

WEFUND

CAHIER DES CHARGES

Contexte

La société WeFund est une startup souhaitant lancer une plateforme de financement participatif (crowdfunding) inspirée de services comme Ulule ou KissKissBankBank.

Cette plateforme permettra :

- à des porteurs de projet de proposer des campagnes de financement
- à des contributeurs de soutenir financièrement ces projets
- à la plateforme d'assurer la confiance, la transparence et la gestion des paiements

Elle souhaite lancer son MVP (Minimum Viable Product) en septembre 2026. Pour cela, elle souhaite réaliser un **POC (Proof of Concept) fonctionnel** d'ici fin mars afin de démontrer la faisabilité technique et la cohérence du système d'information.

Cette première étape vise à poser les premières fondations de la gestion des projets et des contributions des utilisateurs.

Le MVP ajoutera la dimension de recommandation des projets suivant leur popularité **et ...**

Le CTO a présélectionné des prestataires pour réaliser les différents projets, ainsi que les technologies qui pourraient servir à constituer le SI. Dans la mesure du possible, la DSI préfèrera les solutions gratuites et soumises à licence open source, ou les solutions SaaS freemium.

Livrables attendus

Les livrables attendus sont :

- un repository Git (Github ou Gitlab) public contenant le code source, les tests et toute documentation que vous jugerez utile d'ajouter, au format markdown.
- une URL permettant d'accéder à une version déployée du POC, qui servira à la recette du client (ou le cas échéant, une procédure pour démarrer le POC en local)

Considérations techniques globales

- Les notions vues en cours devront apparaître dans le projet (typage avancé, gestion des erreurs, testing, architecture, ...)
- L'architecture globale sera sous forme de micro-services. Ces services communiqueront entre eux via un message broker ([RabbitMQ](#))
- Pour l'infrastructure (base de données et déploiement des services), je peux vous conseiller [render.com](#) qui possède une offre free. Vous pouvez utiliser une autre solution si vous le souhaitez. La conteneurisation est recommandée.
- Si vous utilisez de l'IA, documentez vos prompts, gardez un esprit critique envers ce que l'IA produit et maintenez la qualité du code malgré tout !

Dans la mesure du possible, mettez en place de l'observabilité simple (logging structuré)

Architecture globale attendue

La plateforme reposera sur plusieurs applications interconnectées :

- Microservice gestion des projets / campagnes de financement
- Microservice contributions / paiements / utilisateurs
- Application web frontend
- Application dashboard

Rôles fonctionnels

Porteur de projet

Utilisateur proposant une campagne.

Contributeur

Utilisateur finançant un projet.

Administrateur

Utilisateur supervisant la plateforme.

Visiteur anonyme

Utilisateur consultant les campagnes.

Le terme « utilisateur » désigne, indistinctement, n'importe lequel de ces rôles.

Le terme « système » désigne, le système informatique, indistinctement du service concerné.

Dans la description des projets le terme « prestataire » désigne la société chargée de la réalisation du projet, le terme « client » désigne la société WeFund.

Projet 1 : Micro-service gestion des projets et campagnes

Description

Ce service gère le cycle de vie des projets et des campagnes de financement.

User Stories

Story 1

En tant que porteur de projet, je peux créer un projet afin de présenter mon initiative à la communauté.

Story 2

En tant que porteur de projet, je peux créer une campagne de financement pour mon projet afin de le financer.

Story 3

En tant que porteur de projet, je peux modifier une campagne tant qu'elle n'est pas validée par l'administrateur.

Story 4

En tant qu'utilisateur, je peux consulter les campagnes afin de découvrir des projets.

Story 5

En tant que système, je peux clôturer automatiquement une campagne lorsque la date limite est atteinte.

Story 6

En tant que porteur de projet, je peux publier des actualités sur ma campagne afin d'informer les contributeurs de son avancement.

Story 7

En tant que porteur de projet, je peux consulter les statistiques de ma campagne afin de suivre sa progression.

Story 8

En tant que porteur de projet, je peux dupliquer une campagne terminée afin de relancer un financement.

Règles de gestion

RG1 : Une campagne possède au minimum :

- titre

- description
- objectif financier
- date de fin
- porteur identifié.

RG2 : Un projet possède au minimum :

- titre
- description
- photo

RG3 : Une campagne ne peut plus être modifiée après publication.

RG4 : Une campagne peut être :

- en brouillon
- en attente de validation (par l'administrateur)
- active
- réussie
- échouée (campagne arrivée à échéance sans avoir remplie l'objectif)
- refusée (campagne ayant été refusée par l'administrateur)

Technologies retenues

Démarche

Les communications synchrones via API REST sont autorisées.

