



Ministère de l'Enseignement  
Supérieur et de la Recherche  
Scientifique

Année Universitaire  
2024/2025



**EPI** DIGITAL SCHOOL  
Ecole Internationale Supérieure  
Privée Polytechnique de Sousse

# RAPPORT DE PROJET DE FIN D'ANNÉE

**Spécialité: Génie Logiciel**

## Intitulé

*Conception et réalisation d'une plateforme web dédiée  
aux porteurs de projets pour le financement  
participatif (crowdfunding)*

**Lieu du stage**

**EPI**

**Réalisé par**

**Mohamed Amine**

**Sakli**

**Encadré par**

**Mr Mouhieddine**

**Belguith**

## ***DÉDICACE***

---

À ma chère maman,

Cette dédicace est pour toi, qui as traversé des moments difficiles avec une force et un courage admirable. Malgré les épreuves de santé, tu as toujours été là, source de lumière et d'inspiration. Merci pour ton amour inébranlable et pour m'avoir montré que, même dans les moments les plus sombres, il y a toujours de l'espoir.

Je t'aime profondément et je suis fier d'être ton enfant. Cette réussite est aussi la tienne.

## ***REMERCIEMENTS***

---

Je tiens à exprimer mes vifs remerciements à tous ceux qui ont participé de près ou de loin à l'élaboration de ce travail. Je tiens à exprimer ma profonde gratitude et à remercier en premier lieu Monsieur Mouhedine Belguith pour l'aide et les compétences qu'il m'a apporté, pour ses disponibilités et ses encouragements à finir ce travail.

J'aimerais également remercier tous mes enseignants d'avoir partagé avec moi leur passion pour l'enseignement. J'ai grandement apprécié leur appui, implication et expérience.

Je tiens à remercier vivement les membres du jury qui ont fait l'honneur de juger ce travail et j'espère qu'il sera à la hauteur de la confiance qu'ils m'ont accordée.

Aussi, ses remerciements s'adressent à l'École EPI et au département de Génie informatique pour les efforts employés pour assurer la bonne formation.

## *Tables des matières*

---

<b>INTRODUCTION GENERALE .....</b>	<b>1</b>
<b>CHAPITRE1 : ÉTUDE PREALABLE .....</b>	<b>3</b>
INTRODUCTION.....	3
1. <i>Problématique</i> .....	3
2. <i>Étude de l'existant</i> .....	3
2.1 Analyse de l'existant .....	3
2.2 Critique de l'existant.....	6
2.3 Solution Proposée.....	6
3. <i>Analyse et spécification des besoins</i> .....	7
3.1 Identification des acteurs.....	7
3.2 Besoins fonctionnels .....	8
3.3. Besoins non fonctionnels .....	8
4. <i>Méthodologie de travail</i> .....	10
4.1.Méthodologie 2-tup .....	10
5. <i>Diagramme de Gantt</i> .....	11
CONCLUSION .....	11
<b>CHAPITRE2 : CONCEPTION .....</b>	<b>12</b>
INTRODUCTION.....	12
1. <i>Architecture logicielle</i> .....	12
2. <i>Conception UML</i> .....	13
2.1 Diagramme de cas d'utilisation .....	13
2.2 Diagramme de cas d'utilisation raffinés de l'administrateur système.....	14
2.3 Diagramme de cas d'utilisation raffinés de l'utilisateur (porteur de projet) .....	17
3. <i>Diagramme de classe globale</i> .....	19
4. <i>Diagramme de séquences</i> .....	20
4.1 Diagramme de séquence de cas d'utilisation « l'authentification ».....	20
4.2 Diagramme de séquence de cas d'utilisation « Créer une campagne ».....	21
4.3 Diagramme de séquence de cas d'utilisation « Approuver une campagne » .....	22
CONCLUSION .....	23
<b>CHAPITRE 3 : REALISATION.....</b>	<b>24</b>

INTRODUCTION.....	24
1. <i>Environnement de développement</i> .....	24
1.1 Environnement logiciel .....	24
1.2 Frameworks utilisés.....	27
1.3 Langage de programmation.....	29
2. <i>Démonstration</i> .....	30
2.1 Grille d'écran de la page de « accueil » .....	30
2.2 Grille d'écran de la page d'inscription .....	31
La figure suivante illustre l'interface de la page d'inscription. Les utilisateurs peuvent s'inscrire sur la plateforme en saisissant leurs informations personnelles, leur adresse e-mail ainsi que leur mot de passe.....	31
2.3 Grille d'écran de la page de « connexion » .....	32
2.4 Grille d'écran de la page « Discover » .....	32
2.5 Grille d'écran de la page « Créer une campagne » .....	33
2.6 Grille d'écran de la page de « Détail de la campagne » .....	34
2.7 Grille d'écran de la page de « Dashboard Admin » .....	34
CONCLUSION.....	35
<b>CONCLUSION GENERALE ET PERSPECTIVES.....</b>	<b>36</b>

## *Liste des figures*

---

Figure 1: logo de la plateforme "gofundme" .....	4
Figure 2: logo de la plateforme "kickstarter" .....	4
Figure 3: logo de la plateforme "indiegogo" .....	5
Figure 4: logo de la plateforme "Tunisia Crowdfunding" .....	5
Figure 5: logo de la plateforme "DabaPay" .....	5
Figure 6: Représentation 2tup méthode.....	10
Figure 7: Diagramme de Gantt.....	11
Figure 8: Schéma explicatif du patron MVC .....	12
Figure 9 : Diagramme de cas d'utilisation général .....	14
Figure 10 : Diagramme de cas d'utilisation raffiné de la gestion de projet .....	15
Figure 11 : Maquette de l'interface de l'administrateur de gestion des campagnes (projets)..	15
Figure 12 : Maquette de l'interface de l'administrateur de gestion d'une campagne(projet) ..	16
Figure 13 : Diagramme de cas d'utilisation raffiné de la création d'une campagne(projet) .....	18
Figure 14 : Diagramme de classe globale .....	19
Figure 15 : Diagramme de séquence de cas d'utilisation « l'authentification » .....	21
Figure 16 : Diagramme de séquence de cas d'utilisation « Créer une campagne » .....	22
Figure 17 : Diagramme de séquence de cas d'utilisation « Approuver une campagne » .....	23
Figure 18 : logo Visual Studio Code .....	24
Figure 19 : logo Postman .....	25
Figure 20 : logo Axios .....	25
Figure 21 : logo Github .....	26
Figure 22 : logo MongoDB .....	26
Figure 23 : logo Node.js .....	27
Figure 24 : logo Express.js .....	27
Figure 25 : logo React .....	28
Figure 26 : logo StarUML .....	28
Figure 27 : logo draw.io .....	29
Figure 28 : logo JavaScript.....	29
Figure 29 : logo HTML .....	30
Figure 30 : logo CSS .....	30
Figure 31 : Interface « page d'accueil » .....	31

