



République Tunisienne
Ministère de l'Enseignement
Supérieur et de la Recherche
Scientifique

Université de Tunis El Manar

Faculté des Sciences
Économiques et de Gestion de
Tunis



PROJET TUTORÉ

Master Professionnel en Ingénierie des Systèmes
d'Information et Data Science

Mise en place d'une Application Web
d'analyse de la Santé Mentale

Élaboré par :

EL HAJ ALI Khaled

SANNEN Essia

SAIDANI Emna

NASRI Hadir

AMARA Nour

MAAMER Rania

Sous la direction de :

JELASSI Nidhal

Année universitaire : 2025–2026

Table des matières

1	Introduction	4
2	Contexte	5
2.1	Vision du projet	5
2.2	Objectifs	5
2.3	Besoins fonctionnels	5
2.4	Besoins non fonctionnels	6
3	Analyse du diagramme de cas d'utilisation	7
3.1	Acteurs du système	7
3.2	Diagramme de cas d'utilisation	8
3.3	Analyse	8
4	Frameworks et Technologies utilisées	9
4.1	Architecture générale	9
4.2	Frontend	9
4.3	Backend	10
4.4	Visualisation	10
4.5	Base de données	11
5	Product Backlog et User Stories	12
5.1	Présentation du Product Backlog	12
5.2	Gestion des comptes	12
5.3	Fonctionnalités du journal	12
5.4	Analyse intelligente	13
6	Plannification	14
6.1	Sprints	14
7	Sécurité et confidentialité	15
8	Expérience utilisateur	16

9	Analyse critique	17
9.1	Points forts	17
9.2	Perspectives	17
10	Conclusion	18

Chapitre 1

Introduction

La santé mentale constitue aujourd’hui un enjeu majeur dans les sociétés modernes. L’augmentation du stress, de l’anxiété et de la pression académique et professionnelle impacte directement le bien-être psychologique des individus.

Avec la généralisation des technologies numériques, de nouvelles solutions émergent pour accompagner les personnes dans la gestion de leur santé mentale. Parmi ces solutions, les applications de journalisation personnelle occupent une place importante, car elles permettent une expression libre des émotions et des pensées.

Le projet **Mental Well-being Web App** s’inscrit dans cette dynamique en proposant une application web sécurisée et légère permettant aux utilisateurs de tenir un journal personnel privé, enrichi par une analyse automatique des sentiments. L’objectif est d’offrir un retour immédiat et visuel sur l’état émotionnel de l’utilisateur, sans recourir à des modèles lourds ou coûteux en ressources.

Ce rapport présente une analyse complète du projet, depuis la définition des besoins jusqu’à la conception technique, en passant par l’organisation agile, les choix technologiques et les perspectives d’évolution.

Chapitre 2

Contexte

2.1 Vision du projet

La vision du projet **Mise en place d’une Application Web d’analyse de la Santé Mentale** est de concevoir une application web simple, intuitive et sécurisée permettant à chaque utilisateur de suivre son bien-être mental à travers l’écriture quotidienne.

Contrairement aux applications complexes ou médicalisées, cette solution se veut accessible à tous, centrée sur l’auto-réflexion et la prise de conscience émotionnelle.

2.2 Objectifs

Les objectifs principaux du projet sont :

- Offrir un espace d’écriture personnel et privé
- Analyser automatiquement le sentiment des textes rédigés
- Visualiser l’évolution émotionnelle dans le temps
- Garantir la confidentialité et la sécurité des données

2.3 Besoins fonctionnels

Les besoins fonctionnels couvrent :

- Gestion des comptes utilisateurs
- Journalisation des entrées
- Analyse de sentiment automatisée
- Tableau de bord avec visualisations
- Fonctionnalités avancées (export, rappels – futures)

2.4 Besoins non fonctionnels

Les besoins non fonctionnels incluent :

- Sécurité et confidentialité des données
- Performance et réactivité de l'application
- Accessibilité sur divers appareils
- Scalabilité pour gérer la croissance des utilisateurs

Chapitre 3

Analyse du diagramme de cas d'utilisation

3.1 Acteurs du système

Le diagramme ci-dessous illustre les acteurs principaux de l'application et leurs interactions avec le système. Chaque acteur représente un type d'utilisateur avec des droits et des responsabilités spécifiques.

