

Chapter

0

-10mm

1



# GRADUATION PROJECT REPORT

*presented at*

National Engineering School of Sfax  
(Computer Engineering & Applied Mathematics Department)

*in order to obtain the*

National Engineering Diploma in Computer Science

*by*

**FirstName LASTNAME**

---

My very attractive  
Title

---

Defended on 12/12/1234 in front of the committee composed of

Mr Foulén FOULENI

President

Ms Fouléna FOULENIA

Supervisor

Mr Foulén FOULENI

Reviewer

# Dedication

Put your dedication lines here

And try to be expressive ;)

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

To all of you,

I dedicate this work.

*FirstName* LASTNAME

# Thanks

And put your thanks here.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Table des matières

<b>Dedication</b>	<b>i</b>
<b>Thanks</b>	<b>ii</b>
<b>Table des matières</b>	<b>iv</b>
<b>Table des figures</b>	<b>v</b>
<b>Liste des tableaux</b>	<b>vi</b>
<b>Acronyms</b>	<b>vii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Contexte général</b>	<b>2</b>
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueille . . . . .	3
1.2.1 Présentation générale . . . . .	3
1.2.2 Organigramme de la Société . . . . .	4
1.2.3 Les services de la société . . . . .	4
1.3 Aperçus du projet . . . . .	5
1.3.1 Contexte du projet . . . . .	5
1.3.2 Analyse et critique de l'existant . . . . .	6
1.4 Choix du Méthodologie de travaille . . . . .	11
1.4.1 Différence entre les méthodologie agile et calssique . . . . .	11
1.4.2 Le framework agile adopté SCRUM . . . . .	12
1.4.3 Pilotage du projet avec SCRUM . . . . .	15
1.4.4 Formalisme adopté . . . . .	20
1.5 Conclusion . . . . .	21
<b>2 Analyse et spécification des besoins</b>	<b>22</b>

2.1	Introduction . . . . .	23
2.2	Spécification des besoins . . . . .	23
2.2.1	Identification des acteurs . . . . .	23
2.2.2	Besoins fonctionnels . . . . .	24
2.2.3	Besoins non fonctionnels . . . . .	25
2.3	Diagramme de cas d'utilisation global . . . . .	26
2.4	Architecture globale . . . . .	28
2.4.1	Architecture Logiciel . . . . .	28
2.4.2	Architecture système . . . . .	29
2.5	Environnement du travail . . . . .	30
2.5.1	Environnement matériel . . . . .	30
2.5.2	Environnement logicielle . . . . .	31
<b>Conclusion</b>		<b>37</b>
<b>Appendix</b>		<b>38</b>
<b>Webography</b>		<b>39</b>
<b>Bibliography</b>		<b>39</b>

# Table des figures

1	logo société . . . . .	4
2	Organigramme de Talents Link . . . . .	4
3	Interface de la plateforme FusionSolar . . . . .	6
4	Interface de la plateforme Solax . . . . .	7
5	Interface de la plateforme Growatt . . . . .	8
6	Méthode Agile . . . . .	11
7	Processus du Scrum . . . . .	13
8	Diagramme de cas d'utilisation globale . . . . .	27
9	Architecture MVT . . . . .	28
10	Architecture logiciel . . . . .	29
11	Architecture système . . . . .	30
12	Visual Studio Code . . . . .	31
13	Docker . . . . .	31
14	PostgreSQL . . . . .	32
15	Redis . . . . .	32
16	Vite.js . . . . .	32
17	React.js . . . . .	33
18	Django . . . . .	33
19	Celery . . . . .	34
20	Teams . . . . .	34
21	ClickUp . . . . .	35
22	Notion . . . . .	35
23	GitHub . . . . .	36

# Liste des tableaux

1	Comparaison entre les solutions existantes et la solution proposée . . . . .	10
2	Répartition des rôles dans l'équipe Scrum . . . . .	15
3	Product backlog . . . . .	16
4	Planification des sprints . . . . .	20
5	Identification des acteurs . . . . .	24
6	Outils matériels . . . . .	30



# Acronyms

**ENIS** National Schoole of Engineering of Sfax

# Introduction

Welcome to National Schoole of Engineering of Sfax (ENIS).

Again, welcome to ENIS.

Your introduction goes here.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Chapitre 1

## Contexte général

## **1.1 *Introduction***

Ce chapitre vise à situer le projet dans son contexte en abordant le thème général du système de gestion des installations photovoltaïques, ainsi que l'entité qui l'accueille. En premier lieu, nous présentons une vue d'ensemble du projet, qui a pour objectif d'optimiser le suivi, la maintenance et la performance des systèmes solaires. Ensuite, nous analysons les solutions existantes dans ce domaine afin d'identifier leurs limites et d'en dégager les axes d'amélioration. Cela nous permettra de préciser les objectifs fondamentaux que nous cherchons à atteindre à travers la mise en œuvre de notre propre solution.

## **1.2 *Présentation de l'organisme d'accueil***

### **1.2.1 Présentation générale**

"Talents Link" est une entreprise tunisienne spécialisée dans les ressources humaines, offrant des services complets pour le recrutement, l'intégration et la gestion des talents. Fondée en 2023, elle s'impose comme un partenaire RH de confiance, accompagnant les entreprises à chaque étape de leur processus de gestion des talents.

Sa mission est de connecter les entreprises avec les profils les plus adaptés à leurs besoins, contribuant ainsi à leur croissance et réussite. Talents Link adopte une approche stratégique et personnalisée, qu'il s'agisse de missions ponctuelles ou de collaborations à long terme, garantissant des recrutements efficaces et durables.



FIGURE 1. *logo société*

### 1.2.2 Organigramme de la Société



FIGURE 2. *Organigramme de Talents Link*

### 1.2.3 Les services de la société

Les principaux services proposés par Talents Link incluent :

- **Recrutement et Intégration** : Aider les entreprises à identifier et intégrer les profils

adaptés à leurs besoins.

- **Gestion Administrative et Légale :** Assurer une gestion conforme aux réglementations en vigueur, simplifiant ainsi les démarches administratives pour les clients.
- **Mise à disposition des Talents :** Fournir des professionnels qualifiés pour répondre aux besoins spécifiques des entreprises.
- **Gestion de Carrière :** Accompagner les talents dans leur développement professionnel, en alignement avec les objectifs des entreprises.

