



REPUBLIQUE TUNISIENNE
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
ÉCOLE SUPERIEURE PRIVÉE DE TECHNOLOGIES ET INGENIERIE
TEK-UP



Rapport de Projet BI

NSM VISION

Elaboré par

Mohamed Slim Chouaib
Nour Houda Zaabi
Mohamed Ali Bouhadja

Encadré Par

Madame. Aicha benjrad
Mr. Abd Laaziz Menchari

Année Universitaire

2024-2025

Remerciements

Tout d'abord, nous souhaitons exprimer notre gratitude à tous les membres du jury pour leur présence et leur évaluation de notre travail.

Nous désirons témoigner notre appréciation aussi bien envers notre professeur, Mme Aicha BEN JRAD, pour son encadrement, son soutien infaillible, sa patience inestimable et sa confiance en notre équipe tout au long de ce projet. Les suggestions et les commentaires qu'elle nous a apportés ont été précieux pour mener à bien ce rapport.

Enfin, nous désirons manifester notre profonde reconnaissance envers notre professeur, M. Abdelaziz MENCHARI, pour son expertise, ses conseils précieux ainsi que son soutien généreux. Nous sommes particulièrement reconnaissants du temps qu'il a consacré à discuter et à interpréter les résultats de notre étude, ce qui a grandement contribué à la qualité de notre rapport.

Introduction générale

L'évolution rapide des technologies numériques et la généralisation de l'usage des données dans la prise de décision ont transformé en profondeur les pratiques des entreprises, notamment dans le secteur financier. Aujourd'hui, les sociétés de financement font face à une nécessité croissante d'automatiser l'analyse de données, d'évaluer les risques avec précision, et de disposer d'indicateurs fiables pour piloter leurs activités. C'est dans ce contexte que s'inscrit notre projet **NSM VISION**, qui consiste à développer une solution intelligente dédiée à l'optimisation de la gestion du crédit pour les petites et moyennes entreprises financières.

Le projet repose sur un écosystème technologique moderne et performant. Le *backend* est développé en **Node.js**, garantissant une architecture asynchrone et évolutive, tandis que le *frontend* repose sur **React** avec **Tailwind CSS**, assurant une interface utilisateur dynamique, réactive et ergonomique. Le stockage des données est pris en charge par **MongoDB**, une base *NoSQL* adaptée aux structures de données flexibles et évolutives. Pour l'analyse des performances et la visualisation des données clés, nous avons intégré des tableaux de bord interactifs via **Power BI**.

Afin d'enrichir la plateforme et d'automatiser certaines tâches critiques, plusieurs modules intelligents ont été développés. Un système de prédiction basé sur le *machine learning* permet d'anticiper les risques de défaut de paiement et d'améliorer la qualité des décisions de crédit. Des analyses statistiques avancées et des indicateurs dynamiques aident également les gestionnaires à suivre les tendances du portefeuille client en temps réel.

L'ensemble du développement a été mené selon la méthodologie **Agile Scrum**, découplant le projet en plusieurs sprints afin de garantir une planification claire, un suivi régulier de l'avancement, une adaptation rapide aux retours et une livraison progressive des fonctionnalités. Cette démarche a permis de structurer efficacement notre travail tout en maintenant une grande flexibilité dans l'intégration des exigences fonctionnelles et techniques.

Table des matières

1 Cadre du projet	6
1.1 Contexte général	6
1.2 Problématique	6
1.3 Étude de l'existant	6
1.4 Étude de la concurrence	7
1.5 Solution proposée	7
1.6 Choix méthodologiques	7
1.6.1 Étude comparative entre les méthodes traditionnelles et Agile Scrum	8
1.6.2 Méthodologie Agile	8
1.6.3 Scrum	8
1.6.4 Application au projet NSM Solution	9
1.7 Conclusion	9
2 Analyse et spécification des besoins	11
2.1 Étude fonctionnelle	11
2.1.1 Identification des acteurs	11
2.1.2 Besoins fonctionnels	12
2.1.3 Besoins non fonctionnels	12
2.1.4 Contraintes du projet	12
2.2 Planification et structuration	12
2.2.1 Backlog produit	13
2.2.2 Planification des sprints	14
2.3 Spécifications techniques	14
2.3.1 Technologies utilisées	14
2.4 Spécifications techniques	14
2.4.1 Technologies utilisées	14
2.4.2 Outils utilisés	15
2.5 Spécifications générales des exigences	15
2.5.1 Diagramme de cas d'utilisation global	15
2.5.2 Diagramme de classes global	17
2.6 Architecture logicielle	17
2.6.1 Couche de présentation (Frontend)	17
2.6.2 Couche de logique métier (Backend)	17
2.6.3 Couche de données	17
2.6.4 Composants transverses	18
2.6.5 Schéma simplifié des interactions	18
3 Sprint 1 : Mise en place des fonctionnalités essentielles	19
3.1 Backlog du Sprint 1	19
3.2 Réalisation	19
3.3 Conclusion	20

4 Sprint 2 : Gestion des demandes de crédit	22
4.1 Réalisation	22
4.2 Conclusion	22
5 Sprint 3 :Visualisation et décision	24
5.1 Introduction	24
5.2 Backlog du Sprint 3	24
5.3 Réalisation	24
5.3.1 Intégration d'un Chatbot Intelligent	25
5.4 Conclusion	27
6 Sprint 4 : Visualisation et analyses	28
6.1 Backlog du Sprint 4	28
6.2 Réalisation	28

Chapitre 1

Cadre du projet

Introduction

Dans le contexte actuel de digitalisation du secteur financier, les petites et moyennes sociétés de financement ont besoin de solutions efficaces pour automatiser l'analyse des demandes de crédit. Le projet *NSM Solution* répond à ce besoin à travers le développement d'un site web interactif dédié à la gestion intelligente du crédit. Ce site intègre un modèle de machine learning capable de prédire l'approbation ou le refus d'une demande de prêt, ainsi qu'un chatbot permettant une interaction naturelle et intuitive avec le système.

1.1 Contexte général

L'analyse des risques liés au crédit repose traditionnellement sur des processus manuels et chronophages. Dans un objectif d'optimisation, NSM Solution propose un site web local qui permet de traiter les demandes de crédit de manière automatique et rapide. Le site offre une interface utilisateur accessible, enrichie par un chatbot intégré, et s'appuie sur des algorithmes de machine learning pour fournir des décisions prédictives basées sur les données historiques. Ce projet vise à rendre les outils d'analyse avancée accessibles aux PME .

1.2 Problématique

Les entreprises de financement de petite et moyenne taille sont confrontées à plusieurs défis :

- Comment automatiser l'évaluation des demandes de crédit de manière fiable ?
- Comment tirer parti des données historiques pour entraîner un modèle prédictif pertinent ?
- Comment rendre l'interaction avec le système simple et efficace à travers un site web ?
- Comment intégrer un chatbot dans un environnement web local pour améliorer l'expérience utilisateur ?

Le projet NSM Solution vise à résoudre ces problématiques par une solution web intelligente, locale et intuitive.

1.3 Étude de l'existant

Les outils classiques d'analyse de crédit, comme FICO ou Experian, sont souvent inadaptés aux PME, car ils sont coûteux, complexes ou dépendants d'infrastructures distantes. D'autres solutions comme Excel ou les systèmes ERP standards ne permettent pas une automatisation

avancée ni une visualisation efficace des données. NSM Solution propose une alternative locale, moderne et spécifique, intégrant intelligence artificielle, visualisation via Power BI, et chatbot intelligent.