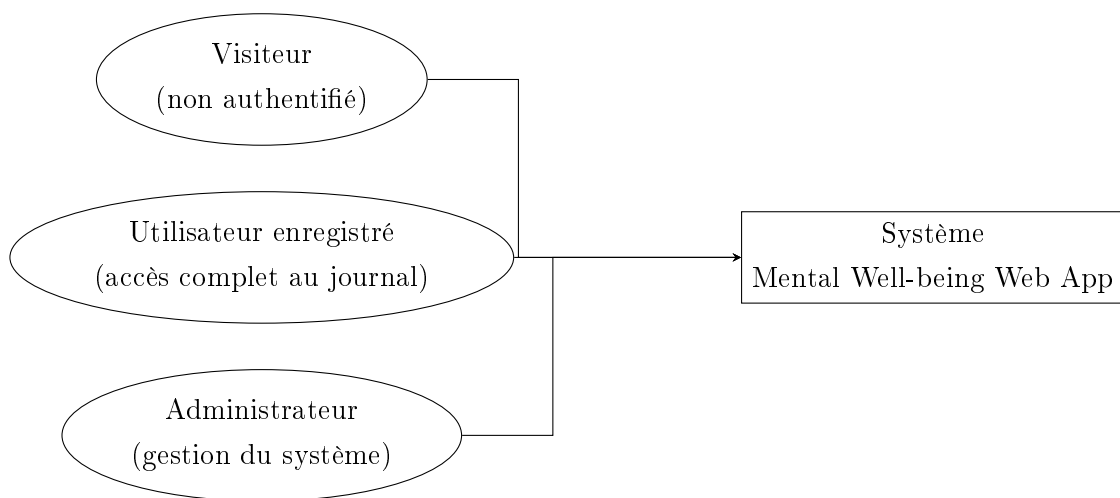


FIGURE 3.1 – Diagramme des acteurs du système

Ce diagramme de cas d'utilisation montre les différents types d'utilisateurs (Visiteur, Utilisateur enregistré, Administrateur) et leurs interactions avec le système.

- Le *Visiteur* peut consulter des informations publiques sans être connecté.
- L'*Utilisateur enregistré* a accès complet à son journal et aux fonctionnalités principales.
- L'*Administrateur* gère l'ensemble du système et supervise les utilisateurs et leurs données.