## 1.3 *Aperçus du projet*

### 1.3.1 Contexte du projet

Le projet consiste en le développement d'une plateforme de gestion des installations photovoltaïques, intégrant des fonctionnalités avancées pour le suivi, l'analyse et l'optimisation des performances des systèmes solaires. L'objectif principal est de concevoir une solution intelligente et intuitive permettant aux administrateurs, installateurs, techniciens et clients de gérer efficacement leurs installations solaires.

L'un des aspects clés du projet est l'intégration d'un module d'intelligence artificielle capable de prédire les anomalies à partir des données collectées. Cette approche vise à améliorer la maintenance préventive et à maximiser l'efficacité des installations solaires.

Enfin, la plateforme offrira une interface utilisateur ergonomique et interactive, permettant aux utilisateurs d'accéder à des tableaux de bord détaillés, des visualisations dynamiques et des outils d'analyse statistique pour une meilleure prise de décision.

## 1.3.2 Analyse et critique de l'existant

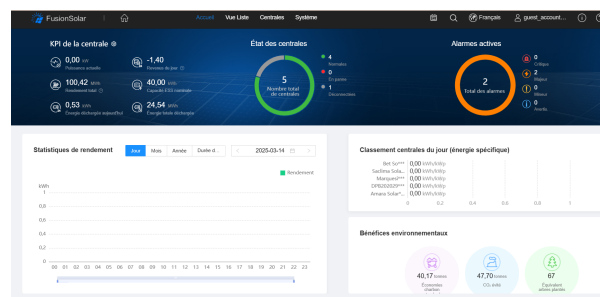
### 1.3.2.1 Étude de l'existant

Dans le domaine de la gestion des installations photovoltaïques, plusieurs plateformes existent déjà sur le marché, proposant diverses fonctionnalités pour le suivi, l'optimisation et la maintenance des systèmes solaires. Parmi les solutions les plus utilisées, on retrouve "Huawei", "Solax", "Kastar" et "Growatt"

#### a) Huawei-FusionSolar

Huawei, à travers sa solution FusionSolar, est l'un des leaders mondiaux en matière de gestion intelligente d'installations solaires.

- Interface intuitive : Disponible sur web et mobile, l'interface permet un accès rapide aux données de production.
- Surveillance en temps réel : Visualisation instantanée des performances de chaque panneau ou onduleur.
- Maintenance prédictive : Grâce à l'IA, le système peut anticiper les pannes et recommander des actions de maintenance avant que les défaillances ne se produisent.
- Compatibilité étendue : Supporte plusieurs marques et types d'équipements solaires.

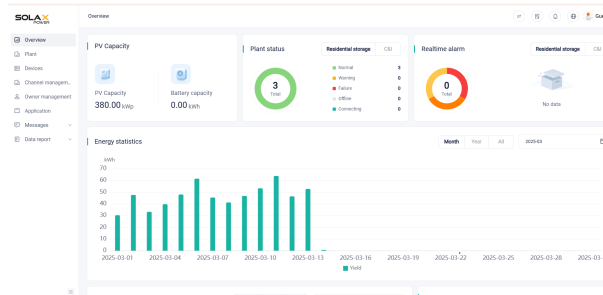


**FIGURE 3.** Interface de la plateforme FusionSolar

#### b) Solax Cloud

Solax est particulièrement reconnu pour ses onduleurs hybrides et son système de stockage intelligent.

- Suivi en temps réel : Données de production et de consommation consultables à distance.
- Optimisation de la consommation : Permet de prioriser l'autoconsommation ou l'injection vers le réseau.
- Stockage intelligent : Gestion automatique des batteries selon les besoins et la production.



**FIGURE 4.** *Interface de la plateforme Solax*

#### c) **Kastar Monitoring**

Kastar propose une solution plus simple mais efficace pour le suivi des installations solaires, souvent utilisée dans les systèmes résidentiels ou de petite taille.

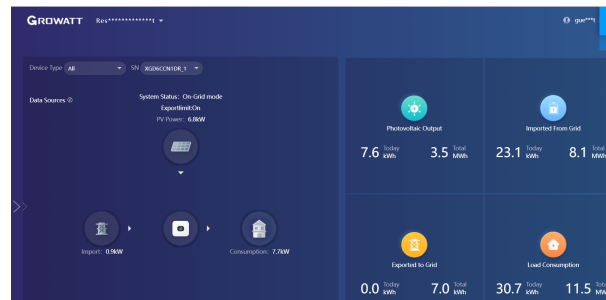
- Interface claire : Visualisation rapide des principales données (tension, courant, puissance ...)
- Alertes intelligentes : Envoi d'alertes par e-mail ou notification mobile en cas d'anomalie.
- Rapports de performance : Génération automatique de rapports journaliers, hebdomadaires ou mensuels.

#### d) **Growatt**

Growatt propose deux interfaces complémentaires : Shine Server (web) et ShinePhone (application mobile).

- Monitoring à distance : Contrôle complet des onduleurs et batteries à partir de n'importe quel appareil.
- Fonctionne avec une grande variété de produits Growatt.
- Affichage graphique des données, comparaison des performances.





**FIGURE 5.** *Interface de la plateforme Growatt*

### 1.3.2.2 Critique de l'existant

Bien que les plateformes existantes telles que "Huawei", "Solax", "Kastar" et "Growatt" proposent des solutions technologiques avancées pour la gestion des installations photovoltaïques, elles présentent plusieurs limites fonctionnelles et contraintes d'usage. Ces lacunes ouvrent la voie au développement de solutions plus flexibles, intelligentes et centrées sur l'utilisateur.

#### — Compatibilité limitée

La majorité des plateformes sont conçues pour fonctionner principalement avec les équipements de leur propre marque, ce qui limite l'interopérabilité avec d'autres fabricants. Cela restreint le choix des installateurs et freine l'adoption dans des environnements mixtes.

#### — Manque de personnalisation

Les interfaces actuelles offrent peu de possibilités de personnaliser les tableaux de bord, les rapports ou les indicateurs de performance.

#### — Interfaces peu intuitives

Certaines solutions, notamment celles destinées aux professionnels, présentent des interfaces complexes, parfois techniques et peu accessibles aux utilisateurs non-experts.

#### — Support client insuffisant

Les utilisateurs rencontrent souvent des difficultés liées à un support technique lent, une documentation peu claire ou incomplète, ce qui complique l'installation initiale, la configuration et le dépannage en cas de problème.