Figure 32 : Interface de la « page d'inscription ».....	31
Figure 33 : Interface de la « page de connexion ».....	32
<i>Figure 34 : Interface de la page « Discover » .....</i>	<i>32</i>
Figure 35 : Interface - Partie 1 de la page « Créer une campagne » .....	33
Figure 36 : Interface - Partie 2 de la page « Créer une campagne » .....	33
Figure 37 : Interface de la page « Détail de la campagne » .....	34
Figure 38 : Interface de la page « Dashboard Admin ».....	35

## *Liste des tableaux*

---

Tableau 1 : Description du cas d'utilisation "Approuver une campagne" .....	17
Tableau 2 : : La description du cas d'utilisation "créer une campagne" .....	19



## **Introduction Générale**

Dans un contexte mondial marqué par une digitalisation croissante et un besoin constant d'innovation, l'entrepreneuriat joue un rôle essentiel dans le dynamisme économique et la création de valeur. De plus en plus, les jeunes porteurs d'idées novatrices aspirent à concrétiser leurs projets, qu'ils soient à vocation technologique, culturelle, sociale ou environnementale. Cependant, l'un des principaux obstacles auxquels ces entrepreneurs sont confrontés reste l'accès au financement. Les méthodes de financement traditionnelles, telles que les prêts bancaires ou les aides institutionnelles, se révèlent souvent inaccessibles, complexes ou insuffisantes, en particulier pour les petites structures ou les projets en phase de démarrage.

Face à cette problématique, le crowdfunding, ou financement participatif, s'est progressivement imposé comme une alternative crédible et innovante. Ce modèle consiste à faire appel à un large public pour financer un projet, généralement via des plateformes en ligne spécialisées. En démocratisant l'accès au financement, le crowdfunding permet de valoriser la créativité, de mobiliser une communauté autour d'une idée et de tester la viabilité d'un projet avant son lancement à grande échelle. Il représente ainsi un levier de développement significatif, notamment dans des contextes comme celui de la Tunisie, où l'écosystème entrepreneurial est en pleine émergence.

C'est dans cette perspective que s'inscrit notre Projet de Fin d'Études (PFA), intitulé TuniFund. Il s'agit d'une plateforme web de financement participatif développée dans le but de mettre en relation des créateurs de projets tunisiens avec des contributeurs potentiels, qu'ils soient locaux ou issus de la diaspora. TuniFund se veut être une solution à la fois accessible, intuitive, fiable et sécurisée, permettant aux porteurs d'idées de présenter leurs initiatives, de recueillir des soutiens financiers, et de bénéficier d'un espace de visibilité et de communication.

Ce projet ne se limite pas à la simple création d'un outil technologique ; il reflète une volonté de répondre à un besoin concret de la société tunisienne, en favorisant l'innovation citoyenne et en renforçant les dynamiques de solidarité à travers le numérique. Il s'inscrit pleinement dans la tendance actuelle du développement d'applications web à fort impact social.

Le présent rapport documente de manière rigoureuse l'ensemble des étapes ayant jalonné la réalisation de ce projet. Il débute par une **étude préalable**, mettant en évidence les besoins du marché, les fonctionnalités attendues, et les plateformes concurrentes. Il enchaîne ensuite avec la **conception fonctionnelle et technique** de la solution, intégrant la modélisation des données, l'architecture logicielle, et l'élaboration de l'interface utilisateur. Enfin, il présente la **mise en œuvre du projet**, à travers le développement, les choix technologiques (stack MERN : MongoDB, Express.js, React.js, Node.js), les tests réalisés, ainsi que les difficultés rencontrées et les solutions apportées.

En conclusion, ce projet représente une opportunité d'apprentissage approfondie dans les domaines du développement web, de la gestion de projet, et de l'innovation numérique. Il ouvre également la voie à de futures évolutions, telles que l'intégration de moyens de paiement locaux, l'ajout de fonctionnalités communautaires ou encore l'extension vers le mobile, dans le but de rendre TuniFund encore plus performant et accessible.

## **Chapitre1 : Étude préalable**

### **Introduction**

L'étude préalable est une phase essentielle dans la gestion de projet. Elle vise les besoins, analyser les solutions existantes, et proposer une réponse adaptée à une problématique spécifique. Elle définit les objectifs, les acteurs, et les fonctionnalités tout en tenant compte des contraintes et des ressources nécessaires.

Dans ce chapitre qui traite l'étude préalable, on va décrire d'abord la problématique à laquelle le site cherche à répondre, ensuite on aborde l'étude de l'existant avec l'analyse et la critique des solutions existantes et présenter une solution. Ensuite, on identifie les besoins fonctionnels et non-fonctionnels, tout en présentant la méthodologie de travail et le planning identifier du projet sous forme de diagramme de Gantt.

#### **1. Problématique**

Le financement participatif, ou crowdfunding, représente une nouvelle dimension pour financer des projets entrepreneuriaux, de petites entreprises, ou même des individus ambitieux souhaitant lancer leur premier projet. Cependant, de nombreux porteurs de projets rencontrent un double défi : le manque de visibilité et la confiance limitée envers les plateformes existantes, souvent étrangères, ce qui complique également les paiements. Il devient donc nécessaire de mettre en place une plateforme de crowdfunding locale, spécifiquement adaptée aux besoins du marché africain, et notamment tunisien, afin de démocratiser cette solution et encourager l'investissement participatif au niveau national.

#### **2. Étude de l'existant**

Cette section se concentre sur l'évaluation des solutions actuelles dans le domaine du **financement participatif** (crowdfunding) en Tunisie. L'objectif est d'examiner les plateformes existantes ainsi que les solutions locales et internationales.

##### **2.1 Analyse de l'existant**

Actuellement, en Tunisie, bien que le financement participatif commence à émerger, l'offre reste limitée, avec une forte dépendance à des plateformes étrangères. Les entrepreneurs

locaux cherchent des solutions pour financer leurs projets, mais les plateformes internationales ne sont pas parfaitement adaptées aux spécificités du marché tunisien.