3.2 Diagramme de cas d'utilisation

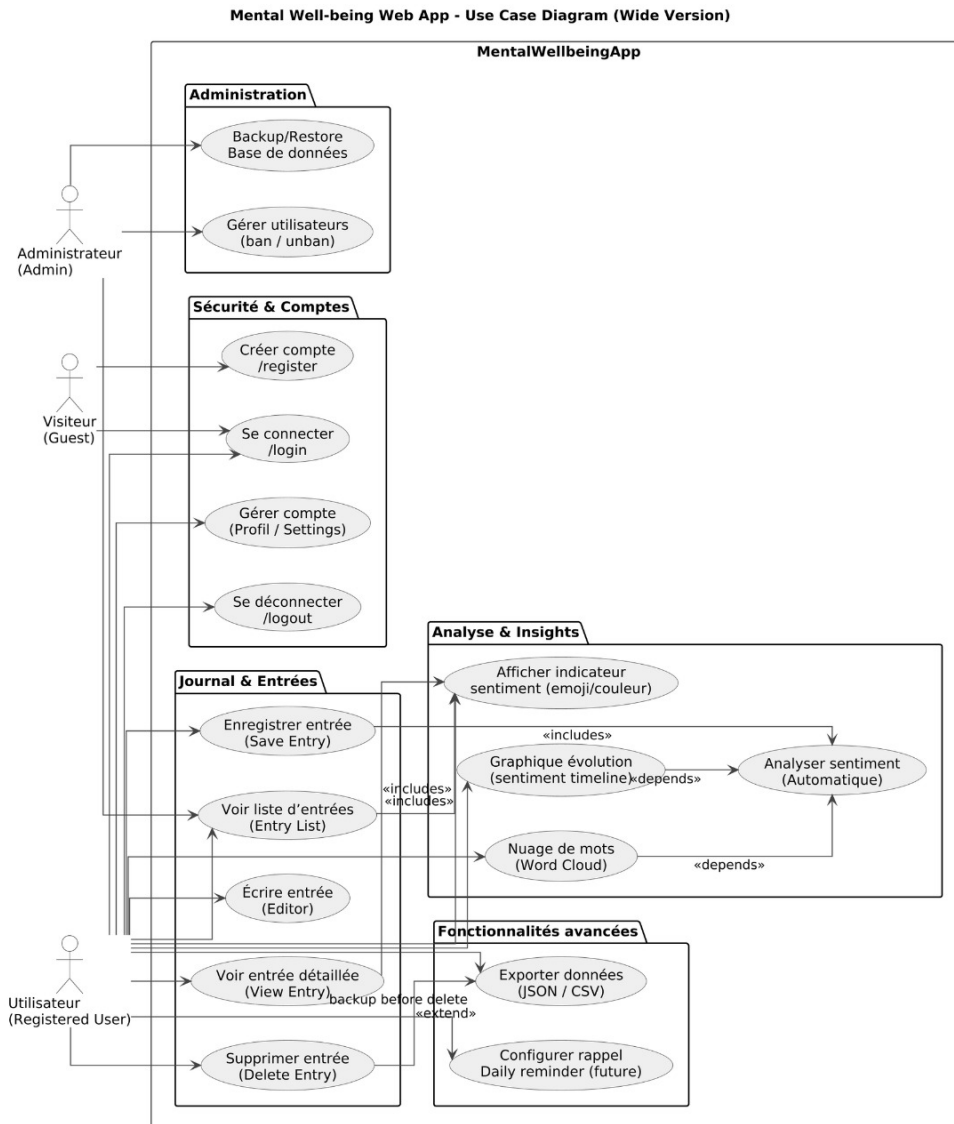


FIGURE 3.2 – Diagramme de cas d'utilisation de l'application

3.3 Analyse

Le diagramme met en évidence une séparation claire des responsabilités. Les fonctionnalités critiques (journal, analyse, visualisation) sont réservées aux utilisateurs authentifiés, garantissant la confidentialité des données.

Chapitre 4

Frameworks et Technologies utilisées

4.1 Architecture générale

Le projet repose sur une architecture client–serveur séparant clairement le frontend et le backend. Cette organisation améliore la maintenabilité, l'évolutivité et la clarté du système. Le frontend gère l'interface utilisateur, tandis que le backend assure le traitement des données, la sécurité, la persistance et l'intelligence artificielle, avec une communication basée sur des API REST au format JSON.

4.2 Frontend



Next.js 14

Le frontend est développé avec **Next.js 14**, un framework basé sur React offrant de hautes performances grâce à la combinaison du rendu côté serveur et côté client. Il permet une structuration claire des composants, une navigation fluide entre les différentes pages et une communication efficace avec les API du backend Flask, améliorant ainsi l'expérience utilisateur et les temps de chargement.

Next.js permet notamment :

- Une interface réactive
- Un rendu performant
- Une navigation fluide

L'utilisation de ce framework facilite également l'intégration de bibliothèques tierces telles que Chart.js et contribue à une expérience utilisateur moderne, responsive et cohérente sur différents types d'appareils.

4.3 Backend



Flask

Le backend est développé avec **Flask**, un micro-framework Python léger et flexible, adapté à la création d'API REST. Il centralise la logique métier du système, notamment l'authentification, la gestion des sessions, le stockage des données et l'analyse de sentiment, tout en assurant une communication sécurisée et structurée avec le frontend.

Le backend **Flask** prend en charge :

- La gestion des utilisateurs
- La gestion des entrées de journal
- L'analyse de sentiment

Ce choix technologique permet également une intégration naturelle avec les bibliothèques de traitement du langage naturel (TextBlob, VADER, spaCy), renforçant ainsi la dimension intelligente de l'application.

4.4 Visualisation



Chart.js

La visualisation des données dans le projet est assurée par **Chart.js**, une bibliothèque JavaScript qui permet de créer des graphiques interactifs et dynamiques.

Grâce à **Chart.js**, le système peut afficher :

- Des graphiques d'évolution des émotions au fil du temps
- Des tableaux de bord synthétiques
- Des nuages de mots basés sur les entrées du journal

Cette visualisation permet à l'utilisateur de mieux comprendre ses tendances émotionnelles et d'analyser ses données de manière intuitive et interactive.

4.5 Base de données



SQLite

La base de données utilisée est **SQLite**, une base de données relationnelle légère intégrée directement à l'application. Ce choix est particulièrement adapté à un prototype ou à un MVP, car il ne nécessite pas de serveur de base de données dédié et offre une grande simplicité de déploiement.

SQLite permet de stocker de manière structurée et sécurisée les données essentielles de l'application tout en garantissant de bonnes performances pour un volume modéré de données.

La base de données **SQLite** stocke :

- Les utilisateurs
- Les entrées de journal
- Les scores de sentiment

Ce choix constitue une base solide pour le projet, avec la possibilité d'une migration future vers une solution plus robuste (comme PostgreSQL) si le volume d'utilisateurs venait à augmenter.