1.4 Étude de la concurrence

Les solutions concurrentes misent principalement sur des plateformes cloud intégrées et des modèles pré-entraînés. Peu d'entre elles ciblent les PME ou intègrent un chatbot directement dans une solution locale. En revanche, NSM Solution offre une prédiction sur mesure, s'appuie sur un front-end moderne, tout en offrant des outils analytiques puissants comme Power BI pour la prise de décision.

1.5 Solution proposée

Pour répondre à ces enjeux, nous avons développé un site web de gestion de crédit intelligent, structuré autour de trois axes principaux :

1. Développement web moderne :

- Frontend développé avec React.js et Tailwind CSS pour une interface fluide, responsive et personnalisée,
- Backend construit avec Node.js pour gérer la logique métier, les requêtes des utilisateurs et la communication avec le modèle IA,
- Base de données MongoDB pour stocker les utilisateurs, les demandes, les historiques de décision et les résultats de prédiction.

2. Intelligence artificielle pour la prédiction :

- Entraînement d'un modèle de machine learning (Random Forest) basé sur des données historiques de demandes de crédit,
- Intégration du modèle dans le backend pour une exécution en temps réel,
- Résultats exploitables par les gestionnaires pour automatiser la décision d'octroi ou de refus de crédit.

3. Chatbot intégré pour l'interaction :

- Chatbot accessible via le site web, développé pour fournir des réponses contextuelles aux utilisateurs,
- Connecté au backend pour fournir des informations dynamiques et pertinentes,
- Contribue à moderniser l'expérience utilisateur sur la plateforme.

Enfin, la visualisation des indicateurs clés (statistiques des crédits, taux d'acceptation, répartition par profil client, etc.) est assurée via des tableaux de bord Power BI, facilitant la lecture des résultats pour les décideurs.

1.6 Choix méthodologiques

Dans cette section, nous allons faire une étude comparative entre les méthodes traditionnelles et Agile Scrum. Ensuite, nous allons définir la méthodologie de travail que nous avons adoptée dans ce projet.

1.6.1 Étude comparative entre les méthodes traditionnelles et Agile Scrum

Pour justifier notre choix de la méthode Agile Scrum, il est important de comparer brièvement ses caractéristiques avec celles des méthodes traditionnelles (comme le cycle en V ou la méthode en cascade). La Table 1.1 explique la différence entre les méthodes traditionnelles et Agile Scrum.

TABLE 1.1 – Comparaison entre méthodes traditionnelles et Agile Scrum

Critère	Méthodes traditionnelles	Méthode Agile Scrum
Approche	Séquentielle (phases fixes)	Itérative et incrémentale
Gestion des changements	Difficile à intégrer en cours de projet	Facile grâce aux sprints flexibles
Interaction avec le client	Faible, principalement en début et fin de projet	Forte, avec une implication continue
Livraison	Unique à la fin du projet	Fréquente à chaque sprint
Documentation	Très formelle et détaillée	Allégée, au profit de la communication
Adaptabilité	Faible, rigide face aux imprévus	Élevée, réactive aux besoins changeants

1.6.2 Méthodologie Agile

La méthodologie Agile se différencie généralement des méthodes traditionnelles de gestion de projet en mettant l'accent sur les exigences du client et en encourageant la communication entre toutes les parties prenantes. Développée à l'origine pour les projets logiciels, cette méthode gagne aujourd'hui en popularité grâce à sa flexibilité et son adaptabilité à divers types de projets.

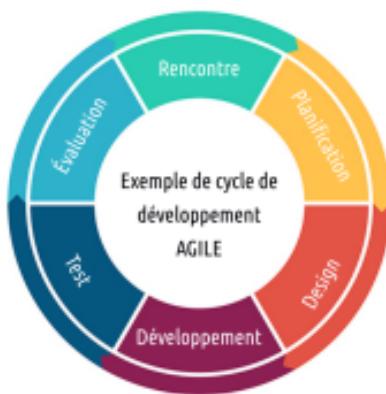


FIGURE 1.1 – Méthodologie Agile : cycle de développement

1.6.3 Scrum

Scrum est l'une des méthodes de gestion de projet Agile, initialement conçue pour le développement de logiciels. Elle favorise la collaboration, la communication régulière, et le travail en équipe. Scrum consiste à découper le projet en sprints courts avec des objectifs précis à atteindre dans un temps limité.

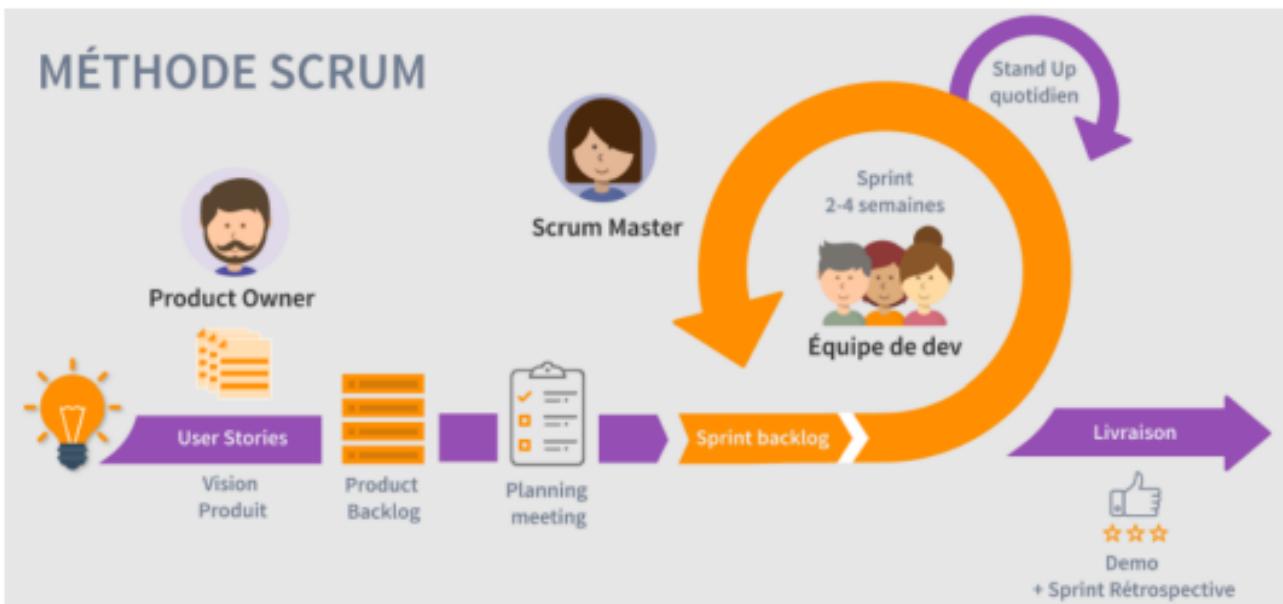


FIGURE 1.2 – Itération selon la méthode Scrum

Les trois types d'intervenants dans Scrum :

- **Product Owner** : responsable de maximiser la valeur du produit et de gérer le backlog.
- **Scrum Master** : responsable du bon déroulement du processus Scrum. Il anime l'équipe et résout les obstacles.
- **Scrum Team** : composée de développeurs, designers et autres membres techniques qui réalisent concrètement le projet.

Les événements Scrum :

- **Sprint** : période de développement (souvent 2 à 4 semaines).
- **Sprint Planning** : planification des tâches au début de chaque sprint.
- **Daily Scrum** : réunion quotidienne de 15 minutes pour synchroniser l'équipe.
- **Sprint Review** : démonstration des résultats à la fin de chaque sprint.
- **Sprint Retrospective** : analyse de l'organisation pour améliorer les sprints suivants.