### 1.3.2.3 Solution Proposée

Pour surmonter les défis identifiés, notre approche consiste à développer une plateforme intelligente et flexible pour la gestion des installations photovoltaïques. Elle permettra d'intégrer plusieurs équipements de différentes marques, garantissant une compatibilité universelle. Grâce à un suivi en temps réel, une analyse avancée des données et une détection prédictive des anomalies via l'intelligence artificielle, les utilisateurs pourront optimiser la gestion de leurs installations.

De plus, la plateforme offrira également des fonctionnalités avancées, telles que la planification des interventions, permettant aux installateurs et techniciens d'organiser, suivre et optimiser leurs actions pour une meilleure efficacité. Les utilisateurs pourront visualiser et filtrer les informations, recevoir des alertes intelligentes en cas de dysfonctionnement et générer des rapports personnalisés. Avec une interface intuitive et des outils d'optimisation, cette solution garantira une gestion centralisée, efficace et proactive des installations photovoltaïques.

**TABLE 1.** *Comparaison entre les solutions existantes et la solution proposée*

<b>Critère</b>	<b>Solutions existantes</b>	<b>Solution proposée (Starck)</b>
Compatibilité avec les équipements	Limitation à leur propre écosystème	Compatible avec plusieurs marques via protocoles standards
Personnalisation de l'interface	Interface figée, peu personnalisable	Tableaux de bord adaptables selon le rôle et les besoins de l'utilisateur
Utilisation de l'IA	Faible ou absente	IA intégrée pour détection des anomalies
Historisation des données	Historisation partielle ou peu exploitable	Historisation complète, filtrable, visualisation temporelle
Alertes et notifications	Alertes basiques ou retardées	Alertes en temps réel
Simplicité d'utilisation	Interfaces souvent complexes, peu intuitives	Interface moderne, ergonomique, pensée pour tous les profils
Coût / accessibilité	Coût élevé, parfois réservé aux grandes structures	Tarification flexible adaptée au marché local, version gratuite pour les petites installations

## 1.4 *Choix du Méthodologie de travail*

### 1.4.1 Différence entre les méthodologie agile et calssique

#### 1.4.1.1 les méthodologies classiques

La méthodologie classique, souvent désignée par les termes "cycle en V" ou "cascade", repose sur une approche linéaire et séquentielle du développement. Chaque étape du projet de l'analyse des besoins jusqu'à la livraison finale est planifiée à l'avance et exécutée de manière rigide, sans retour en arrière. Ce modèle convient particulièrement aux projets dont les exigences sont clairement définies dès le départ et peu susceptibles d'évoluer.

#### 1.4.1.2 Méthodologie Agiles

La méthodologie agile est une approche de gestion de projet qui repose sur l'adaptabilité, l'interaction continue avec le client, et la livraison incrémentale de fonctionnalités. Contrairement aux méthodes traditionnelles, l'agilité ne cherche pas à tout planifier dès le départ, mais à avancer par petites étapes appelées sprints, généralement d'une durée de 1 à 4 semaines. Cette méthode favorise la collaboration entre les membres de l'équipe et les parties prenantes, l'amélioration continue, ainsi que la réactivité face aux changements.

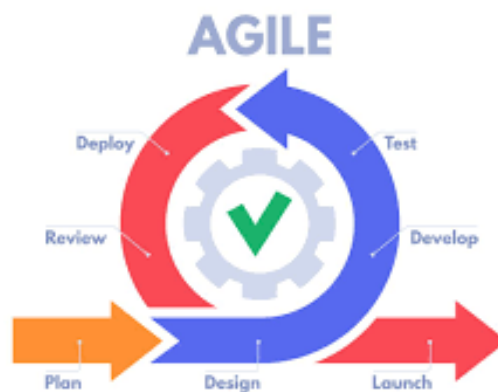


FIGURE 6. *Méthode Agile*

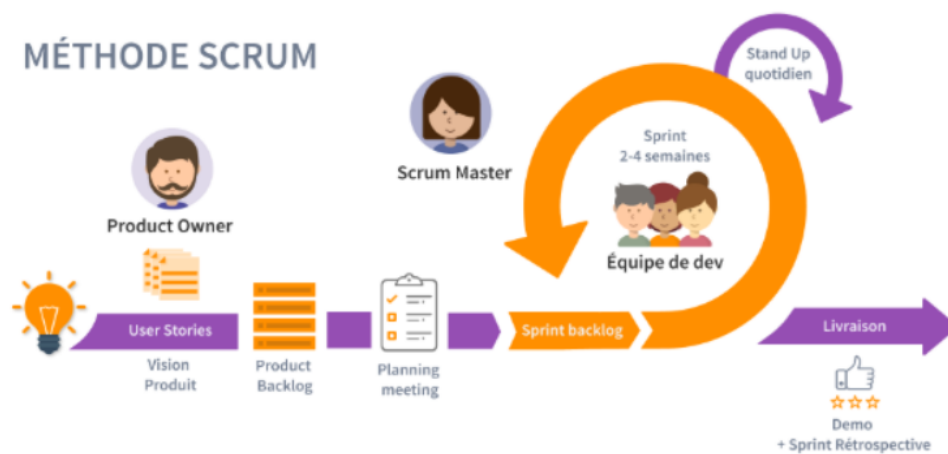
## **1.4.2 Le framework agile adopté SCRUM**

### **1.4.2.1 Pourquoi SCRUM ?**

Afin de mener notre projet à bien dans un cadre structuré et flexible, nous avons opté pour le framework agile Scrum. Ce choix repose sur sa capacité à structurer le travail tout en offrant une grande souplesse face aux évolutions du projet. La méthode consiste à avancer par étapes bien définies, en fixant des objectifs clairs à atteindre dans des délais courts. Elle encourage une communication régulière au sein de l'équipe, ainsi qu'une implication active de chaque membre dans le suivi et l'ajustement des tâches. Grâce à cette dynamique, nous avons pu maintenir un bon rythme de travail, tout en restant réactifs aux éventuels changements ou besoins spécifiques apparus en cours de réalisation.

### **1.4.2.2 Définition de SCRUM**

Scrum est un cadre de travail agile (framework) de gestion de projet utilisée pour développer, livrer et maintenir des produits complexes. Elle repose sur les principes de collaboration, de responsabilité partagée et de progression itérative. Le processus Scrum divise le travail en cycles courts, généralement de deux à quatre semaines, au cours desquels l'équipe se concentre sur la réalisation d'un ensemble précis de fonctionnalités destinées aux utilisateurs. Tout au long du projet, la communication entre les membres de l'équipe joue un rôle central. Elle permet d'assurer une planification efficace, d'ajuster les priorités si nécessaire, et de garantir à la fois la productivité de l'équipe et la qualité du produit livré.

