièrement mon encadrant en entreprise pour sa disponibilité et ses précieux conseils, qui m'ont permis de progresser et d'acquérir de nouvelles compétences dans un environnement de travail stimulant.

Table des matières

Remerciements	1
Introduction Générale	6
1 Cadre de projet	7
1.1 Introduction	7
1.2 Présentation de l'organisme d'accueil	7
1.3 Etude de l'existant	7
1.4 Critique de l'existant	8
1.5 Solution envisagée	8
1.6 Méthodologie de travail et planification	8
1.6.1 Choix de modèle en V	9
1.7 Conclusion	9
2 Spécification des besoins	10
2.1 Introduction	10
2.2 Identification des besoins fonctionnels	10
2.3 Identification des besoins non fonctionnels	10
2.4 Identification des acteurs	11
2.5 Diagramme de cas d'utilisation	11
2.6 Conclusion	11
3 Etude Conceptuelle	12
3.1 Introduction	12
3.2 Détails des cas d'utilisation	12
3.2.1 Raffinement du cas d'utilisation "Consulter Tableau de Bord"	12
3.2.2 Raffinement du cas d'utilisation "Gérer les utilisateurs"	13
3.3 Diagramme De Classes Global	14
3.4 Diagrammes de séquences	15
3.4.1 Diagramme de séquences "Gérer utilisateurs"	16
3.4.2 Diagramme de séquence "Consulter tableBord"	17

3.5	Conclusion	17
4	Réalisation	18
4.1	Introduction	18
4.2	Outils de travail	18
4.3	Interfaces utilisateurs	20
4.3.1	Tableau de bord	20
4.3.2	Interfaces de gestion des utilisateurs	21
4.4	Conclusion	22
	Conclusion	23
	Nétographie	24

Table des figures

1.1	Logo de la société	7
1.2	Processus de développement en V	8
2.1	Diagramme de cas d'utilisation	11
3.1	Diagramme de cas d'utilisation	12
3.2	Diagramme de cas d'utilisation «Gérer utilisateurs»	13
3.3	Diagramme de classes global	15
3.4	Diagramme de séquence "gerer utilisateurs"	16
3.5	Diagramme de séquence "Consulter TableBord"	17
4.1	Outils de travail	19
4.2	Tableau de bord	20
4.3	Widget de réglage de l'horloge.	20
4.4	Design de l'historique des données(non fonctionnel).	20
4.5	Interface d'ajout des utilisateurs.	21
4.6	Interface d'affichage ,recherche et suppression des utilisateurs.	21
4.7	Interface modification d'utilisateur	22

Liste des tableaux

2.1	Identification des acteurs	11
3.1	Raffinement du cas d'utilisation "Consulter Tableau de Bord"	13
3.2	Raffinement du cas d'utilisation "Gérer Utilisateurs"	14
4.1	Tableau des références	24

Introduction Générale

Dans le cadre de l'évolution rapide des technologies numériques, la gestion efficace des ressources énergétiques à l'aide de solutions IoT (Internet of Things) est devenue un enjeu majeur pour les entreprises.

Mon stage s'inscrit dans ce contexte technologique innovant, avec pour objectif principal de contribuer au développement d'interfaces web facilitant la gestion et le suivi des données. Le stage s'est déroulé sous la forme de deux tâches distinctes :

- Première tâche : Développement d'une interface permettant la gestion des utilisateurs à travers un système "CRUD" (Create, Read, Update, Delete).
- Deuxième tâche : Création d'une interface front-end sous forme de "dashboard" affichant en temps réel des indicateurs tels que la température et l'humidité.

Afin de garantir une organisation méthodique du projet et une meilleure traçabilité des différentes étapes de développement, nous avons adopté la méthodologie en V.

Le rapport est structuré comme suit :

Chapitre 1 : Cadre du projet – Analyse de l'existant et présentation de la solution envisagée.

Chapitre 2 : Spécification des besoins – Définition et modélisation des besoins fonctionnels et non fonctionnels.

Chapitre 3 : Étude conceptuelle – Conception détaillée avec des diagrammes UML.

Chapitre 4 : Réalisation – Développement et mise en œuvre des interfaces.

Chapitre 1

Cadre de projet

1.1 Introduction

Ce chapitre présente le cadre du projet réalisé au sein d'Orbit Engineering Solutions. Il décrit l'organisme d'accueil, analyse les limites des systèmes existants, propose une solution adaptée, et détaille la méthodologie de travail adoptée pour mener à bien le développement du projet.

1.2 Présentation de l'organisme d'accueil

Orbit Engineering Solutions est une entreprise spécialisée dans les solutions numériques pour la gestion intelligente de l'énergie et des solutions durables basées sur le cloud et l'Internet des Objets (IoT). Figure 1.1 présente le logo de la société Orbit Engineering Solutions.