Chapitre 5

Product Backlog et User Stories

5.1 Présentation du Product Backlog

Le Product Backlog regroupe l'ensemble des fonctionnalités attendues du système, classées par priorité et organisées sous forme de User Stories. Il sert de référence pour l'équipe de développement et permet de planifier les sprints en fonction des besoins utilisateurs et des contraintes techniques.

5.2 Gestion des comptes

Les User Stories US-1 à US-3 couvrent l'inscription, la connexion et la déconnexion, assurant un accès sécurisé aux données personnelles. Ces fonctionnalités garantissent :

- La création et la gestion sécurisée des comptes utilisateurs
- La protection des informations sensibles
- Une expérience utilisateur fluide et intuitive

5.3 Fonctionnalités du journal

Les User Stories US-4 à US-6 constituent le cœur fonctionnel du projet, permettant l'écriture, la sauvegarde et la consultation des entrées. Ces fonctionnalités offrent :

- La possibilité de noter ses expériences et émotions quotidiennes
- Un stockage fiable et structuré des données
- Un accès facile à l'historique pour suivi et analyse

5.4 Analyse intelligente

Les User Stories US-7 et US-8 introduisent l'intelligence artificielle légère via l'analyse automatique des sentiments et leur visualisation. Elles permettent :

- Une compréhension rapide des tendances émotionnelles
- Des représentations visuelles claires grâce à des graphiques et nuages de mots
- Une aide à la prise de conscience personnelle pour l'utilisateur

5.5 Product Backlog

Le Product Backlog présente de manière synthétique l'ensemble des User Stories du projet, classées par priorité et organisées selon les grands thèmes fonctionnels. Il constitue un outil central de pilotage du projet agile et permet d'assurer une vision globale des fonctionnalités à développer.

ID	Priorité	Description de la User Story
US-1	Élevée	Inscription sécurisée des utilisateurs avec chiffrement des mots de passe.
US-2	Élevée	Connexion sécurisée permettant l'accès au tableau de bord personnel.
US-3	Moyenne	Déconnexion de l'utilisateur afin de garantir la sécurité des données.
US-4	Élevée	Mise à disposition d'un éditeur de texte pour l'écriture du journal.
US-5	Élevée	Sauvegarde des entrées du journal avec horodatage.
US-6	Élevée	Consultation chronologique des entrées précédentes.
US-7	Élevée	Analyse automatique du sentiment des textes enregistrés.
US-8	Moyenne	Affichage d'indicateurs visuels représentant le sentiment associé.
US-9	Élevée	Visualisation de l'évolution émotionnelle via un graphique linéaire.
US-10	Moyenne	Génération d'un nuage de mots à partir des entrées du journal.

TABLE 5.1 – Product Backlog du projet Mental Well-being Web App

Chapitre 6

Plannification

6.1 Sprints

Le tableau ci-dessous présente l'organisation du projet selon une approche agile basée sur des sprints successifs. Chaque sprint correspond à une phase clé du développement et regroupe des objectifs précis. Cette organisation permet une progression graduelle du projet, depuis la mise en place des bases techniques jusqu'à la finalisation du MVP.

Sprint	Objectifs / Contenu
Sprint 1	Établir les bases du système : authentification sécurisée, écriture et sauvegarde des entrées du journal, mise en place de l'architecture frontend-backend.
Sprint 2	Introduction de l'analyse automatique des sentiments et de l'historique émotionnel : traitement des textes, génération de scores de sentiment et affichage des tendances dans l'historique utilisateur.
Sprint 3	Finalisation du MVP : création du tableau de bord interactif, intégration des graphiques d'évolution des émotions et génération de nuages de mots pour visualiser les données de manière synthétique.

TABLE 6.1 – Organisation du projet en sprints et objectifs associés

Chapitre 7

Sécurité et confidentialité

La sécurité constitue un axe fondamental du projet, garantissant la protection des données personnelles et la confiance des utilisateurs.

Chiffrement des mots de passe : Tous les mots de passe sont sécurisés avec l'algorithme **bcrypt** avant stockage. Bcrypt offre une grande résistance aux attaques par force brute grâce au salage et à un temps de calcul ajustable, garantissant que les mots de passe restent illisibles même en cas de fuite de la base de données.

Gestion des sessions : Les sessions utilisateurs sont gérées via **Flask-Login**, assurant une authentification fiable et la limitation de l'accès aux données sensibles. Chaque session est unique et expire automatiquement après une période d'inactivité pour prévenir tout accès non autorisé.

Chiffrement des données sensibles : Une évolution future prévoit l'intégration d'un chiffrement **AES (Advanced Encryption Standard)** pour protéger les informations critiques et renforcer la confidentialité des données stockées.