1.6.4 Application au projet NSM Solution

Dans le cadre de notre projet, nous avons appliqué la méthode Scrum selon quatre sprints :

1. **Sprint 1** : Mise en place du backend avec Node.js et MongoDB ;
2. **Sprint 2** : Développement du frontend en React avec Tailwind CSS ;
3. **Sprint 3** : Intégration du modèle prédictif (machine learning) et du chatbot ;
4. **Sprint 4** : Création des tableaux de bord Power BI et phase de tests.

Chaque sprint a été ponctué par une rétrospective, permettant de revoir les avancées et d'ajuster les priorités pour la suite.

1.7 Conclusion

Dans ce premier chapitre, nous avons présenté le cadre général du projet, en partant de la problématique identifiée jusqu'à la solution envisagée. Nous avons également analysé l'état de

l'existant afin d'identifier les insuffisances, puis détaillé les choix technologiques et méthodologiques. Le recours à la méthode Agile, et plus particulièrement à Scrum, nous a permis de structurer notre approche de manière itérative, en intégrant une flexibilité adaptée aux besoins évolutifs du projet.

Cette base conceptuelle et méthodologique nous permet d'aborder l'analyse fonctionnelle du système avec rigueur, en posant les fondations techniques et organisationnelles du reste du projet.

Chapitre 2

Analyse et spécification des besoins

Introduction

L'analyse et la spécification des besoins constituent une étape cruciale dans le cycle de développement logiciel. Ce chapitre a pour objectif de présenter de manière détaillée les besoins fonctionnels et non fonctionnels du site web de gestion de crédit *NSM Solution*, d'identifier les différents acteurs impliqués, et de structurer les exigences sous forme de backlog produit. Il introduit également les technologies et outils sélectionnés, l'architecture du projet, ainsi que les modèles de spécification générale tels que les cas d'utilisation et la modélisation de la base de données. Cette analyse approfondie permet d'assurer une conception cohérente, robuste et en adéquation avec les objectifs métiers et les attentes des utilisateurs finaux.

2.1 Étude fonctionnelle

Nous décrivons ici les acteurs de notre site web, les fonctionnalités attendues (consultation de demandes, traitement, prédiction, visualisation), ainsi que les exigences non fonctionnelles (performance, sécurité, compatibilité, etc.).

2.1.1 Identification des acteurs

Dans le cadre du projet *NSM Solution*, nous avons identifié quatre principaux types d'acteurs interagissant avec le système :

- **Décideurs (Gestionnaires de Crédit)** : Utilisateurs principaux qui prennent des décisions basées sur les données fournies par le système.
 - Consulter les tableaux de bord.
 - Analyser les risques.
 - Approuver ou refuser des crédits.
- **Analystes Financiers** : Professionnels qui analysent les données financières et préparent des rapports.
 - Générer des rapports.
 - Effectuer des analyses de risques.
 - Segmenter les clients.
- **Administrateurs Système** : Responsables de la maintenance et de la configuration du système.
 - Gérer les utilisateurs.

- Configurer les paramètres du système.
- Assurer la sécurité des données.
- **Client** : Entreprises ou institutions utilisant la plateforme pour gérer les crédits.
 - Soumettre des demandes de crédit.
 - Consulter l'état de leurs demandes.
 - Accéder aux rapports et analyses financières.

2.1.2 Besoins fonctionnels

Les besoins fonctionnels décrivent les principales fonctionnalités du système. Voici les modules à intégrer :

- **Inscription et Connexion** :
 - Authentification sécurisée des utilisateurs ;
 - Attribution de rôles selon les profils.
- **Analyse des Demandes de Crédit** :
 - Consultation des demandes et leurs statut (approuvée, refusée, en cours) ;
 - Prédiction automatique de l'acceptation d'une demande via un modèle ML.
- **Visualisation des données (BI)** :
 - Tableaux de bord Power BI affichant indicateurs clés (volume de crédits, taux d'acceptation, etc.) ;
 - Filtres dynamiques selon période, région, ou catégorie client.
- **Interaction via Chatbot** :
 - Consultation rapide d'indicateurs ou d'états de crédit via un agent conversationnel.

2.1.3 Besoins non fonctionnels

Les exigences non fonctionnelles garantissent la qualité du système :

- **Performance** : temps de réponse optimal même sur de grands volumes de données.
- **Sécurité** : gestion rigoureuse des accès, protection des données sensibles.
- **Compatibilité** : intégration avec Power BI, Node.js, et MongoDB.
- **Scalabilité et fiabilité** : la solution doit évoluer sans perte de performance.

2.1.4 Contraintes du projet

Le projet est soumis aux contraintes suivantes :

- **Coût** : recours prioritaire à des technologies open-source ou déjà disponibles. **Ressources** : utilisation de Power BI, Node.js, React, Python, MongoDB, et outils d'IA/ML en local.

2.2 Planification et structuration

Dans cette partie nous nous intéressons au backlog produit et à la planification des sprints de notre projet.

2.2.1 Backlog produit

Dans cette partie, nous élaborons un backlog produit initial pour prioriser les User Stories selon leur valeur métier. La table 2.1 nous illustre le backlog produit global. Chaque User Story est évaluée selon la suite de Fibonacci : 1 (très simple), 2 (simple), 3 (normale), 5 (difficile), 8 (très difficile), 13 (complexe).

TABLE 2.1 – User Stories principales par acteur

ID	User Story	Priorité	Com.	Critère d'acceptation
US01	En tant qu' Administrateur , je veux gérer les utilisateurs.	Moyenne	5	Accès à la liste des utilisateurs → activer/désactiver, modifier les rôles.
US02	En tant qu' Administrateur , je veux surveiller les connexions suspectes.	Haute	5	Historique de connexions + alertes en cas d'activité inhabituelle (IP, heure, échec de connexion).
US03	En tant qu' Administrateur , je veux connecter la plateforme à Power BI.	Haute	5	Connexion établie avec Power BI → données visibles depuis l'interface Power BI (vérification des jeux de données).
US04	En tant que Client , je veux soumettre une demande de crédit.	Haute	5	Formulaire envoyé → demande enregistrée dans la base + notification de réception.
US05	En tant que Client , je veux consulter l'état de ma demande.	Haute	5	Accès à une section "Suivi" affichant le statut de la demande (en cours, acceptée, refusée).
US06	En tant que Décideur , je veux approuver ou refuser les demandes.	Haute	5	Liste des demandes à traiter + boutons "Approuver"/"Refuser" → mise à jour du statut + justification possible.
US07	En tant que Décideur , je veux visualiser les graphiques Power BI.	Haute	8	Accès au dashboard Power BI : KPIs sur crédits accordés, délais de traitement, taux d'acceptation.
US08	En tant qu' Analyste , je veux générer des rapports de risques.	Moyenne	5	Génération automatique ou manuelle d'un rapport PDF avec résumé des indicateurs de risque.
US09	En tant qu' Analyste , je veux charger et traiter les données financières.	Moyenne	5	Téléversement de fichiers CSV/Excel → affichage des données traitées dans un tableau.
US10	En tant que Système (IA) , je veux calculer le scoring client.	Haute	8	Envoi des données → prédition (score affiché + justification du modèle via SHAP).

2.2.2 Planification des sprints

Nous avons adopté la méthodologie Agile Scrum et réparti le travail de développement en quatre sprints. Cette section présente le découpage temporel, les objectifs de chaque sprint ainsi que les User Stories sélectionnées.

La table 2.2 illustre la planification des sprints selon l'ordre fonctionnel et logique du système.

TABLE 2.2 – Planification des sprints

Sprint	Objectifs	US Réalisées	Points Totals	Durée
1	Administration et sécurité	US01, US02, US03	15	2 semaines
2	Gestion des demandes clients	US04, US05, US06	15	2 semaines
3	Visualisation et décision	US07, US08	16	2 semaines
4	Intelligence artificielle	US09, US10	16	2 semaines

2.3 Spécifications techniques

Dans cette section, nous allons exposer les spécifications techniques utilisées lors de la phase de réalisation.