**Plateformes existantes internationales :**

- **GoFundMe** : Plateforme internationale permettant aux individus de lever des fonds pour des projets personnels, des causes sociales ou des événements. Elle est principalement utilisée pour des collectes de fonds à l'échelle mondiale.



**Figure 1: logo de la plateforme "gofundme"**

- **Kickstarter** : Plateforme de crowdfunding spécialisée dans les projets créatifs, technologiques et innovants. Elle permet de financer des idées et produits avant leur lancement.



**Figure 2: logo de la plateforme "kickstarter"**

- **Indiegogo** : Plateforme de financement participatif similaire à Kickstarter, mais offrant une plus grande flexibilité, permettant de financer aussi bien des projets créatifs que commerciaux.



Figure 3: logo de la plateforme "indiegogo"

### Plateformes locales :

- **Tunisia Crowdfunding** : Plateforme tunisienne dédiée au financement participatif, permettant aux porteurs de projets locaux de lever des fonds en ligne.



Figure 4: logo de la plateforme "Tunisia Crowdfunding"

- **DabaPay** : Solution locale de paiement et de financement participatif, qui facilite les transactions pour les projets tunisiens à travers des moyens de paiement locaux.



Figure 5: logo de la plateforme "DabaPay"

## **2.2 Critique de l'existant**

- **GoFundMe et Kickstarter**

- **Points forts** : Ces plateformes offrent une visibilité internationale aux projets et génèrent confiance grâce à leur notoriété. Elles permettent d'atteindre une large audience d'investisseurs potentiels.
- **Points faibles** : Elles ne sont pas adaptées aux méthodes de paiement locales, ce qui complique l'accès pour les utilisateurs tunisiens. Les frais de transaction sont élevés, et la compétition internationale rend difficile la visibilité des projets tunisiens.

- **Tunisia Crowdfunding**

- **Points forts** : Plateforme locale, offrant une meilleure accessibilité aux utilisateurs tunisiens et adaptée aux méthodes de paiement locales.
- **Points faibles** : Peu connue, le nombre de projets soutenus est restreint, et les outils de soutien pour les porteurs de projets sont peu développés.

- **DabaPay**

- **Points forts** : Solution de paiement local simple et efficace pour les projets tunisiens. Permet des transactions directes et sécurisées.
- **Points faibles** : La plateforme est encore en développement et manque de visibilité et de fonctionnalités avancées par rapport aux plateformes internationales.

## **2.3 Solution Proposée**

Face aux limites des solutions existantes, nous proposons la création d'une plateforme de crowdfunding tunisienne baptisée "TuniFund", adaptée aux besoins spécifiques des porteurs de projets et investisseurs tunisiens. Cette plateforme permettrait de résoudre les principaux défis rencontrés par les utilisateurs locaux tout en soutenant les projets locaux.

Les principales caractéristiques de cette solution incluent :

- **Système de paiement local sécurisé** : Nous intégrerions des options de paiement adaptées aux méthodes tunisiennes, telles que e-Dinar, Carte bancaire nationale, ou

solutions de paiement mobile (D17, Flouci). Cela permettrait de garantir des transactions rapides et sécurisées.

- **Accompagnement des porteurs de projets** : La plateforme offrirait un soutien complet aux porteurs de projets, y compris des conseils sur la création de campagnes efficaces, la gestion des fonds collectés et des stratégies de communication. Un système de coaching ou de mentorat serait mis en place.
- **Projets diversifiés et catégories variées** : TuniFund permettrait de financer des projets dans des secteurs variés (technologie, art, environnement, etc.) en mettant en avant des projets locaux innovants et prometteurs.
- **Suivi et transparence** : Un système de suivi en temps réel permettrait aux investisseurs de suivre l'avancement des projets qu'ils soutiennent, avec des mises à jour régulières des porteurs de projets.

En résumé, **TunisFund** serait la solution pour démocratiser le financement participatif en Tunisie, en offrant une plateforme sécurisée, locale et adaptée aux besoins spécifiques du marché tunisien, tout en facilitant l'accès à des financements pour les projets innovants locaux.

### **3. Analyse et spécification des besoins**

#### **3.1 Identification des acteurs**

Un acteur est une entité externe qui interagit avec le système afin d'accomplir ses besoins.

À la suite de cette interaction, celui-ci lui fournit un service qui correspond à son besoin, en effet, les acteurs qu'interagissent avec notre système sont :

- **Administrateur** : Il gère la plateforme, contrôle les utilisateurs et supervise les campagnes de financement.
- **Utilisateur** : qui peut être
  - **Un porteur de projet** : il crée des campagnes de financement, présente ses projets et collecte des fonds.
  - **Donateur** : il explore les campagnes disponibles et contribue financièrement aux projets qui l'intéressent.

### **3.2 Besoins fonctionnels**

Les besoins fonctionnels décrivent les fonctionnalités spécifiques que le système doit offrir pour répondre aux attentes des utilisateurs.

#### **Pour le porteur de projet :**

- **Créer une campagne** : Le porteur de projet peut soumettre une campagne de financement avec une description, des objectifs financiers, une durée et des visuels.
- **Gérer ses campagnes** : Modifier ou mettre à jour les informations d'une campagne en cours.
- **Suivre les contributions** : Voir en temps réel les fonds collectés et les contributeurs.
- **Interagir avec les donateurs** : Répondre aux questions à travers les commentaires en temps réel.

#### **Pour le donateur :**

- **Consulter les campagnes** : Parcourir les projets disponibles, voir leurs détails et effectuer une recherche ciblée.
- **Faire un don** : Contribuer à un projet via des moyens de paiement sécurisés.
- **Suivre ses contributions** : Consulter l'historique de ses dons et recevoir des mises à jour sur les projets financés.

#### **Pour l'administrateur :**

- **Gérer les utilisateurs** : Ajouter, modifier, supprimer ou suspendre des comptes d'utilisateurs.
- **Gérer les campagnes** : Valider ou rejeter les campagnes soumises par les porteurs de projet.
- **Modérer les contenus** : Contrôler les descriptions et les mises à jour des campagnes pour éviter les fraudes.

### **3.3. Besoins non fonctionnels**

Les besoins non fonctionnels décrivent les qualités attendues du système au-delà de ses fonctionnalités principales. Ils garantissent la fiabilité, la sécurité et l'expérience utilisateur de



l'application. Dans le cadre de notre plateforme de crowdfunding, ces besoins prennent une importance particulière pour garantir la confiance des utilisateurs et la fluidité des campagnes.