FIGURE 1.1 – Logo de la société

1.3 Etude de l'existant

Chez Orbit Engineering Solutions, la gestion des utilisateurs est fragmentée, ce qui complique les opérations CRUD. De plus, les données collectées ne sont pas centralisées dans un tableau de bord, limitant leur exploitation efficace. L'absence d'automatisation alourdit le traitement des informations,

justifiant le besoin d'une solution intégrée et d'un suivi en temps réel pour améliorer la réactivité.

1.4 Critique de l'existant

L'approche actuelle présente plusieurs limitations : l'absence de centralisation des données entrave une gestion fluide des informations et des utilisateurs. Le manque d'automatisation engendre une surcharge de travail manuel, réduisant l'efficacité globale. De plus, l'absence d'un tableau de bord en temps réel empêche une supervision proactive et limite la capacité à réagir rapidement aux anomalies. Ces lacunes freinent l'optimisation des processus et soulignent la nécessité d'une solution plus intégrée et performante.

1.5 Solution envisagée

La solution consiste à créer un système de gestion des utilisateurs pour améliorer l'organisation et le suivi. Pour le deuxième problème, il s'agit de développer une interface avec un tableau de bord permettant de visualiser en temps réel des données comme la température et l'humidité.

1.6 Méthodologie de travail et planification

Pour s'assurer que le projet se déroule comme souhaité et dans les délais, une organisation méthodologique est essentielle. Nous avons choisi d'utiliser le modèle en forme de V pour ce projet. Figure 1.2 présente le processus de développement en V.

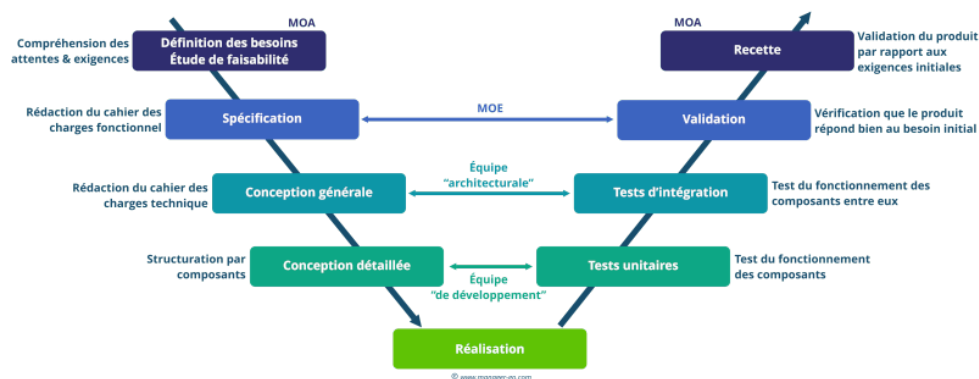


FIGURE 1.2 – Processus de développement en V

1.6.1 Choix de modèle en V

Le principal avantage du cycle en V est qu'il évite de revenir en arrière incessamment pour redéfinir les spécifications initiales, comme un cliquet. Chaque phase de conception demande la rédaction d'une documentation précise et exhaustive, où chaque point doit être validé par le produit final. Dès lors qu'une étape est validée, on ne revient pas en arrière et on passe à l'étape suivante sur une base solide ; c'est la principale force du cycle en V.[?]

1.7 Conclusion

Ce chapitre a mis en évidence les limites des systèmes actuels chez Orbit Engineering Solutions, notamment le manque de centralisation et d'automatisation. Nous proposons de créer un système de gestion des utilisateurs et une interface de tableau de bord pour améliorer l'efficacité et la réactivité. Ces solutions visent à optimiser les processus de l'entreprise.

Chapitre 2

Spécification des besoins

2.1 Introduction

Cette section identifie et décrit les besoins fonctionnels et non fonctionnels du système de gestion des utilisateurs et de l'interface de tableau de bord, orientant ainsi le développement vers des solutions efficaces pour Orbit Engineering Solutions.

2.2 Identification des besoins fonctionnels

Un besoin fonctionnel est un besoin spécifiant une action qu'un système doit être capable d'effectuer, sans considérer aucune contrainte physique. C'est un besoin du point de vue de l'utilisateur. Notre projet consiste à réaliser un système de gestion des utilisateurs qui répond aux besoins suivants :

- Gérer les utilisateurs ;

2.3 Identification des besoins non fonctionnels

Les besoins non fonctionnels concernent la qualité du système sans toucher directement à ses fonctionnalités. Voici les principaux points :

- **Performance** : Le tableau de bord doit afficher les données en temps réel sans délai perceptible.
- **Fiabilité** : Le système doit être accessible 99,9 % du temps avec des données fiables et mises à jour instantanément.
- **Accessibilité** : Le tableau de bord doit être accessible via un navigateur, sur ordinateur et mobile.