2.3.1 Technologies utilisées

Notre projet repose sur un ensemble de technologies modernes, choisies pour leur efficacité et leur complémentarité :

2.4 Spécifications techniques

Dans cette section, nous allons exposer les spécifications techniques utilisées lors de la phase de réalisation.

2.4.1 Technologies utilisées

Notre projet repose sur un ensemble de technologies modernes, choisies pour leur efficacité et leur complémentarité :

-  **MongoDB** : base de données NoSQL flexible, idéale pour le stockage des utilisateurs, crédits, rôles, et historiques.
-  **Node.js** : environnement backend JavaScript utilisé pour la création d'API REST, la gestion de l'authentification et des accès sécurisés.
-  **React.js** : bibliothèque JavaScript utilisée pour développer des interfaces utilisateur interactives et réactives.
-  **Tailwind CSS** : framework CSS utilitaire permettant une conception rapide et cohérente des composants UI.
-  **Python** : utilisé pour l'implémentation du scoring des crédits, l'analyse des risques et l'intégration de modèles de Machine Learning.

2.4.2 Outils utilisés

La table 2.3 illustre les outils utilisés tout au long de la phase de développement.

TABLE 2.3 – Outils utilisés

Outil	Utilisation
 Git	Gestion de version et collaboration
 GitHub	Hébergement et collaboration sur le code
 Visual Studio Code	Environnement de développement
 draw.io	Modélisation UML (diagrammes)
 Power BI	Création de tableaux de bord interactifs

2.5 Spécifications générales des exigences

Dans cette partie, nous allons expliquer nos spécifications générales des exigences.

2.5.1 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation a pour objectif de définir les attentes de chaque utilisateur vis-à-vis du système. En utilisant ce diagramme, nous pouvons représenter l'interaction entre les utilisateurs et les fonctions futures du système, ce qui nous permet d'extraire les besoins nécessaires à la conception du système.

La figure 2.1 nous illustre le diagramme de cas d'utilisation.

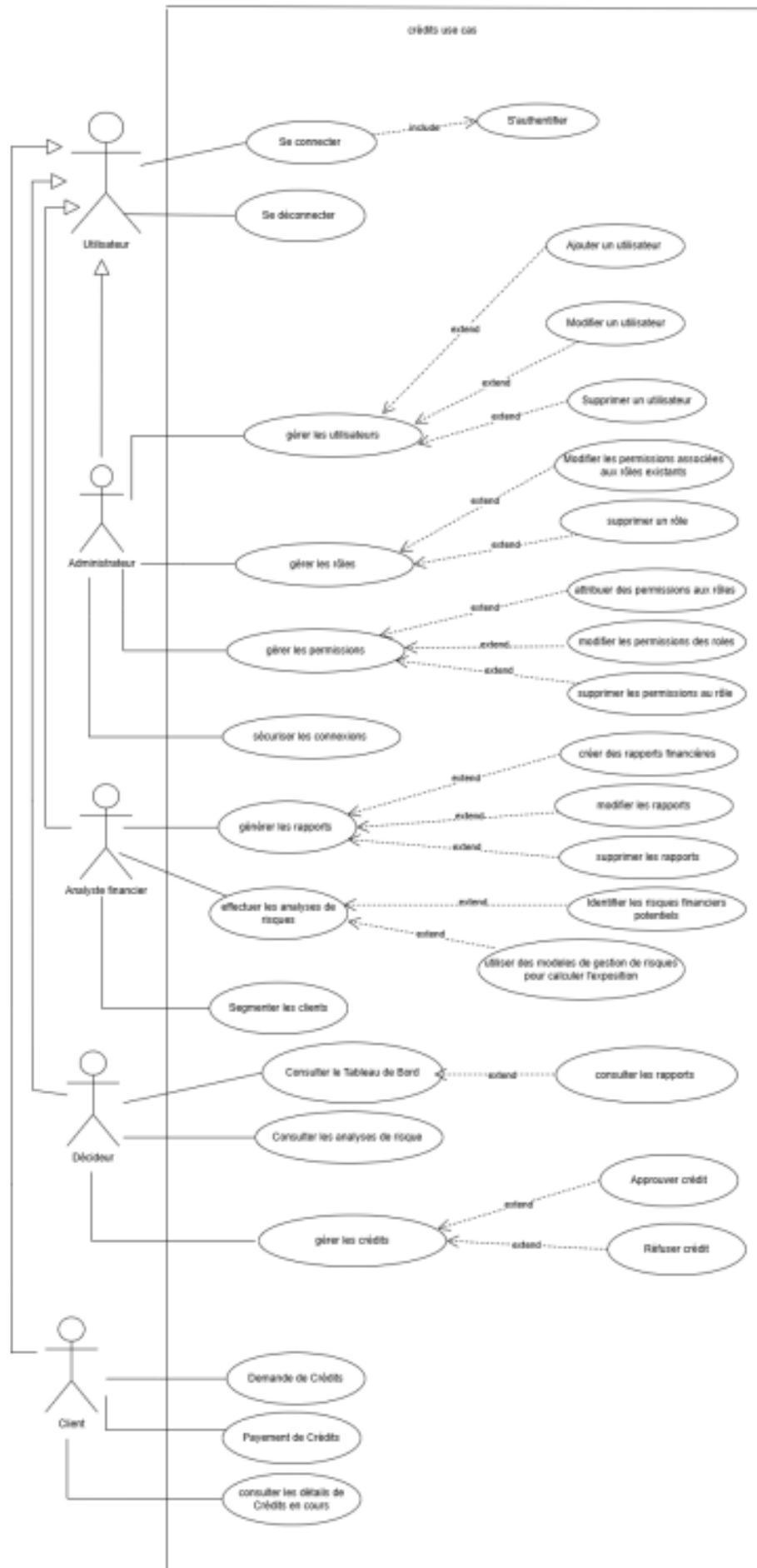


FIGURE 2.1 – Diagramme de cas d'utilisation

2.5.2 Diagramme de classes global

Le diagramme de classes est une illustration qui représente les divers éléments du système et les relations existantes entre eux. La figure 2.2 nous illustre le diagramme de classes global.

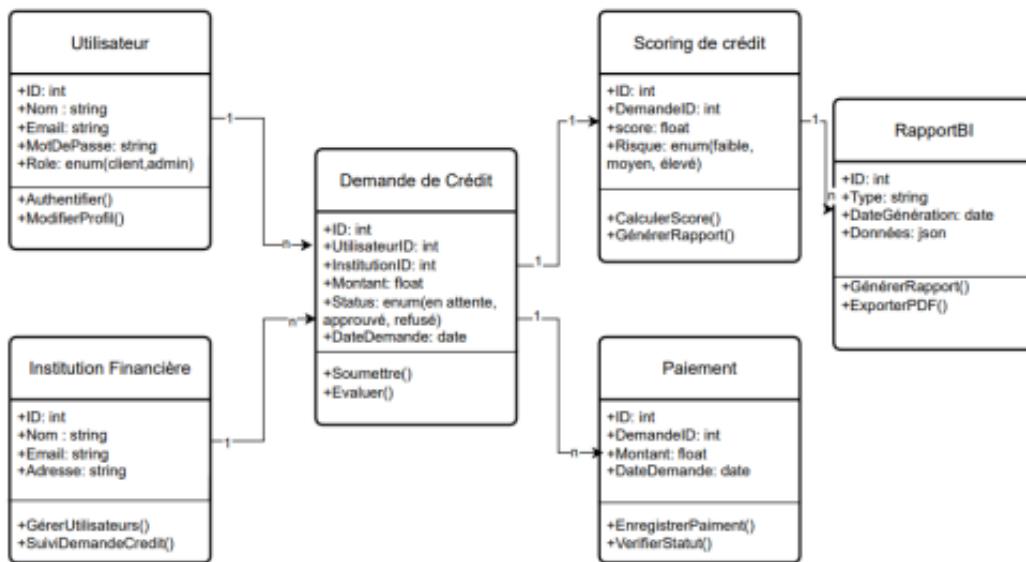


FIGURE 2.2 – Diagramme de classes

2.6 Architecture logicielle

L'architecture logicielle de notre application repose sur une approche en couches, permettant une séparation claire des responsabilités, une modularité accrue et une bonne évolutivité.

2.6.1 Couche de présentation (Frontend)

Développée avec **React.JS** et stylisée avec **Tailwind CSS**, cette couche permet aux utilisateurs (clients, décideurs, administrateurs) d'interagir avec le système via une interface web fluide, moderne et responsive. Elle communique avec le backend via des appels **API REST**.

2.6.2 Couche de logique métier (Backend)

Le backend est développé en **Node.js**, et intègre l'ensemble des règles métier, la gestion des accès et des rôles, ainsi que les modules d'intelligence artificielle (scoring crédit, détection d'anomalies, prédiction des risques). L'authentification est sécurisée grâce au système **JWT (JSON Web Token)**.

2.6.3 Couche de données

Nous utilisons **MongoDB**, une base de données NoSQL, pour stocker les utilisateurs, les crédits, les rôles, les historiques et les journaux. Sa flexibilité permet de gérer efficacement les structures dynamiques et les documents liés aux traitements IA.

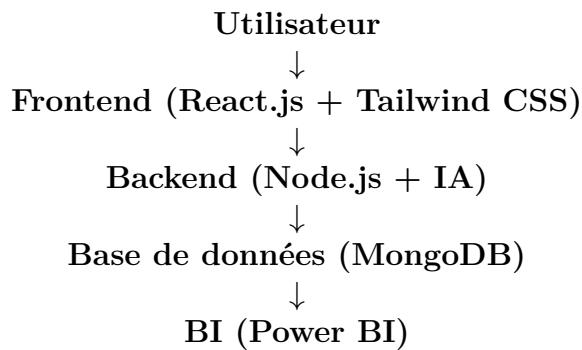
2.6.4 Composants transverses

- **Chatbot** : intégré dans l'interface, il dialogue avec l'utilisateur pour faciliter l'accès aux données ou poser des requêtes simples.
- **Power BI** : utilisé pour la visualisation des indicateurs clés sous forme de tableaux de bord interactifs.
- **Sécurité renforcée** : assurée par des mécanismes de validation, de rôles, d'authentification (JWT) et de vérification d'accès.

2.6.5 Schéma simplifié des interactions

La figure 2.3 présente un aperçu du flux global d'interaction entre les couches du système.

FIGURE 2.3 – Schéma simplifié des interactions



Conclusion

Ce chapitre a défini le contexte du projet, les besoins à couvrir et la solution proposée, en détaillant l'architecture, les technologies et l'organisation du travail. Le chapitre suivant présentera le développement et les premières fonctionnalités mises en œuvre.

Chapitre 3

Sprint 1 : Mise en place des fonctionnalités essentielle

Introduction

Ce premier sprint vise à établir les bases du projet avec la mise en place des configurations initiales, des premières pages (accueil, login, signup) et de l'infrastructure générale

3.1 Backlog du Sprint 1

La table 3.1 nous illustre le backlog product du Sprint 1.

TABLE 3.1 – Backlog du Sprint 1

ID	User Story	Priorité	Com.	Critère d'acceptation
US01	En tant qu' Administrateur , je veux gérer les utilisateurs.	Moyenne	5	Accès à la liste des utilisateurs → activer/désactiver, modifier les rôles.
US02	En tant qu' Administrateur , je veux surveiller les connexions suspectes.	Haute	5	Historique de connexions + alertes en cas d'activité inhabituelle (IP, heure, échec de connexion).
US03	En tant qu' Administrateur , je veux connecter la plateforme à Power BI.	Haute	5	Connexion fiable avec Power BI → données visibles depuis l'interface Power BI (vérification des jeux de données).

3.2 Réalisation

- Création des pages principales avec React.js et Tailwind CSS
- Connexion au backend Node.js pour la gestion des utilisateurs
- Mise en place de MongoDB pour le stockage initial des comptes et produits

La figure 3.1 nous montre la page d'accueil de NSMVISION.

La figure 3.2 nous montre l'interface d'inscription.

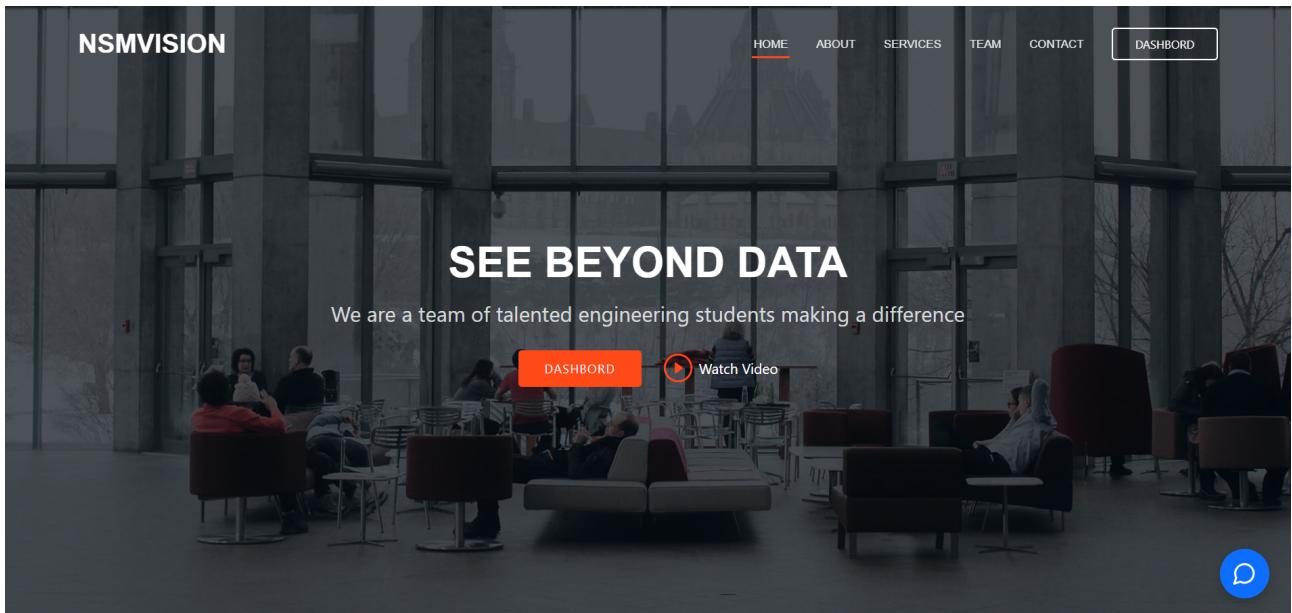


FIGURE 3.1 – Interface de connexion

La figure 3.3 nous montre l'interface de connexion.

3.3 Conclusion

Ce sprint a permis de poser les fondations du projet NSM Vision, d'assurer les premiers parcours utilisateurs, et de tester l'intégration des technologies choisies pour notre plateforme de gestion de crédits.

Les bases étant en place, le sprint suivant se concentrera sur l'optimisation des fonctionnalités de gestion des crédits et des demandes, éléments essentiels à toute plateforme de gestion financière.

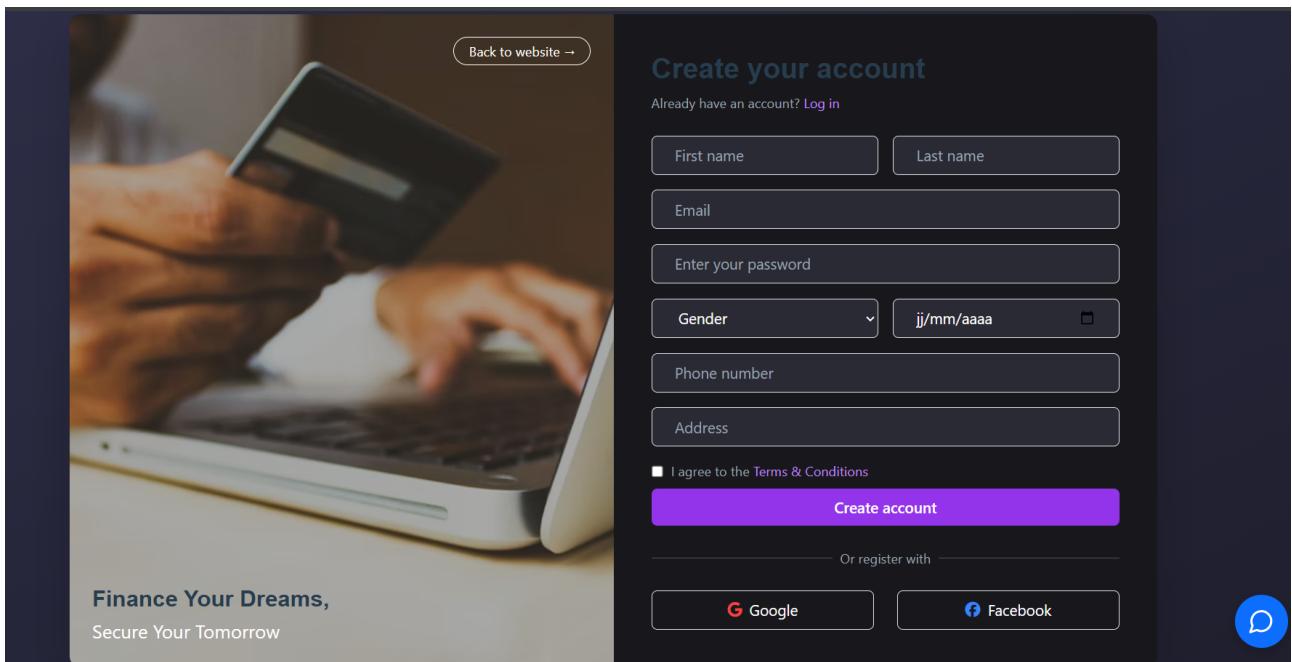


FIGURE 3.2 – Interface d’inscription

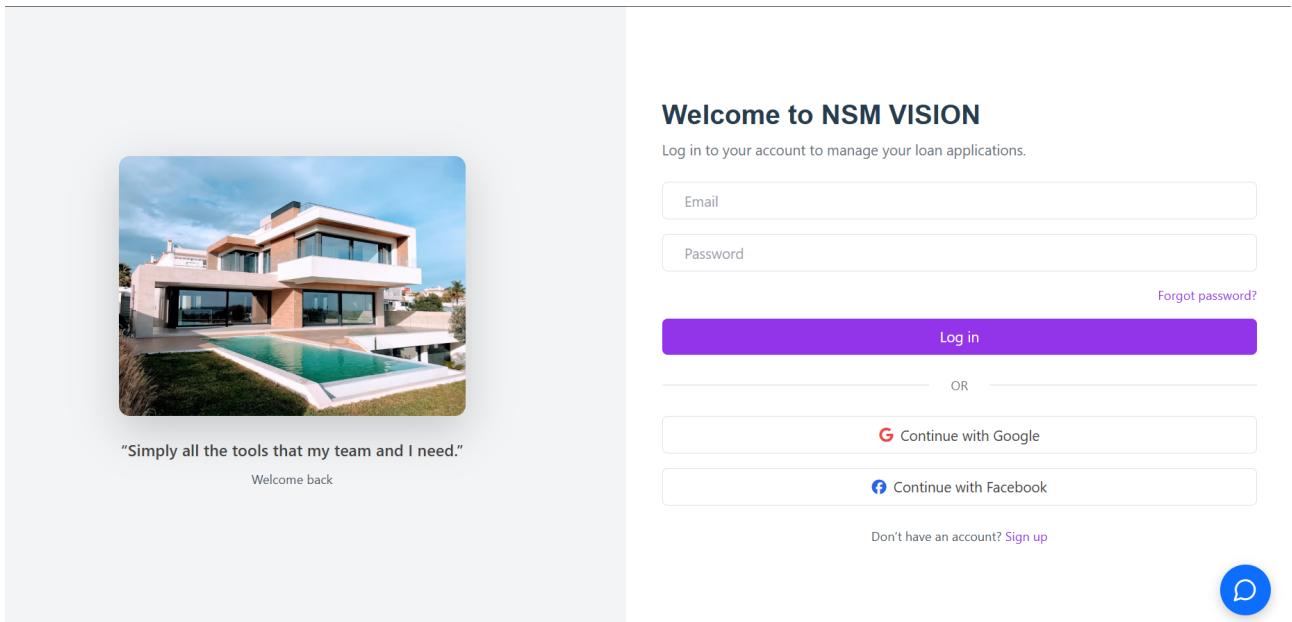


FIGURE 3.3 – Interface de connexion

Chapitre 4

Sprint 2 : Gestion des demandes de crédit

4.1 Introduction

Ce sprint est axé sur la gestion des demandes de crédit des clients, du point de vue utilisateur et administrateur.

4.2 Backlog du Sprint 2

La table 4.1 illustre le backlog produit du Sprint 2.

TABLE 4.1 – Backlog du Sprint 2

ID	User Story	Priorité	Com.	Critère d'a
US04	En tant que Client, je veux soumettre une demande de crédit.	Haute	5	Formulaire de
US05	En tant que Client, je veux consulter l'état de ma demande.	Haute	5	Accès à une

4.1 Réalisation

- Backend API pour la soumission des demandes de crédit et consulter sa statut (Node.js)
- Interface de gestion pour soumettre et suivre des demandes de crédit (React.js)
- Affichage dynamique des demandes de crédits .

La figure 4.1 montre l'interface permettant de soumettre et de suivre les demandes de crédit.

4.2 Conclusion

Les fonctionnalités de gestion des demandes de crédit, permettant aux clients de soumettre une demande et de consulter l'état de celle-ci, sont désormais opérationnelles. Ces fonctionnalités sont entièrement intégrées avec les bases de données MongoDB pour le stockage et la gestion des données des demandes de crédit.

CREDIT APPLICATION

APPLY FOR A NEW LOAN

HOME CREDIT

Age	Gender
<input type="text"/>	<input type="button" value="Select..."/>
Highest Education	Annual Income (€)
<input type="button" value="Select..."/>	<input type="text"/>
Employment Experience (Years)	Home Ownership Status
<input type="text"/>	<input type="button" value="Select..."/>
Loan Amount Requested	Purpose of the Loan
<input type="text"/>	<input type="button" value="Select..."/>
Interest Rate (%)	Loan as % of Income
<input type="text"/>	<input type="text"/>
Credit History Length (Years)	Credit Score
<input type="text"/>	<input type="text"/>



FIGURE 4.1 – Interface des demandes de crédit

NSMVISION

HOME CREDIT

APPLY FOR CREDIT

CREDIT APPLICATION

Fill out the form to submit your credit request.