- **Confidentialité**

La confidentialité est essentielle dans une plateforme de financement participatif. Les utilisateurs (donateurs ou porteurs de projets) saisissent des informations personnelles et financières sensibles. Il est donc impératif de sécuriser les accès à travers un système d'authentification (login/mot de passe) afin que chaque utilisateur accède uniquement à son espace personnel. Cela renforce la confiance envers la plateforme.

- **Performance**

Une bonne performance est cruciale pour garantir une expérience fluide, surtout lors du chargement de projets ou du traitement des dons. Si la plateforme est lente, les utilisateurs risquent de se décourager, ce qui peut impacter négativement la collecte des fonds. Un temps de réponse rapide est donc indispensable pour assurer l'efficacité des interactions.

- **Temps de réponse**

Le système doit répondre dans un délai raisonnable à chaque action de l'utilisateur : consultation d'un projet, envoi d'un don, modification de profil, etc. Des retards peuvent entraîner des abandons de navigation ou des doubles paiements involontaires. C'est donc un facteur clé pour minimiser les erreurs et maximiser la satisfaction utilisateur.

- **Disponibilité**

Les campagnes de financement peuvent être actives 24h/24, ce qui exige que le système soit hautement disponible, notamment pendant les pics de trafic (par exemple, en fin de campagne). Une interruption du service pourrait compromettre les dons et nuire à la crédibilité de la plateforme.

- **Ergonomie**

L'interface doit être intuitive, claire et accessible, même pour des utilisateurs non techniques. Un bon design ergonomique permet de faciliter la navigation, d'augmenter le taux de conversion des visiteurs en donateurs, et de réduire les erreurs lors de la création de projets ou de dons.

## 4. Méthodologie de travail

### 4.1.Méthodologie 2-tup

Nous avons choisi la **méthode 2-tup** pour notre projet afin de bénéficier de ses nombreux avantages. Cette méthode a été choisie pour ce projet, car elle combine une approche itérative et structurée. Elle permet une séparation claire des phases (spécification, conception, développement, test) tout en intégrant des améliorations progressives. Adaptée aux projets individuels, elle assure une gestion efficace et une validation continue, réduisant ainsi les risques d'erreurs.

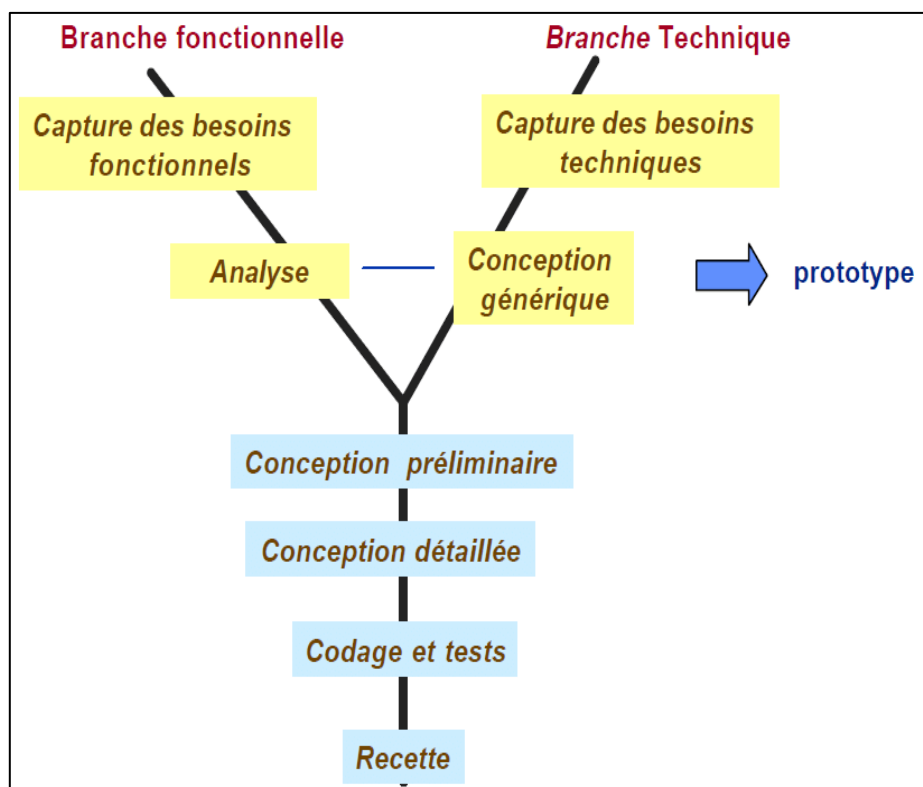


Figure 6: Représentation 2tup méthode

## 5. Diagramme de Gantt

Le diagramme de Gantt est un outil de gestion de projet qui représente visuellement le calendrier d'un projet. Il affiche les tâches à accomplir, leur durée, et les relations entre elles sous la forme d'un graphique à barres. Chaque tâche est listée sur l'axe vertical, tandis que l'axe horizontal indique le temps.



Figure 7: Diagramme de Gantt

## Conclusion

Ce chapitre a été dédié à la définition du contexte générale du projet ainsi que la spécification des besoins divers de l'application. Il nous a servi à bien comprendre le problème posé afin de préciser nos objectifs à atteindre. Cet aperçu général est l'appui et le support de base de notre conception que nous mettrons en évidence dans le chapitre qui suit.

## Chapitre 2 : Conception

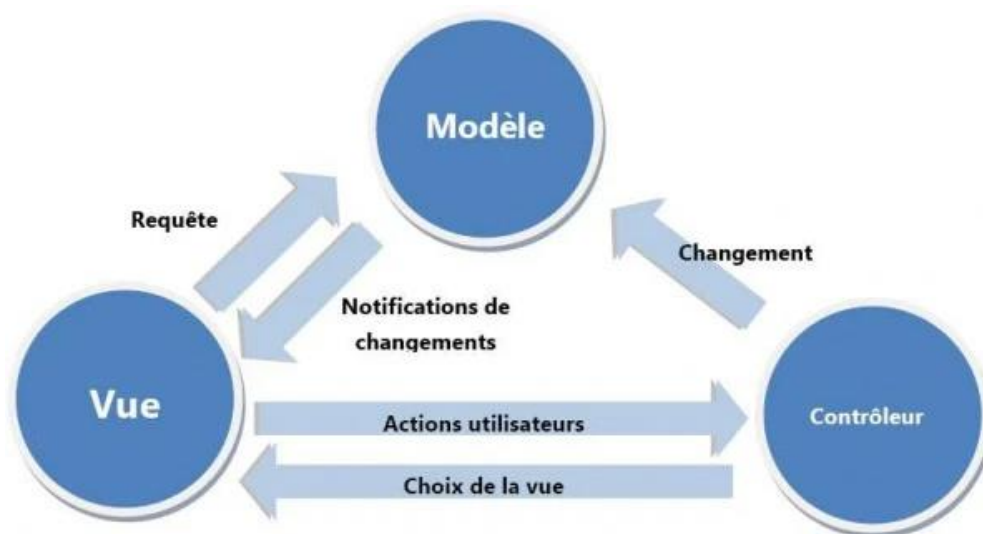
### Introduction

Dans le chapitre précédent, nous avons précisé les différents acteurs de notre solution en détaillant les nombreuses fonctionnalités qu'ils seront capables d'exploiter. Nous arrivons maintenant à l'une des phases les plus importantes lors du développement d'une application, la conception. Dans cette étape nous allons présenter le diagramme de cas d'utilisation général et raffiné, le diagramme de classes et les diagrammes de séquences