FIGURE 7. *Processus du Scrum*

### 1.4.2.3 Les principes fondamentaux de Scrum

Scrum repose sur un ensemble de principes qui guident son fonctionnement et assurent son efficacité dans la gestion de projets complexes. Ces principes sont les suivants :

- **Transparence** : Toutes les informations importantes liées au projet doivent être visibles et comprises par tous les membres de l'équipe. Cela permet une meilleure prise de décision et une compréhension commune de l'état d'avancement du travail.
- **Inspection** : L'équipe doit régulièrement examiner l'avancement du projet, les obstacles rencontrés et les résultats obtenus. Ces moments d'inspection, souvent organisés à travers des réunions planifiées, permettent d'identifier rapidement les écarts par rapport aux objectifs.
- **Adaptation** : En fonction des résultats des inspections, l'équipe ajuste son travail, ses priorités ou sa méthode. Cette capacité à s'adapter en temps réel est essentielle pour faire face aux changements et optimiser la valeur du produit.
- **Engagement** : Chaque membre de l'équipe s'engage activement dans la réussite du projet. L'implication individuelle et collective est essentielle pour respecter les objectifs fixés et garantir la qualité des livrables.
- **Focus** : Scrum met l'accent sur la livraison rapide de fonctionnalités à forte valeur ajoutée pour les utilisateurs. Chaque décision ou tâche est orientée vers la création d'un produit

utile, fonctionnel et satisfaisant pour le client.

#### **1.4.2.4 Les rôles de Scrum**

Scrum définit trois rôles principaux, chacun ayant des responsabilités précises et complémentaires. Ces rôles assurent une organisation efficace, une communication fluide et une répartition claire des tâches au sein de l'équipe.

- Le Product Owner (PO) : il définit les besoins, fixe les priorités et s'assure que l'équipe travaille sur les fonctionnalités les plus importantes.
- Le Scrum Master : il facilite le travail de l'équipe, veille au bon respect de la méthode Scrum et aide à résoudre les blocages.
- L'équipe de développement : elle réalise le travail technique, conçoit et développe les fonctionnalités de manière autonome et collaborative.

#### **1.4.2.5 Les artefacts de Scrum**

Scrum utilise trois artefacts principaux pour organiser et suivre le travail de l'équipe :

- Le Product Backlog : c'est la liste de toutes les fonctionnalités, améliorations ou corrections à réaliser. Il est géré et priorisé par le Product Owner.
- Le Sprint Backlog : c'est la sélection des tâches à réaliser pendant une étape de travail. Il est préparé en début de chaque cycle de développement.
- L'Incrément : c'est le résultat concret du travail effectué, une version fonctionnelle et testée du produit, prête à être livrée ou présentée.

#### **1.4.2.6 Les événements de Scrum**

Scrum comprend plusieurs événements réguliers qui rythment le travail de l'équipe et assurent un bon suivi du projet :

- Sprint Planning : au début de chaque cycle, l'équipe décide des tâches à réaliser et se fixe des objectifs clairs.

- Daily Scrum : chaque jour, l'équipe se réunit brièvement pour faire le point sur l'avancement, les blocages et les prochaines étapes.
- Sprint Review : à la fin de chaque période de travail, l'équipe présente ce qui a été réalisé et recueille les retours.
- Sprint Retrospective : l'équipe analyse ce qui a bien fonctionné ou non, afin d'améliorer son organisation et ses méthodes de travail.

### 1.4.3 Pilotage du projet avec SCRUM

#### 1.4.3.1 Identification de l'équipe de SCRUM

L'équipe scrum joue un rôle très important dans le choix de la meilleure manière d'achever leur travail. Dans cette section, nous présentons les différents acteurs impliqués dans l'avancement du projet à différentes étapes et leurs rôles associés. Dans notre projet, la distribution des rôles aux différents participants est établie comme suit :

**TABLE 2.** Répartition des rôles dans l'équipe Scrum

Product Owner	Scrum Master	Développeurs
M.	Mme Nour Derouich	Aya Majerdi, Tayssir Zayet

#### 1.4.3.2 Backlog du produit de l'application

Le Product Backlog est l'un des artefacts essentiels de Scrum. Il regroupe l'ensemble des besoins du produit, exprimés du point de vue du client. Il s'agit d'une liste ordonnée selon les priorités, décrivant l'état futur souhaité du produit, et servant de référence pour l'équipe Scrum. Chaque élément du backlog correspond à une user story (histoire utilisateur). Pour chaque item, les informations suivantes sont précisées :

- **ID** : L'identifiant des modules.
- **Module** : Fonctionnalité ou domaine auquel l'histoire est rattachée.



- **User Story** : Description de l'interaction entre l'utilisateur et le système, formulée de manière claire et compréhensible.
- **Priorité** : Niveau d'importance assigné à chaque histoire, déterminant son ordre de traitement.

**TABLE 3.** *Product backlog*

ID	Module	User Story	Priorité	Complexité
US1	Infrastructure & Authentification	En tant que développeur, je veux Configurer l'environnement Django avec PostgreSQL	Haute	Moyenne
		En tant que développeur, je veux Configurer le frontend Vite.js	Haute	Moyenne
		En tant que développeur, je veux Implémenter l'authentification JWT	Haute	Moyenne
		En tant qu'administrateur, je veux Créer et gérer des comptes utilisateurs	Haute	Moyenne
		En tant qu'utilisateur, je veux me Connecter et déconnecter de la plateforme	Haute	Faible
		En tant qu'utilisateur, je veux Réinitialiser mon mot de passe	Moyenne	Moyenne
		En tant qu'utilisateur, je veux mettre à jour mes informations de profil	Moyenne	Faible
		En tant qu'administrateur, je veux gérer les permissions basées sur les rôles	Haute	Élevée
		En tant que développeur, je veux Configurer Redis pour la mise en cache	Moyenne	Moyenne
US2	Modèles de Données & API	En tant que développeur, je veux créer des modèles pour les installations solaires	Haute	Moyenne
		En tant que développeur, je veux créer des modèles pour les équipements	Haute	Moyenne
		En tant que développeur, je veux créer des modèles pour les données de production/consommation	Haute	Moyenne
		En tant que développeur, je veux créer des modèles pour les alarmes et notifications	Haute	Moyenne
		En tant que développeur, je veux développer des endpoints API pour la gestion des utilisateurs	Haute	Moyenne
		En tant que développeur, je veux développer des endpoints API pour les installations	Haute	Moyenne
		En tant que développeur, je veux développer des endpoints API pour les données de production/consommation	Haute	Moyenne

