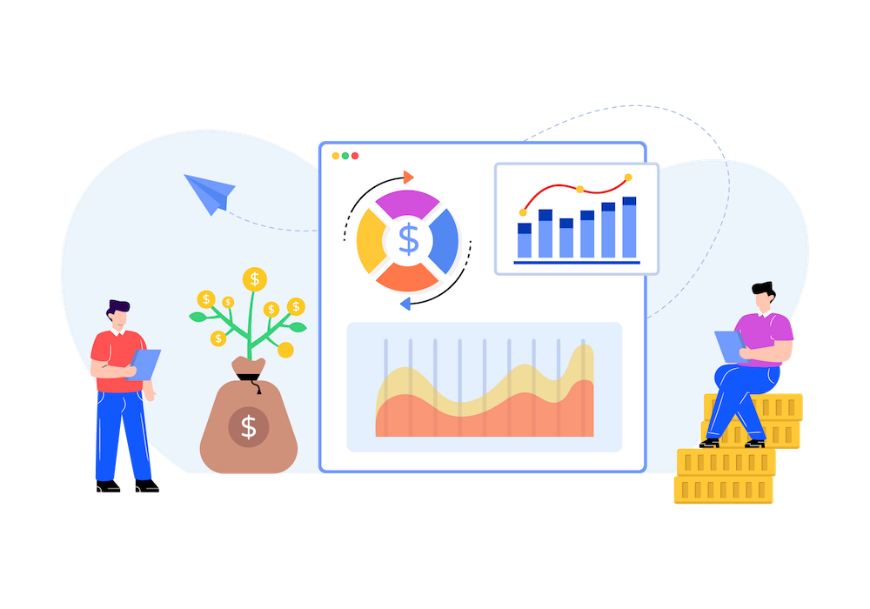


**Plateforme qui met en harmonie les startups et les investisseurs**

Rapport



Réalisé par :

**El fich Rihab**

**Amchaouech wiam**

**El Azouan Aya**

**Akiirn Amal**

**El baraka Nouhaila**

Encadré par :

### Mr. Abdelali Ouchetachen

Encadré par :

### Mr. Ghailani

Table des matières

[Mr. Abdelali Ouchetachen 0](file:///C:\Users\I511\OneDrive\Documents\rapport%5b1%5d%5b1%5d.docx#_Toc160139500)

[Mr. Ghailani 0](file:///C:\Users\I511\OneDrive\Documents\rapport%5b1%5d%5b1%5d.docx#_Toc160139501)

[1. Introduction : 3](#_Toc160139502)

[2. Objectifs : 3](#_Toc160139503)

[3.Conception General du Projet 4](#_Toc160139504)

[1. Diagrmme de cas Utilisateur : 4](#_Toc160139505)

[3. Diagrmme d’activités 5](#_Toc160139506)

[4. Architecture de la Plateforme 6](#_Toc160139507)

[Service de Messagerie : 6](#_Toc160139508)

[Service d'Appels Vocaux : 6](#_Toc160139509)

[Service de Gestion des Utilisateurs et des Organisations (authentification, autorisation,) : 6](#_Toc160139510)

[Service de Gestion des postes : 6](#_Toc160139511)

[Service de Partage de Fichiers : 6](#_Toc160139512)

[Service de Gestion des Notifications : 6](#_Toc160139513)

[Service d’un algorithme de recommandation : 6](#_Toc160139514)

[5.Technologies Utilisées 7](#_Toc160139515)

[1. Frameworks Back-End : 7](#_Toc160139516)

[ .NET 7](#_Toc160139517)

[ Spring Boot 7](#_Toc160139518)

[ Python 7](#_Toc160139519)

[2. Frameworks Front-End : 7](#_Toc160139520)

[ Angular 7](#_Toc160139521)

[3. Conteneurisation : 7](#_Toc160139522)

[ Docker : 7](#_Toc160139523)

[4. Base de Données : 8](#_Toc160139524)

[ SqlServer, PostgreSQL, MongoDB : 8](#_Toc160139525)

[5. Communication entre les Services : 8](#_Toc160139526)

[ RabbitMQ : 8](#_Toc160139527)

[ Web Services : 8](#_Toc160139528)

[6. Gestion de Projet : 8](#_Toc160139529)

[ Jira : 8](#_Toc160139530)

[7. Tests Unitaires : 8](#_Toc160139531)

[xUnit, NUnit : 8](#_Toc160139532)

[8. Déploiement Continu (CI/CD) : 9](#_Toc160139533)

[ GitHub Actions : 9](#_Toc160139534)

[ GitLab : 9](#_Toc160139535)

[ Kubernetes : 9](#_Toc160139536)

[6 . Sécurité et Confidentialité 9](#_Toc160139537)

[Authentification : 9](#_Toc160139538)

[Autorisation : 10](#_Toc160139539)

[Chiffrement des Données : 10](#_Toc160139540)

[7. Planification (Provesoire) 10](#_Toc160139541)

# Introduction :

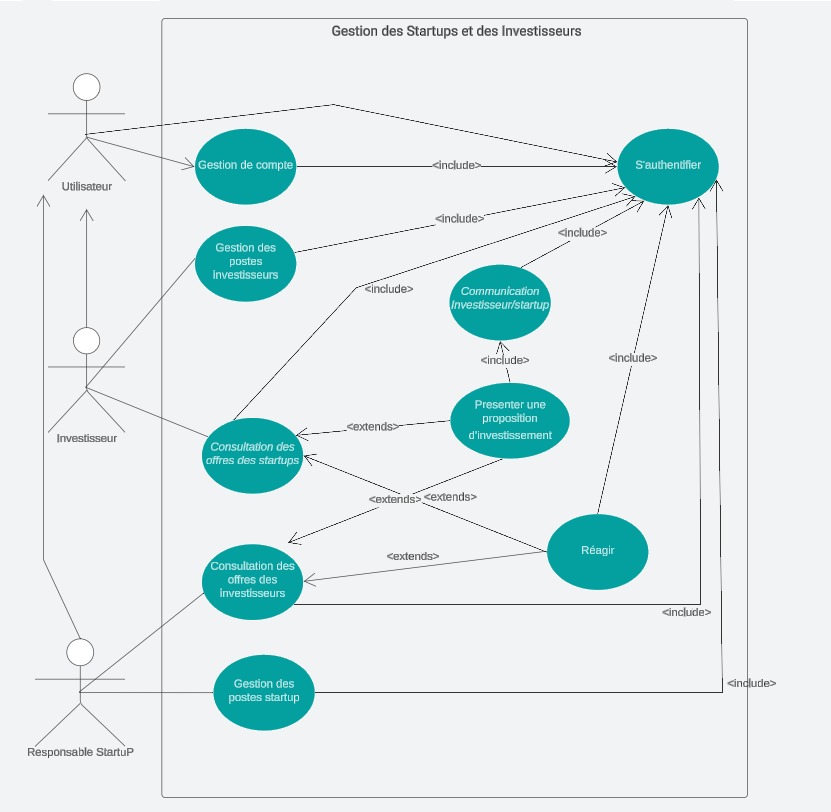
Dans un contexte de croissance exponentielle des startups et d'un marché de l'investissement en perpétuelle évolution, la création d'une plateforme dédiée à mettre en harmonie les startups et les investisseurs devient un impératif. Cette initiative vise à répondre aux besoins croissants des deux parties, en facilitant leurs rencontres et en favorisant un écosystème propice à la croissance et à la réussite.

# Objectifs :

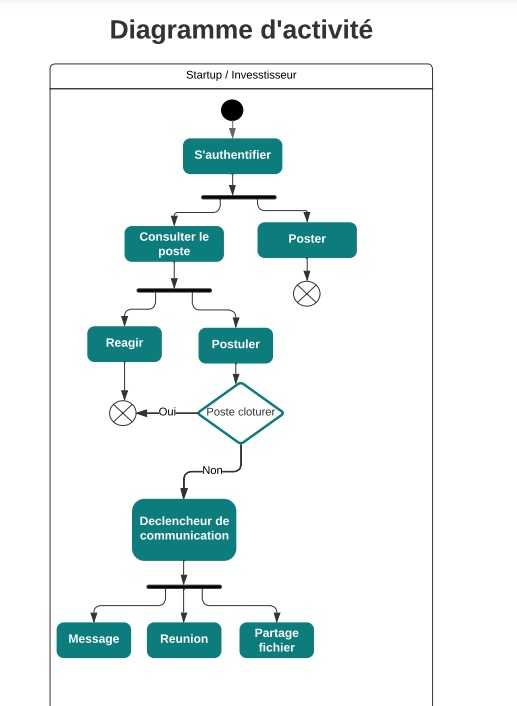
* Faciliter les rencontres entre startups et investisseurs.
* Accélérer la croissance des startups grâce à un soutien financier.
* Maximiser les opportunités d'investissement pour les investisseurs.
* Construire une communauté dynamique favorisant la collaboration et le succès mutuel.
* Assurer la sécurité des données et la confidentialité des utilisateurs.
* Offrir une expérience utilisateur fluide et intuitive.
* Offrir des fonctionnalités de messagerie, d'appels vocaux, de partage de fichiers, de planification de réunions, gestion des postes.

# 3.Conception General du Projet

## 1. Diagrmme de cas Utilisateur :



## Diagrmme d’activités



# 4. Architecture de la Plateforme

La plateforme sera basée sur une architecture de micro services pour garantir la capabilité, la flexibilité et la résilience. Les micro services principaux seront :

Service de Messagerie :

ce service permettra aux utilisateurs de communiquer entre eux par des messages texte. Il gérera l'envoi, la réception et potentiellement le stockage sécurisé des messages. Il peut également inclure des fonctionnalités telles que la recherche de messages.

Service d'Appels Vocaux :

ce service permettra aux utilisateurs de passer des appels vocaux entre eux via la plateforme. Il gérera la connexion entre les utilisateurs pour les appels vocaux, ainsi que les fonctionnalités telles que la mise en attente, la mise en sourdine et la fin des appels.

## Service de Gestion des Utilisateurs et des Organisations (authentification, autorisation,) :

Ce service gérera l'authentification et l'autorisation des utilisateurs sur la plateforme. Il comprendra des fonctionnalités telles que la création de comptes utilisateurs, la gestion des profils, la connexion sécurisée, la réinitialisation des mots de passe et la gestion des autorisations d'accès aux fonctionnalités de la plateforme.

## Service de Gestion des postes :

Ce service gérera les annonces ou les publications de postes de travail par les utilisateurs sur la plateforme. Il inclura des fonctionnalités telles que la création, la modification, la suppression et la recherche de postes, ainsi que la gestion des candidatures.

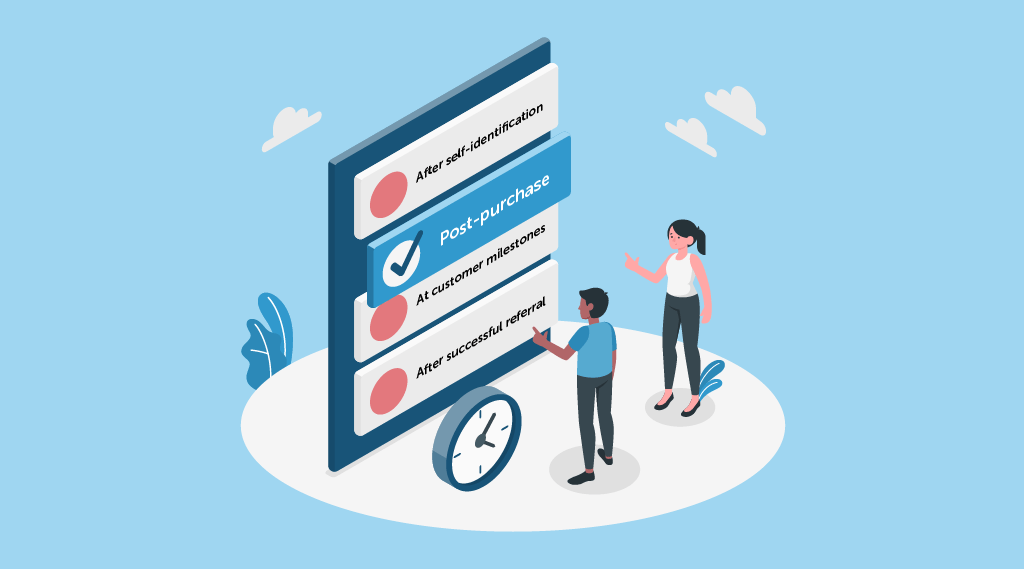
## 5 sites gratuits pour partager des fichiers entre deux ordinateursService de Partage de Fichiers :

Ce service permettra aux utilisateurs de partager des fichiers avec d'autres utilisateurs sur la plateforme. Il gérera le stockage sécurisé des fichiers, ainsi que les fonctionnalités de téléchargement, de partage et de gestion des autorisations d'accès aux fichiers.

## Les BU de Lorraine lancent un nouveau service de messagerie instantanée - IUT Saint-DiéService de Gestion des Notifications :

 Ce service gérera l'envoi de notifications aux utilisateurs de la plateforme. Il inclura des fonctionnalités telles que la configuration des préférences de notification, l'envoi de notifications en temps réel et la gestion des notifications archivées.

## Service d’un algorithme de recommandation :

Ce service utilisera des algorithmes pour recommander des startups aux investisseurs et vice versa, en fonction de leurs préférences, de leur historique d'interaction sur la plateforme et d'autres facteurs pertinents. Il peut également inclure des fonctionnalités de personnalisation des recommandations et d'évaluation de la pertinence des suggestions.

# 5.Technologies Utilisées

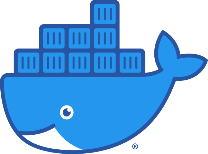
## 1. Frameworks Back-End :

* .NET **:** En utilisant .NET pour le service d'authentification et de gestion des utilisateurs, ainsi que pour le service de gestion des posts, nous bénéficions de la robustesse et de la sécurité offertes par cette plateforme. .NET offre également une intégration aisée avec d'autres technologies Microsoft, ce qui garantit une cohérence et une compatibilité optimales avec notre infrastructure existante. L'utilisation de .NET permettra d'améliorer la stabilité et la sécurité de nos microservices sensibles, tout en simplifiant le développement et la maintenance du code.
* Spring Boot **:** L'emploi de Spring Boot pour les services d'appel vidéo et d'échange de fichiers assure une facilité de développement et une prise en charge des tâches complexes telles que le streaming vidéo et le transfert de fichiers. Grâce à son architecture légère et à ses fonctionnalités de configuration automatique, Spring Boot permet un développement rapide et efficace des microservices, tout en garantissant une performance optimale. L'utilisation de Spring Boot contribuera à améliorer la réactivité et la fiabilité de nos services d'appel vidéo et d'échange de fichiers.
* Python **:** Le choix de Python pour le service de mise en correspondance entre les posts offre plusieurs avantages, notamment sa simplicité, sa lisibilité du code et ses capacités d'analyse de données. Python est largement utilisé dans le domaine de l'analyse de données et de l'apprentissage automatique, ce qui en fait un choix judicieux pour les services nécessitant des algorithmes de correspondance complexes. L'utilisation de Python permettra d'améliorer la précision et l'efficacité de notre service de mise en correspondance, en permettant une analyse plus avancée des données.

## 2. Frameworks Front-End :

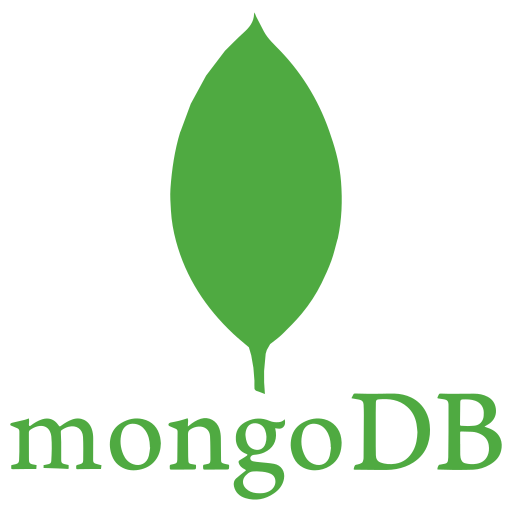
* Angular **:** En utilisant Angular pour la partie client de l'application, nous bénéficions de sa capacité à créer des applications web évolutives et réactives. Angular offre une architecture modulaire, des fonctionnalités de liaison de données bidirectionnelle et une large gamme de bibliothèques et de modules complémentaires, ce qui facilite le développement et la maintenance de l'interface utilisateur. L'utilisation d'Angular permettra d'améliorer l'expérience utilisateur en offrant une interface conviviale et réactive, tout en garantissant une évolutivité et une maintenabilité optimales de notre application.

## 3. Conteneurisation :

* Docker : En conteneurisant tous nos microservices avec Docker, nous garantissons une portabilité, une isolation et une facilité de déploiement optimales. Les conteneurs Docker permettent de packager les applications et leurs dépendances dans des environnements isolés, ce qui facilite le déploiement sur différentes plateformes et garantit une cohérence entre les environnements de développement, de test et de production. L'utilisation de Docker améliorera l'efficacité opérationnelle de notre infrastructure en simplifiant le déploiement et la gestion des microservices, tout en assurant une isolation et une sécurité renforcées.

4. Base de Données :

* SqlServer, PostgreSQL, MongoDB : En utilisant SqlServer, PostgreSQL et MongoDB pour la gestion des données structurées, relationnelles et non structurées, nous choisissons la technologie la mieux adaptée à chaque type de données.

SqlServer offre une robustesse et une performance pour la gestion des données structurées, PostgreSQL fournit une gestion relationnelle avancée,

tandis que MongoDB permet de gérer efficacement des données non structurées. L'utilisation de plusieurs bases de données nous permettra d'optimiser la performance et la flexibilité de nos microservices en choisissant la technologie la mieux adaptée à chaque cas d'utilisation.

## 5. Communication entre les Services :

* RabbitMQ : En utilisant RabbitMQ pour la communication asynchrone entre les microservices, nous garantissons une intégration efficace et une évolutivité accrue. RabbitMQ offre une messagerie fiable et robuste basée sur le modèle de file d'attente, ce qui permet de découpler les microservices et d'améliorer la résilience et la scalabilité de notre architecture. L'utilisation de RabbitMQ permettra d'améliorer la réactivité et la fiabilité de nos microservices en assurant une communication asynchrone et décentralisée.
* Web Services : En utilisant des Web Services pour la communication synchrone, notamment pour l'authentification entre tous les services, nous garantissons une communication directe et immédiate entre les composants du système. Les Web Services offrent une architecture standardisée et interopérable, ce qui facilite l'intégration et l'interopérabilité entre les différents composants du système. L'utilisation de Web Services permettra d'améliorer la cohérence et l'efficacité de nos microservices en assurant une communication synchrone et directe entre les services.

## 6. Gestion de Projet :

* Jira : En utilisant Jira pour la gestion de projet, la planification des tâches et le suivi des problèmes, nous bénéficions d'une visibilité et d'un contrôle complets sur le développement du projet. Jira offre des fonctionnalités avancées de suivi des tâches, de gestion des versions et de collaboration en équipe, ce qui facilite la coordination et la gestion des projets complexes. L'utilisation de Jira permettra d'améliorer l'efficacité et la transparence de notre processus de développement en offrant une plateforme centralisée pour la gestion de projet et la collaboration en équipe.

## 7. Tests Unitaires :

xUnit, NUnit : En utilisant xUnit et NUnit pour les tests unitaires, nous garantissons la qualité et la fiabilité du code. Les tests unitaires permettent de détecter et de corriger les erreurs rapidement, ce qui contribue à la stabilité et à la robustesse de nos microservices. En effectuant des tests unitaires réguliers à l'aide de ces frameworks, nous assurons que chaque composant fonctionne comme prévu, ce qui renforce la confiance dans le système global.