#### 1. Architecture logicielle

L'architecture logicielle est une étape cruciale dans la conception d'un projet, car elle définit la structure globale du système ainsi que la manière dont les différentes parties interagiront. Dans le cadre de notre plateforme **TuniFund**, nous avons choisi d'adopter l'architecture **Modèle-Vue-Contrôleur (MVC)**. Cette architecture est largement utilisée pour la construction d'applications web en raison de sa simplicité, de sa flexibilité et de sa capacité à séparer les différentes préoccupations du système.

Le modèle de conception MVC divise une application en trois aspects principaux : Modèle, Vue et Contrôleur, comme indiqué dans la figure ci-dessous



**Figure 8: Schéma explicatif du patron MVC**

- **Model (Modèle)** : Le modèle représente une collection de classes qui explique la logique métier, à savoir le modèle métier et le modèle de données (opérations d'accès aux

données). Il définit également les règles métier pour les moyens de données comme la façon dont les données peuvent être modifiées et manipulées.

- **View** (Vue) : La vue représente les composants de l'interface utilisateur tels que CSS, jQuery, HTML, etc (dans le web). Elle affiche les données reçues du contrôleur comme résultat.
- **Controller** (Contrôleur) : La responsabilité du contrôleur est de traiter les demandes entrantes. Il obtient l'entrée des utilisateurs via la vue, puis traite les données de l'utilisateur via le modèle, en renvoyant les résultats à View. Il agit comme un médiateur entre la vue et le modèle

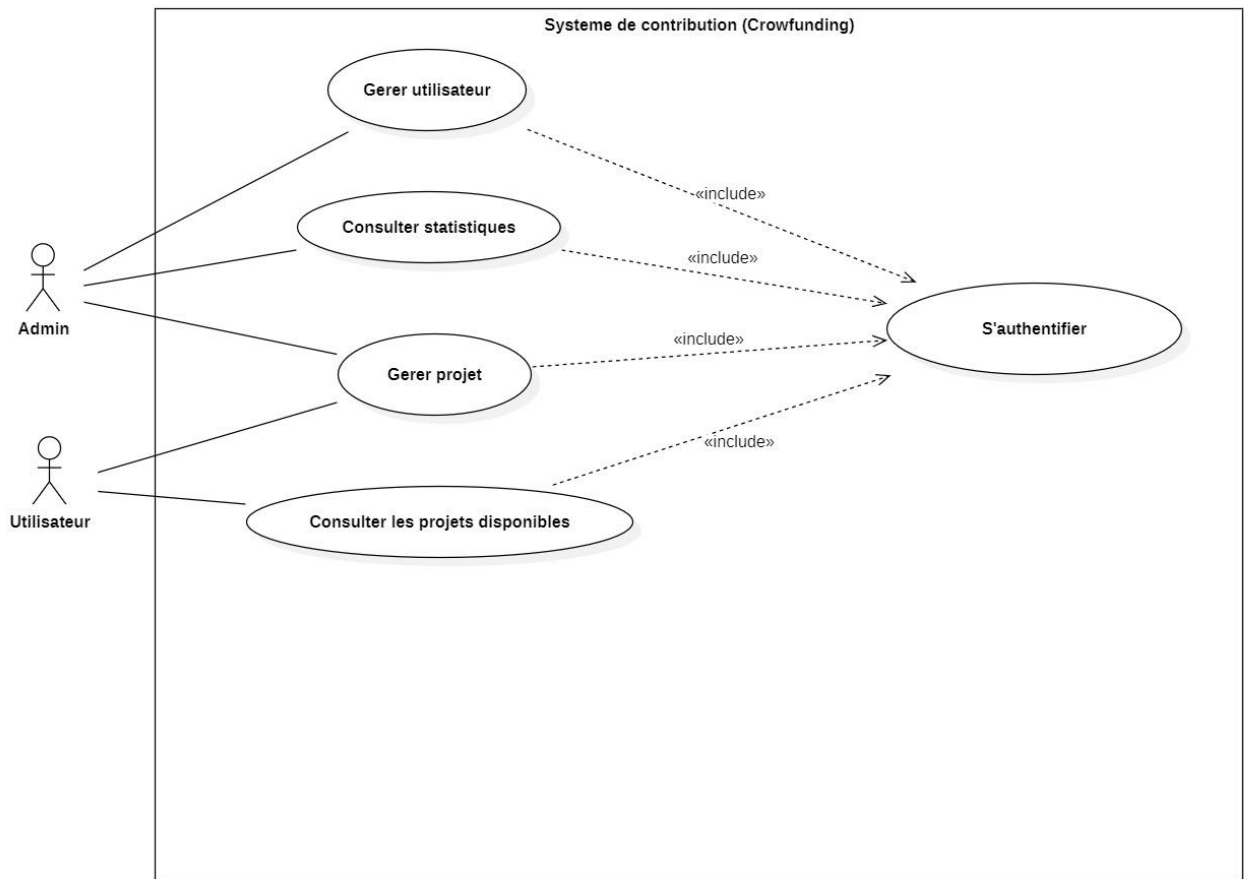
## **2. Conception UML**

La conception UML est une méthode de modélisation visuelle qui décrit la structure et le comportement d'un système logiciel, facilitant la compréhension et la planification avant le développement.

### **2.1 Diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation représente les interactions entre les utilisateurs (acteurs) et le système, décrivant les fonctionnalités principales du système du point de vue de l'utilisateur. Il sert à modéliser un des aspects statiques du système.

