



*Rapport de Projet de Fin d'Année*

*Réalisé par*

**HOUSSAM YOUSSEF  
NIYYA HAJAR**

---

Plateforme  
d'Incubation Start-ups avec Intelligence  
Artificielle

---

**Encadrant académique :**

M.LARHLIMI

École Marocaine des Sciences de l'Ingénieur (EMSI)

Année universitaire : 2025/2026

# Remerciements

Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué de manière significative à la réussite de ce projet de fin d'année.

Nous remercions chaleureusement **l'École Marocaine des Sciences de l'Ingénieur (EMSI)** pour l'excellente formation reçue tout au long de notre parcours. Les connaissances et compétences fondamentales acquises durant ces années ont été essentielles pour mener à bien ce projet académique.

Nous souhaitons tout particulièrement remercier notre encadrant académique, **M. LARHLIMI**, pour son précieux guidage, sa disponibilité constante et ses conseils avisés qui ont été déterminants pour l'orientation et la réussite de ce projet.

Nous avons eu le plaisir de travailler ensemble sur ce projet dans un esprit de collaboration et d'entraide. Cette expérience nous a permis de développer nos compétences techniques et de relever ensemble les défis rencontrés, ce qui a rendu ce travail d'autant plus enrichissant.

Enfin, nous souhaitons remercier nos familles et nos amis pour leur soutien inconditionnel et leurs encouragements constants, qui ont été des sources de motivation essentielles tout au long de cette expérience.

Cordialement,  
HOUSSAM Youssef & NIYYA Hajar

# Résumé et Abstract

## Résumé

Le travail présenté dans ce rapport s'inscrit dans le cadre du projet de fin d'année (PFA) réalisé à l'**École Marocaine des Sciences de l'Ingénieur (EMSI)**. Il concerne la conception, le développement et la réalisation d'une plateforme web d'incubation pour les start-ups marocaines avec intégration d'intelligence artificielle.

Cette plateforme permet aux jeunes entrepreneurs de structurer leur activité, de créer un profil professionnel complet, et de générer automatiquement des pitches d'investissement personnalisés grâce à l'intelligence artificielle . Elle offre également un système de matching intelligent avec des investisseurs potentiels basé sur la compatibilité sectorielle et les critères d'investissement. L'application facilite ainsi le suivi de progression des start-ups, la gestion d'équipe et de jalons, et favorise la mise en relation avec l'écosystème entrepreneurial marocain.

Le projet a été développé selon une architecture microservices avec Spring Boot pour le backend, React.js pour le frontend, et PostgreSQL pour la base de données.

## Abstract

The work presented in this report is part of the end-of-year project (PFA) carried out at the **Moroccan School of Engineering Sciences (EMSI)**. It concerns the design, development, and implementation of a web-based incubation platform for Moroccan start-ups with artificial intelligence integration.

This platform allows young entrepreneurs to structure their business, create a comprehensive professional profile, and automatically generate personalized investment pitches using artificial intelligence . It also provides an intelligent matching system with potential investors based on sector compatibility and investment criteria. The application facilitates start-up progress tracking, team and milestone management, and promotes networking within the Moroccan entrepreneurial ecosystem.

The project was developed using a microservices architecture with Spring Boot for the backend, React.js for the frontend, and PostgreSQL for the database.

# Table des matières

<b>Table des figures</b>	<b>v</b>
<b>1 Présentation générale</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Problématique . . . . .	3
1.3 Solution proposée . . . . .	3
1.4 Étude Préliminaire . . . . .	3
1.4.1 Cahier des Charges . . . . .	4
1.4.2 Spécifications Fonctionnelles Détaillées . . . . .	5
1.5 Conclusion . . . . .	7
<b>2 Conception et modélisation</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 Diagramme de cas d'utilisation . . . . .	8
2.2.1 Définition . . . . .	8
2.2.2 Modélisation : Diagramme de Cas d'Utilisation . . . . .	8
2.3 Diagramme de Classe . . . . .	9
2.3.1 Définition . . . . .	9
2.3.2 Modélisation : Diagramme de Classe . . . . .	9
2.4 Conclusion . . . . .	11
<b>3 Implémentation :</b>	<b>12</b>
3.1 Introduction . . . . .	12
3.2 Architecture Microservices . . . . .	12
3.3 Technologies utilisées . . . . .	13
3.3.1 Partie Backend . . . . .	13
3.3.2 Partie Frontend . . . . .	14
3.3.3 Intégration Intelligence Artificielle . . . . .	15
3.3.4 Outils de Développement . . . . .	16
3.4 Interface de l'application : . . . . .	17
3.4.1 Système d'authentification . . . . .	17
3.4.2 Vue Start-up . . . . .	18
3.4.3 Vue Investisseur . . . . .	24
3.5 Conclusion . . . . .	29
Conclusion Générale . . . . .	31
<b>Bibliographie</b>	<b>33</b>

# Table des figures

2.1	Diagramme de cas d'utilisation . . . . .	9
2.2	Diagramme de Classe . . . . .	10
3.1	Logo du Framework Spring Boot . . . . .	14
3.2	Logo de la bibliothèque React.js . . . . .	15
3.3	* . . . . .	17
3.4	* . . . . .	17
3.5	Tableau de bord principal de la start-up . . . . .	18
3.6	* . . . . .	19
3.7	* . . . . .	19
3.8	* . . . . .	19
3.9	Interface de Profil Startup - Gestion complète des informations . . . . .	19
3.10	* . . . . .	20
3.11	* . . . . .	20
3.12	Interface de Matching et Connexion avec Investisseurs . . . . .	20
3.13	* . . . . .	22
3.14	* . . . . .	22
3.15	Interface de Gestion des Réunions et Calendrier . . . . .	22
3.16	Page de Paramètres et Préférences . . . . .	23
3.17	Tableau de bord principal de l'investisseur . . . . .	24
3.18	Page de profil investisseur avec critères d'investissement . . . . .	25
3.19	* . . . . .	26
3.20	* . . . . .	26
3.21	Interface de Gestion des Startups et Demandes de Connexion . . . . .	26
3.22	* . . . . .	28
3.23	* . . . . .	28
3.24	* . . . . .	28
3.25	* . . . . .	28
3.26	Interface Complète de Gestion du Calendrier et des Réunions (Vue Investisseur) . . . .	28

# Introduction

## Introduction

Aujourd'hui, l'entrepreneuriat et l'innovation jouent un rôle essentiel dans le développement économique et la compétitivité des nations. Au Maroc, l'écosystème des start-ups connaît une croissance remarquable, portée par une nouvelle génération d'entrepreneurs ambitieux. Cependant, ces jeunes entreprises font face à de nombreux défis, notamment la structuration de leur proposition de valeur, la préparation de pitches professionnels pour convaincre les investisseurs, et l'accès au financement. L'intelligence artificielle, en pleine expansion, offre aujourd'hui des opportunités sans précédent pour accompagner et faciliter le parcours entrepreneurial.

## Contexte académique

Ce rapport présente les travaux réalisés dans le cadre du **Projet de Fin d'Année (PFA)** de cinquième année de la filière Ingénierie Informatique et Réseaux à l'**École Marocaine des Sciences de l'Ingénieur (EMSI)**. L'EMSI est une école d'ingénierie reconnue par l'État, qui délivre une formation d'excellence en ingénierie et dispose de 19 campus à travers le Maroc.

Dans un contexte où la transformation digitale redéfinit les modèles d'affaires et où l'intelligence artificielle devient un levier stratégique pour l'innovation, notre projet s'inscrit dans cette dynamique de modernisation de l'écosystème entrepreneurial marocain. L'objectif est de concevoir une plateforme web innovante destinée à accompagner les start-ups marocaines dans leur développement, en leur offrant des outils intelligents de structuration de leur activité et de mise en relation avec des investisseurs potentiels.

## Objectifs du PFA

Les projets de fin d'année (PFA) constituent un élément essentiel de la formation EMSI, permettant aux étudiants d'appliquer leurs connaissances théoriques dans la conception de solutions concrètes répondant à des besoins réels. Les objectifs de ce PFA sont les suivants :

- **Objectif technique** : Concevoir et développer une plateforme web d'incubation pour start-ups avec architecture microservices
- **Objectif technologique** : Intégrer l'intelligence artificielle pour la génération automatique de pitches d'investissement personnalisés
- **Objectif fonctionnel** : Créer un système de matching intelligent entre start-ups et investisseurs basé sur des critères de compatibilité
- **Objectif analytique** : Développer des tableaux de bord pour le suivi de progression et l'analyse des performances des start-ups

- **Objectif pédagogique** : Maîtriser les architectures distribuées modernes avec Spring Boot pour le backend et React.js pour le frontend
- **Objectif d'impact** : Contribuer au développement de l'écosystème entrepreneurial marocain en facilitant l'accès au financement

## Structure du rapport

Ce rapport présente le développement de cette plateforme d'incubation avec intelligence artificielle et est organisé selon la structure académique suivante :

Le **premier chapitre** expose le cadre général du projet, incluant l'analyse approfondie des besoins fonctionnels et non-fonctionnels, l'étude comparative des solutions existantes, et la justification des choix techniques effectués.

Le **deuxième chapitre** présente la conception et la modélisation de l'architecture microservices, en détaillant les diagrammes UML (cas d'utilisation, classes), la conception de la base de données PostgreSQL avec schémas séparés, et l'architecture logicielle distribuée retenue.

Le **troisième chapitre** décrit l'implémentation de l'application, en détaillant l'environnement de développement Spring Boot/React.js, les différents microservices développés (authentification, gestion start-ups, génération IA, matching investisseurs, analytics), accompagnés de captures d'écran et d'exemples d'utilisation concrets.

Enfin, une **conclusion générale** synthétise les résultats obtenus, les compétences acquises, les difficultés rencontrées, et propose des perspectives d'évolution pour enrichir la plateforme.

# Chapitre 1

## Présentation générale

### 1.1 Introduction

Nous présentons dans ce chapitre une étude préliminaire du projet. Nous commencerons par définir le contexte et la problématique, puis nous détaillerons les objectifs et la solution proposée à travers le cahier des charges et les spécifications fonctionnelles.

### 1.2 Problématique

Nous avons constaté que de nombreux entrepreneurs marocains rencontrent des difficultés majeures pour structurer leur projet de start-up et convaincre les investisseurs. Ils manquent d'outils adaptés pour présenter leur proposition de valeur de manière professionnelle, créer des pitches d'investissement percutants, et identifier les investisseurs compatibles avec leur secteur d'activité. Cette situation entraîne une perte de temps considérable, des opportunités de financement manquées, et un frein au développement de l'écosystème entrepreneurial marocain. De plus, l'absence de plateformes centralisées limite la visibilité des start-ups auprès des investisseurs et complique le suivi de leur progression.