ID	Module	User Story	Priorité	Complexité
		En tant que développeur, je veux développer des endpoints API pour les alarmes	Haute	Moyenne
		En tant que développeur, je veux implémenter des WebSockets pour les mises à jour en temps réel	Haute	Élevée
US3	Interface Administrateur	En tant qu'administrateur, je veux voir une Vue d'ensemble des installations sur mon tableau de bord	Haute	Élevée
		En tant qu'administrateur, je veux voir le nombre d'alarmes actives avec criticité	Haute	Moyenne
		En tant qu'administrateur, je veux un tableau récapitulatif des installations	Haute	Moyenne
		En tant qu'administrateur, je veux ajout des nouvelles centrales	Haute	Moyenne
		En tant qu'administrateur, je veux voir importer des données historiques	Haute	Élevée
		En tant qu'administrateur, je veux voir une Vue détaillée par installation	Haute	Moyenne
		En tant qu'administrateur, je veux accéder Tableau de bord technique	Haute	Élevée
		En tant qu'administrateur, je veux Filtrer les installations	Haute	Moyenne
		En tant qu'administrateur, je veux être notifié en temps réel	Haute	Élevée
US4	Système d'Alarme & Carte	En tant qu'administrateur, je veux une base de données des codes d'alarme par marque d'onduleur	Haute	Moyenne
		En tant qu'administrateur, je veux recevoir des notifications d'alarmes	Haute	Moyenne
		En tant qu'administrateur, je veux une interface de suivi des alarmes	Haute	Moyenne
		En tant qu'administrateur, je veux Visualiser géographiquement les installations sur une carte	Haute	Élevée
		En tant qu'administrateur, je veux voir la production journalière par installation sur la carte	Moyenne	Moyenne
		En tant qu'administrateur, je veux voir la production mensuelle sur la carte	Moyenne	Moyenne
		En tant qu'administrateur, je veux voir les revenus générés sur la carte	Moyenne	Moyenne
		En tant qu'administrateur, je veux voir l'état des alarmes sur la carte	Haute	Moyenne
		En tant qu'administrateur, je veux filtrer les données affichées sur la carte	Moyenne	Moyenne

ID	Module	User Story	Priorité	Complexité
US5	Interventions & Rapports	En tant qu'administrateur, je veux créer des fiches d'intervention	Haute	Moyenne
		En tant qu'administrateur, je veux planifier des entretiens	Haute	Moyenne
		En tant qu'administrateur, je veux suivre les réclamations clients	Haute	Moyenne
		En tant qu'administrateur, je veux générer des rapports de production (mensuelle, annuelle, totale)	Haute	Élevée
		En tant qu'administrateur, je veux générer des rapports de consommation (mensuelle, annuelle, totale)	Haute	Élevée
		En tant qu'administrateur, je veux générer des rapports d'historique des alarmes	Haute	Moyenne
		En tant qu'administrateur, je veux personnaliser les rapports selon mes besoins	Haute	Élevée
		En tant qu'administrateur, je veux exporter les rapports (PDF, Excel)	Moyenne	Moyenne
		En tant qu'administrateur, je veux programmer l'envoi automatique des rapports	Moyenne	Moyenne
US6	Interface Client	En tant que client, je veux une vue générale de mon installation client	Haute	Moyenne
		En tant que client, je veux voir l'état de fonctionnement de mon installation	Haute	Moyenne
		En tant que client, je veux voir les photos de mon installation	Haute	Faible
		En tant que client, je veux voir ma production journalière en temps réel	Haute	Élevée
		En tant que client, je veux voir ma production mensuelle	Haute	Moyenne
		En tant que client, je veux voir ma production totale	Haute	Moyenne
		En tant que client, je veux voir ma consommation totale	Haute	Moyenne
		En tant que client, je veux voir l'état de mes équipements	Haute	Moyenne
		En tant que client, je veux être alerté en cas d'alarmes sur mon installation	Haute	Moyenne
US7	Gestion Administrative Client	En tant que client, je veux voir des graphiques de ma production/consommation	Haute	Élevée
		En tant que client, je veux générer des rapports pour mon installation	Haute	Moyenne
		En tant que client, je veux utiliser un système d'auto-relève	Haute	Moyenne
		En tant que client, je veux suivre mes réclamations	Haute	Moyenne
		En tant que client, je veux gérer les paramètres de mon compte	Haute	Faible
		En tant que client, je veux comparer ma production	Moyenne	Moyenne

ID	Module	User Story	Priorité	Complexité
		En tant que client, je veux estimer mes économies d'énergie	Haute	Élevée
		En tant que client, je veux recevoir des conseils d'optimisation	Basse	Moyenne
		En tant que client, je veux être notifié d'interventions	Moyenne	Faible
US8	Performance & Sécurité	En tant que développeur, je veux optimiser les temps de chargement (<3 secondes)	Haute	Élevée
		En tant que développeur, je veux assurer le support simultané de 100+ utilisateurs	Haute	Élevée
		En tant que développeur, je veux implémenter des mises à jour en temps réel	Haute	Élevée
		En tant que développeur, je veux sécuriser l'authentification	Haute	Moyenne
		En tant que développeur, je veux chiffrer les données sensibles	Haute	Moyenne
		En tant que développeur, je veux protéger contre les injections SQL	Haute	Moyenne
		En tant que développeur, je veux configurer des sauvegardes quotidiennes	Haute	Moyenne
		En tant que développeur, je veux assurer la compatibilité multi-navigateurs	Haute	Moyenne
		En tant que développeur, je veux implémenter un design responsive	Haute	Élevée

### 1.4.3.3 Planification des sprints

La planification de sprint est un pilier du développement agile. Elle consiste à définir, ordonnancer et estimer les tâches à accomplir pour un sprint donné. Notre approche s'articule autour de quatre étapes clés, imbriquées dans le processus de développement, pour garantir une exécution optimale. Objectifs de la planification de sprint :

- Clarté des user stories : Définir précisément les fonctionnalités attendues pour chaque user story.
- Évaluation de la faisabilité : S'assurer que les objectifs du sprint sont réalistes et atteignables dans le temps imparti.
- Suivi et communication transparente : Mettre en place des outils et des processus pour

suivre l'avancement du projet et communiquer régulièrement avec les parties prenantes (tableaux de bord, réunions d'avancement...).