La figure 7 (ci-dessous) représente le **diagramme des cas d'utilisation générale** de notre application ce qui permet de décrire les fonctionnalités globales. Ensuite nous détaillerons les cas d'utilisation les plus importants.



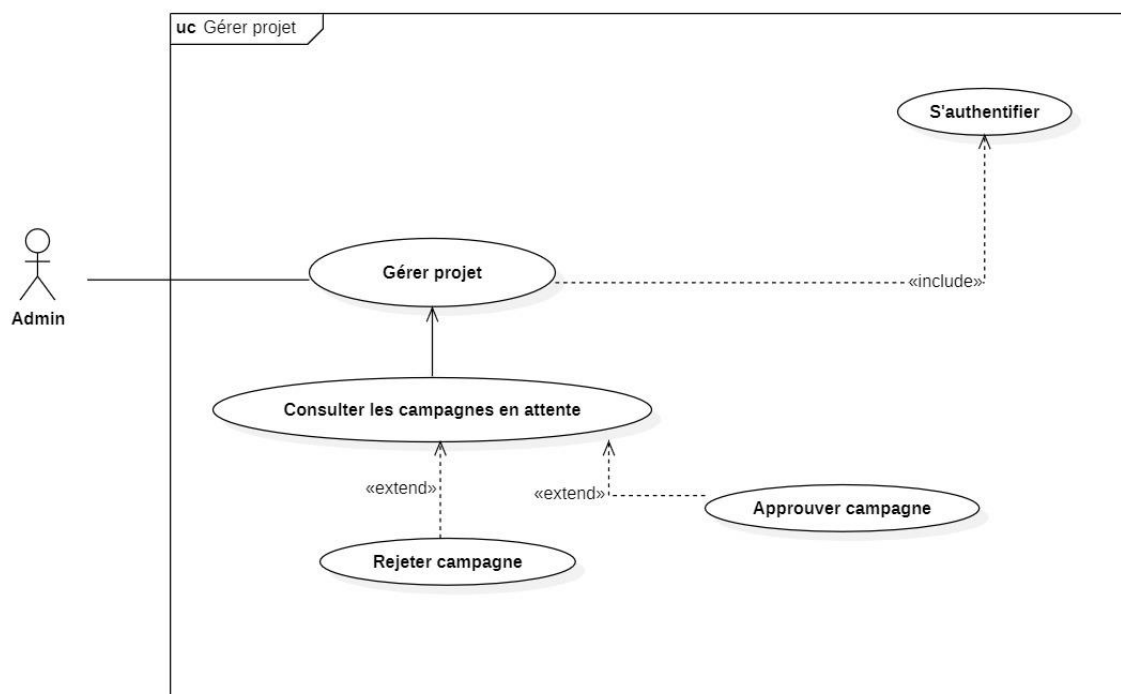
**Figure 9 : Diagramme de cas d'utilisation général**

## 2.2 Diagramme de cas d'utilisation raffinés de l'administrateur système

Dans notre application, il existe plusieurs cas d'utilisation pour l'administrateur système. Nous allons présenter les cas les plus marquants tout en les analysant.

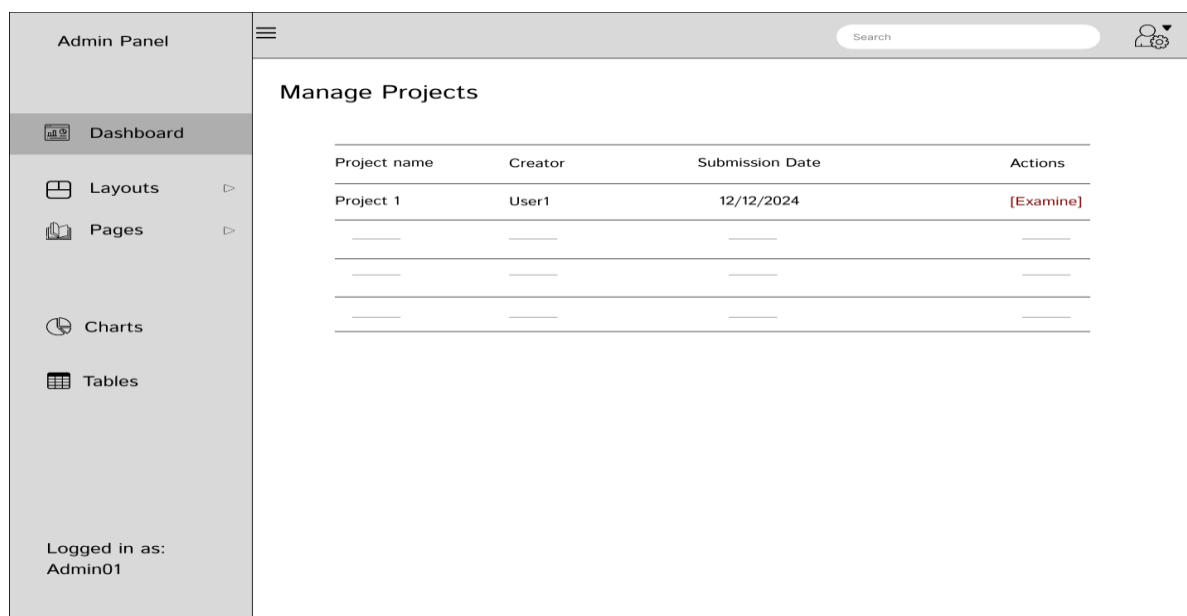
### Diagramme de cas d'utilisation raffiné de la gestion de projets

Le diagramme de cas d'utilisation raffiné de la gestion de projets est illustré par la figure ci-dessous :