2.4 Identification des acteurs

Dans le cadre de notre projet, deux acteurs principaux ont été identifiés :

- Administrateur : Responsable de la gestion des utilisateurs. L'administrateur a également accès au tableau de bord pour visualiser les données en temps réel et assurer le suivi des équipements.
- Client/Utilisateur : Ce rôle concerne les clients du dispositif IoT, qui peuvent accéder au tableau de bord pour visualiser les données en temps réel.

TABLE 2.1 – Identification des acteurs

Acteur	Rôles
Administrateur	Gérer les utilisateurs Consulter le tableau de bord
Utilisateur	Consulter le tableau de bord

2.5 Diagramme de cas d'utilisation

Ce diagramme 2.1 est un diagramme de cas d'utilisation illustrant les interactions entre les utilisateurs et le système.

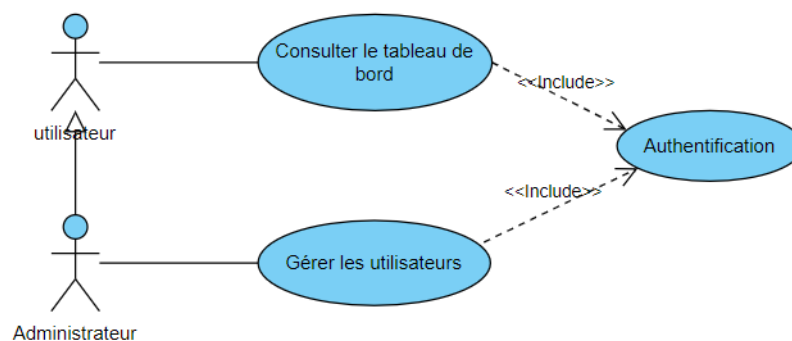


FIGURE 2.1 – Diagramme de cas d'utilisation

2.6 Conclusion

Ce chapitre a défini les besoins fonctionnels et non fonctionnels du projet, identifié les acteurs clés, et présenté un diagramme de cas d'utilisation. Ces éléments servent de base pour guider le développement vers une solution adaptée aux attentes.

Chapitre 3

Etude Conceptuelle

3.1 Introduction

Ce chapitre présente l'étude conceptuelle du projet, qui constitue une étape essentielle pour organiser et clarifier le fonctionnement du système avant sa réalisation.

3.2 Détails des cas d'utilisation

Dans cette partie en vas détailler les cas d'utilisations.

3.2.1 Raffinement du cas d'utilisation "Consulter Tableau de Bord"

Le diagramme 3.1 représente le cas d'utilisation "Consulter le tableau de bord", accessible à la fois par un utilisateur et un administrateur.

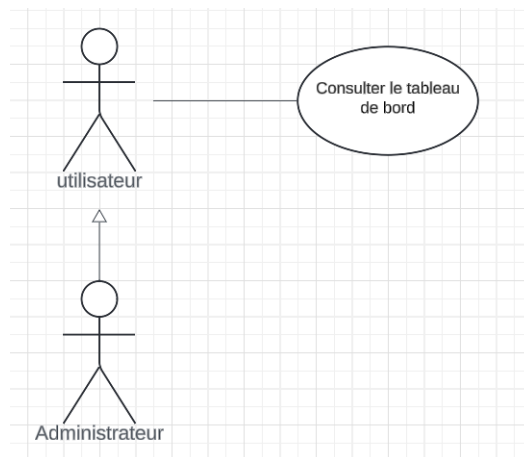


FIGURE 3.1 – Diagramme de cas d'utilisation

Le tableau décrit le cas d'utilisation "Consulter Tableau de Bord", en précisant les acteurs, les préconditions, les postconditions, le scénario principal, et un scénario d'exception.

TABLE 3.1 – Raffinement du cas d'utilisation "Consulter Tableau de Bord"

Cas d'utilisation	Consulter Tableau de Bord
Acteur	- Utilisateur - administrateur
Pré-condition	- Système en marche - Connexion établie
Post-condition	Tableau de bord affiché
Description de scénario principal	- L'utilisateur accède au tableau de bord. - Le tableau de bord affiche des graphiques et des données en temps réel. - L'utilisateur peut ajuster les valeurs minimales et maximales de l'horloge.
Scénario d'exception	- les valeurs s'affichent comme null.

3.2.2 Raffinement du cas d'utilisation "Gérer les utilisateurs"

Le diagramme 3.2 illustre les cas d'utilisation liés à la gestion des utilisateurs, permettant à un administrateur d'ajouter, consulter, modifier ou supprimer des utilisateurs.