- Calendrier précis : Déterminer la durée du sprint, généralement entre une et quatre semaines.

Les différentes étapes du projet sont structurées et organisées dans un tableau de planification, comme illustré dans le tableau ci-dessous.

**TABLE 4.** *Planification des sprints*

Story ID	Nom du sprint
US1	Infrastructure et Authentification
US2	Modèles de Données et APIs
US3	Tableau de Bord Administrateur
US4 + US5	Alarmes et Carte Interactive
US5 + US6	Rapports et Interface Client
US6 + US7 + US8	Finalisations et Optimisations

#### 1.4.4 Formalisme adopté

Le langage de modélisation UML a été adopté dans ce projet afin de garantir une conception claire, structurée et efficace. Sa standardisation visuelle, sa documentation synthétique et sa flexibilité indépendante des langages de programmation en font un outil particulièrement adapté

à la collaboration entre les différents intervenants. De plus, les simulations qu'il permet en amont réduisent considérablement les risques d'erreurs et les coûts de modifications.

Ce rapport s'appuie ainsi sur trois types de diagrammes UML essentiels : le diagramme de cas d'utilisation, le diagramme de classes et le diagramme de séquences. Ces outils permettent de représenter les fonctionnalités du système, les interactions entre utilisateurs et objets, ainsi que la dynamique temporelle des échanges.

En définitive, le choix d'UML constitue une décision stratégique, contribuant à une communication fluide, une documentation claire, une plus grande souplesse de conception et une réduction notable des risques tout au long du cycle de développement.

## **1.5 Conclusion**

En résumé, ce chapitre a permis de définir le cadre général du projet, en présentant l'organisme d'accueil, les problématiques identifiées, les solutions existantes ainsi que la méthodologie retenue. Il pose ainsi les bases nécessaires à la bonne compréhension des enjeux du projet et des orientations méthodologiques à suivre. Le prochain chapitre sera consacré à la préparation des éléments essentiels à la mise en œuvre concrète de notre solution.

# **Chapitre 2**

## **Analyse et spécification des besoins**

## ***2.1 Introduction***

Avant de commencer le développement de notre application, il est crucial de passer par la phase de spécification des besoins pour une meilleure organisation et identification des tâches. Ce chapitre se concentre sur les exigences fonctionnelles et non fonctionnelles, les intervenants du projet, et l'architecture technique de l'application. Nous y présenterons également un diagramme global des cas d'utilisation et un aperçu de l'environnement de développement et des technologies utilisées.

## ***2.2 Spécification des besoins***

### **2.2.1 Identification des acteurs**

Les principaux acteurs de notre plateforme sont l'administrateur, le client, le technicien et l'installateur. Chacun joue un rôle essentiel dans la gestion et le suivi des installations photovoltaïques. Le tableau suivant présente une description détaillée de chaque acteur.



**TABLE 5.** *Identification des acteurs*

<b>Acteur</b>	<b>Description</b>
<b>Administrateur</b>	Responsable de la gestion globale de la plateforme. Il peut ajouter des utilisateurs (installateurs, techniciens, clients), superviser les installations, consulter les statistiques générales, gérer les alarmes critiques et configurer les paramètres du système.
<b>Client</b>	Propriétaire d'une ou plusieurs installations photovoltaïques. Il peut consulter la production, la consommation, les alarmes, les interventions effectuées, et recevoir des notifications en temps réel.
<b>Technicien</b>	Chargé de l'entretien et des interventions techniques sur les installations. Il peut consulter les alarmes actives, effectuer des actions correctives, et mettre à jour le statut des interventions.
<b>Installateur</b>	Responsable de l'installation initiale des systèmes photovoltaïques. Il peut créer de nouvelles installations, associer des techniciens et des clients, et assurer le suivi de l'installation jusqu'à sa validation finale.

## 2.2.2 Besoins fonctionnels

### a) Administrateur

- Authentification sécurisée
- Gestion des utilisateurs (client, installateur, technicien)
- Gestion des installation
- Visualisation des statistiques globales
- Gestion des codes alarmes
- Consultation des alarmes
- Gestion des rapports

### b) Client

- Authentification sécurisée
- Consultation de l'état de fonctionnement de l'installation
- Consultation des données de production et de consommation
- Être notifié en cas d'alarme
- Visualisation de l'historique des alarmes et interventions

c) **Installateur**

- Authentification sécurisée
- Gestion des installation
- Consultation des alarmes
- Visualisation des statistiques globales

d) **Technicien**

- Authentification sécurisée
- Gestion des rapports technique
- Consultation de liste des interventions à effectuer
- Visualisation des Alarmes

### 2.2.3 Besoins non fonctionnels

Les besoins non fonctionnels sont indispensables et permettent l'amélioration de la qualité logicielle de notre système. Ils agissent comme des contraintes sur les solutions, mais leur prise en considération fait éviter plusieurs incohérences dans le système. Celui-ci doit répondre aux exigences suivantes :