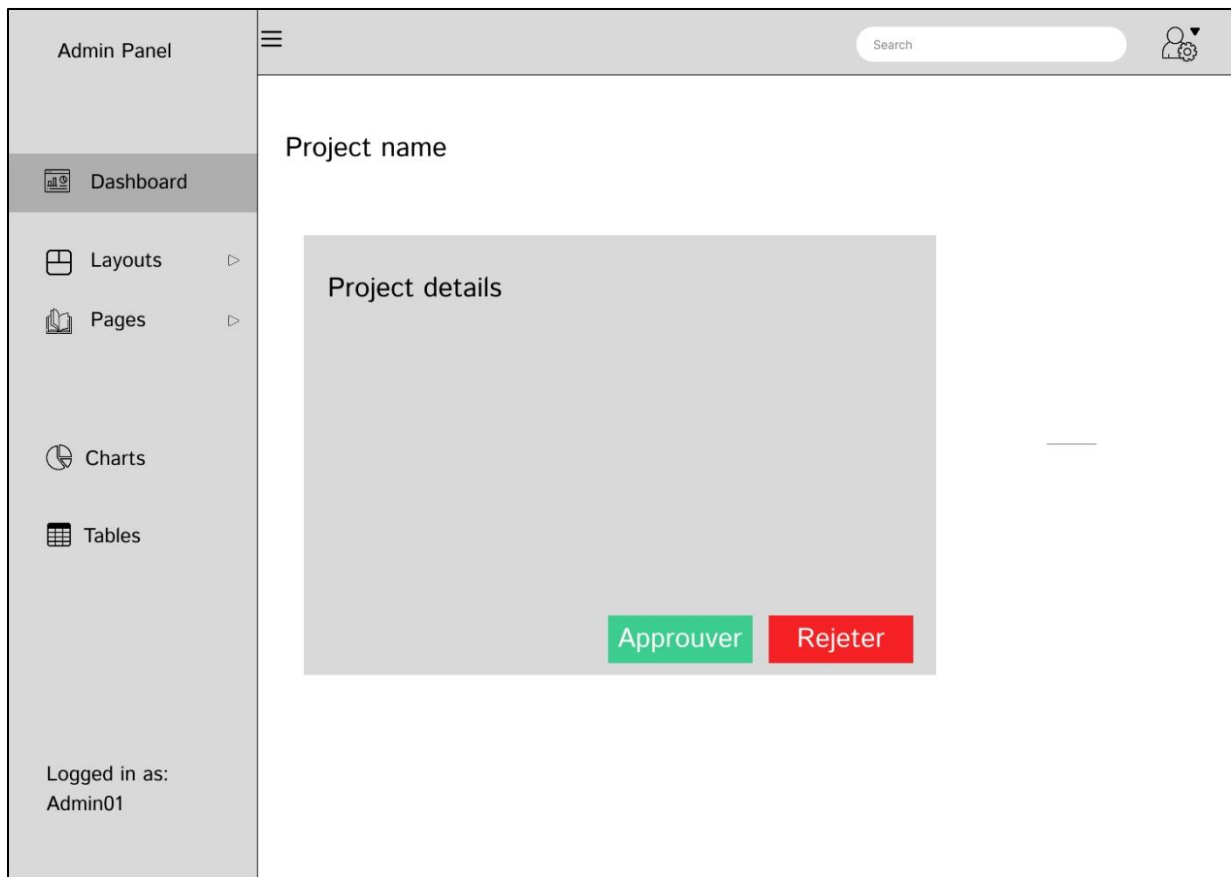


**Figure 10 : Diagramme de cas d'utilisation raffiné de la gestion de projet**

La maquette de cas d'utilisation raffiné de la gestion de campagnes est illustrée par la figure suivante :



**Figure 11 : Maquette de l'interface de l'administrateur de gestion des campagnes (projets)**



**Figure 12 : Maquette de l'interface de l'administrateur de gestion d'une campagne(projet)**

La description du cas de gestion des projets est donnée par le tableau suivant :

Cas utilisation	Approuver une campagne.
Acteur principale	Administrateur Système.
Pré conditions	Administrateur authentifié.
Scénario nominal	<p><b>[S1] : L'administrateur valide une campagne</b></p> <ol style="list-style-type: none"> <li>1. L'administrateur consulte la liste des campagnes en attente de validation.</li> <li>2. L'admin clique sur examiner l'une des campagnes en attente</li> <li>3. Il clique sur le bouton "Valider"</li> <li>4. Le système affiche une boîte de confirmation pour valider la campagne.</li> <li>5. L'administrateur confirme l'action.</li> <li>6. Le système met à jour le statut de la campagne en</li> </ol>



	"validée". 7. Un message de confirmation est affiché, et la campagne devient visible au public.
Enchainements alternatifs	A1 : Si l'utilisateur n'est pas authentifié : Le système affiche l'interface d'authentification. A2 : Le système détecte que certains champs vides ou invalides Il affiche un message d'erreur

**Tableau 1 : Description du cas d'utilisation "Approuver une campagne"**

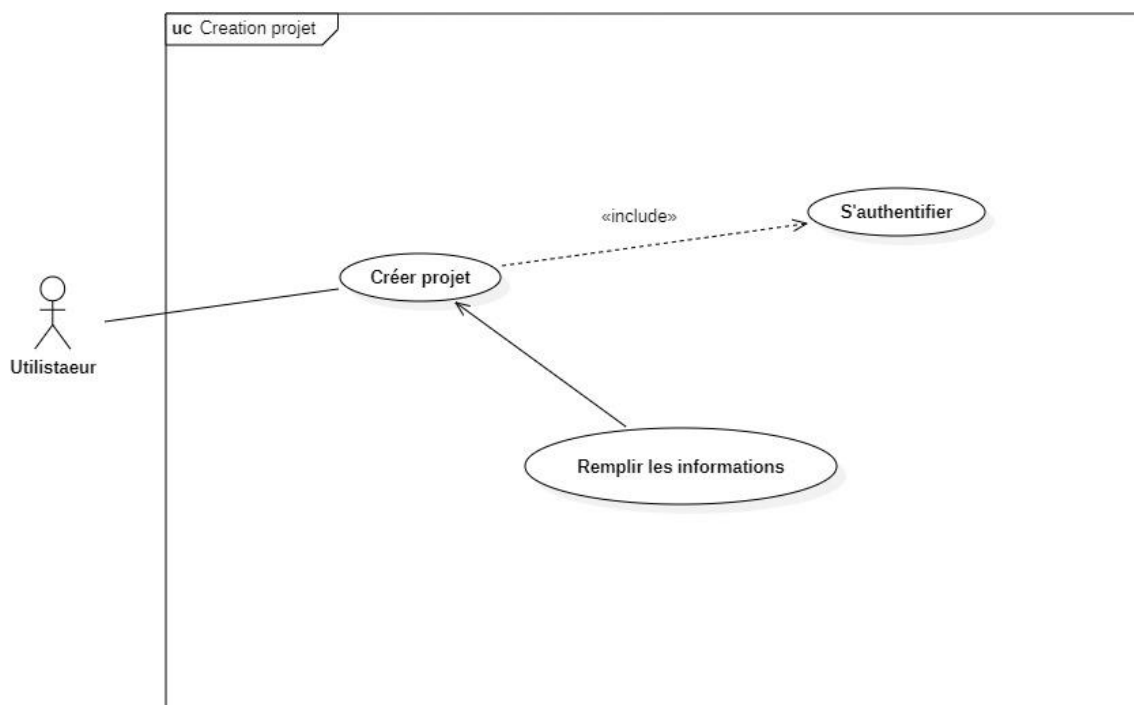
Post Conditions : La campagne est enregistrée avec état validé.