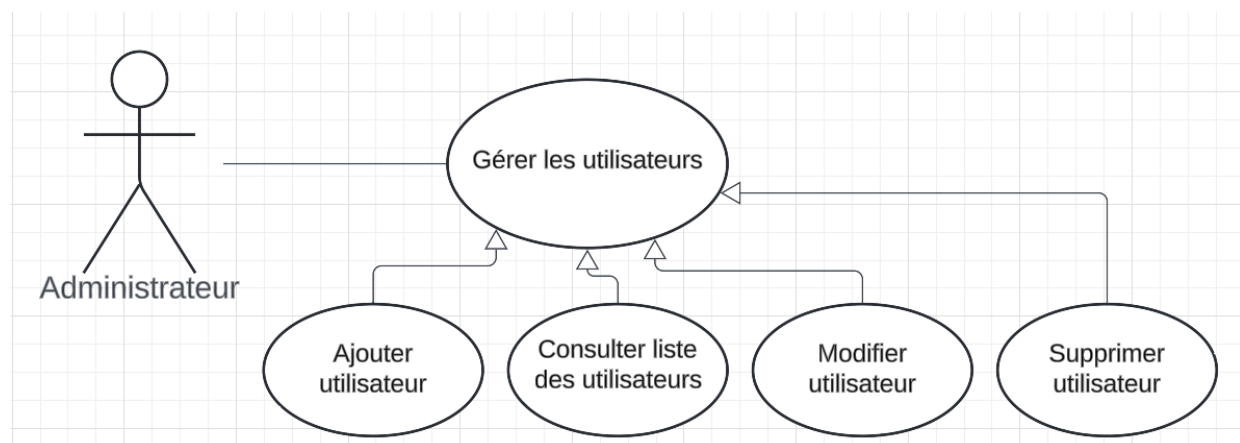


FIGURE 3.2 – Diagramme de cas d'utilisation «Gérer utilisateurs»

Le tableau détaille le cas d'utilisation "Gérer les Utilisateurs", précisant l'acteur (administrateur), les préconditions, les postconditions et les scénarios principaux.

TABLE 3.2 – Raffinement du cas d'utilisation "Gérer Utilisateurs"

Cas d'utilisation	Gérer Les Utilisateur
Acteur	-Administrateur
Pré-condition	<ul style="list-style-type: none"> - Système en marche - AgentRéception est authentifié - Connexion établie
Post-condition	Utilisateur géré
Description de scénario principal	<ul style="list-style-type: none"> - Le système affiche l'interface. - L'administrateur choisit l'opération de gestion des dossiers. - Le système affiche l'interface approprié selon le choix de L'admin .
Scénarii	<ul style="list-style-type: none"> -Consulter liste des utilisateur -Créer utilisateur -Modifier utilisateur -Supprimer utilisateur

3.3 Diagramme De Classes Global

Le diagramme de classes global offre une vue synthétique de l'organisation des entités et de leurs interactions au sein du projet fédéré. En résumant les classes, leurs attributs et leurs relations, il constitue un outil clé pour comprendre l'architecture du système.[?]