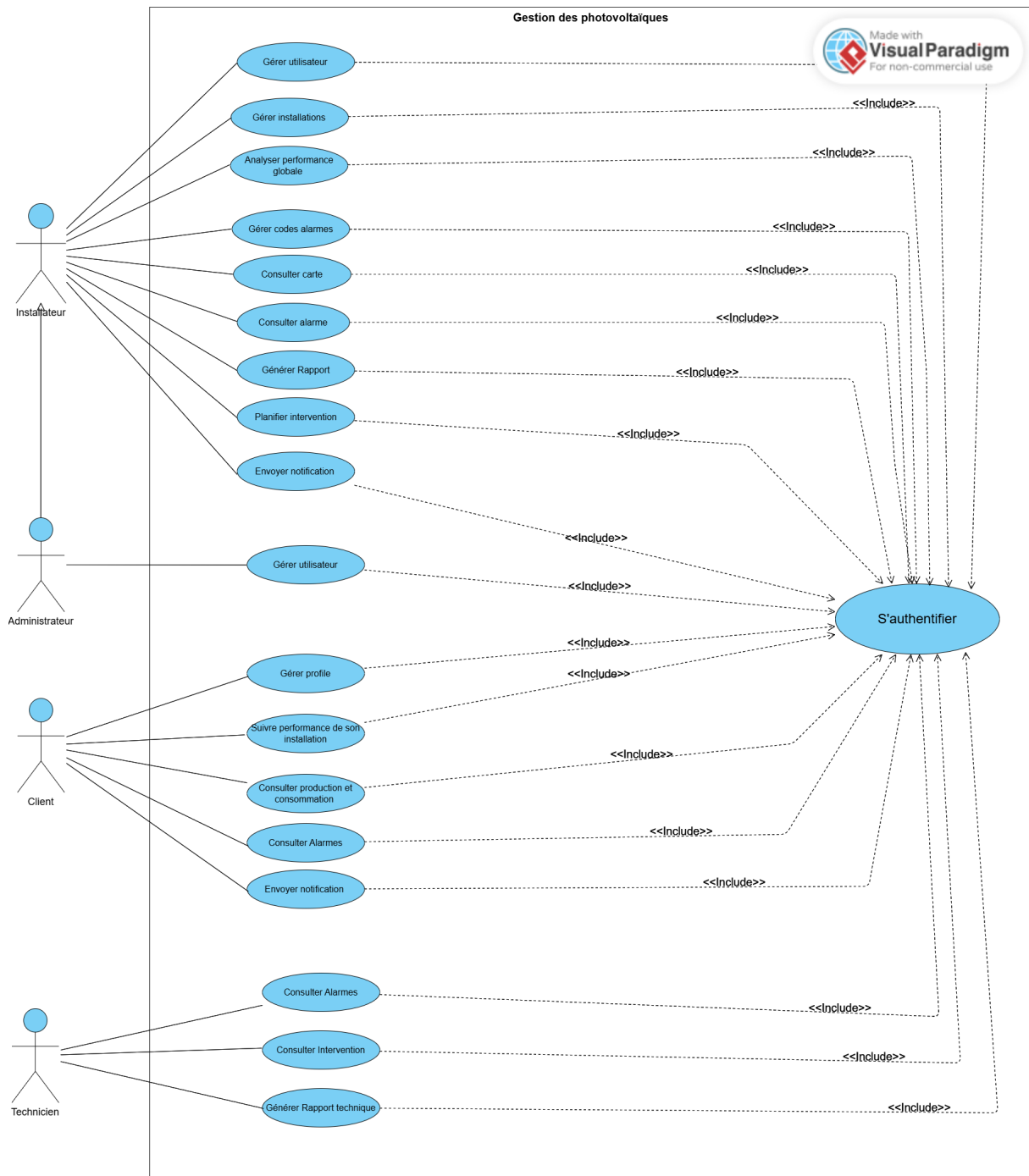
- ✓ **Sécurité** : Seuls les utilisateurs enregistrés auront accès aux services de la plateforme après la vérification de leurs comptes à l'aide de codes envoyés par email. Pour protéger la vie privée des utilisateurs, l'authentification par email et mot de passe est également requise afin de garantir la confidentialité.
- ✓ **Responsive Design** : Un design responsive est mis en œuvre pour permettre à l'application de s'adapter parfaitement à toutes les résolutions d'écran, assurant ainsi une

expérience utilisateur cohérente.

- ✓ **Interface conviviale** : L'interface est conçue avec une mise en page graphique simple et intuitive, garantissant une navigation facile et agréable pour les utilisateurs.
- ✓ **Évolutivité et maintenance** : L'application est conçue pour permettre une évolution flexible et une maintenance aisée, facilitant l'ajout, la suppression ou la mise à jour de différents modules.

## ***2.3 Diagramme de cas d'utilisation global***

une fois nous avons analysé en détail les besoins fonctionnels, nous les avons intégrés dans un diagramme de cas d'utilisation global, comme illustré ci-dessous.

**FIGURE 8.** *Diagramme de cas d'utilisation globale*

## 2.4 Architecture globale

### 2.4.1 Architecture Logiciel

#### 2.4.1.1 Architecture MVT

L'architecture MVT (Model – View – Template) est un pattern de conception logicielle utilisé par le framework Django. Elle permet de séparer les responsabilités entre la gestion des données, la logique métier, et l'interface utilisateur, facilitant ainsi la maintenabilité, l'évolutivité et la clarté du code.

- **Modèle (Model)** : Représente la structure des données et interagit directement avec la base de données. Il définit les champs, les relations et les contraintes des objets manipulés dans l'application.
- **Vue (View)** : Contient la logique métier. Elle traite les requêtes HTTP, interagit avec les modèles si nécessaire, et retourne une réponse. Elle agit comme un intermédiaire entre le modèle et l'interface utilisateur.
- **Template** : Correspond à la couche de présentation. Il s'agit de fichiers HTML enrichis avec le langage de template Django, permettant d'afficher dynamiquement les données fournies par la vue.

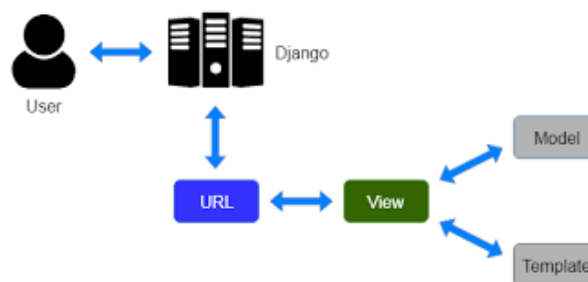


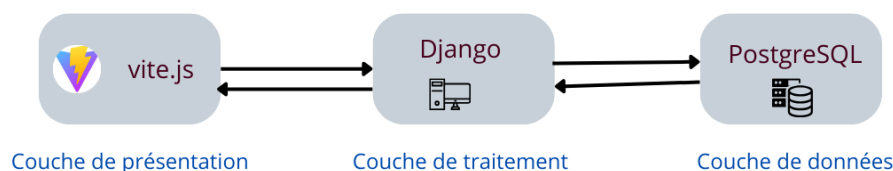
FIGURE 9. Architecture MVT

#### 2.4.1.2 Application du l'architecture MVT dans notre cas

Notre application suit le modèle MVT (Model – View – Template) fourni par le framework Django, en l'adaptant à une architecture moderne où le frontend est séparé via Vite.js. Cette

séparation permet une meilleure organisation du code et une évolutivité accrue. Voici les différentes couches :

- **Couche de données** : Elle est assurée par la base de données PostgreSQL. Les modèles Django définissent les structures des tables, assurent la cohérence des données et permettent leur manipulation via l'ORM de Django.
- **Couche de présentation** : Cette couche est assurée par Vite.js (React), qui constitue l'interface utilisateur (UI) de notre application. Grâce aux composants interactifs et à une gestion dynamique des états, elle offre une expérience fluide, moderne et réactive à l'utilisateur.
- **Couche de traitement** : Django joue le rôle de couche de traitement. Il gère la logique métier, l'accès aux données et la communication via des API REST exposées par Django REST Framework. Cette couche traite les requêtes du frontend, applique les règles du métier et renvoie les réponses appropriées.