FIGURE 4.2 – Interface des demandes de crédit

Chapitre 5

Sprint 3 : Visualisation et décision

5.1 Introduction

Ce sprint est dédié à la sécurisation de l'accès, à la gestion des crédits , à la visualisation des demandes de crédits et à l'intégration de l'IA (chatbot et recommandations).

5.2 Backlog du Sprint 3

La table 5.1 illustre le backlog product du Sprint 3.

ID	User Story	Priorité	Estimation (P)
US06	En tant que Décideur, je veux approuver ou refuser les demandes.	Haute	5
US07	En tant que Décideur, je veux visualiser les graphiques Power BI.	Haute	8

TABLE 5.1 – Backlog du Sprint 3

5.3 Réalisation

- **US06 – Approuver ou refuser une demande de crédit :**
 - Un module d'interface web affiche la liste des demandes de crédit, avec les détails nécessaires pour chaque demande : informations personnelles, montant du prêt, taux d'intérêt, score de crédit, etc.
 - Chaque demande est automatiquement évaluée par un modèle de machine learning entraîné avec TensorFlow (modèle de classification binaire).
 - Le résultat de prédiction (prêt approuvé ou refusé) ainsi que la confiance du modèle sont affichés.
 - Le décideur peut manuellement **valider ou refuser** la demande, indépendamment de la prédiction automatique.
 - La décision est enregistrée dans la base de données pour un suivi ultérieur.
- **US07 – Visualiser les graphiques Power BI :**
 - Des rapports interactifs sont créés dans Power BI pour faciliter l'analyse des demandes de crédit (taux d'approbation, profils de clients, taux de défaut, etc.).
 - Le décideur accède à ces rapports via un tableau de bord Power BI intégré ou accessible via un lien sécurisé.

- Les graphiques permettent une vue d'ensemble des performances du système de crédit, des tendances et de la distribution des profils.
- Ces visualisations aident à la prise de décision stratégique (ex. ajuster les seuils, détecter des anomalies, etc.).

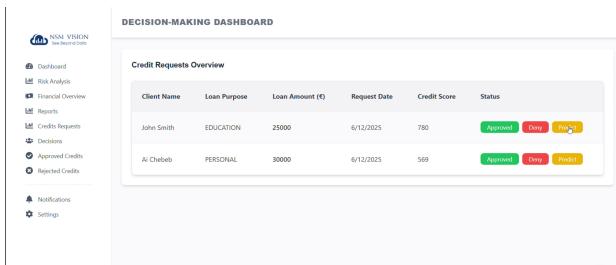


FIGURE 5.1 – Tableau de demande de crédit

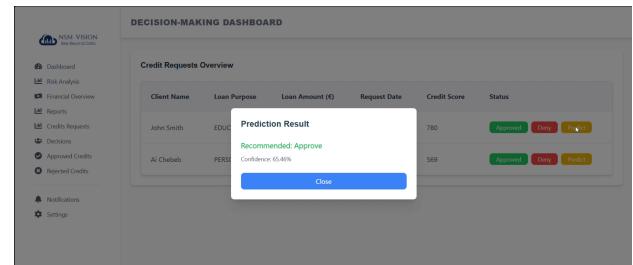


FIGURE 5.2 – Prédiction du modèle

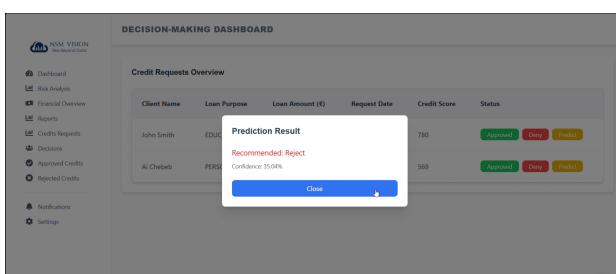


FIGURE 5.3 – Réfus

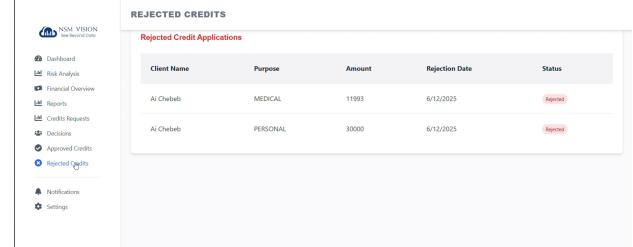


FIGURE 5.4 – Validation manuelle par le déicideur

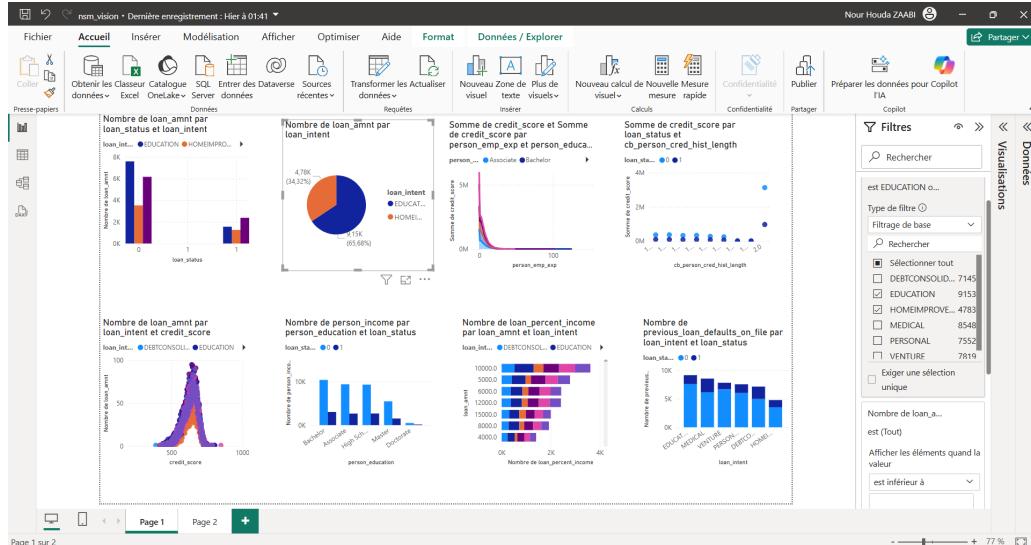


FIGURE 5.5 – Tableau de bord Power BI des demandes de crédit

5.3.1 Intégration d'un Chatbot Intelligent

Dans le cadre du projet de prédition de crédit, un **chatbot intelligent** a été intégré afin d'assister les utilisateurs dans la compréhension des démarches liées aux demandes de crédit, aux conditions d'éligibilité, aux documents nécessaires, etc. Ce chatbot permet également de répondre à des questions fréquentes de manière interactive et contextuelle.

Objectif

L'objectif est d'offrir une **interface conversationnelle intuitive**, capable de fournir des **réponses précises et contextualisées** à partir de documents internes, comme les politiques de crédit ou les guides bancaires.

Architecture du Système

Le chatbot repose sur une architecture moderne combinant :

- **FastAPI** : pour créer une API REST performante.
- **Groq API** (modèle : `llama-3.3-70b-versatile`) : pour la génération de texte.
- **RAG** (Retrieval-Augmented Generation) : pour la récupération de contexte pertinent depuis une base documentaire.
- **LlamaIndex** : pour l'indexation et la recherche de documents.

Fonctionnement

1. L'utilisateur pose une question via l'interface.
2. Le système interroge une base documentaire locale (via RAG).
3. Les passages pertinents sont extraits et injectés dans un prompt.
4. Le prompt est envoyé à l'API Groq pour générer une réponse claire.
5. La réponse est renvoyée à l'utilisateur.

Exemple d'utilisation

Un utilisateur peut poser la question :

“Quels sont les critères pour obtenir un crédit immobilier ?”