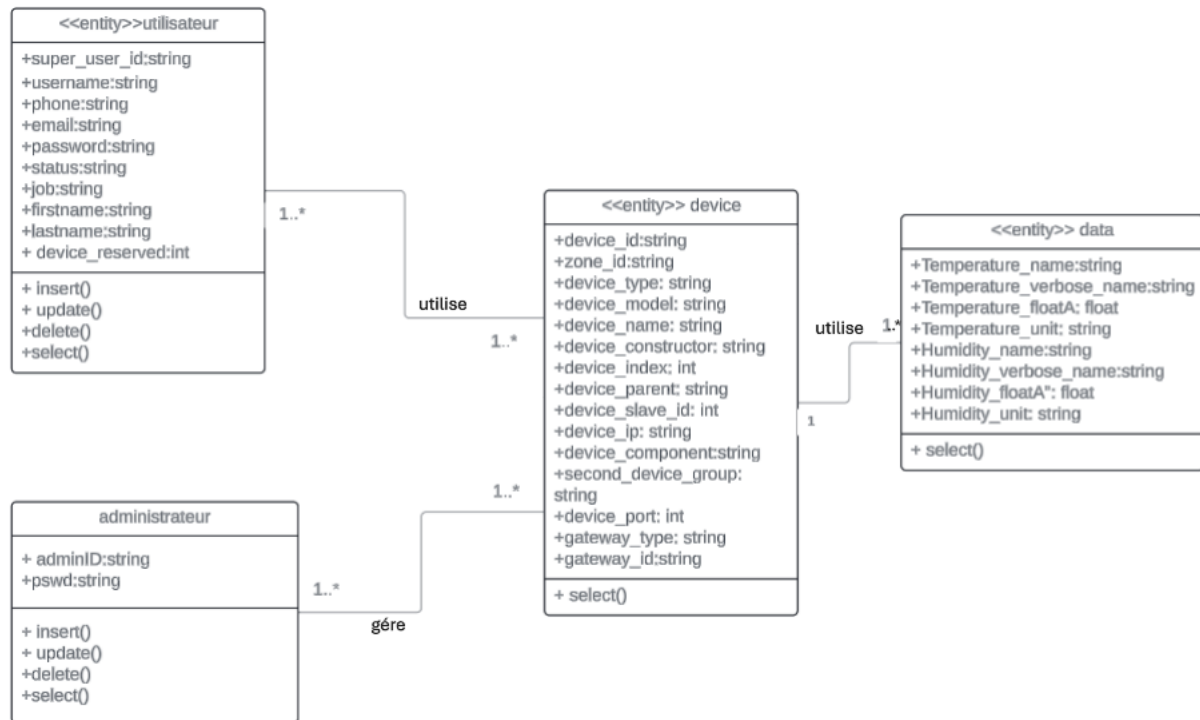


FIGURE 3.3 – Diagramme de classes global

3.4 Diagrammes de séquences

Les diagrammes de séquence sont une solution populaire de modélisation dynamique en langage UML, car ils se concentrent plus précisément sur les lignes de vie, les processus et les objets qui vivent simultanément, et les messages qu'ils échangent entre eux pour exercer une fonction avant la fin de la ligne de vie.[?]

3.4.1 Diagramme de séquences "Gérer utilisateurs"

Le diagramme de séquences "Gerer utilisateur" montre les interactions entre l'administrateur et l'interface de gestion des utilisateurs pour créer, modifier, consulter ou supprimer des utilisateurs.

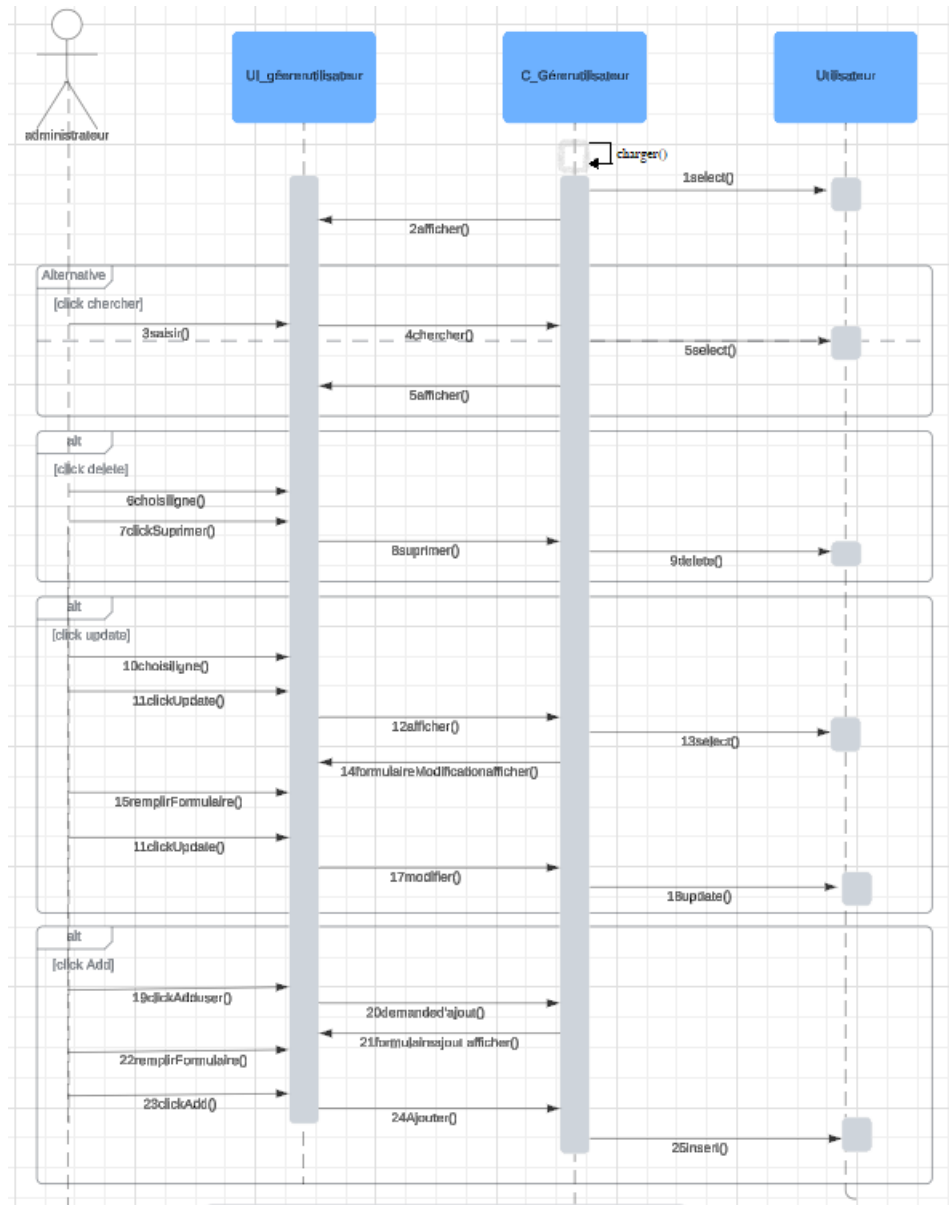


FIGURE 3.4 – Diagramme de séquence "gerer utilisateurs"

3.4.2 Diagramme de séquence "Consulter tableBord"

ce diagramme illustre le processus où l'administrateur ajuste les paramètres de l'horloge ou visualise les données IoT en temps réel affichées sur le tableau de bord.