### 1.3 Solution proposée

Notre solution est une plateforme web d'incubation intelligente pour start-ups marocaines, simple et intuitive à utiliser. Cette plateforme permet aux entrepreneurs de créer un profil professionnel complet, de structurer leur activité avec gestion d'équipe et de jalons, et de générer automatiquement des pitches d'investissement personnalisés grâce à l'intelligence artificielle en quelques minutes seulement. Elle offre également un système de matching intelligent avec des investisseurs potentiels, basé sur des critères de compatibilité (secteur, montant, localisation). Grâce à un tableau de bord analytique et un suivi de progression, les entrepreneurs peuvent optimiser leur parcours et maximiser leurs chances d'obtenir un financement.

### 1.4 Étude Préliminaire

Cette section présente les documents fondateurs du projet : le cahier des charges qui en définit le cadre et les objectifs généraux, suivi des spécifications fonctionnelles détaillées qui précisent les choix techniques et le périmètre exact des fonctionnalités.



### 1.4.1 Cahier des Charges

#### Présentation du Projet

Développement d'une plateforme web d'incubation pour start-ups marocaines avec intégration d'intelligence artificielle. La plateforme permet aux entrepreneurs de structurer leur projet, de générer des pitches professionnels automatiquement, et de se connecter avec des investisseurs compatibles selon une architecture microservices moderne.

#### Objectifs

- Permettre aux start-ups de créer un profil professionnel complet et structuré.
- Générer automatiquement des pitches d'investissement via intelligence artificielle.
- Matcher les start-ups avec des investisseurs compatibles selon leur secteur d'activité.
- Fournir un tableau de bord avec statistiques et suivi de progression.
- Proposer une interface utilisateur moderne, intuitive et responsive.

#### Utilisateurs & Rôles

Rôle	Description
Administrateur	Gère la plateforme, supervise les start-ups et investisseurs, modère le contenu.
Start-up	Crée et gère son profil, génère des pitches IA, consulte les investisseurs matchés.
Investisseur	Crée son profil, définit ses critères d'investissement, consulte les start-ups compatibles.

#### Fonctionnalités

##### Administrateur

- Tableau de bord global : statistiques plateforme (nombre de start-ups, pitches générés, matchings réussis).
- Gestion des utilisateurs (validation, modération, désactivation).
- Supervision des contenus générés par IA.
- Consultation des logs et activités.
- Exportation des données analytiques (CSV, Excel, PDF).

##### Start-up

- Création et édition du profil entreprise (nom, secteur, description, tags).
- Gestion de l'équipe fondatrice (membres avec rôles et photos).
- Gestion des jalons/milestones (tâches à accomplir avec statuts).
- Génération automatique de pitches via IA (formulaire en 4 questions).
- Historique et gestion des pitches générés (modification, suppression, favoris).
- Consultation des investisseurs matchés avec score de compatibilité.
- Envoi de demandes de connexion aux investisseurs.
- Tableau de bord personnel (score de complétion, progression, métriques).

## Investisseur

- Création et édition du profil investisseur (type, secteurs d'intérêt, montants).
- Définition des critères d'investissement (secteurs, montants min/max, localisation).
- Consultation des start-ups matchées avec score de compatibilité.
- Gestion des demandes de connexion reçues (accepter/rejeter).
- Visualisation des pitches des start-ups connectées.
- Tableau de bord avec statistiques de matching.

## Architecture Technique

Élément	Choix
Architecture	Microservices (5 services indépendants)
Backend	Spring Boot avec API REST
Frontend	React.js avec Hooks
Base de données	PostgreSQL avec schémas séparés
Authentification	JWT tokens avec expiration
API Gateway	Spring Cloud Gateway (port 8080)
IA	Intégration API pour génération de pitches
Communication	OpenFeign (REST synchrone)

## Contraintes et Spécificités

- Responsive design pour utilisation sur tablette/mobile.
- Génération de pitches en moins de 15 secondes.
- Calcul automatique du score de complétion du profil (0-100%).
- Algorithme de matching basé sur plusieurs critères pondérés.
- Scalabilité et indépendance des microservices.

## 1.4.2 Spécifications Fonctionnelles Détaillées

### Présentation Générale

**Contexte** Le projet consiste à développer une plateforme web d'incubation pour start-ups marocaines avec architecture microservices, permettant la génération automatique de pitches d'investissement via intelligence artificielle et le matching intelligent avec des investisseurs potentiels.

### Objectifs

- Structurer les profils de start-ups avec calcul automatique de maturité.
- Générer des pitches professionnels personnalisés via IA en quelques minutes.
- Matcher intelligemment start-ups et investisseurs selon critères multiples.
- Fournir des tableaux de bord analytiques pour suivi de progression.
- Faciliter les connexions entre entrepreneurs et investisseurs.
- Garantir scalabilité et maintenabilité via architecture microservices.

### Périmètre Fonctionnel

#### Utilisateurs et Rôles

- **Administrateur** : Supervise la plateforme, gère les utilisateurs, accède aux analytics globaux.

- **Start-up** : Crée son profil, génère des pitches IA, consulte les investisseurs matchés, envoie des demandes de connexion.
- **Investisseur** : Définit ses critères, consulte les start-ups compatibles, gère les demandes de connexion.

## Modules Fonctionnels

- Authentication & Gestion Utilisateurs** Inscription avec email/mot de passe. Connexion sécurisée avec JWT tokens. Gestion de profil utilisateur. Attribution de rôles (STARTUP, INVESTOR, ADMIN).
- Gestion des Profils Start-ups** Création et édition du profil entreprise (nom, secteur, description, tags, logo, site web). Gestion de l'équipe fondatrice avec rôles. Gestion des jalons/milestones avec statuts. Calcul automatique du score de complétion (0-100%).
- Générateur de Pitch IA** Formulaire avec 4 questions clés (problème, solution, cible, avantage). Génération automatique de pitches personnalisés via IA. Sauvegarde et historique de tous les pitches. Fonctionnalités d'édition, suppression, favoris, notation.
- Gestion des Investisseurs** Création de profil investisseur (type, secteurs, montants, localisation). Définition des critères d'investissement. Portfolio des investissements réalisés.
- Matching Intelligent** Algorithme de matching avec score pondéré (secteur 70%, montant 20%, localisation 10%). Affichage des investisseurs/start-ups classés par compatibilité. Système de demande de connexion avec statuts (PENDING, ACCEPTED, REJECTED).
- Dashboard & Analytics** Tableau de bord avec métriques clés (score complétion, pitches générés, matchings, connexions). Graphiques de progression. Historique des activités récentes. Vue consolidée pour administrateur.
- Communication Inter-Services** Communication via OpenFeign (REST synchrone). Gestion des erreurs avec fallbacks. Validation des tokens JWT entre services.

## Architecture Microservices

- **API Gateway (8080)** : Routing, authentication, load balancing, CORS
- **Auth-Service (8081)** : Utilisateurs, authentication JWT, gestion des tokens
- **Startup-Service (8082)** : Profils start-ups, équipes, jalons, calcul complétion
- **Pitch-Service (8083)** : Génération IA de pitches, historique, templates
- **Investor-Service (8084)** : Profils investisseurs, algorithme matching, connexions
- **Analytics-Service (8085)** : Dashboard, statistiques, agrégation de données

## Données & Modèles Principaux

- **User** : id, email, password (hashed), role (STARTUP/INVESTOR/ADMIN), isActive
- **Startup** : id, userId, nom, secteur, description, tags, profileCompletion, logo, dateCreation
- **FounderMember** : id, startupId, nom, role (CEO/CTO/CMO), linkedIn, photo
- **Milestone** : id, startupId, titre, description, statut (TODO/IN\_PROGRESS/COMPLETED), dateEcheance
- **Pitch** : id, startupId, probleme, solution, cible, avantage, pitchGenere, type, rating, isFavorite
- **Investor** : id, userId, nom, type (VC/BUSINESS\_ANGEL/INCUBATOR), secteursInterets, montantMin, montantMax, localisation
- **MatchingResult** : id, startupId, investorId, score (0-100), criteria (JSON), isViewed

- **ConnectionRequest** : id, startupId, investorId, message, statut (PENDING/ACCEPTED/REJECT)

## Base de Données

- **PostgreSQL** avec schémas séparés par service
- **Schémas** : auth\_schema, startup\_schema, pitch\_schema, investor\_schema, analytics\_schema
- Relations via Foreign Keys cross-schema
- Indexation sur les champs fréquemment requis (secteur, userId, dates)

## Contraintes

- Interface responsive (mobile/tablette/desktop)
- API REST JSON pour frontend React
- Protection des routes via JWT Middleware dans Gateway
- Génération IA avec timeout 15 secondes max
- Retry automatique en cas d'échec (max 2 fois)
- Logging de toutes les requêtes IA pour audit

## Livrables

- Code source Backend (Spring Boot) - 5 microservices + Gateway
- Code source Frontend (React.js)
- Scripts de création de la base de données PostgreSQL
- Documentation technique (architecture, API, modèles)
- Documentation utilisateur
- Rapport de projet académique

## Planning (4 semaines)

- **Semaine 1** : Architecture microservices, Auth-Service, API Gateway, setup PostgreSQL.
- **Semaine 2** : Startup-Service (profils, équipes, jalons), Frontend React (authentification, profils).
- **Semaine 3** : Pitch-Service avec intégration IA, Investor-Service, algorithme de matching.
- **Semaine 4** : Analytics-Service, Dashboard, tests d'intégration, documentation, déploiement.

## 1.5 Conclusion

Ce chapitre nous a permis de situer le projet dans son cadre et d'en définir les fondements à travers le cahier des charges et les spécifications fonctionnelles. Nous avons présenté la problématique rencontrée par les entrepreneurs marocains, la solution innovante proposée avec intelligence artificielle, ainsi que l'architecture microservices retenue. Nous allons continuer, dans le chapitre suivant, par la présentation détaillée de la conception et de la modélisation de notre plateforme d'incubation.

# Chapitre 2

## Conception et modélisation

### 2.1 Introduction

Ce chapitre est consacré à la modélisation conceptuelle et dynamique du système à l'aide du langage de modélisation unifié UML (Unified Modeling Language). Cette phase de modélisation est une étape cruciale du cycle de développement, car elle permet de formaliser les besoins fonctionnels identifiés dans le cahier des charges, de conceptualiser l'architecture logicielle et de décrire les interactions entre les différents composants du système avant toute implémentation.

Deux diagrammes UML essentiels seront présentés et analysés afin de fournir une vision complète et structurée de l'application :

- Le **diagramme de cas d'utilisation** qui délimite le périmètre fonctionnel du système du point de vue des utilisateurs.
- Le **diagramme de classes** qui décrit la structure statique du système, les entités métiers et leurs relations.

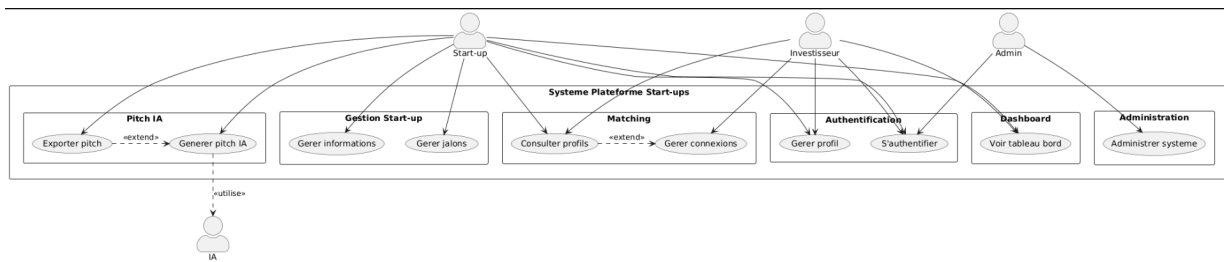
L'objectif de ce chapitre est de fournir une blueprint détaillée et non ambiguë qui servira de guide de référence pour la phase de développement qui suivra.