## 8. Déploiement Continu (CI/CD) :

* GitHub Actions : En utilisant GitHub Actions pour l'intégration continue et le déploiement continu, nous automatisons efficacement les workflows de développement, de test et de déploiement de nos microservices. Cette automatisation nous permet de détecter et de corriger les problèmes rapidement, tout en assurant un déploiement rapide et fiable des nouvelles fonctionnalités. Grâce à GitHub Actions, nous pouvons définir des flux de travail personnalisés en utilisant la syntaxe YAML, déclenchés par des événements tels que des poussées de code ou des pull requests. Cette automatisation simplifie le processus de développement et de déploiement, améliorant ainsi l'efficacité globale du cycle de vie de développement.
* GitLab : GitLab est également une plateforme web de contrôle de version et de gestion de code source. Tout comme GitHub, GitLab prend en charge les dépôts Git et propose des fonctionnalités de collaboration et de gestion de projet. Cependant, GitLab se distingue en offrant des fonctionnalités supplémentaires, notamment l'intégration continue et le déploiement continu (CI/CD), le registre de conteneurs, et plus encore. GitLab peut être utilisé en tant que service auto-hébergé ou basé sur le cloud.

**Améliorations Potentielles pour le Déploiement Continu :**

* Kubernetes : L'introduction de Kubernetes en complément de Docker pourrait permettre une gestion plus avancée et une orchestration automatisée des conteneurs. Kubernetes offre des fonctionnalités avancées telles que le scaling automatique, la gestion des ressources et la tolérance aux pannes, ce qui garantit une haute disponibilité et une résilience accrue de nos microservices.

En mettant en œuvre ces améliorations potentielles, nous pourrons renforcer davantage la fiabilité, la performance et l'évolutivité de nos microservices, garantissant ainsi une expérience utilisateur optimale et une valeur ajoutée continue pour nos utilisateurs finaux.

# 

# 6 . Sécurité et Confidentialité

La sécurité des données des utilisateurs est une priorité absolue dans notre plateforme, et nous mettons en œuvre des pratiques de sécurité robustes pour garantir la protection des informations sensibles. Voici comment nous assurons la sécurité des données :

Authentification : Nous utilisons des mécanismes d'authentification solides pour vérifier l'identité des utilisateurs et garantir qu'ils ont les autorisations appropriées pour accéder aux fonctionnalités de la plateforme. Cela peut inclure l'utilisation de protocoles d'authentification tels que OAuth, OpenID Connect ou JWT (JSON Web Tokens), ainsi que des méthodes d'authentification multifactorielle pour renforcer la sécurité des comptes utilisateur.

Autorisation : Une fois qu'un utilisateur est authentifié, nous appliquons des contrôles d'autorisation pour déterminer les actions auxquelles il est autorisé à effectuer dans la plateforme. Cela implique la définition de rôles et de permissions spécifiques, ainsi que la mise en place de politiques d'accès granulaires pour limiter l'accès aux données sensibles uniquement aux utilisateurs autorisés.

Chiffrement des Données : Toutes les données sensibles stockées dans notre système sont chiffrées pour empêcher tout accès non autorisé. Nous utilisons des algorithmes de chiffrement robustes tels que AES (Advanced Encryption Standard) pour chiffrer les données au repos, ainsi que des protocoles de chiffrement SSL/TLS pour sécuriser les communications entre les clients et les serveurs.

Protection contre les Attaques : Nous mettons en place des mesures de protection contre les attaques courantes telles que les attaques par injection SQL, les attaques par force brute, les attaques de type cross-site scripting (XSS) et cross-site request forgery (CSRF), ainsi que les attaques de déni de service (DDoS). Cela peut inclure l'utilisation de pare-feu, de filtres de sécurité, de mécanismes de détection d'intrusion et de surveillance active du trafic réseau pour détecter et bloquer les activités suspectes.

En mettant en œuvre ces pratiques de sécurité robustes, nous nous engageons à garantir la confidentialité, l'intégrité et la disponibilité des données des utilisateurs sur notre plateforme. Nous restons également vigilants en surveillant constamment les menaces émergentes et en mettant à jour nos mesures de sécurité pour faire face aux nouvelles vulnérabilités et aux attaques potentielles.

# 

# 7. Planification (Provesoire)