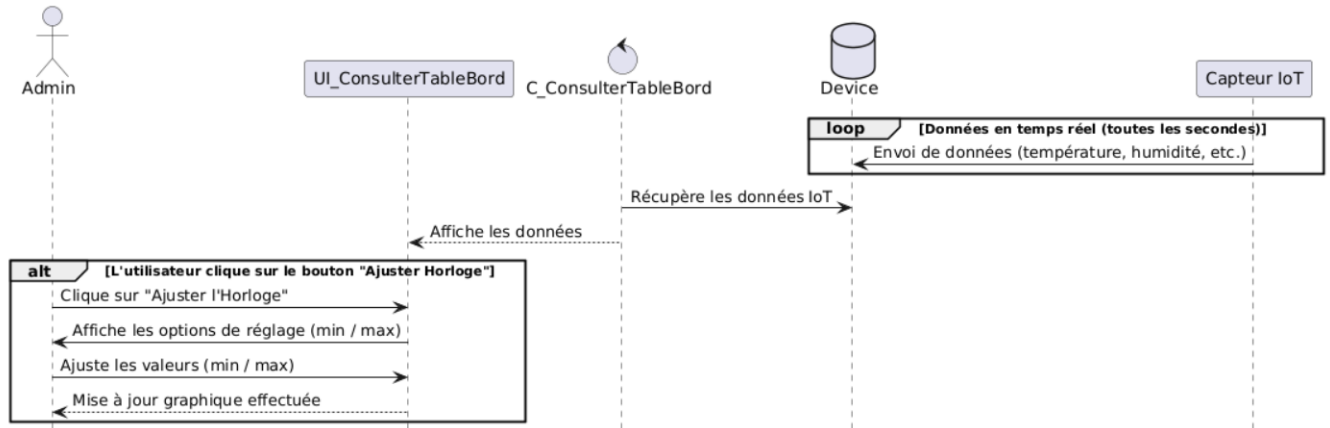


FIGURE 3.5 – Diagramme de séquence "Consulter TableBord"

3.5 Conclusion

En résumé, l'identification des acteurs et leurs besoins nous a permis de réaliser la modélisation de la solution qu'on a proposé et de la présenté dans des différents types de diagrammes (classe, séquence ...).

Chapitre 4

Réalisation

4.1 Introduction

Jusqu'à présent, nous avons donné un aperçu des différents aspects du projet, allant de la spécification des besoins à la conception. Un autre aspect tout aussi essentiel est de rassembler la pile technologique et les outils qui ont permis de mener à bien le développement. Dans ce chapitre, nous présenterons les langages, technologies et matériels utilisés pour réaliser les tâches du projet. En outre, nous partagerons quelques captures d'écran des interfaces développées, notamment celles permettant la gestion des utilisateurs et la visualisation des données IoT en temps réel.

4.2 Outils de travail

Dans mon projet, j'ai utilisé les logiciels et les Technologies de développement suivants :

FIGURE 4.1 – Outils de travail

Logiciel	Définition
	Visual Studio Code (VSCode) est un éditeur de code open-source de Microsoft offrant autocomplétion, débogage, Git intégré et un terminal.
	Postman est un outil d'API permettant de tester, documenter et automatiser les requêtes HTTP.
	MongoDB Compass est une interface graphique permettant d'interagir avec des bases de données MongoDB.
	Vuexy est un template d'interface utilisé pour créer des tableaux de bord modernes et dynamiques.
	React est une bibliothèque open source JavaScript pour créer des interfaces utilisateurs. Elle est maintenue par Meta ainsi que par une communauté de développeurs individuels et d'entreprises depuis 2013
	MERN est un stack technologique combinant MongoDB, Express.js, React.js et Node.js, souvent utilisé pour le développement d'applications web complètes.
	Bootstrap est un framework CSS facilitant la création d'interfaces web responsives.
	Apache ECharts est une bibliothèque de visualisation de données open source, écrite en JavaScript, qui vous permet de créer et d'ajouter des graphiques et des diagrammes personnalisables et interactifs à vos applications Web .

4.3 Interfaces utilisateurs

4.3.1 Tableau de bord

Le tableau de bord permet la visualisation en temps réel des données de température et d'humidité collectées par les dispositifs IoT. Cette interface est accessible à l'administrateur et aux clients concernés.