### 2.2 Diagramme de cas d'utilisation

#### 2.2.1 Définition

En UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et aident à capturer les exigences du système. Les diagrammes de cas d'utilisation décrivent les fonctions de haut niveau et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que fait le système et comment les acteurs l'utilisent, mais pas comment le système fonctionne en interne. [1]

#### 2.2.2 Modélisation : Diagramme de Cas d'Utilisation



### Description :

Ce diagramme identifie les acteurs du système et leurs interactions fonctionnelles. Il distingue quatre profils utilisateurs : Start-up (gestion de profil et génération de pitch), Investisseur (consultation de profils et gestion des connexions), Administrateur (modération du système) et Système IA (génération automatique de contenu). Les cas d'usage couvrent l'ensemble des fonctionnalités métier : authentification, gestion des informations et jalons, génération de pitch avec IA, export de documents, matching entre start-ups et investisseurs, gestion des connexions, consultation du tableau de bord et administration du système. Les relations d'extension et d'inclusion montrent les dépendances fonctionnelles, notamment entre la génération de pitch et l'utilisation du système IA, ainsi qu'entre la consultation de profils et la gestion des connexions.

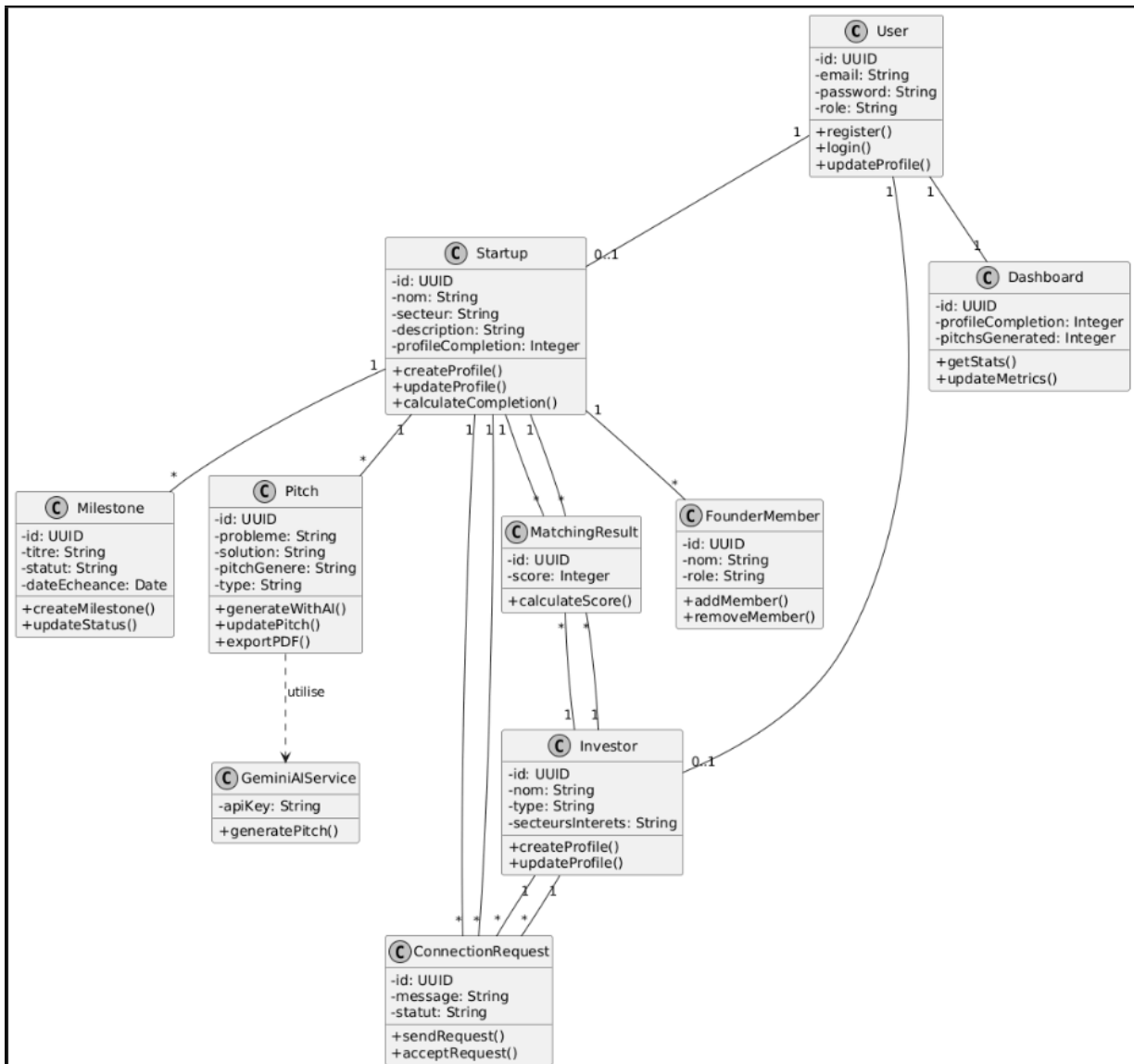
FIGURE 2.1 – Diagramme de cas d'utilisation

## 2.3 Diagramme de Classe

### 2.3.1 Définition

Dans le langage UML, les diagrammes de classes appartiennent à l'un des six types de diagramme structurel. Les diagrammes de classes sont fondamentaux pour le processus de modélisation des objets et modélisent la structure statique d'un système. Suivant la complexité d'un système, vous pouvez utiliser un seul diagramme de classes pour modéliser un système complet ou bien vous pouvez utiliser différents diagrammes de classes pour modéliser les composants d'un système. [2]

### 2.3.2 Modélisation : Diagramme de Classe



### Description :

Ce diagramme présente l'architecture orientée objet de la plateforme, mettant en évidence les entités principales du système. Il illustre les relations entre les modèles User, Startup, FounderMember, Milestone, Pitch, Investor, MatchingResult, ConnectionRequest et Dashboard. Chaque classe contient ses attributs essentiels et méthodes principales, démontrant la structure de données avec identification unique via UUID. Les cardinalités (1-n, 0..1-1) définissent les contraintes relationnelles entre les entités : un utilisateur peut être soit une start-up soit un investisseur, une start-up possède plusieurs membres fondateurs, jalons et pitches, et participe à plusieurs résultats de matching et demandes de connexion. L'intégration de l'intelligence artificielle pour la génération automatique de pitches est représentée par le service dédié qui traite les données des start-ups.

FIGURE 2.2 – Diagramme de Classe

## 2.4 Conclusion

À travers ce chapitre, nous avons mené une analyse approfondie et une modélisation rigoureuse de la plateforme d'incubation pour start-ups à l'aide de diagrammes UML fondamentaux. Cette étude nous a permis de passer d'une expression des besoins en langage naturel à une spécification formelle et structurée, limitant ainsi les risques d'incompréhension et d'ambiguïté lors du développement.

Le diagramme de cas d'utilisation a permis de circonscrire les fonctionnalités offertes aux acteurs principaux (start-ups, investisseurs, administrateurs) ainsi que l'intégration du système d'intelligence artificielle. Le diagramme de classes a défini l'ossature statique et les relations entre les entités métiers centrales : utilisateurs, profils de start-ups, équipes fondatrices, jalons, pitches générés, profils investisseurs, résultats de matching et demandes de connexion.

Ensemble, ces modèles constituent une documentation technique complète et cohérente qui valide la faisabilité conceptuelle de la solution et jette les bases solides sur lesquelles s'appuiera l'implémentation de l'architecture microservices, qui fera l'objet du chapitre suivant.



# Chapitre 3

## Implémentation :

### 3.1 Introduction

Ce chapitre présente la concrétisation pratique des modèles conceptuels et des spécifications techniques définis dans les chapitres précédents. Il a pour objectif principal d'exposer l'architecture microservices mise en place ainsi que les interfaces utilisateur de la plateforme d'incubation pour start-ups qui a été développée.

La réalisation de cette plateforme constitue une étape cruciale dans le cycle de développement, car elle matérialise l'ensemble des fonctionnalités conçues : la gestion des profils de start-ups, la génération automatique de pitches via intelligence artificielle, le système de matching avec les investisseurs, et les tableaux de bord analytiques. Une architecture microservices bien conçue doit garantir scalabilité, maintenabilité et indépendance des services, tandis que l'interface utilisateur doit allier fonctionnalité, intuitivité et efficacité pour assurer une adoption réussie.

Dans les pages qui suivent, nous présenterons d'abord l'architecture technique et les technologies utilisées, puis nous exposerons systématiquement les écrans principaux de l'application. Pour chaque interface, une capture d'écran sera accompagnée d'une analyse détaillée mettant en lumière les choix de conception, l'agencement des éléments et leur adéquation avec les besoins métier préalablement identifiés.

### 3.2 Architecture Microservices

L'architecture de notre plateforme repose sur le principe des microservices, une approche architecturale qui décompose l'application en plusieurs services indépendants, chacun responsable d'un domaine métier spécifique. Cette décision architecturale a été motivée par plusieurs facteurs :

- **Scalabilité indépendante** : Chaque service peut être dimensionné selon ses propres besoins de charge
- **Maintenance facilitée** : Les services étant de taille réduite, leur compréhension et leur maintenance sont simplifiées
- **Déploiement indépendant** : Possibilité de déployer et mettre à jour chaque service sans affecter les autres
- **Résilience** : Une défaillance dans un service n'affecte pas nécessairement l'ensemble du système
- **Technologies hétérogènes** : Possibilité d'utiliser différentes technologies selon les besoins spécifiques

Notre architecture comprend cinq microservices principaux communiquant via REST et orchestrés par une API Gateway :

- Auth-Service (Port 8081)** : Gestion des utilisateurs et authentification JWT
- Startup-Service (Port 8082)** : Gestion des profils start-ups, équipes et jalons
- Pitch-Service (Port 8083)** : Génération de pitches via intelligence artificielle
- Investor-Service (Port 8084)** : Gestion des investisseurs et algorithme de matching
- Analytics-Service (Port 8085)** : Tableaux de bord et statistiques agrégées
- API Gateway (Port 8080)** : Point d'entrée unique, routage et sécurité

## 3.3 Technologies utilisées

### 3.3.1 Partie Backend

Pour le développement du backend, nous avons opté pour une stack technologique moderne et robuste basée sur l'écosystème Java/Spring.

#### Spring Boot

Spring Boot a été choisi comme framework principal pour le développement des microservices. Cette décision repose sur plusieurs avantages majeurs :

- **Architecture microservices native** : Spring Boot facilite la création de services autonomes avec serveur embarqué
- **Écosystème Spring Cloud** : Outils dédiés pour la gestion de microservices (Gateway, Open-Feign, Config Server)
- **Injection de dépendances** : Architecture modulaire et testable grâce au conteneur IoC
- **Spring Data JPA** : Simplification de l'accès aux données avec Hibernate
- **Spring Security** : Sécurisation robuste avec support JWT natif
- **Convention over Configuration** : Configuration minimale grâce aux annotations

Chaque microservice est une application Spring Boot indépendante, packagée en JAR exécutable avec serveur Tomcat embarqué. Cette approche permet un déploiement simplifié et une gestion facilitée des dépendances.

#### PostgreSQL

PostgreSQL a été sélectionné comme système de gestion de base de données relationnelle pour ses caractéristiques techniques avancées :

- **Support des schémas** : Isolation logique des données par service (auth\_schema, startup\_schema, etc.)
- **Type UUID natif** : Identifiants uniques distribués sans collision
- **JSON/JSONB** : Stockage flexible pour données semi-structurées (critères de matching)
- **Performances** : Indexation avancée et optimiseur de requêtes puissant
- **Contraintes référentielles** : Relations cross-schema pour maintenir l'intégrité
- **Open Source** : Solution robuste et éprouvée sans coûts de licence

La base de données unique est structurée en cinq schémas séparés, un par microservice, garantissant une séparation logique claire tout en permettant les relations entre entités de différents services.

## Spring Cloud Gateway

Spring Cloud Gateway assure le rôle d'API Gateway avec plusieurs responsabilités critiques :

- **Routage dynamique** : Redirection des requêtes vers les microservices appropriés
- **Authentification centralisée** : Validation des JWT avant transmission aux services
- **Load Balancing** : Répartition de charge entre instances de services
- **CORS** : Gestion des autorisations cross-origin pour le frontend
- **Rate Limiting** : Protection contre les abus (max 100 requêtes/min)
- **Logging** : Traçabilité complète des requêtes entrantes

## OpenFeign

La communication inter-services est gérée par OpenFeign, un client HTTP déclaratif qui simplifie les appels REST :

- **Déclaratif** : Définition des clients via interfaces Java annotées
- **Intégration Spring** : Injection automatique dans les services
- **Fallback** : Gestion des erreurs avec mécanismes de repli
- **Load Balancing** : Distribution automatique des appels

## JWT (JSON Web Tokens)

L'authentification est basée sur JWT pour ses avantages dans une architecture distribuée :

- **Stateless** : Pas de session serveur, idéal pour microservices
- **Self-contained** : Token contient les informations utilisateur (id, role)
- **Sécurisé** : Signature HMAC-SHA256 avec clé secrète
- **Expiration** : Durée de vie configurable (24h pour access token)



FIGURE 3.1 – Logo du Framework Spring Boot

### 3.3.2 Partie Frontend

Pour l'interface utilisateur, nous avons choisi React.js, une bibliothèque JavaScript moderne et performante pour la construction d'interfaces réactives.

#### React.js

React.js a été sélectionné comme bibliothèque frontend pour plusieurs raisons stratégiques :

- **Composants réutilisables** : Architecture modulaire facilitant la maintenance
- **Virtual DOM** : Performances optimales avec mises à jour sélectives

- **Hooks** : Gestion d'état moderne (useState, useEffect, useContext)
- **Écosystème riche** : Nombreuses bibliothèques compatibles (React Router, Axios)
- **Single Page Application** : Navigation fluide sans rechargement de page
- **JSX** : Syntaxe intuitive mêlant HTML et JavaScript

Notre application utilise les React Hooks pour la gestion d'état, React Router pour la navigation, et Axios pour les appels API vers le backend via l'API Gateway.

## Tailwind CSS

Tailwind CSS a été employé comme framework CSS utility-first pour la conception visuelle :

- **Utility-first** : Classes CSS atomiques pour un développement rapide
- **Responsive** : Design adaptatif mobile-first intégré
- **Personnalisable** : Configuration centralisée des couleurs, espacements, etc.
- **Performance** : PurgeCSS intégré pour éliminer le CSS inutilisé
- **Consistance** : Design system cohérent sur toute l'application

Cette approche nous a permis de créer rapidement des interfaces modernes et responsives tout en maintenant une cohérence visuelle sur l'ensemble de la plateforme.

## Axios

Axios gère les communications HTTP avec le backend :

- **Promise-based** : API asynchrone moderne avec async/await
- **Interceptors** : Ajout automatique du token JWT dans les headers
- **Error handling** : Gestion centralisée des erreurs HTTP
- **Request/Response transformation** : Sérialisation JSON automatique



FIGURE 3.2 – Logo de la bibliothèque React.js

### 3.3.3 Intégration Intelligence Artificielle

La fonctionnalité phare de génération automatique de pitches repose sur l'intégration d'une API d'intelligence artificielle externe :

- **API REST** : Communication via requêtes HTTP POST
- **Prompt Engineering** : Templates de prompts optimisés par secteur
- **Génération contextuelle** : Pitches personnalisés basés sur les données de la start-up
- **Timeout** : Limitation à 15 secondes maximum par requête
- **Retry mechanism** : Nouvelle tentative automatique en cas d'échec (max 2)
- **Fallback** : Message d'erreur explicite si génération impossible
- **Logging** : Traçabilité complète pour audit et amélioration

Cette intégration permet de générer des pitches professionnels de 120-150 mots en quelques secondes, offrant ainsi un gain de temps considérable aux entrepreneurs.

### 3.3.4 Outils de Développement

- **Maven** : Gestion des dépendances et build des microservices Spring Boot
- **npm** : Gestion des packages JavaScript et build frontend
- **Git** : Versioning du code source
- **Postman** : Tests des APIs REST
- **IntelliJ IDEA** : IDE pour le développement backend Java
- **VS Code** : Éditeur pour le développement frontend React

## 3.4 Interface de l'application :

### 3.4.1 Système d'authentification

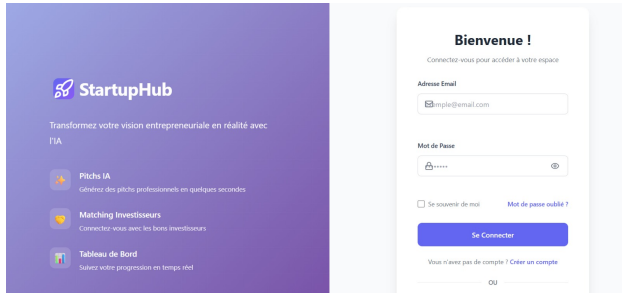


FIGURE 3.3 – \*

(a) Formulaire de connexion (Login)

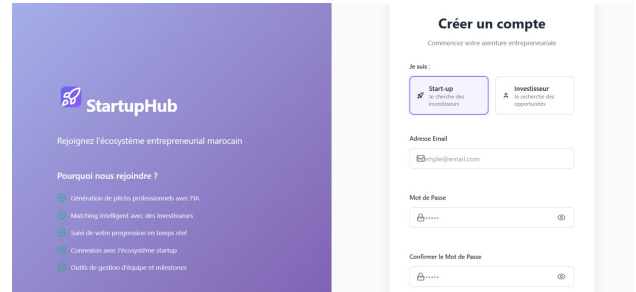


FIGURE 3.4 – \*

(b) Formulaire d'inscription (Register)

#### Description :

Cette section présente l'interface d'authentification de la plateforme **StartupHub**, une application dédiée à l'accompagnement entrepreneurial au Maroc. Les deux écrans assurent l'accès sécurisé et personnalisé aux portails des startups et des investisseurs.

- (a) **Écran de connexion** : Permet aux utilisateurs déjà inscrits de se connecter à leur espace personnel. L'interface met en avant les fonctionnalités phares de la plateforme : pitches IA, matching avec investisseurs et tableau de bord de suivi.
- (b) **Écran d'inscription** : Offre aux nouveaux utilisateurs (startups ou investisseurs) la possibilité de créer un compte. Il comprend une sélection du profil, la saisie de l'email et du mot de passe avec confirmation, ainsi qu'un aperçu des services proposés (génération de pitches IA, suivi de progression, outils de gestion d'équipe, etc.).

L'implémentation repose sur une authentification sécurisée avec validation des données et expérience utilisateur adaptée aux besoins des entrepreneurs et des investisseurs.

### 3.4.2 Vue Start-up

#### Tableau de Bord Start-up

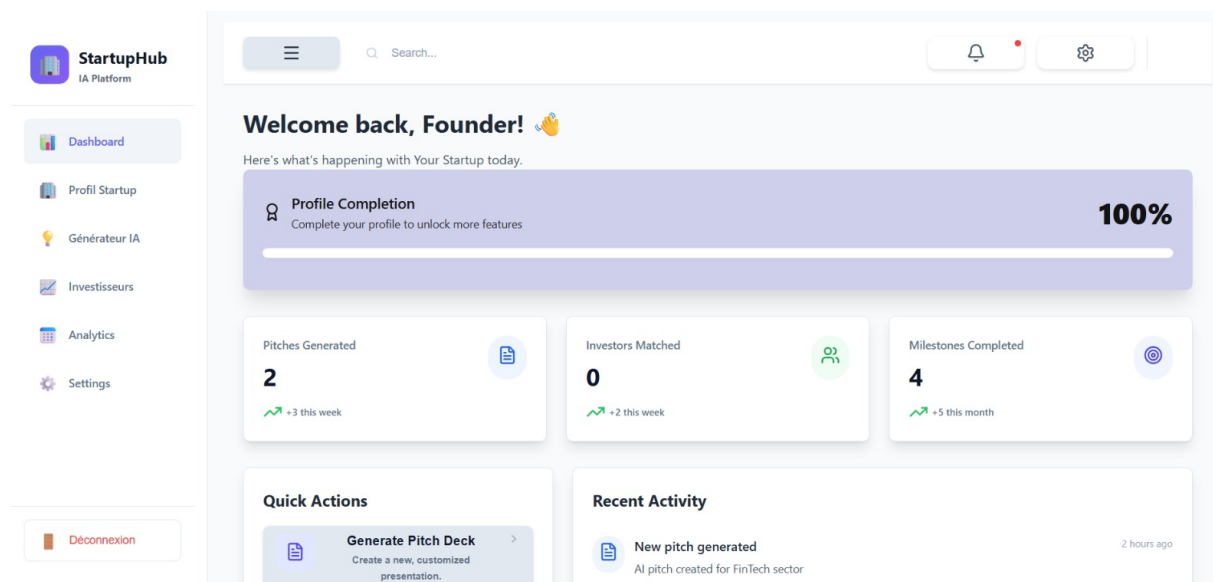


FIGURE 3.5 – Tableau de bord principal de la start-up

#### Description :

Le tableau de bord start-up constitue la page d'accueil principale après authentification. Il offre une vue d'ensemble de l'activité de la start-up avec les éléments suivants :

- **Complétion du profil** : Affiche le pourcentage d'avancement du profil (100% dans cet exemple) avec une barre de progression, encourageant l'entrepreneur à renseigner toutes les informations nécessaires.
- **Métriques clés** : Trois indicateurs principaux présentent les statistiques essentielles :
  - Pitches Generated : 2 pitches créés (+3 cette semaine)
  - Investors Matched : 0 investisseur compatible (+2 cette semaine)
  - Milestones Completed : 4 jalons complétés (+5 ce mois)
- **Quick Actions** : Section permettant d'accéder rapidement à la fonctionnalité principale "Generate Pitch Deck" pour créer une nouvelle présentation personnalisée via l'intelligence artificielle.
- **Recent Activity** : Historique des dernières actions avec horodatage, montrant par exemple "New pitch generated - AI pitch created for FinTech sector" (il y a 2 heures).
- **Navigation** : Menu latéral donnant accès aux différentes sections (Dashboard, Profil Startup, Générateur IA, Investisseurs, Analytics, Settings).

Cette page centralise les informations essentielles permettant à l'entrepreneur de suivre sa progression, d'accéder rapidement au générateur de pitches IA, et de consulter les investisseurs matchés.

# Interface de Profil Startup

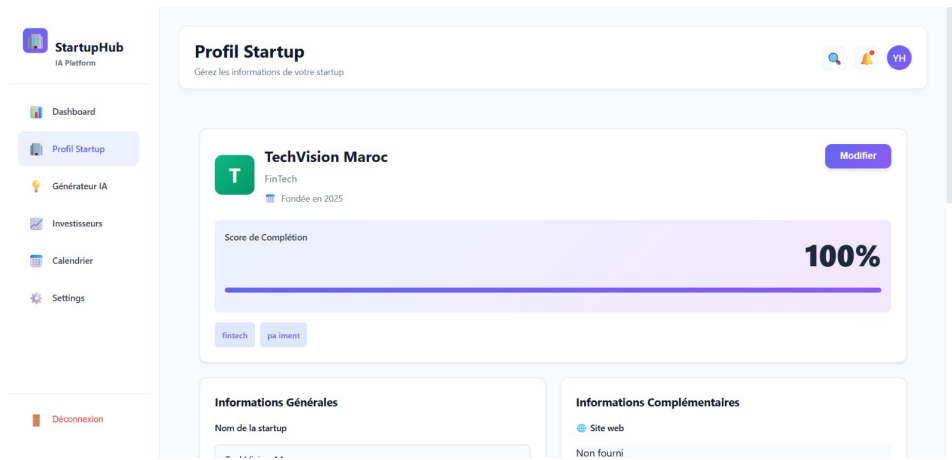


FIGURE 3.6 – \*  
(a) Vue d'ensemble du profil startup

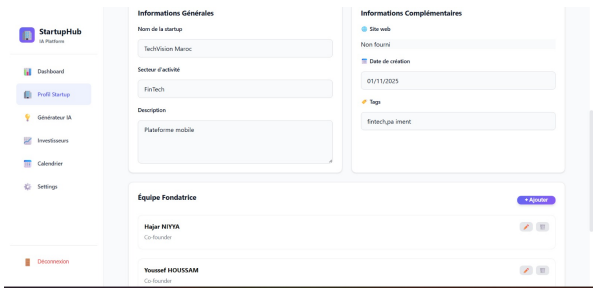


FIGURE 3.7 – \*  
(b) Informations générales et équipe

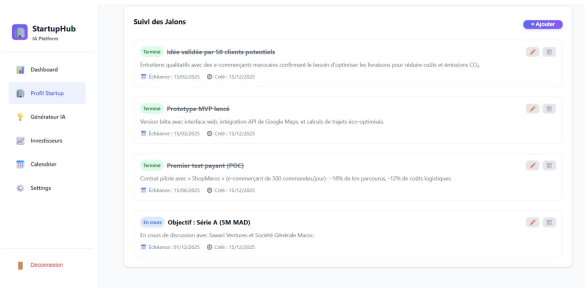


FIGURE 3.8 – \*  
(c) Suivi des jalons (milestones)

FIGURE 3.9 – Interface de Profil Startup - Gestion complète des informations

## Description :

L'interface de profil startup permet aux entrepreneurs de créer et gérer un profil professionnel complet pour leur entreprise. Cette page centralise toutes les informations essentielles de la start-up.

- (a) **Vue d'ensemble** : Affiche le nom de la startup "TechVision Maroc", le secteur d'activité "FinTech", et l'année de fondation "2025". Le score de complétion du profil est affiché à 100% avec une barre de progression visuelle. Les tags associés ("fintech", "paiement") permettent d'améliorer la visibilité et le matching avec les investisseurs. Un bouton "Modifier" permet d'éditer les informations.
- (b) **Informations détaillées** : Section "Informations Générales" contenant le nom de la startup, le secteur d'activité, et une description courte. Section "Informations Complémentaires" avec le site web, la date de création et les tags. La section "Équipe Fondatrice" liste les co-fondateurs avec leurs rôles (Hajar NIYYA - Co-founder, Youssef HOUSSAM - Co-founder), avec possibilité d'ajouter de nouveaux membres via le bouton "+ Ajouter".
- (c) **Suivi des jalons** : Page "Suivi des Jalons" présentant la liste des milestones avec leurs



statuts. Les jalons terminés affichent un badge vert "Terminé" avec dates d'échéance et de création (exemple : "Idée validée par 50 clients potentiels", "Prototype MVP lancé", "Premier test payant (POC)"). Les jalons en cours sont marqués "En cours" avec badge bleu (exemple : "Objectif : Série A (5M MAD)"). Chaque jalon dispose d'actions d'édition et de suppression, et un bouton "+ Ajouter" permet de créer de nouveaux jalons.

Cette interface permet aux start-ups de structurer leur profil de manière professionnelle, de présenter leur équipe, de suivre leur progression via les jalons, et de maximiser leur score de complétion pour améliorer leur visibilité auprès des investisseurs.

## Interface Investisseurs

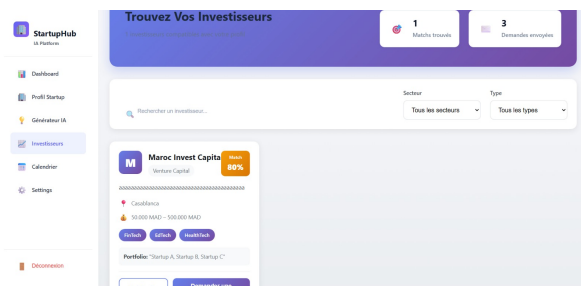


FIGURE 3.10 – \*  
(a) Liste des investisseurs matchés

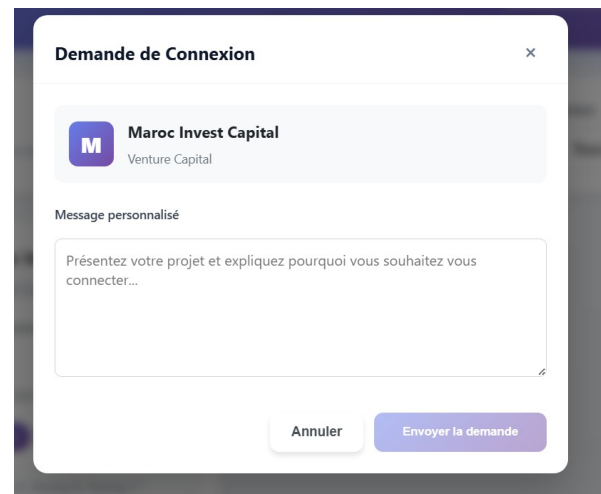


FIGURE 3.11 – \*  
(b) Demande de connexion

FIGURE 3.12 – Interface de Matching et Connexion avec Investisseurs

### Description :

L'interface investisseurs permet aux start-ups de découvrir les investisseurs compatibles avec leur profil et d'établir des connexions directes.

- **(a) Liste des investisseurs matchés** : En-tête "Trouvez Vos Investisseurs" avec sous-titre "1. Investisseurs compatibles avec votre profil". Statistiques affichées : 1 match trouvé et 3 demandes envoyées. Barre de recherche pour filtrer par nom, et filtres par secteur ("Tous les secteurs") et type ("Tous les types"). Carte investisseur présentant "Maroc Invest Capital" avec badge orange "Match 80%" indiquant le score de compatibilité calculé par l'algorithme. Informations détaillées : type "Venture Capital", localisation "Casablanca", fourchette d'investissement "50.000 MAD - 500.000 MAD", secteurs d'intérêt (badges "FinTech", "EdTech", "HealthTech"), et portfolio "Startup A, Startup B, Startup C". Deux boutons d'action : "Voir le site" pour consulter le site web de l'investisseur, et "Demander une connexion" pour initier un contact.
- **(b) Modal de demande de connexion** : Fenêtre pop-up "Demande de Connexion" s'ouvrant lors du clic sur "Demander une connexion". Affiche les informations de l'investisseur ciblé : "Maroc Invest Capital - Venture Capital". Champ de texte "Message personnalisé" permettant

de rédiger un message d'introduction avec placeholder "Présentez votre projet et expliquez pourquoi vous souhaitez vous connecter...". Deux boutons en bas : "Annuler" pour fermer sans envoyer, et "Envoyer la demande" (violet) pour soumettre la requête de connexion à l'investisseur.

Cette interface facilite la découverte d'investisseurs pertinents grâce à l'algorithme de matching avec score de compatibilité, permet de filtrer par secteur et type d'investisseur, et offre la possibilité d'envoyer des demandes de connexion personnalisées pour établir un premier contact professionnel.

# Interface Calendrier et Réunions

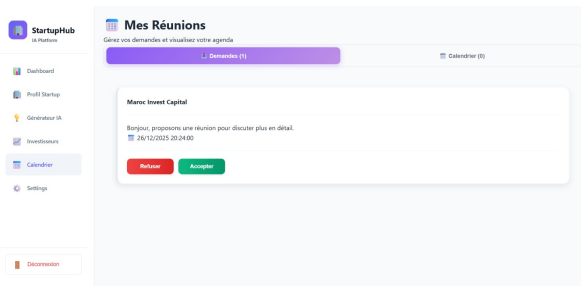


FIGURE 3.13 – \*

(a) Demandes de connexion reçues

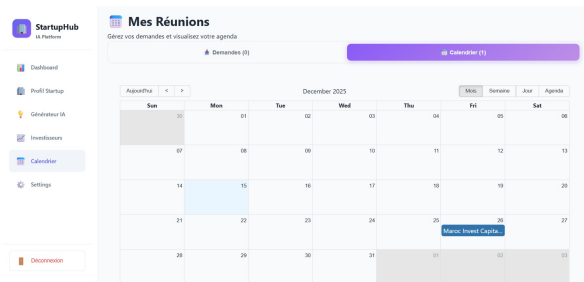


FIGURE 3.14 – \*

(b) Vue calendrier des réunions

FIGURE 3.15 – Interface de Gestion des Réunions et Calendrier

## Description :

L'interface calendrier permet aux start-ups de gérer les demandes de connexion des investisseurs et d'organiser leurs réunions.

- **(a) Gestion des demandes** : Page "Mes Réunions" avec sous-titre "Gérez vos demandes et visualisez votre agenda". Deux onglets disponibles : "Demandes (1)" actif en violet, et "Calendrier (0)" en gris. Affiche une carte de demande de connexion de "Maroc Invest Capital" contenant le message "Bonjour, proposons une réunion pour discuter plus en détail." avec la date et heure proposées "26/12/2025 20 :24 :00". Deux boutons d'action : "Refuser" (rouge) pour décliner la demande, et "Accepter" (vert) pour confirmer la réunion. Cette interface permet de traiter les demandes reçues suite aux connexions établies via la page investisseurs.
- **(b) Vue calendrier** : Onglet "Calendrier (1)" actif affichant le calendrier mensuel de Décembre 2025. Navigation temporelle avec boutons "Aujourd'hui", "<" et ">" pour changer de période. Sélecteur de vue : "Mois", "Semaine", "Jour", "Agenda". Le calendrier présente une grille hebdomadaire (Sun-Sat) avec les dates du mois. Une réunion planifiée apparaît le vendredi 26 décembre sous forme de pastille bleue "Maroc Invest Capita..." (nom tronqué pour s'adapter à la cellule). Les jours 15-16 sont mis en évidence avec fond bleu clair. Cette vue permet de visualiser rapidement l'agenda des réunions acceptées avec les investisseurs.

Cette interface centralise la gestion des demandes de connexion reçues des investisseurs, permet d'accepter ou refuser les propositions de réunion, et offre une vue calendrier pour organiser et suivre les rendez-vous planifiés avec différentes options de visualisation (mois, semaine, jour, agenda).

## Interface Paramètres

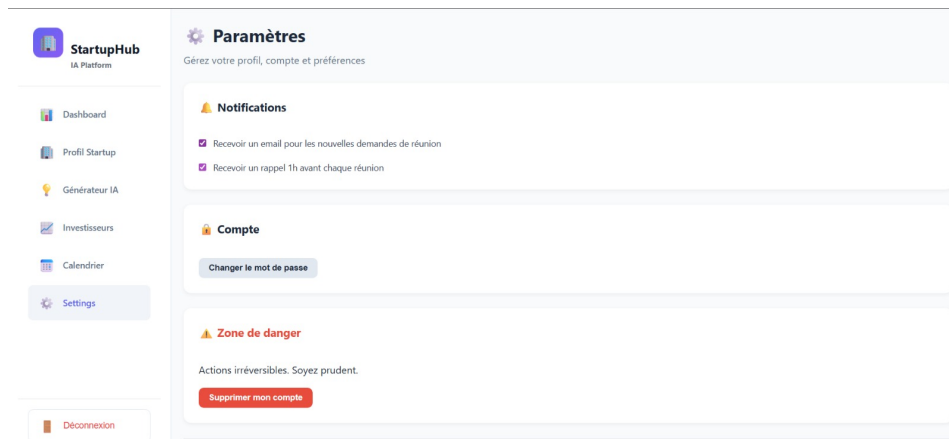


FIGURE 3.16 – Page de Paramètres et Préférences

### Description :

La page de paramètres permet aux utilisateurs de gérer leur compte, leurs préférences de notifications et d'effectuer des actions sensibles sur leur profil. Elle est structurée en trois sections principales :

- **Notifications** : Section avec icône de cloche permettant de configurer les alertes et notifications. Deux options disponibles avec cases à cocher activées : "Recevoir un email pour les nouvelles demandes de réunion" pour être notifié par email lorsqu'un investisseur envoie une demande de connexion ou de réunion, et "Recevoir un rappel 1h avant chaque réunion" pour obtenir une notification de rappel automatique une heure avant chaque rendez-vous planifié dans le calendrier. Ces paramètres permettent de ne manquer aucune opportunité de connexion avec les investisseurs.
- **Compte** : Section avec icône de cadenas pour la gestion de la sécurité du compte. Contient un bouton "Changer le mot de passe" (fond gris clair) permettant d'accéder à l'interface de modification du mot de passe pour renforcer la sécurité du compte utilisateur.
- **Zone de danger** : Section critique marquée en rouge avec icône d'avertissement et texte "Actions irréversibles. Soyez prudent." Contient le bouton rouge "Supprimer mon compte" permettant de supprimer définitivement le compte utilisateur et toutes les données associées (profil startup, pitches générés, connexions avec investisseurs). Cette action destructive nécessite une confirmation supplémentaire pour éviter les suppressions accidentelles.

Cette interface centralise la gestion du compte utilisateur avec des paramètres de notifications pour optimiser la communication avec les investisseurs, des options de sécurité pour protéger le compte, et une zone de danger clairement identifiée pour les actions irréversibles.

### 3.4.3 Vue Investisseur

#### Tableau de Bord Investisseur

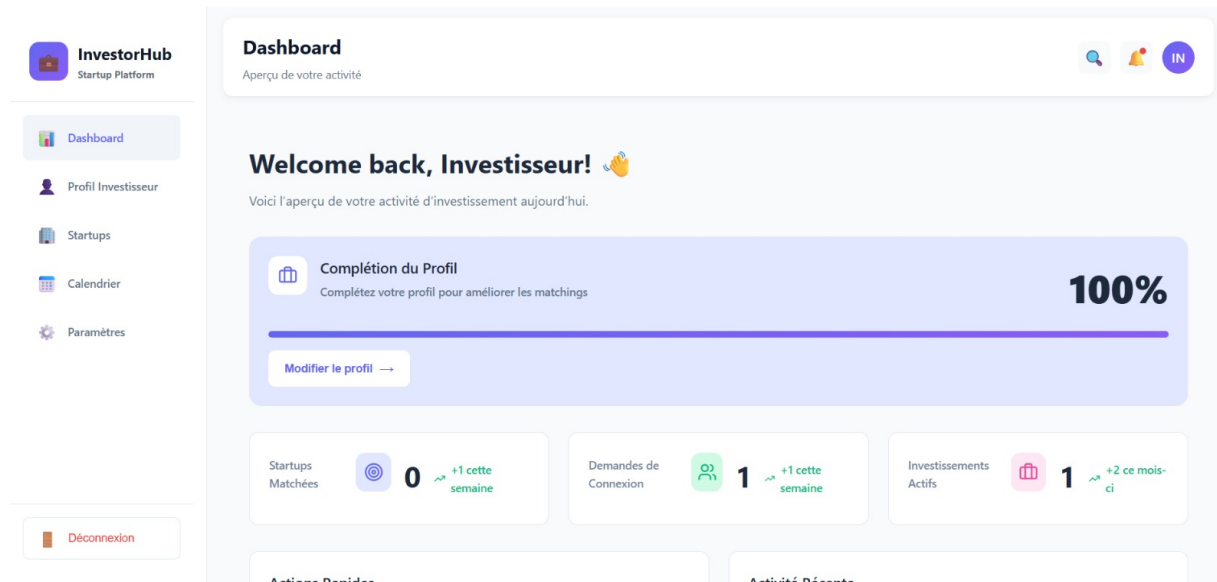


FIGURE 3.17 – Tableau de bord principal de l'investisseur

#### Description :

Le tableau de bord investisseur constitue la page d'accueil après authentification avec le rôle INVESTOR. Il offre une vue d'ensemble de l'activité d'investissement avec les éléments suivants :

- **Identification de la plateforme** : L'en-tête affiche "InvestorHub - Startup Platform" avec un logo distinct de celui des start-ups, différenciant clairement l'interface investisseur. Le badge "IN" (initiales de l'investisseur) apparaît en haut à droite avec icône de recherche et notifications.
- **Message de bienvenue** : "Welcome back, Investisseur !" avec sous-titre "Voici l'aperçu de votre activité d'investissement aujourd'hui." Cette personnalisation crée une expérience utilisateur adaptée au profil investisseur.
- **Complétion du profil** : Section proéminente affichant "Complétion du Profil 100%" avec message "Complétez votre profil pour améliorer les matchings". Barre de progression complète en violet. Bouton "Modifier le profil →" permettant d'accéder à l'édition des critères d'investissement (secteurs, montants, localisation). Un profil complet améliore la qualité des start-ups matchées par l'algorithme.
- **Métriques clés** : Trois indicateurs principaux présentent les statistiques d'activité :
  - Startups Matchées : 0 start-up compatible (+1 cette semaine)
  - Demandes de Connexion : 1 demande reçue des start-ups (+1 cette semaine)
  - Investissements Actifs : 1 investissement en cours (+2 ce mois-ci)

Chaque métrique inclut un indicateur de tendance (flèche verte) avec évolution temporelle.

- **Navigation** : Menu latéral adapté au rôle investisseur avec sections Dashboard, Profil Investisseur, Startups (liste des start-ups matchées), Calendrier (gestion des réunions), et Paramètres.
- **Sections visibles partiellement** : En bas de page apparaissent les sections "Actions Rapides" et "Activités Récentes" (non visibles entièrement dans la capture) permettant respectivement d'accéder rapidement aux fonctionnalités principales et de consulter l'historique des

interactions avec les start-ups.

Cette interface permet aux investisseurs de suivre leur activité, de visualiser les start-ups compatibles identifiées par l'algorithme de matching, de gérer les demandes de connexion reçues des entrepreneurs, et de suivre leurs investissements en cours.

## Interface de Profil Investisseur

The screenshot displays the 'Profil Investisseur' page on the InvestorHub platform. The header includes the InvestorHub logo and a search icon. The main content area is divided into two columns. The left column, titled 'Informations Générales', contains fields for 'Nom' (Maroc Invest Capital), 'Type' (Capital-Risque), and 'Email' (contact@marocinvest.ma). The right column, titled 'Préférences d'Investissement', shows 'Secteurs d'intérêt' (FinTech, EdTech, HealthTech) and 'Montant minimum' (50000 €) and 'Montant maximum' (500000 €). A 'Modifier' button is located at the top right of the profile section. A sidebar on the left contains navigation links: Dashboard, Profil Investisseur, Startups, Calendrier, Paramètres, and a Déconnexion button.

FIGURE 3.18 – Page de profil investisseur avec critères d'investissement

### Description :

L'interface de profil investisseur permet de gérer les informations personnelles et de définir les critères d'investissement utilisés par l'algorithme de matching. La page est structurée en deux sections principales :

- **En-tête du profil** : Présente l'identité de l'investisseur avec avatar "M" (Maroc Invest Capital), nom "Maroc Invest Capital", type "Fonds de Capital-Risque", et localisation "Casablanca" avec icône de localisation. Bouton "Modifier" (bleu foncé) en haut à droite permettant d'éditer le profil.
- **Informations Générales** : Section à gauche contenant les données de base de l'investisseur. Champ "Nom" affichant "Maroc Invest Capital", champ "Type" indiquant "Capital-Risque" (catégorie d'investisseur parmi VC, Business Angel, Incubator, Corporate VC), et champ "Email" montrant l'adresse de contact "contact@marocinvest.ma". Ces informations permettent aux start-ups d'identifier le type d'investisseur et de le contacter.
- **Préférences d'Investissement** : Section à droite définissant les critères utilisés par l'algorithme de matching pour identifier les start-ups compatibles. "Secteurs d'intérêt" sous forme de badges sélectionnables : "FinTech", "EdTech", "HealthTech" (en bleu clair), permettant de cibler les domaines d'investissement privilégiés. Ces secteurs représentent 70% du score de matching dans l'algorithme. "Montant minimum" fixé à "50000 €" et "Montant maximum" à "500000 €" (partiellement visible), définissant la fourchette d'investissement et représentant 20% du score de matching. Ces critères permettent d'automatiser le filtrage des start-ups pertinentes.

Cette interface permet aux investisseurs de structurer leur profil professionnel, de définir précisément leurs critères d'investissement (secteurs, montants, localisation), et d'optimiser la qualité des start-ups proposées par l'algorithme de matching automatique. Un profil complet et bien paramétré améliore significativement la pertinence des matchings.

### Interface Startups Matchées

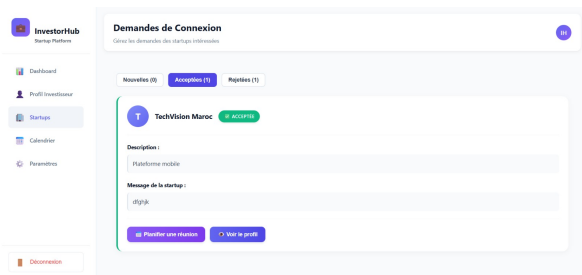


FIGURE 3.19 – \*  
(a) Gestion des demandes de connexion

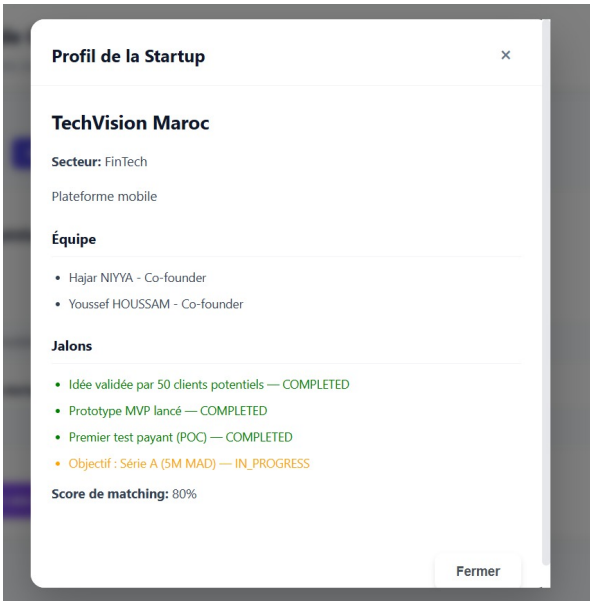


FIGURE 3.20 – \*  
(b) Détails profil startup

FIGURE 3.21 – Interface de Gestion des Startups et Demandes de Connexion

### Description :

L'interface Startups permet aux investisseurs de consulter les demandes de connexion reçues et de visualiser les profils détaillés des start-ups intéressées.

- (a) **Gestion des demandes** : Page "Demandes de Connexion" avec sous-titre "Gérez les demandes des startups intéressées". Trois onglets de filtrage : "Nouvelles (0)" pour les demandes en attente, "Acceptées (1)" actif en violet montrant les connexions validées, et "Rejetées (1)" pour les demandes refusées. Carte de demande présentant "TechVision Maroc" avec avatar "T" et badge vert "ACCEPTÉE" indiquant que la connexion a été validée. Section "Description" affichant "Plateforme mobile" (description courte de la startup). Section "Message de la startup" montrant le message personnalisé envoyé lors de la demande initiale (exemple : "dfghjk"). Deux boutons d'action : "Planifier une réunion" (violet) pour programmer un rendez-vous avec la start-up, et "Voir le profil" (bleu) pour consulter les détails complets incluant équipe, jalons et score de matching.
- (b) **Modal profil détaillé** : Pop-up "Profil de la Startup" s'ouvrant lors du clic sur "Voir le profil". En-tête avec nom "TechVision Maroc", secteur "FinTech", et description "Plateforme mobile". Section "Équipé" listant les co-fondateurs : "Hajar NIYYA - Co-founder" et "Youssef HOUSSAM - Co-founder". Section "Jalons" présentant la progression avec statuts : trois

jalons terminés en vert "COMPLETED" ("Idée validée par 50 clients potentiels", "Prototype MVP lancé", "Premier test payant (POC)"), et un jalon en cours en orange "IN\_PROGRESS" ("Objectif : Série A (5M MAD)"). Section "Score de matching : 80%" affichant le score de compatibilité calculé par l'algorithme basé sur les critères de l'investisseur. Bouton "Fermer" en bas pour revenir à la liste des demandes.

Cette interface permet aux investisseurs de trier les demandes par statut (nouvelles/acceptées/rejetées), de consulter les profils détaillés des start-ups avec équipe et jalons, de visualiser le score de compatibilité algorithmique, et de planifier des réunions avec les entrepreneurs acceptés pour discuter d'opportunités d'investissement.



## Interface Calendrier et Gestion des Réunions

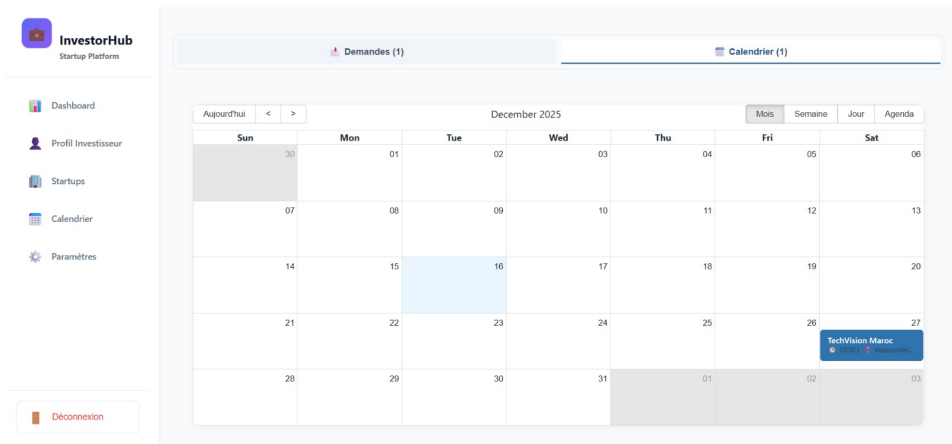


FIGURE 3.22 – \*  
(a) Vue calendrier mensuel

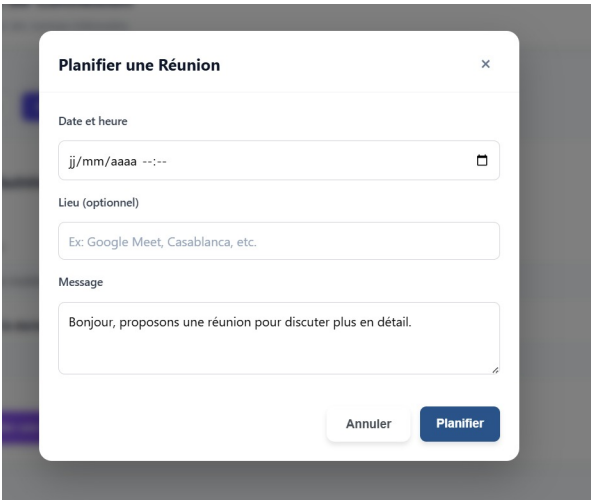


FIGURE 3.23 – \*  
(b) Modal planification de réunion

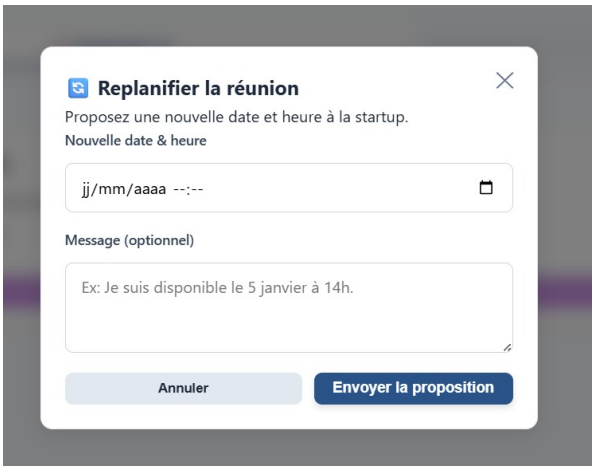


FIGURE 3.24 – \*  
(c) Modal replanification

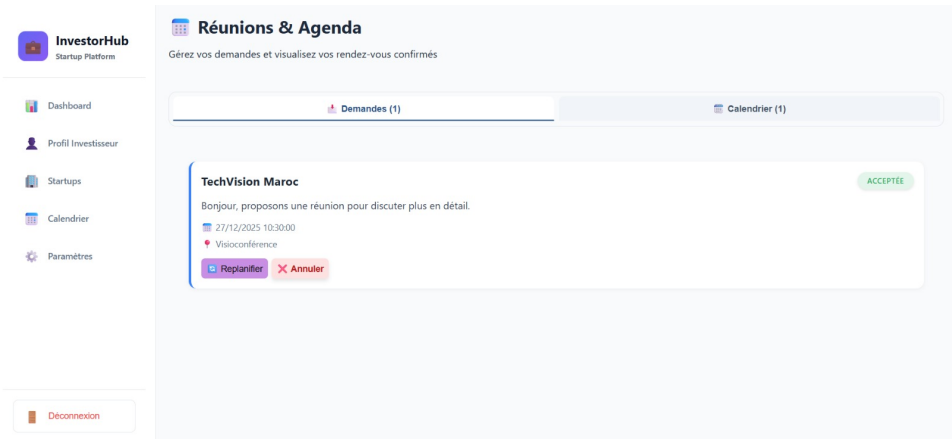


FIGURE 3.25 – \*  
(d) Gestion des demandes de réunion

FIGURE 3.26 – Interface Complète de Gestion du Calendrier et des Réunions (Vue Investisseur)

## Description :

L'interface calendrier permet aux investisseurs de gérer leurs rendez-vous avec les start-ups et de planifier de nouvelles réunions suite aux connexions acceptées.

- **(a) Vue calendrier** : Deux onglets disponibles : "Demandes (1)" et "Calendrier (1)" actif. Calendrier mensuel de Décembre 2025 avec navigation (Aujourd'hui, < >). Sélecteur de vues : Mois, Semaine, Jour, Agenda. Grille hebdomadaire (Sun-Sat) affichant les dates du mois. Réunion planifiée visible le samedi 27 décembre sous forme de pastille bleue "TechVision Maroc" avec heure "14 :30" et icône de vidéoconférence indiquant le mode de réunion. Les jours 15-16 sont mis en évidence avec fond bleu clair. Cette vue permet de visualiser rapidement l'agenda des réunions confirmées avec les start-ups.
- **(b) Modal planification initiale** : Pop-up "Planifier une Réunion" s'ouvrant depuis la page Startups après acceptation d'une connexion. Formulaire avec trois champs : "Date et heure" (format jj/mm/aaaa – :-) avec sélecteur de date/heure, "Lieu (optionnel)" avec placeholder "Ex : Google Meet, Casablanca, etc." pour spécifier le mode (visio, présentiel, téléphone), et "Message" pré-rempli avec "Bonjour, proposons une réunion pour discuter plus en détail." personnalisable par l'investisseur. Deux boutons : "Annuler" (gris) et "Planifier" (bleu foncé) pour envoyer la proposition à la start-up.
- **(c) Modal replanification** : Pop-up "Replanifier la réunion" avec icône calendrier, permettant de proposer une nouvelle date suite à un conflit ou modification. Instruction "Proposez une nouvelle date et heure à la startup." Champ "Nouvelle date & heure" (format jj/mm/aaaa – :-), et champ "Message (optionnel)" avec placeholder "Ex : Je suis disponible le 5 janvier à 14h." pour expliquer le changement. Boutons "Annuler" (gris clair) et "Envoyer la proposition" (bleu foncé) pour notifier la start-up du report.
- **(d) Onglet demandes** : Page "Réunions & Agenda" avec sous-titre "Gérez vos demandes et visualisez vos rendez-vous confirmés". Onglet "Demandes (1)" actif affichant les réunions en attente ou confirmées. Carte réunion pour "TechVision Maroc" avec badge vert "ACCEPTÉE" indiquant que la start-up a confirmé. Message "Bonjour, proposons une réunion pour discuter plus en détail." Date et heure "27/12/2025 10 :30 :00", et lieu "Visioconférence" avec icône. Deux boutons d'action : "Replanifier" (rose) pour modifier la date/heure via le modal (c), et "Annuler" (rouge) pour supprimer définitivement la réunion.

Cette interface centralise la gestion complète du calendrier investisseur : visualisation mensuelle des réunions, planification de nouveaux rendez-vous avec les start-ups connectées, replanification en cas de conflit, et gestion des demandes avec confirmation/annulation. Le système intègre différents modes de réunion (visioconférence, présentiel) et permet une communication fluide entre investisseurs et entrepreneurs.

## 3.5 Conclusion

Ce chapitre a permis de présenter la matérialisation finale du projet à travers son architecture microservices et ses différentes interfaces utilisateur. Chaque composant a été conçu et développé pour répondre précisément aux cas d'usage et aux workflows décrits dans le diagramme de cas d'utilisation, tout en s'appuyant sur la structure de données formalisée par le diagramme de classes.

L'architecture microservices mise en place garantit la scalabilité, la maintenabilité et l'évolutivité du système. La séparation en cinq services indépendants (Auth, Startup, Pitch, Investor, Analytics) orchestrés par une API Gateway assure une résilience accrue et permet des déploiements indépendants. La communication via OpenFeign et l'authentification JWT distribuée constituent une base solide

pour une application moderne et sécurisée.

L'analyse des interfaces a démontré comment les principes de conception UX/UI ont été appliqués pour créer une expérience utilisateur fluide, cohérente et efficace. Les fonctionnalités essentielles—comme la gestion des profils de start-ups, la génération automatique de pitches via intelligence artificielle, le système de matching avec les investisseurs, et les tableaux de bord analytiques—sont accessibles via des écrans intuitifs qui guident l'utilisateur et minimisent la charge cognitive.

L'intégration de l'intelligence artificielle pour la génération de pitches constitue la valeur ajoutée majeure de cette plateforme, permettant aux entrepreneurs de créer en quelques minutes des pitches professionnels qui nécessiteraient normalement plusieurs heures de travail. Le système de matching intelligent facilite quant à lui la mise en relation avec les investisseurs pertinents, optimisant ainsi les chances d'obtenir un financement.

La réalisation de cette plateforme marque l'aboutissement du processus de développement et valide la faisabilité technique de la solution. L'application web finale est non seulement fonctionnelle, mais aussi alignée sur les standards modernes de développement avec son architecture distribuée, offrant ainsi une base robuste et évolutive pour une éventuelle industrialisation ou pour enrichir davantage l'écosystème entrepreneurial marocain.

## Apports du Projet

**Apports Professionnels :** Ce projet de fin d'année nous a permis d'acquérir une expertise technique solide en développement d'applications distribuées modernes avec une architecture microservices. Nous avons maîtrisé les concepts avancés d'architecture orientée services avec Spring Boot pour le backend et React.js pour le frontend, notamment la communication inter-services via OpenFeign, la gestion d'authentification JWT distribuée, et l'isolation des données par schémas PostgreSQL. L'intégration d'une API d'intelligence artificielle pour la génération automatique de pitches nous a familiarisés avec les enjeux pratiques de l'IA générative et du prompt engineering. L'utilisation d'outils professionnels comme Maven, Git, Postman, et les méthodologies de développement agile ont enrichi notre profil technique. L'exposition aux enjeux réels de l'écosystème entrepreneurial marocain et aux problématiques de financement des start-ups nous a donné une vision concrète des besoins du marché.

**Apports Personnels :** Cette expérience a renforcé notre capacité de collaboration en binôme et notre résolution de problèmes complexes dans un contexte académique exigeant. Nous avons développé notre autonomie technique tout en apprenant à nous répartir efficacement les tâches selon nos spécialisations respectives (backend/frontend). La gestion du temps et des priorités dans un projet avec des échéances strictes et en parallèle d'autres projets académiques a considérablement amélioré nos compétences organisationnelles. Cette collaboration nous a également permis de développer des qualités humaines essentielles telles que l'esprit d'équipe, la communication technique, l'entraide et la capacité à trouver des compromis constructifs face aux choix architecturaux.

**Comparaison Théorie/Pratique :** L'écart entre la théorie académique et la pratique de développement s'est révélé instructif. Si les concepts fondamentaux enseignés à l'EMSI restent valides, leur application concrète nécessite une adaptation aux contraintes réelles : performance des API, sécurité des microservices, gestion des erreurs distribuées, latence réseau et évolutivité du système. Les problématiques d'intégration d'intelligence artificielle externe, d'optimisation des algorithmes de matching, et de gestion des timeouts dans les communications inter-services ne sont que partiellement abordées en cours théoriques. Cette expérience a souligné l'importance de l'apprentissage continu, de la consultation de documentation technique, et de la veille technologique dans notre domaine en constante évolution. Cette mise en pratique nous a confortés dans notre choix de carrière et renforcé notre intérêt pour le développement d'applications distribuées et l'innovation technologique au service de l'entrepreneuriat.

## Conclusion Générale

Ce projet de fin d'année nous a permis de concevoir et développer une plateforme d'incubation innovante pour start-ups marocaines, combinant architecture microservices et intelligence artificielle. À travers ce travail collaboratif, nous avons relevé le défi de créer un système distribué complet comprenant cinq microservices indépendants orchestrés par une API Gateway, une interface utilisateur React moderne et responsive, et une intégration IA pour la génération automatique de pitches d'investissement.

Ce projet nous a également permis de développer des compétences essentielles en architecture microservices, notamment en matière de conception distribuée, de communication inter-services, et d'isolation des données par schémas PostgreSQL. Nous avons appris à gérer efficacement les contraintes de sécurité (JWT), de performance (timeout, retry), et de résilience (fallback) dans un environnement distribué, en restant flexibles et adaptables face aux défis techniques rencontrés lors du développement.

L'intégration de l'intelligence artificielle constitue l'innovation majeure de notre projet, permettant de réduire drastiquement le temps de création de pitches professionnels (de plusieurs heures à quelques minutes). Le système de matching intelligent facilite la mise en relation entre entrepreneurs et investisseurs, contribuant ainsi potentiellement au développement de l'écosystème entrepreneurial marocain. En conclusion, ce projet de fin d'année à l'**École Marocaine des Sciences de l'Ingénieur (EMSI)** nous a offert une expérience académique enrichissante qui nous prépare efficacement pour notre future carrière professionnelle. Il nous a permis de consolider nos compétences techniques en développement full-stack moderne (Spring Boot/React.js), de maîtriser les concepts d'architecture microservices distribuée, et de développer notre capacité à concevoir des solutions scalables, sécurisées et innovantes intégrant l'intelligence artificielle. Cette expérience nous a également sensibilisés aux enjeux réels de l'entrepreneuriat et du financement des start-ups au Maroc. Nous sommes reconnaissants envers notre encadrant **M. LARHLIMI** pour son guidage précieux tout au long de ce projet, et confiants dans notre capacité à apporter une contribution significative dans le domaine du développement d'applications distribuées et de l'innovation technologique.

# Bibliographie

- [1] Richardson, C. *Microservices Patterns : With examples in Java*. 1st ed. Shelter Island : Manning Publications, 2018.
- [2] Newman, S. *Building Microservices : Designing Fine-Grained Systems*. 2nd ed. Sebastopol : O'Reilly Media, 2021.
- [3] Booch, G., Rumbaugh, J., & Jacobson, I. *The Unified Modeling Language User Guide*. 2nd ed. Boston : Addison-Wesley, 2005.
- [4] Fowler, M. *UML Distilled : A Brief Guide to the Standard Object Modeling Language*. 3rd ed. Boston : Addison-Wesley, 2003.
- [5] Larman, C. *Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design*. 3rd ed. Upper Saddle River : Prentice Hall, 2004.
- [6] Walls, C. *Spring Boot in Action*. 2nd ed. Shelter Island : Manning Publications, 2022.
- [7] Fielding, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.
- [8] Object Management Group. *OMG Unified Modeling Language (UML) Specification*. Version 2.5.1. Milford : OMG, 2017. [En ligne]. Disponible : <https://www.omg.org/spec/UML/> [Consulté le :04 novembre 2025].
- [9] VMware, Inc. *Spring Framework Documentation*. Version 6.x. [En ligne]. Disponible : <https://docs.spring.io/spring-framework/reference/> [Consulté le : 10 novembre 2025].
- [10] VMware, Inc. *Spring Boot Reference Documentation*. Version 3.x. [En ligne]. Disponible : <https://docs.spring.io/spring-boot/docs/current/reference/html/> [Consulté le : 10 novembre 2025].
- [11] Meta Platforms, Inc. *React Documentation*. Version 18.x. [En ligne]. Disponible : <https://react.dev/> [Consulté le : 14 novembre 2025].
- [12] PostgreSQL Global Development Group. *PostgreSQL 16 Documentation*. [En ligne]. Disponible : <https://www.postgresql.org/docs/16/> [Consulté le :10 novembre 2025].
- [13] OpenFeign. *OpenFeign Documentation*. [En ligne]. Disponible : <https://github.com/OpenFeign/feign> [Consulté le : 11 novembre 2024].
- [14] Jones, M., Bradley, J., & Sakimura, N. *JSON Web Token (JWT)*. RFC 7519. Internet Engineering Task Force, May 2015. [En ligne]. Disponible : <https://datatracker.ietf.org/doc/html/rfc7519>
- [15] Brynjolfsson, E. & McAfee, A. *The Second Machine Age : Work, Progress, and Prosperity in a Time of Brilliant Technologies*. New York : W. W. Norton & Company, 2014.
- [16] Ministère de l'Industrie et du Commerce du Maroc. *Stratégie Nationale pour l'Entrepreneuriat et l'Innovation 2021-2025*. Rabat, 2021.