Cependant, les événements métier inter-services devraient préférentiellement utiliser une communication asynchrone (message broker), afin d'améliorer le découplage, la résilience et la scalabilité du SI.

L'approche contract-first est à adopter afin de communiquer efficacement avec les autres équipes. L'architecture hexagonale est encouragée.

Framework et langage

L'application doit être réalisée en Typescript et exécutée à l'aide du runtime Node.js, en version 24.

Le prestataire est libre d'utiliser ou non un framework, bien que cela soit encouragé par le client.

Les frameworks acceptés par le client sont :

- Express, si vous choisissez d'utiliser un framework « unopinionated »
<https://expressjs.com/>
- NestJS, si vous choisissez d'utiliser un framework « opinionated »
<https://docs.nestjs.com/>

- Fastify <https://fastify.dev/>

Base de données

La base de données à utiliser est PostgreSQL.

Le prestataire est libre d'utiliser ou non un ORM, bien que cela soit encouragé par le client.

Projet 2 : Microservice contributions / paiements / utilisateurs

Description

Ce service a pour but de gérer les contributions financières pour les projets et les paiements associés.

User Stories

Story 1

En tant que contributeur, je peux financer une campagne afin de soutenir un projet.

Story 2

En tant que contributeur, je peux consulter mes contributions afin de suivre mes engagements.

Story 3

En tant que système, je peux déclencher un remboursement si la campagne échoue.

Story 4

En tant que contributeur, je peux demander un remboursement de ma contribution afin de retirer mon soutien du projet.

Story 5

En tant que contributeur, je peux modifier le montant de ma contribution tant que la campagne est active afin d'ajuster mon soutien au projet.

Story 6

En tant qu'utilisateur, je peux créer un compte afin d'utiliser la plateforme.

Story 7

En tant qu'utilisateur, je peux m'authentifier afin d'accéder à mes fonctionnalités.

Story 8

En tant qu'administrateur, je peux modérer une campagne afin d'assurer la conformité.

Règles de gestion

RG1 : Une contribution est associée :

- à un utilisateur
- à une campagne
- à un montant.

RG2 : Une contribution est toujours datée.

RG3 : La modification et annulation (remboursement) de contribution ne peuvent se faire si, et seulement si, la campagne est active.

RG4 : Les paiements et remboursements doivent être tracés.

RG5 : Les contributions restent sur un compte de séquestre jusqu'au succès de la campagne, c'est-à-dire que le porteur de projet ne touche l'argent que lorsque sa campagne est réussie (=date arrivée à échéance et objectif atteint)

RG6 : La modération reste simple pour le POC : acceptée ou refusée. Pas de notification, de commentaire ou boucle de retour.

Technologies retenues

Démarche

Les communications synchrones via API REST sont autorisées.

Cependant, les événements métier inter-services devraient préférentiellement utiliser une communication asynchrone (message broker), afin d'améliorer le découplage, la résilience et la scalabilité du SI.

L'approche contract-first est à adopter afin de communiquer efficacement avec les autres équipes.

L'architecture hexagonale est encouragée.

Framework et langage

L'application doit être réalisée en Typescript et exécutée à l'aide du runtime Node.js, en version 24.

Le prestataire est libre d'utiliser ou non un framework, bien que cela soit encouragé par le client.

Les frameworks acceptés par le client sont :

- Express, si vous choisissez d'utiliser un framework « unopinionated »
<https://expressjs.com/>
- NestJS, si vous choisissez d'utiliser un framework « opinionated »
<https://docs.nestjs.com/>
- Fastify <https://fastify.dev/>

Pour simuler les transactions financières, le prestataire est encouragé à utiliser la sanbox de Stripe (ou un équivalent). Les paiements seront évidemment simulés.

Base de données

La base de données à utiliser est PostgreSQL.

Le prestataire est libre d'utiliser ou non un ORM, bien que cela soit encouragé par le client.

Projet 3 : Application Web Frontend

Description

Ce service a pour but d'exposer une interface graphique afin que les utilisateurs puissent utiliser WeFund.

User Stories

Story 1

En tant que visiteur, je peux créer un compte depuis l'interface web afin d'accéder aux fonctionnalités de la plateforme.

Story 2

En tant qu'utilisateur, je peux me connecter via l'application web afin d'accéder à mon espace personnel.