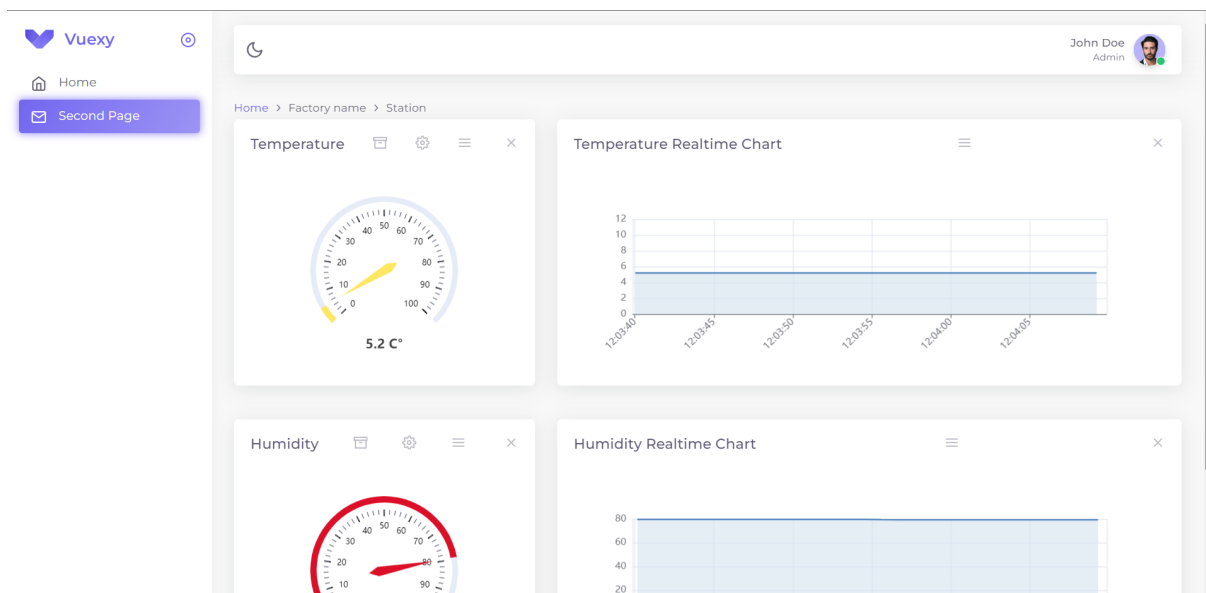


FIGURE 4.2 – Tableau de bord

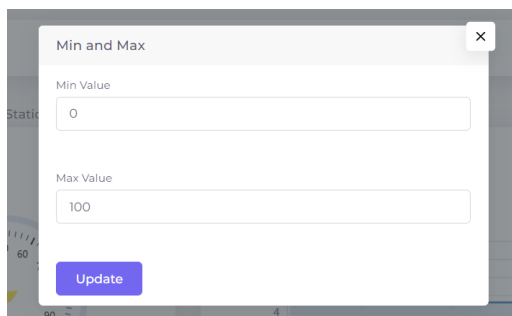


FIGURE 4.3 – Widget de réglage de l'horloge.

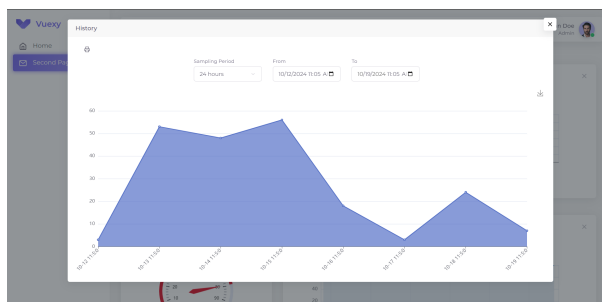


FIGURE 4.4 – Design de l'historique des données(non fonctionnel).

4.3.2 Interfaces de gestion des utilisateurs

La partie de gestion des utilisateurs comprend trois interfaces principales. La première permet d'ajouter un utilisateur en renseignant les informations nécessaires. La deuxième affiche la liste des utilisateurs et offre la possibilité de les supprimer ou de les modifier. Enfin, le bouton "update" permet d'afficher un interface pour modifier les informations d'un utilisateur.

Add User

User Name:

First name:

Email:

Job:

Phone:

Last name:

Password:

Device reserved:

FIGURE 4.5 – Interface d'ajout des utilisateurs.

Users List						
<div>Search <input type="text" value="Search"/></div>						
<input type="checkbox"/>	Username	Name	Email	Job	Device Reserved	Actions
<input type="checkbox"/>	Delice Bouzid	Bouzid Delice	delice-bouzid@delice.com	ing	55	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test connection	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test 2	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test 3	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test 55	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test 33	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test 73	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	test AA8	test test	test@test.com	ing	11	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	foulen test	foulen foulen	bentaloubainsaf@gmail.com	ing	22	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	foulen	foulen henj	foulen@gmail.com	ing	8	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	foulen test	foulen foulen	bentaloubainsaf@gmail.com	ing	1	<input type="button" value="Delete"/> <input type="button" value="Update"/>
<input type="checkbox"/>	final	final final	EMAIL@gmail.org	test	22	<input type="button" value="Delete"/> <input type="button" value="Update"/>

FIGURE 4.6 – Interface d'affichage ,recherche et suppression des utilisateurs.