Bonnes pratiques : Des mesures complémentaires sont également mises en place, telles que :

- Validation des entrées utilisateurs pour éviter les injections SQL et attaques XSS.
- Protection des formulaires contre les attaques CSRF.
- Limitation des tentatives de connexion pour prévenir les intrusions.
- Journalisation des accès et actions critiques pour détecter tout comportement suspect.

Bénéfices pour l'utilisateur : Cette approche multi-niveaux permet aux utilisateurs de saisir leurs informations personnelles et émotionnelles en toute confiance, avec une expérience à la fois simple, fluide et sécurisée.

En combinant chiffrement des mots de passe, gestion sécurisée des sessions et protections avancées, le projet place la sécurité au cœur de sa conception, garantissant ainsi une utilisation fiable et protégée.

Chapitre 8

Expérience utilisateur

L'interface utilisateur est conçue pour être à la fois minimaliste et intuitive, afin de faciliter la prise en main pour tout type d'utilisateur, même ceux ayant peu d'expérience avec les applications web. Chaque composant a été pensé pour réduire la complexité et éviter la surcharge d'informations.

Les composants **Tailwind CSS** et **Shadcn/UI** garantissent une expérience cohérente, responsive et esthétiquement agréable sur tous les types d'appareils, que ce soit sur ordinateur, tablette ou smartphone. Grâce à leur intégration, l'interface offre :

- Des formulaires clairs et faciles à remplir pour les entrées de journal
- Une navigation fluide entre les différentes sections de l'application
- Des boutons, menus et éléments interactifs uniformes et facilement identifiables

Les indicateurs visuels, tels que les couleurs, icônes et emojis, permettent une lecture rapide et intuitive de l'état émotionnel de l'utilisateur. Par exemple :

- Des couleurs chaudes pour les émotions positives et des couleurs froides pour les émotions négatives
- Des emojis ou pictogrammes associés à chaque type d'émotion pour une interprétation immédiate
- Des graphiques simples et clairs, facilement compréhensibles, même pour un utilisateur novice

Cette approche visuelle favorise l'engagement de l'utilisateur et facilite la compréhension des tendances émotionnelles sur le long terme. De plus, l'interface est conçue pour être évolutive, permettant l'ajout futur de nouvelles fonctionnalités sans compromettre la simplicité et la clarté de l'expérience utilisateur.

Enfin, chaque choix de design a été guidé par la volonté de créer une application accessible, agréable et efficace, combinant esthétique moderne et ergonomie optimale, ce qui contribue à la satisfaction et à la fidélisation des utilisateurs.

Chapitre 9

Analyse critique

9.1 Points forts

Le projet se distingue par :

- Sa simplicité, permettant une prise en main rapide pour les utilisateurs et un déploiement facile.
- Son approche légère en IA, utilisant des modèles accessibles qui garantissent un traitement rapide des données sans surcharger le système.
- Son respect de la confidentialité, avec un stockage local sécurisé des données et aucune dépendance à des services externes pour l'analyse des entrées de journal.

9.2 Perspectives

Les limites incluent :

- Une analyse de sentiment basique, qui ne prend pas encore en compte certaines nuances émotionnelles ou contextuelles.
- Une base SQLite peu scalable, adaptée aux projets de petite à moyenne taille mais limitée si le volume de données ou le nombre d'utilisateurs augmente fortement.

Chapitre 10

Conclusion

Le projet **Mental Well-being Web App** répond pleinement aux objectifs définis dans le cahier des charges. Il propose une solution fonctionnelle, sécurisée et intelligente pour le suivi du bien-être mental, combinant une interface utilisateur intuitive avec des fonctionnalités d'analyse émotionnelle fiables.

Ce travail constitue une base solide pour des évolutions futures et illustre la pertinence de l'utilisation d'une intelligence artificielle légère dans des applications à impact humain. Il démontre également l'efficacité d'une architecture web moderne, intégrant des technologies comme **Next.js**, **Flask** et **Chart.js**, pour offrir une expérience fluide et engageante.

Les perspectives d'évolution comprennent :

- **Migration vers PostgreSQL** pour améliorer la scalabilité et la robustesse de la base de données.
- **Utilisation de modèles NLP avancés** afin d'obtenir une analyse plus fine et contextuelle des sentiments.
- **Développement d'une application mobile** pour offrir un suivi du bien-être accessible à tout moment.
- **Notifications intelligentes** pour alerter et accompagner l'utilisateur de manière personnalisée selon son état émotionnel.

Dans l'ensemble, le projet illustre la capacité d'un système léger et bien conçu à répondre aux besoins des utilisateurs tout en restant évolutif et performant.