Story 3

En tant qu'utilisateur, je peux parcourir et consulter les campagnes depuis l'application web afin de découvrir des projets.

Story 4

En tant que porteur de projet, je peux créer et gérer mes projets et campagnes depuis l'interface web afin de piloter mon financement.

Story 5

En tant qu'administrateur, je peux consulter et modérer les campagnes depuis une interface dédiée afin d'assurer leur conformité.

Story 6

En tant qu'utilisateur authentifié, je peux financer une campagne depuis l'interface web afin de soutenir un projet.

Story 7

En tant que contributeur, je peux demander le remboursement d'une contribution depuis mon espace personnel afin de retirer mon soutien.

Story 8

En tant que contributeur, je peux modifier ma contribution tant que la campagne est active.

Story 9

En tant que contributeur, je peux consulter mes contributions depuis mon espace personnel afin de suivre mes engagements.

Story 10

En tant que porteur de projet, je peux publier des actualités sur ma campagne via l'interface web afin d'informer les contributeurs.

Story 11

En tant que porteur de projet, je peux visualiser les statistiques de ma campagne dans un tableau de bord afin de suivre sa progression.

Technologies retenues

Démarche

L'architecture hexagonale est encouragée, ainsi que l'accessibilité.

Le responsive design est un plus.

Framework et langage

L'application doit être réalisée en Typescript et ReactJS et exécutée à l'aide du runtime Node.js, en version 24.

Vous avez la possibilité d'utiliser NextJS.

Vous avez la possibilité d'utiliser l'IA pour générer vos composants graphiques à partir de vos maquettes, afin d'accélérer le développement mais veillez à documenter votre démarche et vos prompts.

Projet 4 : Dashboard SI / Gestion de projet

Description

Cette application a pour objectif de fournir des graphiques de suivi à la DSI.

La réalisation de ce lot est confiée par le client à sa Direction des Services Informatiques (DSI).

La DSI est également chargée de s'assurer que l'ensemble des projets s'intègrent et soit déployé pour le POC.

La DSI est libre de fixer l'infrastructure technique d'hébergement cloud du POC.

User Stories

Story 1

En tant qu'utilisateur, je peux sélectionner un indicateur, afin de visualiser le graphique associé

Indicateurs possibles

- Nombre de campagnes actives sur une période donnée
- Montant collecté au total sur une période donnée
- Taux de succès global
- Nombre de contributions total sur une période donnée
- Nombre moyen de contributions par campagne
- Durée moyenne avant succès/échec
- Montant moyen par contribution
- Taux d'atteinte moyen des objectifs (collecté / objectif cible)
- Volume remboursé total sur une période

Limites pour le POC du logiciel

Bien que principalement destinés à la DSI, l'ensemble des graphiques sont publics.

Aucune gestion d'accès n'est donc à implémenter dans le cadre du POC.

Technologies retenues

Framework et langage

L'application doit être réalisée en Typescript.

Vous avez la possibilité d'utiliser NextJS ou tout autre librairie / framework frontend que vous jugerez pertinent.

Vous avez la possibilité d'utiliser l'IA pour générer vos composants graphiques à partir de vos maquettes, afin d'accélérer le développement mais veillez à documenter votre démarche et vos prompts.

Si vous utilisez React, les graphiques peuvent être produits, au choix du prestataire avec la librairie (ex: Rechart <https://recharts.org>)

Lexique

Projet

Initiative portée par une personne ou une organisation visant à produire un bien, un service ou une action.

Un projet peut exister indépendamment d'une campagne de financement.

Campagne de financement

Dispositif temporaire permettant de collecter des fonds pour un projet. Une campagne est caractérisée par :

- un objectif financier
- une durée
- des contreparties éventuelles (non traité dans le POC)

Une campagne est toujours associée à un projet.

Porteur de projet

Personne ou organisation responsable d'un projet et éventuellement de sa campagne de financement.

Contributeur

Personne finançant un projet via une campagne.

Un contributeur peut :

- financer plusieurs projets
- contribuer anonymement (selon règles plateforme)

Contribution

Montant financier versé par un contributeur à une campagne.

Une contribution peut être :

- associée à une contrepartie (non traité dans le POC)
- remboursée si la campagne échoue

Objectif financier

Montant cible que la campagne doit atteindre pour être considérée comme réussie.

Clôture de campagne

Fin officielle d'une campagne entraînant :

- transfert des fonds au porteur de projet, si succès
- remboursement des contributions, si échec

Administrateur plateforme

Utilisateur chargé de la validation des projets.