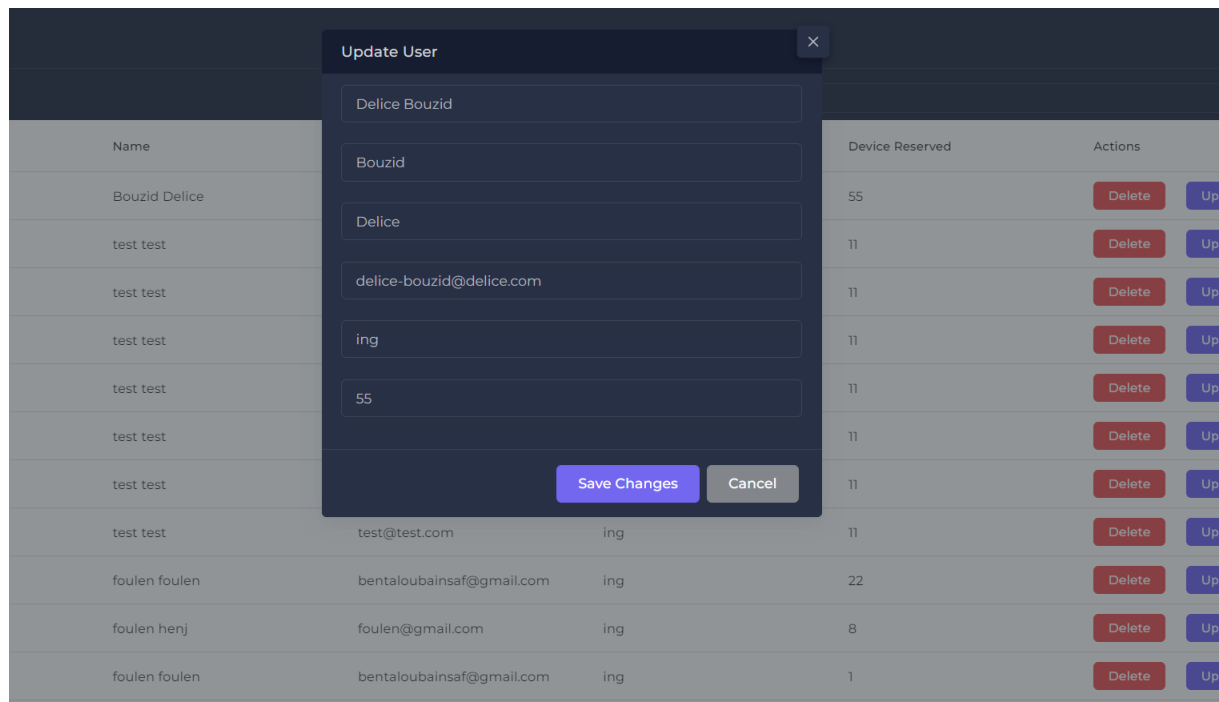


FIGURE 4.7 – Interface modification d'utilisateur

4.4 Conclusion

Dans ce chapitre, nous avons présenté la réalisation du projet en décrivant les outils et technologies utilisés ainsi que les interfaces développées. L'utilisation de technologies telles que le stack MERN, le template Vuexy, et ECharts a permis d'assurer une gestion efficace des utilisateurs et une visualisation en temps réel des données issues des dispositifs IoT.

Ces choix technologiques ont offert une grande flexibilité et facilité de développement, tout en assurant une interface utilisateur moderne et fonctionnelle.

Conclusion

En résumé, ce projet a permis de mettre en pratique les connaissances théoriques tout en développant des compétences techniques autour de technologies modernes telles que MERN, ECharts... . La première tâche a porté sur la mise en place de l'ajout, la modification et la suppression des utilisateurs, ce qui constitue le cœur du système de gestion. La deuxième tâche a été tout aussi importante, car elle s'est concentrée sur l'intégration de la visualisation en temps réel des données via des graphiques interactifs, permettant de mieux suivre et analyser les informations collectées. Bien que certaines fonctionnalités, comme l'historique des données, soient encore au stade de développement, ce projet a enrichi nos compétences en développement web et en gestion de tâches techniques, tout en constituant une expérience formatrice pour des projets futurs.

Nétographie

TABLE 4.1 – Tableau des références

[1]	https://www.manager-go.com/gestion-de-projet/cycle-en-v.html
[2]	https://www.lucidchart.com/pages/diagramme-de-classes-uml
[3]	https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml