

UNIVERSITE MOHAMMED V DE RABAT

FACULTÉ DES SCIENCES



Université Mohammed V
Faculté des Sciences
Rabat

PROJET DE FIN D'ETUDES

Pour l'obtention du diplôme de Licence d'Etudes fondamentales

Option : Sciences Mathématiques et Informatique

**Mise en place d'une solution de
data visualization basée sur un
modèle de machine learning**

Réalisé par :

- OUHMAD Inass
- ACHIR Nihad

Encadré par :

Mr. KHALIFI Hamid

Soutenu le 07 Juin 2023, Devant les jurys composé de :

Mme. Lamyae SARDI : - Présidente

Mr. Faissal OUARDI : - Examinateur

Promotion : 2022/2023

Remerciements

Nous remercions le bon Dieu qui nous a donné le courage et la volonté d'élaborer ce travail.

Au terme de ce travail, nous tenons à exprimer notre profonde gratitude et nos sincères Remerciements à notre encadrant Monsieur KHALIFI Hamid, pour tout le temps qu'il nous a consacré, ses conseils précieux, et pour la qualité de son suivi durant toute la période de notre projet.

Nos remerciements vont aussi aux membres du jury pour avoir accepté de juger notre travail.

À tous ceux qui me sont chers, à vous tous

Merci.

Résumé

L'objectif de ce projet consiste de mise en place d'une solution de Data Visualization d'un système d'irrigation basé sur un modèle de machine Learning.

Le projet est réalisé en python à l'aide de Framework Django afin de pouvoir visualiser les données et les analyser à l'aide du modèle de machine Learning.

La première partie consiste à la création de l'interface utilisateur qui contient : un espace d'identification, un tableau de bord avec la visualisation et reporting des données, un aperçu historique des données enregistrées dans la base de données.

La deuxième partie consiste à prédire les données statistiques qui sont enregistrées dans la base de données chaque heure à base de réseaux de neurones récurrents LSTM Série chronologique à court terme et à long terme.

Mots clés : Django, Python, MYSQL, Java Script, Data visualization, Reporting, Redis, LSTM, Série chronologique.

Liste des Abréviations

HTML	<i>Hypertext Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
JS	<i>Java Script</i>
DB	<i>Data base</i>
SQL	<i>Structured Query Language</i>
SGBD	<i>Système de Gestion de Base de Données</i>
MVC	<i>Model-View-Controller</i>
UML	<i>Unified Modeling Language</i>
CRUD	<i>Create, Read, Update, Delete</i>
CSRF	<i>Cross-Site Request Forgery</i>
ML	<i>Machine Learning</i>
LSTM	<i>Long Short-Term Memory</i>
RAD	<i>Rapid Application Development</i>

Table des matières

Remerciements	1
Résumé	2
Liste des Abréviations	3
Liste des figures	7
Liste des tableaux	8
Introduction générale	9
1 Contexte général du projet	10
1.1 Cadre général du projet	10
1.2 Management du projet	11
2 Spécification Fonctionnelle des besoins	13
2.1 Expression des besoins	13
2.1.1 Les services qu'il va rendre :	14
2.2 Schéma global	15
2.3 Exigences et contraintes	16
2.3.1 Description des fonctionnalités attendues	16

2.3.2	Répartition des fonctionnalités par profil utilisateur	17
3	Conception et Analyse	19
3.1	Outils de travail	19
3.2	Modélisation du système	20
3.2.1	Diagramme de cas d'utilisation	20
3.2.2	Diagramme de classe	25
3.2.3	Diagramme de séquence	26
3.3	Conception générale	28
4	Mise en oeuvre du projet	32
4.1	Technologies et outils de développement	32
4.2	Architecture technique	41
4.3	Présentation des interfaces Homme Machine	42
4.3.1	Accueil:	42
4.3.2	Inscription:	43
4.3.3	Connexion:	44
4.3.4	Regroupement:	44
4.3.5	Gestion des profils:	45
4.3.6	Gestion des Terrains:	46
4.3.7	Gestion des Statistiques:	48
4.3.8	Prédiction:	53
	Conclusion & Perspective	55
	Bibliographie	56

Liste des figures

1.1	Méthodologie RAD	12
2.1	Schéma global de système d'irrigation	15
3.1	Association entre back-end et front-end	19
3.2	Diagramme de cas d'utilisation	21
3.3	Diagramme de Classe	25
3.4	Diagramme de séquence	27
3.5	La Table «Utilisateur»	28
3.6	La Table «Propriétaire»	28
3.7	La Table «Adresse»	29
3.8	La Table «Terrain»	29
3.9	La Table «Statistique»	30
3.10	La Table «Admin»	30
3.11	La Table «Celerybeatperiodictasks»	31
3.12	La Table «Celerybeatperiodictask»	31
4.1	PYTHON	32
4.2	Framework DJANGO	33
4.3	MYSQL	33

4.4	MODEL VIEW CONTROLLER	34
4.5	ARCHITECTURE DJANGO CELERY	36
4.6	Javascript	37
4.7	DATA VISUALIZATION	37
4.8	CHART JS	38
4.9	BOOTSTRAP	38
4.10	MACHINE LEARNING	39
4.11	RNN-LSTM-SERIE CHRONOLOGIQUE	40
4.12	Architecture générale de système d'irrigation	41
4.13	Page d'accueil	42
4.14	Page d'inscription	43
4.15	Page de connexion	44
4.16	Page Regroupement des terrains	44
4.17	Page de profil	45
4.18	Page de modification de profil	45
4.19	Page de consultation des terrains	46
4.20	Ajout d'un nouveau terrain	47
4.21	Editer Terrain	48
4.22	Page de consultation des statistiques	48
4.23	Page de statistiques	49
4.24	Page de statistiques	50
4.25	Page ajouter statistique	51
4.26	Page de modification d'une statistique	52
4.27	Page de suppression d'une statistique	53
4.28	Page de prédiction	54

Liste des tableaux

3.1	Raffinement de cas d'utilisation "S'inscrire"	22
3.2	Raffinement de cas d'utilisation "S'authentifier"	22
3.3	Raffinement de cas d'utilisation "consulter profile"	23
3.4	Raffinement de cas d'utilisation "consulter les terrains des propriétaires"	23
3.5	Raffinement de cas d'utilisation "consulter les statistiques d'un terrain"	24

Introduction générale

Les entreprises ont aujourd’hui accès à de vastes quantités de données générées à l’intérieur et à l’extérieur de leurs organisations, et la recherche montre que les humains réagissent mieux aux stimuli visuels qu’à tout autre type de stimulus. Le cerveau humain traite les informations visuelles 60 000 fois plus rapidement que le texte. En fait, les données visuelles représentent 90 % des informations transmises au cerveau. Par conséquent, il faut imaginer des techniques pour traiter correctement ce volume d’information. Parmi les solutions, la data visualization facilite à transformer des données complexes en informations compréhensibles basées sur des graphiques, des tableaux, des cartes, et d’autres représentations visuelles.

Pour rendre la visualisation des données plus fiable et faire des prédictions au niveau des données, des techniques d’apprentissage automatique sont utilisées, qui permettent d’utiliser les données en entrée et de les préparer à l’aide des algorithmes pour gérer les procédures automatiquement sans intervention humaine.

Dans ce sens, on va réaliser une application web d’un système d’irrigation qui permet aux utilisateurs d’avoir une représentation visuelle de l’ensemble des données représenté sous forme d’un tableau de bord et avec la configuration de modèle prédictif à base de réseaux de neurones récurrents , et bien donc ce système permet de faire l’irrigation automatique sans l’intervention humaine avec une réduction de la consommation d’eau et la prédiction des données à l’aide de LSTM et Série chronologique.

Notre plan de projet est comme suit. Tout d’abord on va attaquer le fonctionnement de système en générale après on se concentre sur l’analyse et la conception des diagrammes et vers la fin on va traiter la phase de réalisation qui affiche les outils utilisés et la présentations des interfaces.

Chapter 1

Contexte général du projet

Ce premier chapitre a pour but de mettre une vue globale sur le cadre général du projet.

1.1 Cadre général du projet

L'arrosage manuel peut être fastidieux et consomme beaucoup de temps. Pour faciliter cette tâche, on a développé une application web d'irrigation automatique pour contrôler efficacement et économiquement l'arrosage. Une application web d'irrigation automatique utilise des capteurs pour collecter des données environnementales telles que l'humidité du sol, la température, l'humidité, la force de vent, le rayonnement solaire et les utilise pour automatiser le système. Cela garantit que le terrain reçoit la quantité optimale d'eau sans gaspiller les ressources naturelles.

Le cadre de développement de cette application web peut inclure :

1-Analyse des besoins:

Pour identifier les besoins spécifiques de l'application d'irrigation automatique, on considère les facteurs suivants :

–Zones d'irrigation : on identifie les différentes zones de jardin ou de terrain du propriétaire qui nécessitent une irrigation automatique. Chaque zone peut avoir des besoins différents en termes de quantité et de durée d'arrosage.

–Conditions climatiques locales : Les conditions climatiques locales de région de terrain pour concevoir un système d'irrigation automatique efficace. Les

conditions climatiques telles que la température, l'humidité, l'humidité de sol, la force de vent et le rayonnement solaire peuvent affecter les besoins d'eau.

2-Conception : La conception est le processus de création d'un plan détaillé de l'application qui inclut :

–la conception de l'interface utilisateur : Les différentes interfaces qui permettent à l'utilisateur de se connecter, ajouter des terrains avec leur propriétaires, ajouter statistiques.

–la conception de la base de données : La conception en basant sur le processus général de l'application.

3-Développement : Dans cette étape, on va utiliser des langages de programmation tels que JavaScript, Python avec les Framework Django et Bootstrap pour créer une interface utilisateur attrayante et facile à utiliser. En plus de l'algorithme de réseau de neurones récurrents pour la prédition En utilisant les dernières technologies, cette application web peut être conçue pour répondre à des besoins spécifiques et de piloter et superviser le système d'irrigation de manière intelligente tout en permettant l'économie d'eau.

1.2 Management du projet

Le développement d'une application web d'irrigation automatique nécessite une gestion de projet efficace pour s'assurer que le produit final répond aux besoins.

Comme objectif il permet à l'utilisateur d'avoir une représentation visuelle de l'ensemble de données clés pour qu'elles soient consultable de manière instantané avec un tableau de bord de data visualisation et reporting personnalisé configurable avec un modèle prédictif à base de réseaux de neurones récurrents.

Pour cela on va utiliser la méthodologie RAD qui est une stratégie de gestion de projet agile qui permet de réaliser un projet en le découplant en petites parties.

La méthode RAD structure le cycle de vie du projet en 5 phases (dont 3 systématisques) :

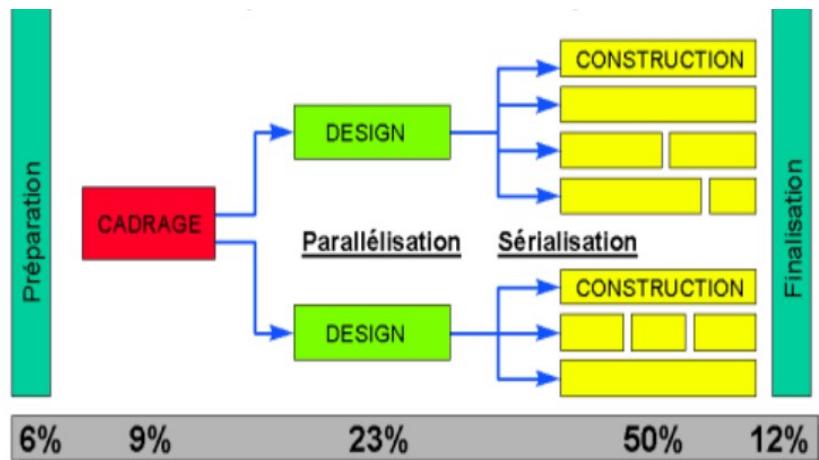


Figure 1.1: Méthodologie RAD

1-Initialisation: Cette phase permet de définir le périmètre général du projet, de structurer le travail par thèmes, de sélectionner les acteurs pertinents et d'amorcer une dynamique de projet.

Dans cette phase on a consacré notre temps à l'apprentissage et à la définition des aspects globale du projet. Cela nous a appris 6% du projet.

2-Cadrage: Le CADRAGE définit un espace d'objectifs, de solutions et de moyens.

Dans cette phase on a réalisé une architecture générale qui englobe le sujet de l'application. Cela nous a appris 9% du projet.

3-Design: Le DESIGN modélise la solution et valide sa cohérence systémique.

Dans cette phase on a choisi le design convenable à cette application. Cela nous a appris 23%.

4-Construction: Durant cette phase, le développement du système réel commence. On va utiliser les spécifications et les modèles créés lors de la phase d'initialisation pour coder les fonctionnalités du système. Des tests unitaires sont également effectués pour s'assurer de la qualité du code développé. Cela nous a appris 50% du projet.

5-Finalisation : La finalisation est réduite à un contrôle final de qualité en site pilote. Cela nous a appris 12% du projet.

Chapter 2

Spécification Fonctionnelle des besoins

Ce deuxième chapitre a pour but d'avoir une idée générale sur le fonctionnement global du système.

2.1 Expression des besoins

Il est aujourd’hui admis que jusqu’à 50% de l’eau d’irrigation est gaspillée en raison d’un excès d’arrosage causé par l’inefficacité des méthodes et systèmes d’irrigation traditionnels.

On va réaliser une application web qui permet à l’utilisateur d’accéder à un espace contenant l’ensemble des informations significatives sous forme d’un tableau de bord qui va répondre aux besoins :

Besoin en eau : Le système d’irrigation doit être dimensionné pour répondre aux besoins en eau. Cela dépendra de facteurs tels que le type de culture, les conditions climatiques.

Distribution uniforme de l’eau : Le système doit être capable de fournir une répartition uniforme de l’eau sur la zone irriguée. Cela garantit que tous les terrains reçoivent la quantité d’eau nécessaire.

Programmation flexible : Il devrait être possible de définir des horaires d’irrigation adaptés aux conditions météorologiques, à la saison et aux exigences spécifiques des cultures.

Contrôle des mesures telles que Température, humidité, humidité de sol, force de vent, rayonnement solaire et la consommation d'eau : Cela permet de fournir la quantité d'eau nécessaire en fonction des besoins réels des terrains, évitant ainsi le gaspillage d'eau et les problèmes de sous-irrigation ou de sur-irrigation.

Conservation de l'eau : L'efficacité de l'utilisation de l'eau est un facteur important. Les systèmes d'irrigation modernes intègrent souvent des technologies telles que l'irrigation goutte à goutte ou l'irrigation par aspersion qui réduisent les pertes d'eau par évaporation ou ruissellement.

Contrôle automatisé : Les systèmes d'irrigation peuvent être équipés de contrôles automatisés pour faciliter leur fonctionnement. Cela peut inclure des minuteries, des capteurs de pluie pour suspendre l'irrigation.

En résumé, un système d'irrigation efficace doit répondre aux besoins spécifiques en eau des terrains, assurer une distribution uniforme de l'eau, offrir une programmation flexible, contrôler les mesures climatiques, préserver l'eau, être facilement contrôlable et entretenir pour garantir sa durabilité.

2.1.1 Les services qu'il va rendre :

Optimiser l'utilisation de l'eau : Un système d'irrigation bien conçu permet d'optimiser l'utilisation de l'eau en fournissant la quantité nécessaire au bon moment.

Programmation et automatisation : Les systèmes d'irrigation modernes offrent la possibilité de programmer et d'automatiser les opérations d'irrigation. Cela permet de définir des horaires d'irrigation adaptés aux besoins spécifiques des cultures, d'ajuster la fréquence et la durée de l'irrigation en fonction des conditions météorologiques.

Économie de main-d'œuvre : Les systèmes d'irrigation automatisés réduisent la dépendance à la main-d'œuvre humaine pour les opérations d'irrigation. Une fois le système installé et programmé, il peut fonctionner de manière autonome, ce qui permet aux agriculteurs de se concentrer sur d'autres tâches importantes.

Amélioration des rendements et de la qualité des cultures : En fournissant une irrigation régulière et adéquate, un système d'irrigation contribue à améliorer les rendements des cultures. Il permet également de contrôler la distribution de l'eau, évitant ainsi les problèmes d'excès ou de manque d'eau, ce qui peut avoir un impact positif sur la qualité des cultures.

Durabilité et préservation des ressources : Les systèmes d'irrigation efficaces contribuent à la durabilité en optimisant l'utilisation de l'eau et en réduisant les pertes. Cela permet de préserver les ressources en eau et de minimiser

l'impact sur l'environnement, en particulier dans les régions où l'eau est limitée ou soumise à des restrictions.

2.2 Schéma global

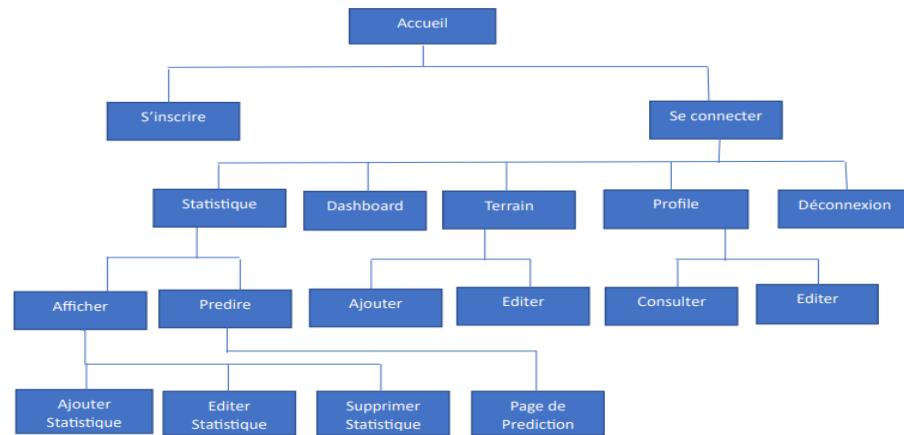


Figure 2.1: Schéma global de système d'irrigation

- 1-L'utilisateur visite la page d'accueil en premier temps.
- 2-Il va choisir entre s'inscrire s'il n'a pas de compte sinon il va se connecter.
- 3- L'utilisateur est redirigé vers la page de Dashboard ou il va consulter tous les Terrains des propriétaires.
- 4- L'utilisateur peut consulter la page de Terrains des propriétaires, il peut aussi ajouter plusieurs terrains bien qu'éditer le terrain concerné.
- 5-L'utilisateur peut consulter la page de son profil ou il peut éditer ses informations personnelles.
- 6-L'utilisateur peut consulter la page des statistiques dans cette dernière il peut soit afficher les statistiques :

(Un tableau de bord contient :

*Un aperçu de l'historique des éléments stockés dans le modèle statistiques avec les options de CRUD.

*La visualisation des données Température, humidité, humidité de sol, force de vent, rayonnement solaire et la consommation d'eau sur un échelle de 1 semaine, 1 mois, 6mois,1 an, 5ans sous forme des graphes.

*Si la valeur de l'arrosage du terrain concerné est positionnée en true on va afficher le débit de l'irrigation sinon on affiche la date de fin d'irrigation.)

Soit prédire (l'utilisateur va être rediriger à la page de prédiction qui contient l'affichage des 5 mesures prédis à court terme et à long terme en utilisant un modèle prédictif à base de réseau de neurones récurrents LSTM Série chronologique).

2.3 Exigences et contraintes

Les exigences et contraintes d'un système d'irrigation peuvent varier en fonction des besoins spécifiques des cultures, ainsi que des ressources disponibles. Voici quelques exigences et contraintes :

Disponibilité de l'eau : La source d'eau disponible est une contrainte essentielle. Il peut s'agir d'une source d'eau souterraine, d'un cours d'eau, d'un réservoir ou d'un système de collecte des eaux de pluie. La quantité d'eau disponible détermine la capacité du système d'irrigation.

Débit d'eau : Le débit de l'eau détermine la quantité d'eau qui peut être distribuée par le système d'irrigation sur une période donnée. Un débit élevé permet une distribution plus rapide, tandis qu'un débit faible peut ralentir le processus d'irrigation. La capacité du système d'irrigation doit être adaptée au débit disponible afin de garantir une distribution adéquate d'eau sur toute la zone irriguée.

Topographie du terrain : La configuration du terrain peut imposer des contraintes sur la conception du système d'irrigation. Les pentes, les zones basses et les obstacles naturels doivent être pris en compte pour s'assurer que l'eau est distribuée de manière uniforme et efficace sur l'ensemble de la zone irriguée.

Conditions climatiques : Les conditions climatiques, telles que Température, humidité, humidité de sol, force de vent, rayonnement solaire et la consommation d'eau, peuvent influencer les exigences d'irrigation.

2.3.1 Description des fonctionnalités attendues

Notre application web est conçue pour aider les propriétaires de jardins et de terrains à maintenir leur système d'irrigation automatique à distance. Voici les

fonctionnalités attendues de cette application :

- 1-Une interface utilisateur fiable, réactive et facile à utiliser.
- 2-Les fonctionnalités d'authentification et d'autorisation Permettent aux utilisateurs de s'inscrire ou de se connecter et de gérer leurs comptes crée et assurer la sécurisation d'accès à l'application.
- 3-Gestion des données : L'application stocke, récupéré Mettre à jour et supprimer (CRUD) des données. Cela peut inclure la gestion des bases de données, la validation des données entrées par les utilisateurs.
- 4-Les fonctionnalités de gestion des utilisateurs Et des permissions permettent aux administrateurs de l'application de gérer les utilisateurs, Les rôles d'utilisateur, les autorisations d'accès, Les paramètres de sécurité.
- 5- Les fonctionnalité de Sécurité : garantie la confidentialité et l'intégrité des données.

2.3.2 Répartition des fonctionnalités par profil utilisateur

Pour qu'une application soit bien structuré et organisé on doit faire une répartition des fonctionnalités par profil utilisateur et ça va être utilisé comme ceci:

- 1-Administrateurs (l'interface d'admin donné par Django) : ils sont accès aux fonctionnalités suivantes :
 - La gestion des utilisateurs : les administrateurs peuvent gèrent les comptes utilisateurs, Les autorisations d'accès, les permissions et les rôles des différents utilisateurs.
 - La configuration des zones d'irrigation : les administrateurs peuvent créer et Modifier les zones d'irrigation, et modifier tout ce qui est statistique : Définir les heures d'irrigation, les quantités d'eau à fournir.
 - La gestion des paramètres de sécurité : les administrateurs peuvent configurer Les paramètres de sécurité de l'application, tels que les mots de passes.

2-Utilisateur :

- Visualisation de l'état d'amélioration de l'irrigation : L'utilisateur peut visualiser les statistiques des terrains des propriétaire et les prédictions de cette dernière.

Cette fonctionnalité permet de suivre l'évolution de l'arrosage et de savoir

quand il est terminé.

-La Consultation de l'historique d'irrigation : L'utilisateur peut consulter l'historique d'irrigation.

-La gestion de profil, de terrain et de statistique.

Chapter 3

Conception et Analyse

Ce chapitre est l'occasion pour nous de présenter l'analyse et les étapes de conception, ce qui nous a permis de connaitre les acteurs, les interfaces, les règles de gestions de notre système ainsi que la manière dont celui-ci devrait fonctionner.

3.1 Outils de travail

Au niveau de cette partie, nous intéressons aux Logiciels utilisés pour la réalisation de notre application et la stratégie suivi durant l'élaboration de notre application web.



Figure 3.1: Association entre back-end et front-end

Dans notre application web on va utiliser le Framework Django qui va en-glober tous le backend de notre application.

On revanche Django est un cadre de développement Web complet qui offre de nombreux avantages pour le développement rapide et efficace d'applications Web.

En l'associer avec PhpMyAdmin qui est une application Web populaire qui permet de gérer les bases de données MySQL. Bien qu'il ne soit pas directement lié à Django.

Et pour pouvoir visualiser les données on va utiliser chart.js qui est une bibliothèque JavaScript open-source qui permet de créer des graphiques interactifs dans des applications web. Elle est facile à utiliser et dispose d'une grande variété de graphiques et de styles pour visualiser les données, tels que les graphiques en barres, en lignes, en secteurs, en radar et en nuages de points.

Pour le frontend on a utilisé le fameux Bootstrap est une bibliothèque open-source de CSS, de JavaScript et de HTML pour la création d'interfaces utilisateur (UI) dans des applications web.

3.2 Modélisation du système

Loin du côté fonctionnel, ce chapitre traite tous les rapports conceptuels et leurs relations avec l'aspect technique et physique, nous présentons alors les spécifications du projet ayant ces aspects.

3.2.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation graphique des interactions entre les utilisateurs (acteurs) et un système. Il permet de décrire les fonctionnalités de l'application web sous forme de scénarios d'utilisation.

a. Diagramme cas d'utilisation:

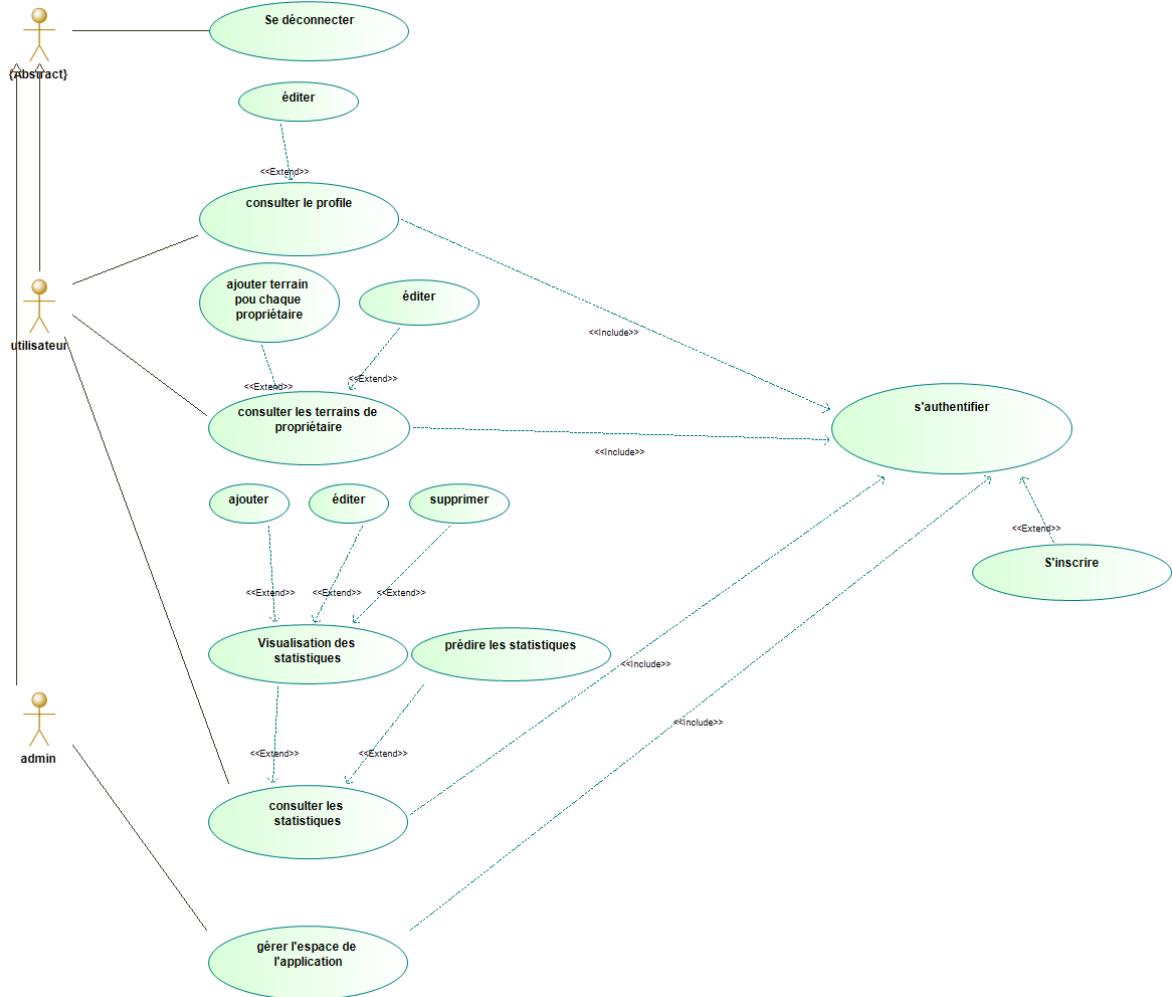


Figure 3.2: Diagramme de cas d'utilisation

b. Raffinements des cas d'utilisation:

Table 3.1: Raffinement de cas d'utilisation "S'inscrire"

Acteurs:	Utilisateur
Préconditions	Chaque utilisateur doit s'inscrire s'il n'a pas un compte Pour bénéficier des services de l'application.
Postconditions :	Cet utilisateur accédera à la page d'inscription.
Scénario :	<ol style="list-style-type: none"> 1. Le système affiche le formulaire d'inscription avec l'implémentation de CSRF. 2.L'utilisateur saisit son login, nom, prénom, adresse mail, Et mot de passe avec la confirmation de ce dernier. 3.L'utilisateur est redirigé vers la page de connexion.
Exception :	<ol style="list-style-type: none"> 1-En cas d'un champ vide un message d'erreur est affiché. 2-En cas d'inégalité des deux champs (mot de passe, confirmation de mot de passe) un message d'erreur est affiché. 3-En cas de platitude du login Le système affiche un message d'erreur si l'utilisateur existe déjà dans la base de données. Dans chaque cas d'erreur l'utilisateur est redirigé vers la page d'inscription.

Table 3.2: Raffinement de cas d'utilisation "S'authentifier"

Acteurs :	Utilisateur/Admin
Préconditions	Chaque utilisateur désirant bénéficier des services de L'application doit s'authentifier d'abord. Chaque utilisateur doit Saisir son login et mot de passe puis se connecter.
Postconditions :	Cet utilisateur accédera aux fonctionnalités de L'application selon ces privilèges.
Scénario :	<ol style="list-style-type: none"> 1. Le système affiche le formulaire de connexion avec l'implémentation de CSRF. 2.L'utilisateur saisit son login et mot de passe. 3.L'utilisateur bénéficiera des fonctionnalités de l'application selon son privilège (administrateur / utilisateur). 4.L'utilisateur est redirigé vers la page de regroupement des terrains des propriétaires.
Exception :	En cas d'inexistence du mot de passe ou du login, un Message d'erreur est affiché en le redirige vers la page de connexion.

Table 3.3: Raffinement de cas d'utilisation "consulter profile"

Acteurs :	Utilisateur/Admin
Préconditions	Chaque utilisateur connecté peut consulter son profil avec la Possibilité de modification.
Postconditions :	Cet utilisateur accédera à son profil.
Scénario :	<ul style="list-style-type: none"> 1. Le système affiche la page de profil de cet utilisateur. 2.L'utilisateur peut modifier son profil en cliquant sur éditer. 3.le système enregistre les données éditées.
Exception :	Le système affiche un message d'erreur si la connexion Avec le serveur base des données est inaccessible.

Table 3.4: Raffinement de cas d'utilisation "consulter les terrains des propriétaires"

Acteurs :	Utilisateur/Admin
Préconditions	Chaque utilisateur connecté peut consulter les terrains des Propriétaires avec la possibilité de l'ajout d'un nouveau terrain d'un propriétaire, modification d'état d'arrosage des terrains.
Postconditions :	l'utilisateur peut consulter l'ensemble des terrains Disponibles de chaque propriétaire.
Scénario :	<ul style="list-style-type: none"> 1. Le système affiche la page de terrain. 2.L'utilisateur consulte les terrains. 3-l'utilisateur peut ajouter un terrain d'un propriétaire. 4-l'utilisateur peut modifier les terrains. 5.le système enregistre les données ajoutées et les données éditées.
Exception :	Le système affiche un message d'erreur si la connexion Avec le serveur base des données est inaccessible.

Table 3.5: Raffinement de cas d'utilisation "consulter les statistiques d'un terrain"

Acteurs :	Utilisateur/Admin
Préconditions	Chaque utilisateur connecté peut consulter les statistiques D'un terrain avec la possibilité de la visualisation des statistiques ou il peut ajouter une nouvelle statistique si l'état d'arrosage est positionné(oui), modifier une statistique, supprimer une statistique. Et la possibilité de la prédiction des statistiques.
Postconditions :	l'utilisateur peut consulter l'ensemble des statistiques Disponibles de chaque terrain.
Scénario :	1-Le système affiche la page de consultation des statistiques. 2-L'utilisateur consulte les statistiques. 3-l'utilisateur peut visualiser les statistiques d'un terrain. 3-1-l'utilisateur ajoute une statistique. si l'état d'arrosage est positionné en (True), à chaque ajout d'une nouvelle statistique le système collecte et enregistre les données chaque heure. 3-2- l'utilisateur modifiée une statistique. Le système enregistre les données éditées. 3-3- l'utilisateur supprime une statistique. 4-l'utilisateur peut prédire les statistiques. Le système utilise les algorithmes de machine learning : Lstm Série chronologique de réseaux de neurones récurrents pour faire la prédiction.
Exception :	Le système affiche un message d'erreur si la connexion Avec le serveur base des données et le serveur redis est inaccessible.

3.2.2 Diagramme de classe

Un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes. Ci-dessous, le diagramme de classe de notre système : Nous pouvons distinguer entre cinq classe priomordiales :

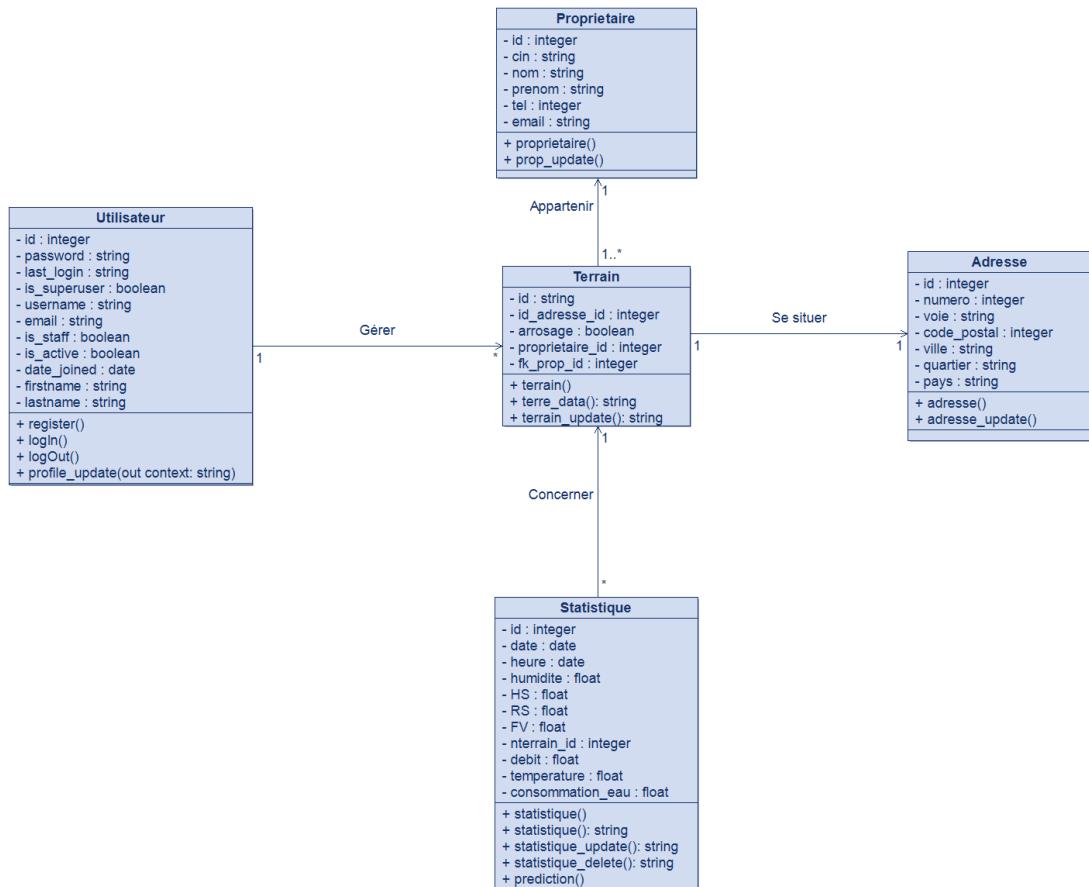


Figure 3.3: Diagramme de Classe

"**Utilisateur**": Un utilisateur qui a accès à interagir avec le système et gère les terrains de chaque propriétaire.

"**Terrain**": Regroupe l'ensemble des terrains disposés par un utilisateur.

"**Proprietaire**": Représente le propriétaire d'un terrain.

"Adresse": Représente l'adresse d'un terrain.

"Statistique": Représente les statistiques d'un terrain disposées par un utilisateur.

3.2.3 Diagramme de séquence

Les diagrammes de séquence sont souvent utilisés pour modéliser des scénarios d'utilisation des systèmes, des interactions entre les composants logiciels et matériels, des processus métiers, des protocoles de communication, etc. Ils sont utiles pour comprendre et communiquer les interactions entre les différents éléments d'un système et pour identifier les problèmes potentiels liés à la séquence des messages échangés entre ces éléments.

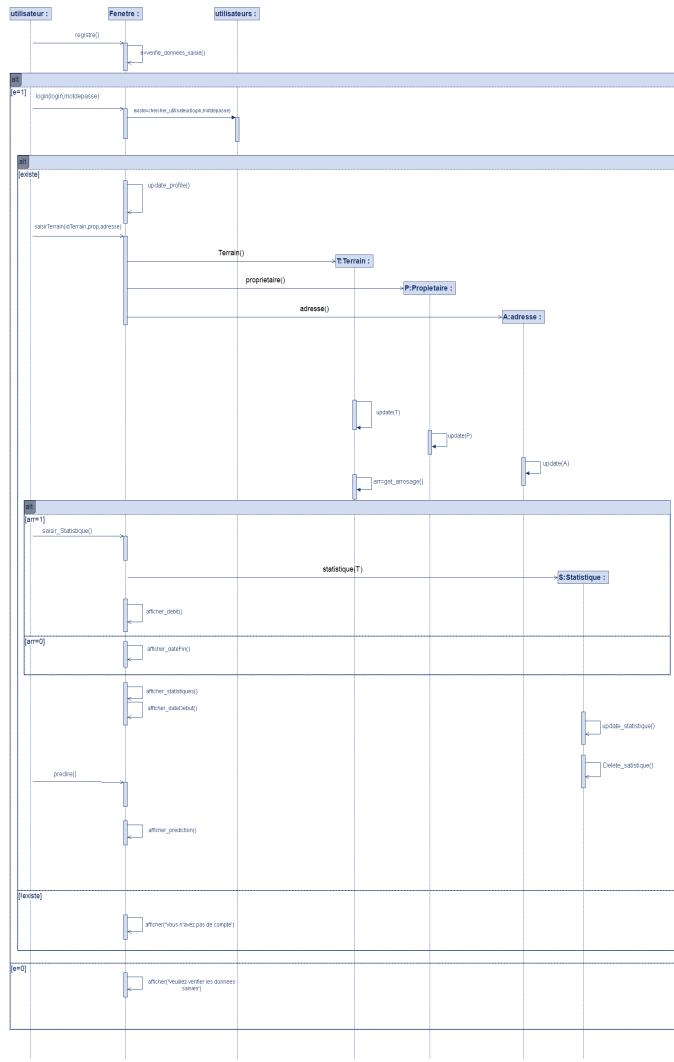


Figure 3.4: Diagramme de séquence

3.3 Conception générale

Dans cette partie on va traiter la conception générale du système SGBD:

Cette Table représente l'utilisateur qui interagie avec le système qui a pour attributs:

id, password, last_login, is_superuser, username, first_name, last_name, email, is_staff, is_active, date_joined.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	password	varchar(128)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
3	last_login	datetime(6)			Oui	NULL			Modifier Supprimer Plus
4	is_superuser	tinyint(1)			Non	Aucun(e)			Modifier Supprimer Plus
5	username	varchar(150)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
6	first_name	varchar(150)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
7	last_name	varchar(150)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
8	email	varchar(254)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
9	is_staff	tinyint(1)			Non	Aucun(e)			Modifier Supprimer Plus
10	is_active	tinyint(1)			Non	Aucun(e)			Modifier Supprimer Plus
11	date_joined	datetime(6)			Non	Aucun(e)			Modifier Supprimer Plus

Figure 3.5: La Table «Utilisateur»

La table représente le Propriétaire du terrain saisie par l'utilisateur. Elle a les attributs suivantes:

id, cin, email, nom, prenom, tel.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	bigint(20)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	cin	varchar(200)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
3	email	varchar(200)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
4	nom	varchar(200)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
5	prenom	varchar(200)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
6	tel	int(11)			Non	Aucun(e)			Modifier Supprimer Plus

Figure 3.6: La Table «Propriétaire»

C'est la table Adresse du terrain associé au propriétaire. Elle contient les attributs suivantes:

id, numero, quartier, voie, code postal, ville, pays.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	bigint(20)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	numero	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
3	quartier	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
4	voie	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
5	code_postale	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
6	ville	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
7	pays	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

Figure 3.7: La Table «Adresse»

La représentation des terrains est représenté par la table Terrain. Elle contient les attributs suivantes:

id, arrosage, proprietaire_id, id_adresse_id, fk_prop_id.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	bigint(20)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	arrosage	tinyint(1)			Non	Aucun(e)			Modifier Supprimer Plus
3	proprietaire_id	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
4	id_adresse_id	bigint(20)			Oui	NULL			Modifier Supprimer Plus
5	fk_prop_id	bigint(20)			Oui	NULL			Modifier Supprimer Plus

Figure 3.8: La Table «Terrain»

Pour représenter les statistiques des terrains de chaque propriétaire en utilise la table Statistique. Elle contient les attributs suivantes:

id, date, heure, humidite, HS, FV, RS, Nterrain_id, debit, temperature, consommation_eau.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id 📄	bigint(20)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	date	date			Non	Aucun(e)			Modifier Supprimer Plus
3	heure	time(6)			Non	Aucun(e)			Modifier Supprimer Plus
4	humidite	double			Non	Aucun(e)			Modifier Supprimer Plus
5	HS	double			Non	Aucun(e)			Modifier Supprimer Plus
6	FV	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
7	RS	double			Non	Aucun(e)			Modifier Supprimer Plus
8	Nterrain_id 🔑	bigint(20)			Non	Aucun(e)			Modifier Supprimer Plus
9	debit	double			Non	Aucun(e)			Modifier Supprimer Plus
10	temperature	double			Non	Aucun(e)			Modifier Supprimer Plus
11	consommation_eau	double			Non	Aucun(e)			Modifier Supprimer Plus

Figure 3.9: La Table «Statistique»

La gestion de l'application se fait par la table Admin. Elle contient les attributs suivantes:

id, action_time, object_id, object_repr, action_flag, change_message, content_type_id, user_id.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id 📄	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
2	action_time	datetime(6)			Non	Aucun(e)			Modifier Supprimer Plus
3	object_id	longtext	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
4	object_repr	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
5	action_flag	smallint(5)		UNSIGNED	Non	Aucun(e)			Modifier Supprimer Plus
6	change_message	longtext	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
7	content_type_id 🔑	int(11)			Oui	NULL			Modifier Supprimer Plus
8	user_id 🔑	int(11)			Non	Aucun(e)			Modifier Supprimer Plus

Figure 3.10: La Table «Admin»

cette table représente la durée de collection des données. Elle contient les attributs suivantes:

ident, last_update.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	ident	smallint(6)			Non	Aucun(e)		 Modifier  Supprimer 	Plus
2	last_update	datetime(6)			Non	Aucun(e)		 Modifier  Supprimer 	Plus

Figure 3.11: La Table «Celerybeatperiodictasks»

La collection des statistiques chaque heure se fait par la table Celerybeatperiodictask. Elle contient les attributs suivantes:

id, name, task, args, kwargs, queue, exohange, routing_key, expires, enabled, last_run_at, total_run_ocount, date_ohanged, desorption, orantab_id, interval_id, scolar_id, one_off, start_time, priority, headers, olookad_id, expire_seconds.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int(11)			Non	Aucun(e)	AUTO_INCREMENT	 Modifier  Supprimer 	Plus
2	name	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)		 Modifier  Supprimer 	Plus
3	task	varchar(200)	utf8mb4_general_ci		Non	Aucun(e)		 Modifier  Supprimer 	Plus
4	args	longtext	utf8mb4_general_ci		Non	Aucun(e)		 Modifier  Supprimer 	Plus
5	kwargs	longtext	utf8mb4_general_ci		Non	Aucun(e)		 Modifier  Supprimer 	Plus
6	queue	varchar(200)	utf8mb4_general_ci		Oui	NULL		 Modifier  Supprimer 	Plus
7	exchange	varchar(200)	utf8mb4_general_ci		Oui	NULL		 Modifier  Supprimer 	Plus
8	routing_key	varchar(200)	utf8mb4_general_ci		Oui	NULL		 Modifier  Supprimer 	Plus
9	expires	datetime(6)			Oui	NULL		 Modifier  Supprimer 	Plus
10	enabled	tinyint(1)			Non	Aucun(e)		 Modifier  Supprimer 	Plus
11	last_run_at	datetime(6)			Oui	NULL		 Modifier  Supprimer 	Plus
12	total_run_count	int(10)		UNSIGNED	Non	Aucun(e)		 Modifier  Supprimer 	Plus
13	date_changed	datetime(6)			Non	Aucun(e)		 Modifier  Supprimer 	Plus
14	description	longtext	utf8mb4_general_ci		Non	Aucun(e)		 Modifier  Supprimer 	Plus
15	orantab_id	int(11)			Oui	NULL		 Modifier  Supprimer 	Plus
16	interval_id	int(11)			Oui	NULL		 Modifier  Supprimer 	Plus
17	solar_id	int(11)			Oui	NULL		 Modifier  Supprimer 	Plus
18	one_off	tinyint(1)			Non	Aucun(e)		 Modifier  Supprimer 	Plus
19	start_time	datetime(6)			Oui	NULL		 Modifier  Supprimer 	Plus
20	priority	int(10)		UNSIGNED	Oui	NULL		 Modifier  Supprimer 	Plus
21	headers	longtext	utf8mb4_general_ci		Non	Aucun(e)		 Modifier  Supprimer 	Plus
22	clocked_id	int(11)			Oui	NULL		 Modifier  Supprimer 	Plus
23	expire_seconds	int(10)		UNSIGNED	Oui	NULL		 Modifier  Supprimer 	Plus

Figure 3.12: La Table «Celerybeatperiodictask»

Chapter 4

Mise en oeuvre du projet

Après la mise en oeuvre de l'étude et la conception de notre application, nous passons à la phase d'implémentation. ce chapitre présente le résultat du travail effectué durant ce projet de fin d'études. Nous allons présenter, aussi, environnement matériels et les outils de développement utilisés. Nous clôturons ce chapitre par quelques captures d'écran démontrant les fonctionnalités de notre application.

4.1 Technologies et outils de développement

On va consacré cette partie à la définition de toutes les outils utilisées.

a.PYTHON:

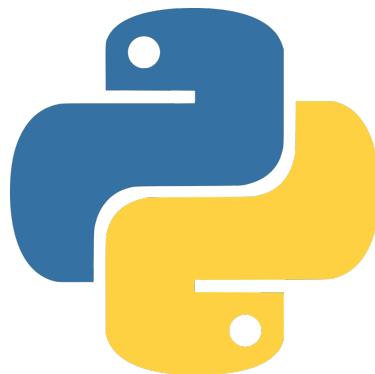


Figure 4.1: PYTHON

Python est un langage de programmation interprété, polyvalent et convivial. Il a été créé par Guido van Rossum et publié pour la première fois en 1991. Python est réputé pour sa lisibilité, sa syntaxe claire et concise, ce qui en fait un choix populaire parmi les développeurs, en particulier pour les débutants en programmation.

Le backend, en revanche, est la partie de l'application qui fonctionne en coulisse pour supporter le frontend. Le backend est responsable de la gestion des données de l'application, de la communication avec les serveurs et les bases de données, ainsi que de l'exécution des calculs et de la logique métier de l'application. comme back-end on a utilisé le framework django.

DJAGNO:



Figure 4.2: Framework DJANGO

Django est un framework Web Python de haut niveau qui suit le modèle de conception MVT (Model View Template). Il est conçu pour aider les développeurs à créer des applications Web rapidement et efficacement.

Virtuel Environnement: venv, également connu sous le nom de "Virtual Environment" en anglais, est un outil intégré à Python qui permet de créer des environnements virtuels isolés. Un environnement virtuel est un environnement Python autonome qui contient sa propre installation de Python et ses propres packages.

MYSQL:



Figure 4.3: MYSQL

MySQL est un système de gestion de base de données relationnelle (SGBDR)

open-source largement utilisé. Il est populaire en raison de sa facilité d'utilisation, de sa fiabilité et de sa performance. MySQL permet de stocker, d'organiser et de gérer efficacement de grandes quantités de données structurées.

MVC:

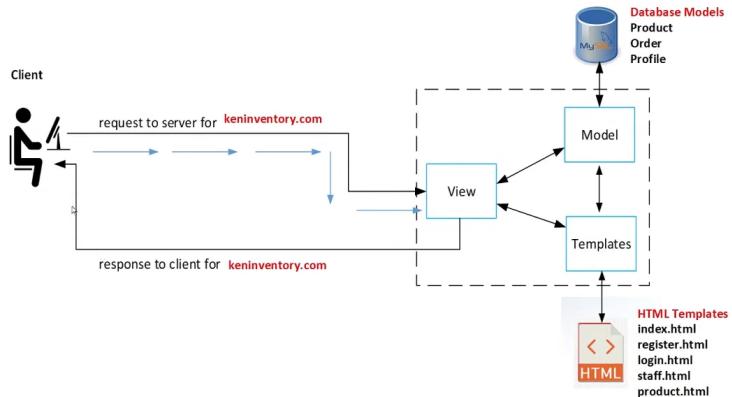


Figure 4.4: MODEL VIEW CONTROLLER

Modèle : Les données qu'on doit présenter, généralement des données provenant d'une base de données. pour cela on a utilisé PhpMyAdmin qui une application Web open source écrite en PHP qui permet d'administrer les bases de données MySQL. Elle est largement utilisée dans le développement web pour gérer des bases de données MySQL de manière graphique plutôt que par l'intermédiaire de la ligne de commande. Les modèles sont généralement situés dans un fichier appelé models.py

Les URLs : elles sont utilisées pour router les requêtes HTTP vers les vues appropriées en fonction de l'URL demandée. Les URLs de Django sont définies dans un fichier de configuration et utilisent des expressions régulières pour identifier les motifs d'URL à correspondre. Les urls sont généralement situés dans un fichier appelé urls.py

Vue: Une vue est une fonction ou une méthode qui prend les requêtes http comme arguments, importe le ou les modèles pertinents, détermine les données à envoyer au modèle et renvoie le résultat final. Les vues sont généralement situés dans un fichier appelé views.py

Les formulaires: ils permettent de créer des formulaires HTML à partir des modèles de Django et de gérer les données soumises par les utilisateurs. Les formulaires sont généralement situés dans un fichier appelé forms.py

Settings: Un fichier de paramètres Django contient toute la configuration de l'installation Django que les informations de base de données, les clés secrètes, les

adresses IP autorisées, etc. C'est l'un des fichiers les plus importants du projet Django car il définit le comportement global de l'application. Les paramètres sont généralement situés dans un fichier appelé `settings.py`

`init.py`: ce fichier est utilisé pour définir un package Python. Il est présent dans chaque dossier qui doit être considéré comme un package et peut contenir du code Python à exécuter lors de l'importation du package.

`manage.py`: c'est le script principal qui est utilisé pour lancer différentes commandes en rapport avec le projet Django. Il peut être utilisé pour créer une nouvelle application, effectuer des migrations de base de données, lancer un serveur de développement, etc.

`static/`: ce dossier contient tous les fichiers statiques (CSS, JavaScript, images, etc.) utilisés par l'application.

`migrations/`: ce dossier contient tous les fichiers de migration de base de données générés automatiquement par Django lorsque la structure de la base de données change.

REPORTING:

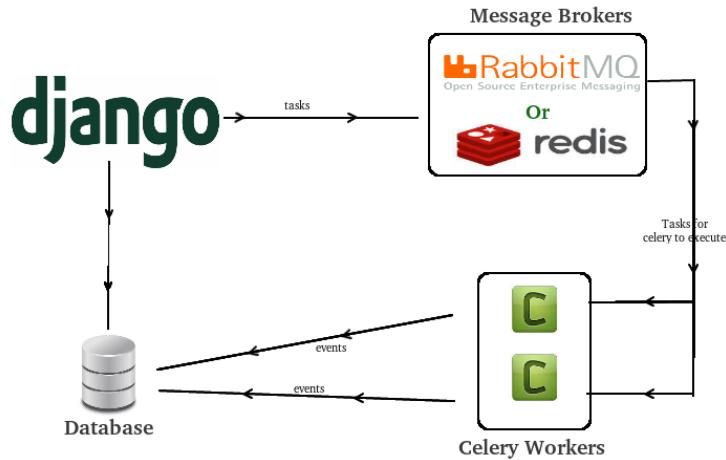


Figure 4.5: ARCHITECTURE DJANGO CELERY

Reporting: Le reporting de données est le processus de collecte, d'analyse et de présentation de données dans un format compréhensible et utile. Le reporting est souvent utilisé pour surveiller les performances d'une entreprise ou d'un système, pour identifier les tendances et les opportunités, pour prendre des décisions éclairées, et pour communiquer des informations à d'autres parties prenantes. Pour cela on va utiliser celery qui permet d'exécuter des tâches de traitement en arrière-plan, sans bloquer le processus principal de l'application web. On va utiliser le serveur Redis qui est un système de base de données en mémoire open-source, qui fonctionne comme une base de données clé-valeur. Le nom Redis est l'acronyme de "REmote DIctionary Server" (serveur de dictionnaire distant). Il a été conçu pour être rapide, performant et simple à utiliser.

Le frontend se réfère à la partie de l'application qui est visible pour l'utilisateur final. Cela inclut l'interface utilisateur, le design graphique, les animations et tous les autres éléments que l'utilisateur peut interagir avec. Le frontend est généralement développé en utilisant des langages de programmation comme HTML, CSS, JavaScript et chart.js. Comme front-end on a utilisé les templates Bootstrap.

JAVASCRIPT:

JavaScript

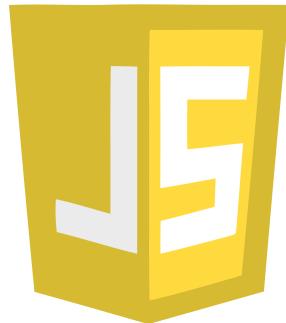


Figure 4.6: Javascript

Javascript: JavaScript (« JS » en abrégé) est un langage de programmation dynamique complet qui, appliqué à un document HTML, peut fournir une interactivité dynamique sur les sites Web.

DATA VISUALIZATION:



Figure 4.7: DATA VISUALIZATION

DataVis: La datavisualisation, ou datavis, est l'utilisation de graphiques, de tableaux, de cartes et d'autres outils visuels pour représenter les données de manière claire et compréhensible. Il existe de nombreux outils de datavis qui permettent de créer des visualisations interactives, des tableaux de bord, des infographies et d'autres représentations graphiques des données. Pour cela on a utilisé :

CHART JS:

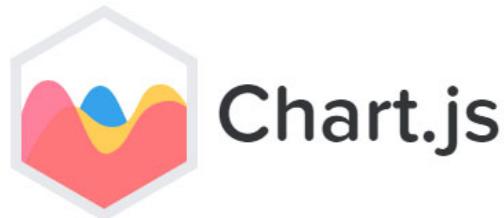


Figure 4.8: CHART JS

chart.js - est une bibliothèque JavaScript open source qui permet de créer des graphiques et des diagrammes dynamiques sur une page web. Elle permet de représenter des données de manière claire et visuelle, en utilisant différents types de graphiques tels que les barres, les courbes, les camemberts.

BOOTSTRAP:



Figure 4.9: BOOTSTRAP

Templates Bootstrap: Les templates Bootstrap sont des modèles de conception préconçus pour créer des sites web ou des applications web avec Bootstrap. Ils incluent souvent des éléments d'interface utilisateur tels que des bannières, des carrousels, des onglets, des cartes, des listes, etc., ainsi que des styles CSS pour personnaliser l'apparence de ces éléments. Les templates Bootstrap sont utiles car ils permettent de gagner du temps et de l'effort dans la conception de l'interface utilisateur d'un site web ou d'une application web. Les développeurs peuvent utiliser ces modèles comme point de départ pour leur propre conception, en modifiant les éléments et les styles selon leurs besoins.

MACHINE LEARNING :

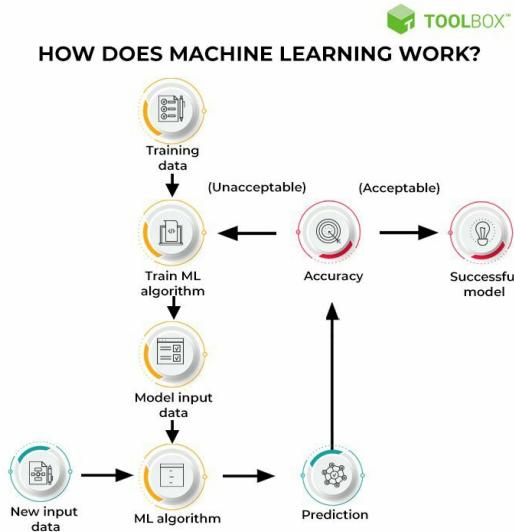


Figure 4.10: MACHINE LEARNING

Apprentissage automatique : est une branche de l'intelligence artificielle qui se concentre sur le développement de techniques et d'algorithmes permettant aux ordinateurs d'apprendre et de s'améliorer à partir des données, sans être explicitement programmés. L'apprentissage automatique permet aux machines d'identifier des modèles et des relations cachés dans les données, et de prendre des décisions ou de faire des prédictions basées sur ces modèles.

Modèle prédictif : est un modèle statistique ou algorithmique utilisé pour faire des prédictions ou des estimations sur de nouvelles données, en se basant sur des données existantes. Les modèles prédictifs sont largement utilisés dans de nombreux domaines, tels que la science des données, l'apprentissage automatique, la finance, l'économie, la météorologie, la santé, le marketing.

RNN-LSTM-SERIE CHRONOLOGIQUE:

RNN: Les réseaux de neurones récurrents (RNN - Recurrent Neural Networks en anglais) sont un type de réseau de neurones artificiels qui permettent de traiter des données séquentielles ou temporelles. Contrairement aux réseaux de neurones classiques, les RNNs possèdent des connexions récurrentes, ce qui leur permet de conserver une mémoire interne et de prendre en compte les informations passées lors du traitement des données actuelles.

LSTM : Long Short-Term Memory est une variante de cellule récurrente utilisée dans les réseaux de neurones récurrents (RNNs). Les LSTMs ont été conçus pour

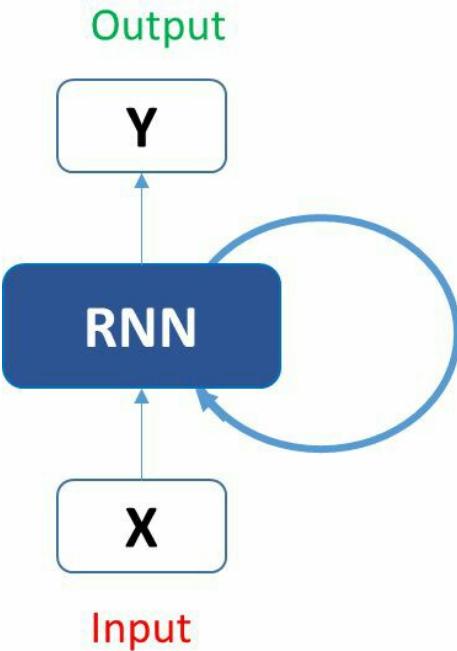


Figure 4.11: RNN-LSTM-SERIE CHRONOLOGIQUE

résoudre le problème du gradient qui explose ou qui disparaît, qui peut survenir lors de l'entraînement des RNNs traditionnels. Ils sont particulièrement efficaces pour modéliser des dépendances à long terme dans les données séquentielles.

Série chronologique : (ou série temporelle) est une séquence de données numériques qui sont collectées et enregistrées dans l'ordre chronologique. Chaque observation de la série est associée à un moment spécifique dans le temps. Les séries chronologiques se trouvent dans de nombreux domaines, notamment l'économie, la finance, la météorologie, la santé, l'ingénierie et bien d'autres.

4.2 Architecture technique

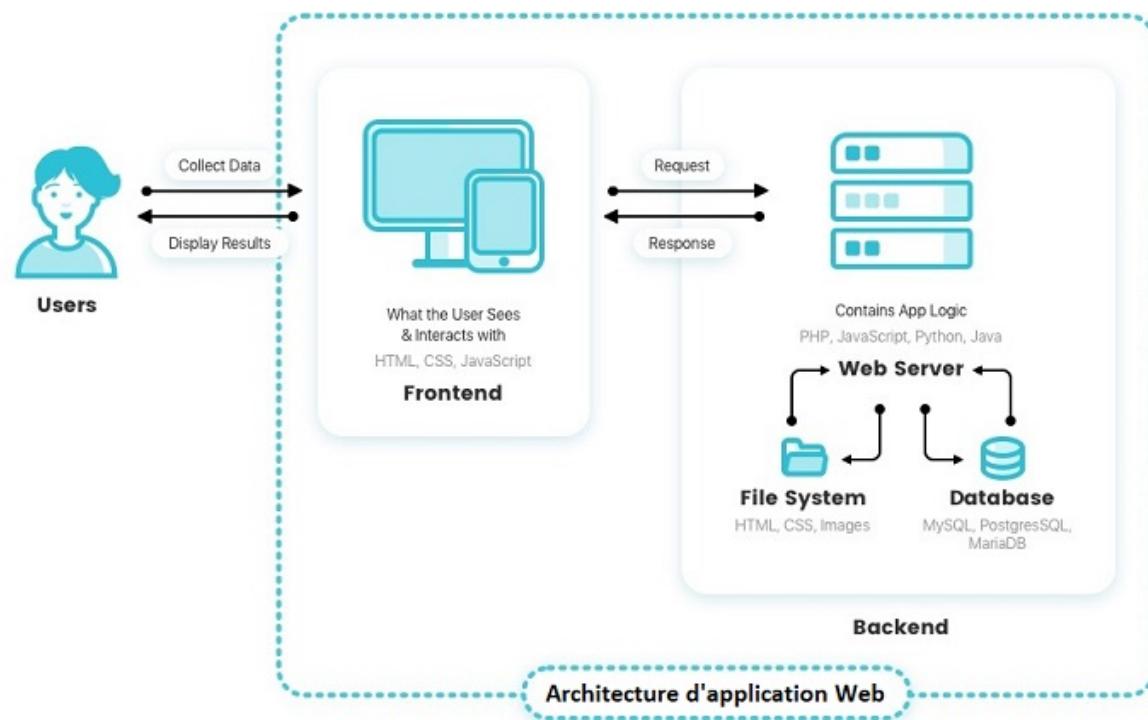


Figure 4.12: Architecture générale de système d'irrigation

Notre site contient plusieurs couches et composants qui travaillent ensemble pour offrir une expérience utilisateur cohérente et fluide.

1-Couche de présentation qui est responsable de la couche de l'interface d'un utilisateur Elle inclut des technologies telles que HTML, CSS et JavaScript pour créer des pages web interactives et réactives.

2-Couche de traitement côté client Elle inclut des technologies pour envoyer des demandes asynchrones au serveur et récupérer des données en temps réel avec des bibliothèques de framework pour créer des applications webs modernes.

3-Couche de traitement côté serveur : Cette couche est responsable de la logique de traitement des données côté serveur. Elle inclut des technologies telles que python.

4-Couche de gestion de données : Cette couche est responsable de la gestion des données stockées par l'application. Elle inclut des technologies telles que MySQL pour stocker et récupérer les données.

5-Couche d'infrastructure : Cette couche est responsable de l'infrastructure technique qui soutient l'ensemble du système.

4.3 Présentation des interfaces Homme Machine

Dans cette partie on va passer à la partie de présentation des interfaces réalisées.

4.3.1 Accueil:



Figure 4.13: Page d'accueil

Afin de répondre aux besoins et attentes de nos clients, nous avons conçu une interface Simple à manipuler et à la portée de tous les utilisateurs.

La page d'accueil présente le logo de l'application et une brève description de ses fonctionnalités.

Un bouton "Connexion" est visible pour les utilisateurs existants, avec un lien vers la page de connexion.

Un bouton "Inscription" est disponible pour les nouveaux utilisateurs, redirigeant vers la page d'inscription.

4.3.2 Inscription:



Figure 4.14: Page d'inscription

Le système d'irrigation automatique propose l'inscription pour un utilisateur si ce dernier n'a pas de compte pour bénéficier des services de l'application.

La page d'inscription contient des champs pour saisir les informations nécessaires, telles que Nom d'utilisateur, Nom, Prénom, l'adresse e-mail et le mot de passe, et la confirmation de mot de passe.

Des validations sont effectuées pour vérifier la validité des informations saisies.

Un bouton "S'inscrire" permet de créer un nouveau compte utilisateur dans l'application.

4.3.3 Connexion:



Figure 4.15: Page de connexion

Le système d'irrigation automatique propose la connexion pour un utilisateur inscrit.

La page de connexion comporte des champs pour saisir le nom d'utilisateur ou l'adresse e-mail et le mot de passe.

Un bouton "Se connecter" permet de valider les informations et de se connecter à l'application.

4.3.4 Regroupement:

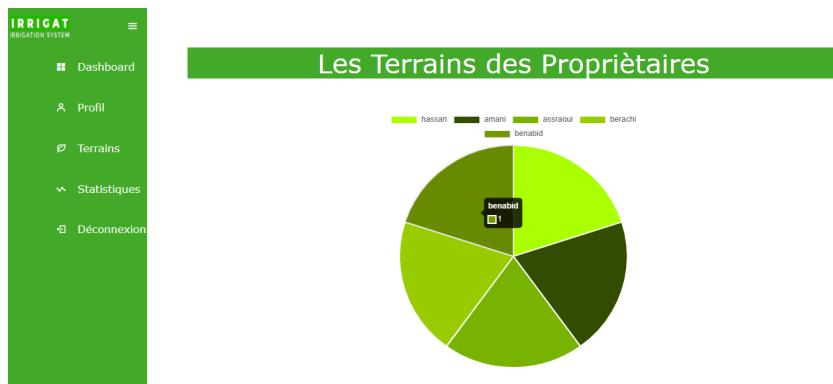


Figure 4.16: Page Regroupement des terrains

L'application fournie un regroupement des terrains des propriétaires. Lorsque l'utilisateur se connecte il est redirigé vers cette page qui représente les différents terrains saisis par l'utilisateur.

4.3.5 Gestion des profils:

- PROFIL:

The screenshot shows a mobile application interface. On the left is a vertical green sidebar with icons for navigation. The main area has a green header bar with the text "Page de Profil". Below the header, the title "Vos informations Personnelles" is displayed, followed by an "Editor" button. A table lists the user's personal information:

Nom d'utilisateur	nihad
Prénom	nihad1
Nom	achir
Email	achirnihad@gmail.com

Figure 4.17: Page de profil

Notre système d'irrigation automatique propose une consultation de profil de l'utilisateur connecté il présente les différentes informations personnelles de cet utilisateur telles que : Nom d'utilisateur, Nom, Prénom, Email.

- EDITER PROFIL

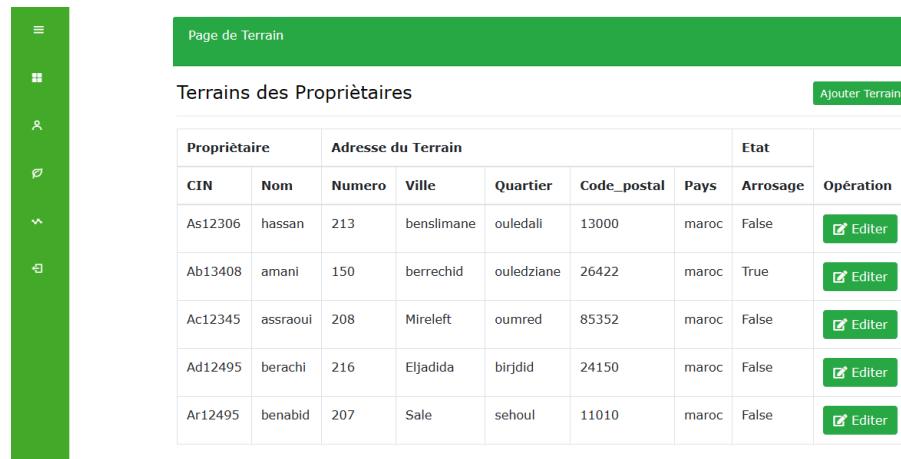
The screenshot shows the same mobile application interface. The sidebar is visible on the left. The main area has a green header bar with the text "Page de Profil". Below it, the title "Votre Profil" is displayed. The user's information is shown in input fields, which are currently filled with the same data as in Figure 4.17. At the bottom of the form is a green "Editor" button.

Figure 4.18: Page de modification de profil

Une autre proposition que notre système offre: la modification des champs des informations personnelles qui sont récupérer tout en respectant CRUD (create, read, update, delete).

4.3.6 Gestion des Terrains:

- TERRAINS:



The screenshot shows a web application interface titled "Page de Terrain". On the left, there is a vertical sidebar with icons for navigation. The main content area has a green header bar with the title "Page de Terrain" and a button "Ajouter Terrain". Below this, a sub-header "Terrains des Propriétaires" is displayed. A table lists several land records, each with an "Editer" button. The table columns are: Propriétaire, Adresse du Terrain, Etat, and Opération.

Propriétaire		Adresse du Terrain					Etat	Opération
CIN	Nom	Numero	Ville	Quartier	Code_postal	Pays	Arrosage	
As12306	hassan	213	benslimane	ouledali	13000	maroc	False	<button>Editer</button>
Ab13408	amani	150	berrechid	ouledziane	26422	maroc	True	<button>Editer</button>
Ac12345	assraoui	208	Mireleft	oumred	85352	maroc	False	<button>Editer</button>
Ad12495	berachi	216	Eljadida	birjdid	24150	maroc	False	<button>Editer</button>
Ar12495	benabid	207	Sale	sehoul	11010	maroc	False	<button>Editer</button>

Figure 4.19: Page de consultation des terrains

Cette page propose une consultation des terrains des propriétaires saisis par l'utilisateur connecté il présente les différentes informations des terrains telles que :

concernant les Propriétaires :

1-CIN.

2-Nom.

concernant les Adresses :

1-Numero.

2-Ville.

3-Quartier.

4-Code postal.

l'état d'arrosage des terrains :

1-Arrosage.

- AJOUTER TERRAIN:

The screenshot shows a mobile application interface with a green header bar labeled "Page de Terrain". Below the header is a section titled "Terrain_d'utilisateur". The form contains the following fields:

- Numero: 2
- Cin: (empty)
- Quartier: (empty)
- Nom: (empty)
- Voie: (empty)
- Prenom: (empty)
- Code postal: 150
- Tel: 144
- Ville: (empty)
- Email: (empty)
- Pays: (empty)
- Arrosage: (checkbox) (unchecked)
- Enregistrer** (green button)

Figure 4.20: Ajout d'un nouveau terrain

Le système d’irrigation automatique propose également l’ajout des terrains qui contient les champs suivants :

consternant les Propriétaires :

1-CIN.

2-Nom.

3-Prenom.

4-tel.

5-Email.

consternant les Adresses :

1-Numero.

2-Ville.

3-Quartier.

4-Code postal.

5-voie.

6-pays.

l’état d’arrosage des terrains :

1-Arrosage.

- EDITER TERRAIN:

Page de Terrain

Cin	Nom
As12306	hassan
Prénom	Email
mrabti	hassan@gmail.com
Téléphone	
667833896	
Numéro	Quartier
213	ouledali
Voie	Code postal
rue	13000
Ville	Pays
benslimane	maroc
Arrosage	
Non	
Editor	

Figure 4.21: Editer Terrain

L'utilisateur a la possibilité de faire la modification des champs des terrains qui sont récupérer tout en respectant CRUD (create, read, update, delete).

4.3.7 Gestion des Statistiques:

- STATISTIQUE:

Page de Statistiques

Numéro_terrain	CIN
213	As12306
150	Ab13408
208	Ac12345
216	Ad12495
207	Ar12495

Afficher **Prédire**

Figure 4.22: Page de consultation des statistiques

Cette application fournit une page de statistique qui affiche le numéro de terrain, le CIN du propriétaire du terrain avec la possibilité de l'affichage des statistiques ou bien la prédition.

- AFFICHER STATISTIQUE:

L'application permet la consultation des statistiques des terrains saisies par l'utilisateur connecté. Il présente un tableau de bord qui contient date de début, durée, date de fin si l'état d'arrosage positionnée en "False" sinon débit total sous forme de cartes avec des graphiques des mesures (température, humidité, humidité de sol, force de vent, rayonnement solaire) et un graphe qui affiche la consommation d'eau sur une échelle d'une semaine, un mois, six mois, un an, cinq ans.

Il présente également un aperçu historique des mesures (température, humidité, humidité de sol, force de vent, rayonnement solaire, date, heure, débit, consommation d'eau) avec les opérations possibles (Edition, Suppression).