Le chatbot effectue une recherche dans les documents de référence et génère une réponse du type :

“Pour être éligible à un crédit immobilier, le client doit disposer d'un revenu stable, avoir un taux d'endettement inférieur à 35%, et fournir les pièces justificatives suivantes : fiche de paie, relevé bancaire, pièce d'identité, etc.”

Avantages de l'approche RAG + Groq

- **Réponses contextuelles** : meilleures que celles générées par des modèles isolés.
- **Mise à jour dynamique** : il suffit d'ajouter de nouveaux documents dans le dossier prévu.
- **Temps de réponse rapide** grâce à Groq (LLM ultra-performant).

Déploiement

Le chatbot est déployé en tant que microservice, sur le port 8001. Il est accessible via une interface web ou intégrable dans une application mobile. La configuration est sécurisée à l'aide de clés d'API et de middleware CORS.

Conclusion

Ce chatbot offre une **expérience utilisateur enrichie** et contribue à l'**automatisation du support client** dans le cadre des demandes de crédit. Il constitue un exemple d'intégration réussie de l'intelligence artificielle dans un système financier.

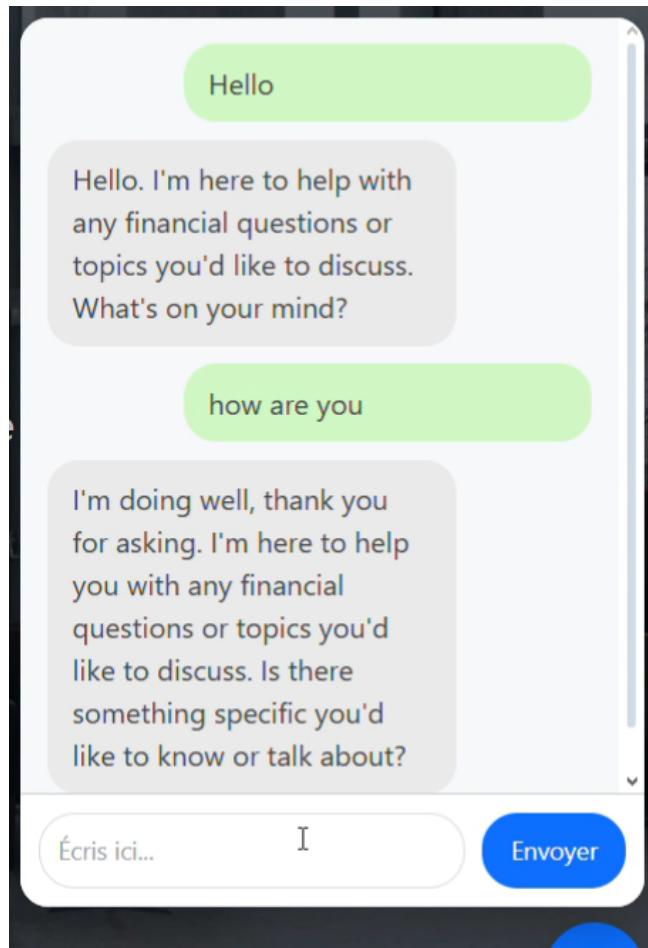


FIGURE 5.6 – Chatbot basé sur RAG et Groq

5.4 Conclusion

Les fonctionnalités de gestion des demandes de crédit, permettant aux clients de soumettre une demande et de consulter l'état de celle-ci, sont désormais opérationnelles. Ces fonctionnalités sont entièrement intégrées avec les bases de données MongoDB pour le stockage et la gestion des données des demandes de crédit.

Chapitre 6

Sprint 4 : Visualisation et analyses

Introduction

Ce sprint vise à exploiter les données collectées via une couche analytique : analyse, prévisions, tableau de bord avec Power BI.

6.1 Backlog du Sprint 4

La table 6.1 nous illustre le backlog product du Sprint 4.

ID	Rôle	Priorité	Description
US08	En tant qu'Analyste, je veux générer des rapports de risques.	Moyenne	Génération de rapports de risques avec résumé et tableau de bord.
US09	En tant qu'Analyste, je veux charger et traiter les données financières.	Moyenne	Téléversement et traitement des données financières.

TABLE 6.1 – Backlog du Sprint 4

6.2 Réalisation

La réalisation du Sprint 4 s'est concentrée sur l'analyse des données collectées et la création de rapports et de tableaux de bord pour aider à la prise de décisions. Les étapes principales ont consisté à créer des rapports de risques, à importer des données financières, et à intégrer des visualisations dynamiques avec Power BI. Les étapes de réalisation comprenaient l'analyse des besoins, le développement de rapports PDF sur les risques, ainsi que l'importation et le traitement des données financières pour les afficher sous forme de tableaux interactifs. Les données traitées ont ensuite été utilisées dans Power BI pour une visualisation dynamique des risques et des prévisions.

Conclusion

L'analyse des données, la mise en œuvre de modèles de Machine Learning pour le scoring des crédits, et l'intégration de **Power BI** pour la visualisation des résultats ont permis

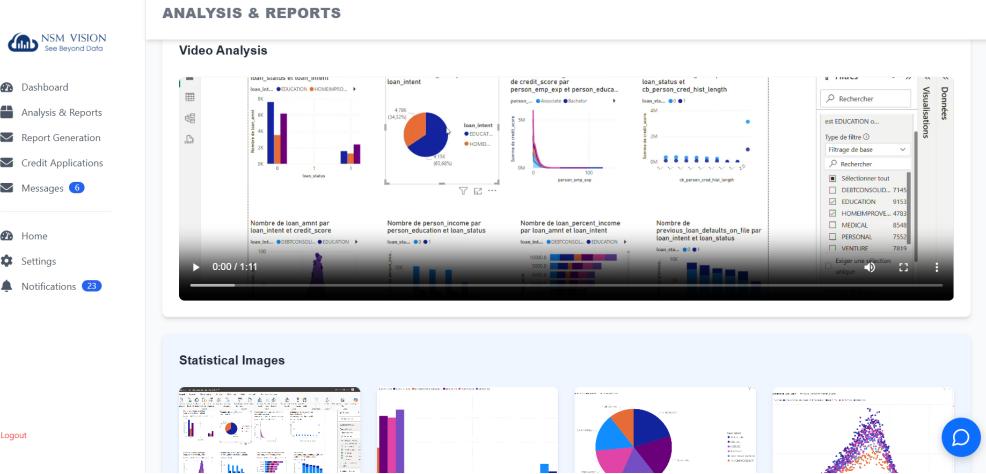


FIGURE 6.1 – Exemple de rapport généré avec les indicateurs de risque.



FIGURE 6.2 – Tableau de bord Power BI pour la visualisation des données financières.

de répondre efficacement aux besoins d’analyse des utilisateurs. La combinaison de ces technologies a assuré une performance optimale et une interface utilisateur réactive, facilitant l’expérience de navigation et d’interaction avec le système.

En conclusion, ce chapitre a mis en lumière l’importance d’une architecture bien conçue, de la gestion sécurisée des données et de l’intégration fluide de l’analytique dans une application web moderne.

Conclusion générale

Le projet NSM Solution a été conçu pour offrir une plateforme intuitive et performante dédiée à la gestion du crédit pour les petites et moyennes sociétés de financement. En intégrant des outils d'intelligence artificielle et de machine learning, NSM Solution permet d'automatiser l'analyse des risques et d'optimiser la prise de décision. Sa structure garantit une scalabilité optimale et une sécurité renforcée, tout en étant accessible et adaptée aux besoins spécifiques des PME. Ce projet vise à simplifier la gestion du crédit, améliorer l'analyse financière et fournir des rapports détaillés pour un accompagnement décisionnel efficace.