**FIGURE 10.** *Architecture logiciel*

## 2.4.2 Architecture système

Une "architecture système" est un terme couramment employé pour décrire l'ensemble des composants matériels et logiciels d'un système, ainsi leur interaction. voici donc notre illustration de l'architecture système :

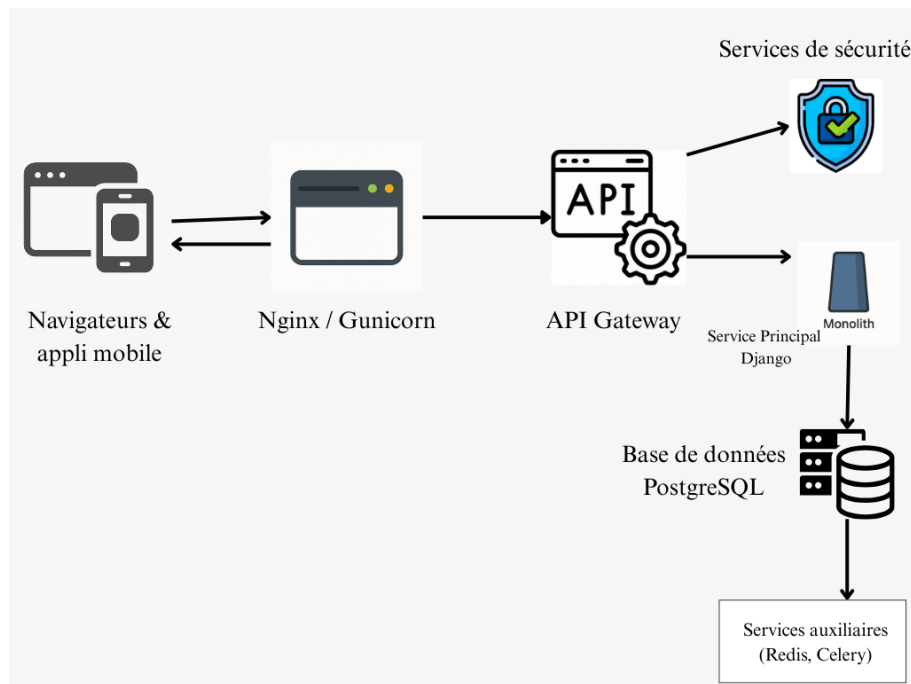


FIGURE 11. Architecture système

## 2.5 Environnement du travail

### 2.5.1 Environnement matériel

les caractéristiques des machines utilisées pour la réalisation de notre projet sont évoquées dans le tableau suivant :

Composant	Lenovo IdeaPad Gaming 3	Dell
Marque	Lenovo IdeaPad Gaming 3	DELL VOSTRO
Processeur	11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz 3.19 GHz	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
RAM	8,00 Go	16.0 Go
Système d'exploitation	Windows 11 Famille	Windows 11 Famille

TABLE 6. Outils matériels

## 2.5.2 Environnement logicielle

### 2.5.2.1 Environnement de développement

- **Visual Studio Code** : Est un éditeur de code source léger, multi-plateforme et open-source, développé par Microsoft. Il est largement utilisé par la communauté des développeurs pour sa flexibilité, sa richesse fonctionnelle et sa personnalisation avancée.



**FIGURE 12.** *Visual Studio Code*

- **Docker desktop** : Docker Desktop est un outil de conteneurisation permettant de créer, exécuter et gérer des environnements logiciels isolés appelés conteneurs. Dans notre projet, il a joué un rôle essentiel pour assurer un environnement de développement stable, reproductible et portable.



**FIGURE 13.** *Docker*

### 2.5.2.2 Outils d'infrastructure et de données

- **PostgreSQL** : PostgreSQL est un système de gestion de base de données relationnelle open-source, reconnu pour sa fiabilité, ses performances et son support des requêtes complexes.





FIGURE 14. *PostgreSQL*

- **Redis** : Redis est une base de données en mémoire clé-valeur, utilisée ici comme broker pour Celery. Elle permet une gestion rapide et efficace des files de tâches asynchrones.



FIGURE 15. *Redis*

### 2.5.2.3 Outils de développement

- **Vite.js** : Vite.js est un bundler moderne et ultra-rapide pour les projets frontend. Il permet un rechargement instantané lors du développement et une compilation optimisée pour la production.



FIGURE 16. *Vite.js*

- **React** : React est une bibliothèque JavaScript développée par Meta pour construire des interfaces utilisateur dynamiques et réactives en composant des éléments visuels réutilisables.



**FIGURE 17.** *React.js*

- **Django** : Django est un framework web Python puissant et sécurisé, basé sur l'architecture MVT. Il permet un développement rapide du backend et expose des API robustes via Django REST Framework.



**FIGURE 18.** *Django*

- **Celery** : Celery est une bibliothèque Python permettant d'exécuter des tâches de manière asynchrone et distribuée. Elle est utilisée avec Redis pour le traitement des actions en arrière-plan (exports, envois de mails, etc.).



**FIGURE 19.** *Celery*

#### 2.5.2.4 Outils de collaboration et documentation

- **Microsoft Teams** : Teams est une plateforme de communication collaborative utilisée pour les réunions, le chat d'équipe, et la coordination en temps réel pendant tout le cycle de développement.



**FIGURE 20.** *Teams*

- **ClickUp** : ClickUp est un outil de gestion de projet tout-en-un, utilisé pour organiser les tâches, suivre l'avancement des sprints, et gérer la productivité de l'équipe.



**FIGURE 21.** *ClickUp*

- **Notion** : Notion est une plateforme collaborative qui centralise la documentation, les notes techniques, les plans de projet, et facilite la structuration de l'information.



**FIGURE 22.** *Notion*

- **Github** : GitHub est une plateforme de gestion de code source basée sur Git. Elle facilite le travail collaboratif, le versionnage, et l'intégration continue à travers des dépôts privés ou publics.



**FIGURE 23.** *GitHub*

# Conclusion

And a very interesting conclusion here.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Appendix

An appedix if you need it.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[title=Webography,type=online]

[title=Bibliography,notype=online]



# Hello World!

Here is my test PDF document.