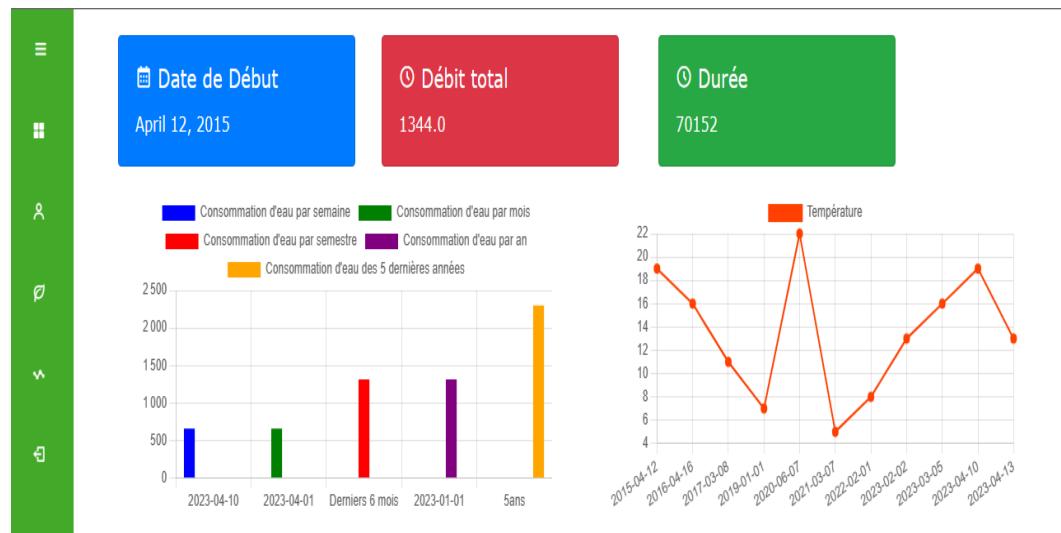


Figure 4.23: Page de statistiques

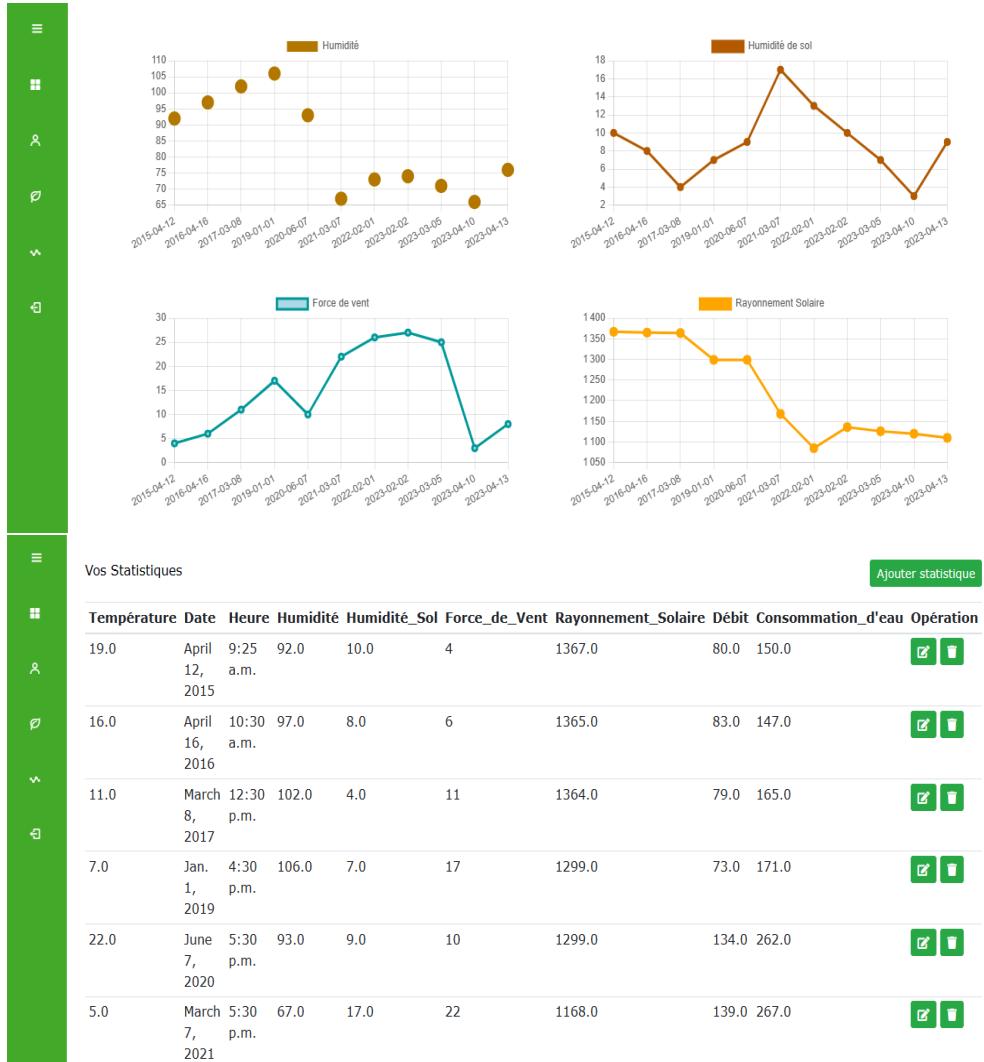


Figure 4.24: Page de statistiques

- AJOUTER STATISTIQUE:

The screenshot shows a mobile application interface for adding user statistics. The top navigation bar is green with the text "Page de Statistiques". On the left, there is a vertical sidebar with a green background containing icons for a menu, a user profile, a search function, and other settings. The main content area is titled "Statistiques d'utilisateur". It features a grid of input fields for various environmental and usage parameters. Row 1: "Temperature:" with value "10", "Humidité:" (empty), "HS:" (empty). Row 2: "FV:" (empty), "RS:" (empty), "Date:" (empty). Row 3: "Heure:" (empty), "Débit:" with value "100", "Consommation eau:" with value "100". At the bottom is a green rounded rectangular button labeled "Enregistrer".

Figure 4.25: Page ajouter statistique

Le système suggère l'ajout d'une nouvelle statistique si l'état d'arrosage positionnée en "True", qui contient:

- Température.
- Humidité.
- HS : Humidité du sol.
- FV : Force de vent.
- RS : Rayonnement solaire.
- Date.
- Heure.
- Débit.
- Consommation d'eau.

- EDITER STATISTIQUE :

Température	Date
5,0	March 7, 2021
Heure	Humidité
5:30 p.m.	67,0
Humidité de sol	Force de vent
17,0	22
Rayonnement Solaire	Débit
1168,0	139,0
Consommation d'eau	
267,0	

Editer

Figure 4.26: Page de modification d'une statistique

Notre système propose également la modification des champs des statistiques qui sont récupérer tout en respectant CRUD (create, read, update, delete). Ces données :

- Température.
- Humidité.
- HS : Humidité du sol.
- FV : Force de vent.
- RS : Rayonnement solaire.
- Date.
- Heure.
- Débit.
- Consommation d'eau.

- SUPPRIMER STATISTIQUE :

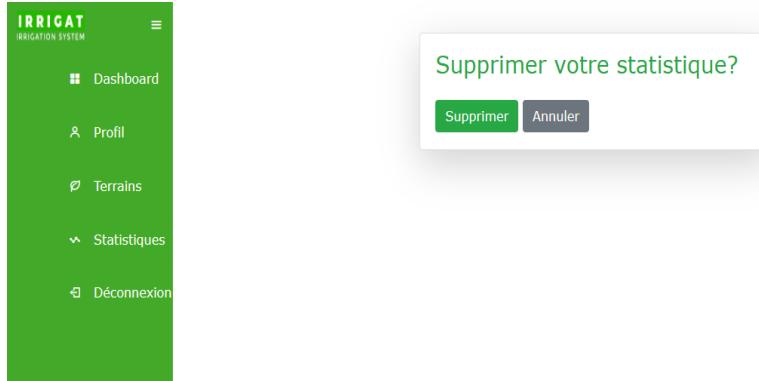


Figure 4.27: Page de suppression d'une statistique

Pour avoir la possibilité de supprimer une statistique Notre système d'irrigation automatique propose la suppression d'une statistique. Cette page contient la suppression de la statistique avec deux boutons : annuler et supprimer Si l'utilisateur clique sur le bouton annuler il est redirigé vers la page de statistique. Si l'utilisateur clique sur le bouton supprimer un message alerte est affiche "les données ont été supprimer avec succès" tout en respectant CRUD (create, read, update, delete).

4.3.8 Prédiction:

Une autre intéressante proposition que notre système peut offrir : l'affichage des prédictions de chaque mesure (température, humidité, humidité de sol, force de vent, rayonnement solaire, consommation d'eau) sous forme des graphiques qui présente les mesures passé, la mesure prédictive. En se basant sur un modèle prédictif à base de réseaux de neurones récurrents LSTM Série chronologique.

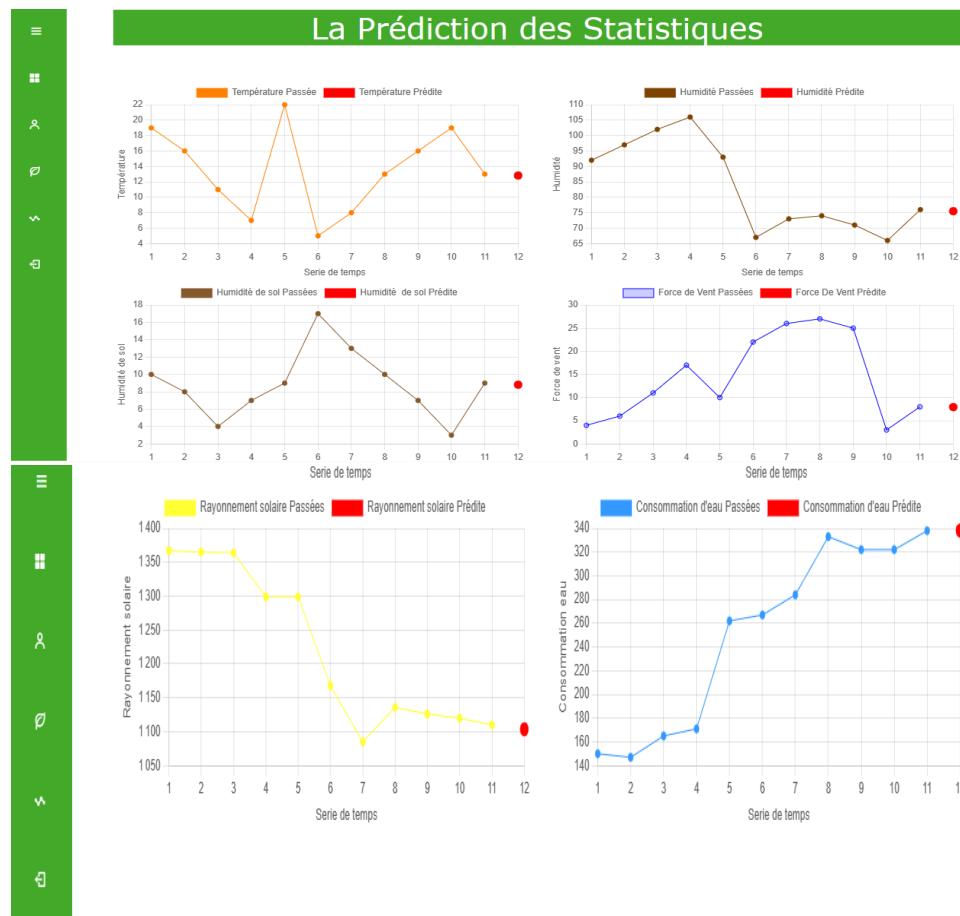


Figure 4.28: Page de prédition

Conclusion & Perspective

1-Conclusion:

Durant toute la période de notre projet de fin d'études on a pu acquérir de nombreuses Techniques. A part le fait d'améliorer l'habileté à travailler en équipe, ce projet là nous a Projété dans le domaine de la modélisation et de la programmation.

On a pu exploiter tous ce qu'on a appris dans le module de la modélisation orienté Objet et par la même occasion apprendre un nouveau langage de programmation qui S'est avéré très demandé dans le marché du travail.

Notre projet permet de réaliser une application web d'un système d'irrigation en langage python Qui permet aux utilisateurs d'avoir une représentation visuelle de l'ensemble des Statistiques représentées sous forme d'un tableau de bord avec la configuration d'un modèle prédictif à base d'un algorithme d'apprentissage automatique ((LSTM Série chronologique).

Enfin, on souhaite que ce modeste travail soit à la hauteur des espérances de notre Encadrant et qu'il soit au niveau des attentes des honorables membres du jury.

2-Perspective:

Notre application d'irrigation automatique peut bénéficier de plusieurs améliorations pour optimiser l'efficacité et la facilité d'utilisation. Voici quelques améliorations possibles

-Utilisation d'un autre algorithme de Prédiction : les modèles de forêt aléatoire sont plus polyvalents et peuvent être utilisés dans différents types de problèmes. Ils sont plus simples et peuvent être plus facilement interprétés.

-Une version Mobile : on peut ajouter une application mobile (Android) pour que les utilisateurs peuvent les utiliser n'importe où et à tout moment, tant qu'ils ont leur appareil mobile avec eux. Cela permet une accessibilité et une disponibilité optimales pour les utilisateurs.

-Intégration d'Assistant vocal : on peut également intégrer notre application d'irrigation automatique avec des assistants vocaux tels que Google Assistant ou Amazon Alexa. Cela nous a permis de contrôler l'arrosage à l'aide de commandes vocales, rendant l'expérience plus pratique et intuitive.

BIBLIOGRAPHIE

Documentation Framework Django :

- <https://www.djangoproject.com/>
- <https://developer.mozilla.org/>
- <https://www.classroomopen.com/>
- <https://www.udemy.com>

Documentation Python :

- <https://www.python.org/>
- <https://www.coursera.org>
- <https://www.w3schools.com>
- <https://www.codecademy.com>

Livres :

- UML : Support de Cours de Mme. MOULINE Salma
- Base de Données : Support de Cours de Mr. BENKHALIFA Mohammed
- Zeste de savoir développez votre site avec le Framework Django
- Apprendre à programmer avec Python3