### **2.3 Diagramme de cas d'utilisation raffinés de l'utilisateur (porteur de projet)**

Dans notre application TuniFund, les porteurs de projets disposent de plusieurs fonctionnalités leur permettant de gérer leurs campagnes. L'une des plus importantes est la création d'une nouvelle campagne de financement participatif. Ce processus constitue un point de départ fondamental dans le parcours utilisateur.

#### **Diagramme de cas d'utilisation raffiné de la création d'une campagne**

Le diagramme de cas d'utilisation raffiné de la création d'une campagne(projet) est illustré par la figure ci-dessous :



**Figure 13 : Diagramme de cas d'utilisation raffiné de la création d'une campagne(projet)**

La description du cas de la création d'une campagne (projet) est donnée par le tableau suivant

Cas utilisation	Créer une campagne (un projet)
Acteur principale	Porteur de projet (utilisateur authentifié)
Post conditions	Le porteur de projet est authentifié.
Pré Conditions	La campagne est enregistrée avec le statut "en attente".
Scénario nominal	<p>[S1] : Le porteur de projet crée une nouvelle campagne</p> <ol style="list-style-type: none"> <li>1. Le porteur de projet clique sur le bouton "Créer une campagne" depuis son tableau de bord.</li> <li>2. Le système affiche un formulaire de création avec les champs nécessaires (titre, description, montant, date de fin, image, etc.).</li> <li>3. Le porteur de projet remplit le formulaire et clique sur "Soumettre".</li> <li>4. Le système vérifie la validité des données saisies.</li> <li>5. Si toutes les données sont valides, le système enregistre la campagne avec le statut "en attente de validation".</li> <li>6. Un message de confirmation est affiché à l'utilisateur.</li> </ol>

Enchainements alternatifs	<p>A1 : Si le porteur de projet n'est pas authentifié : Le système redirige vers l'interface d'authentification.</p> <p>A2 : Si certains champs sont vides ou invalides : Le système affiche un message d'erreur précisant les champs à corriger.</p> <p>Le scénario reprend à l'étape 2.</p>
---------------------------	---

Tableau 2 : : La description du cas d'utilisation "créer une campagne"

Post condition : La campagne est enregistrée avec le statut "en attente".

### 3. Diagramme de classe globale

Dans la figure 3.1 on représente le diagramme de classe relatif à notre application

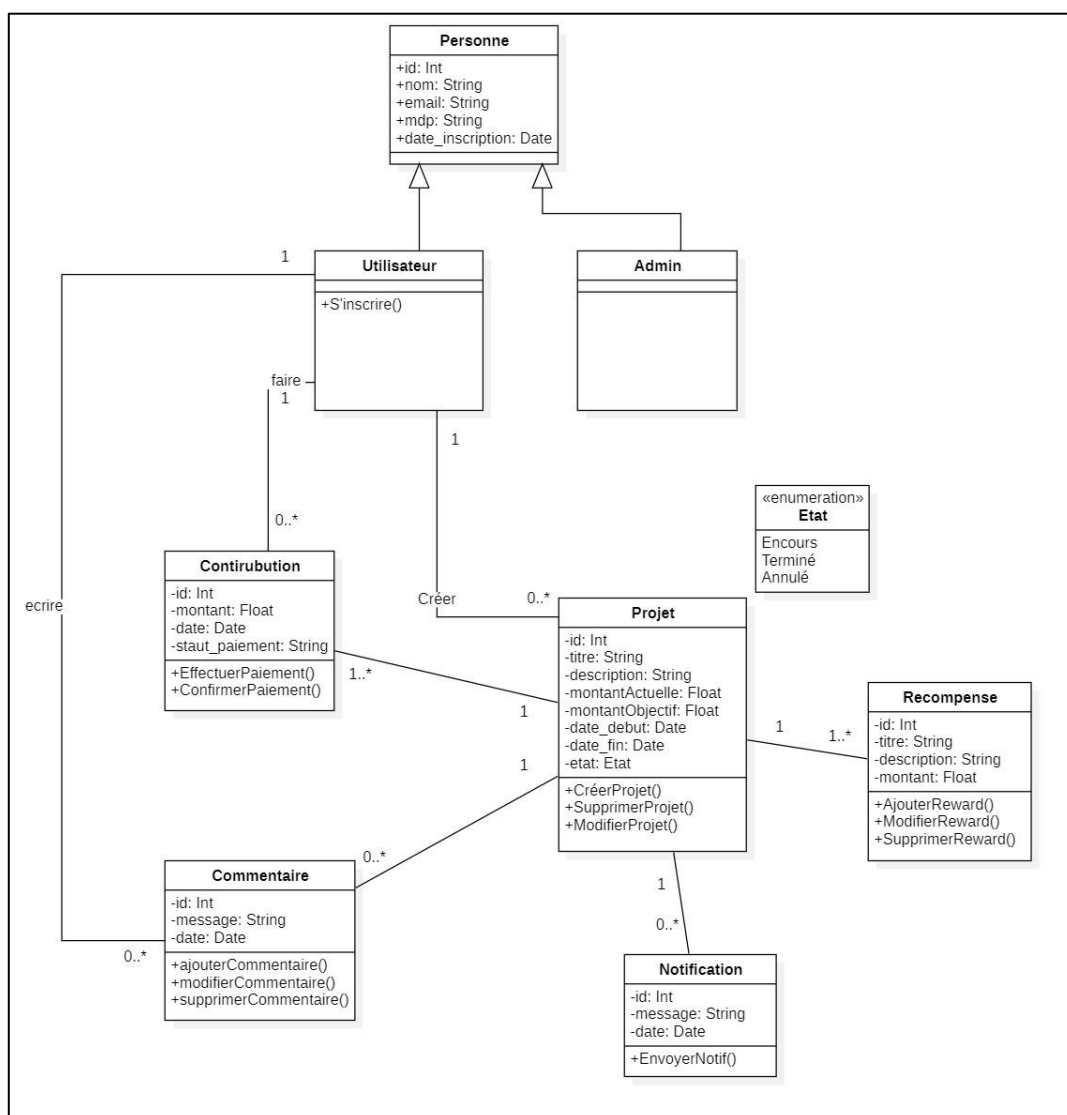


Figure 14 : Diagramme de classe globale

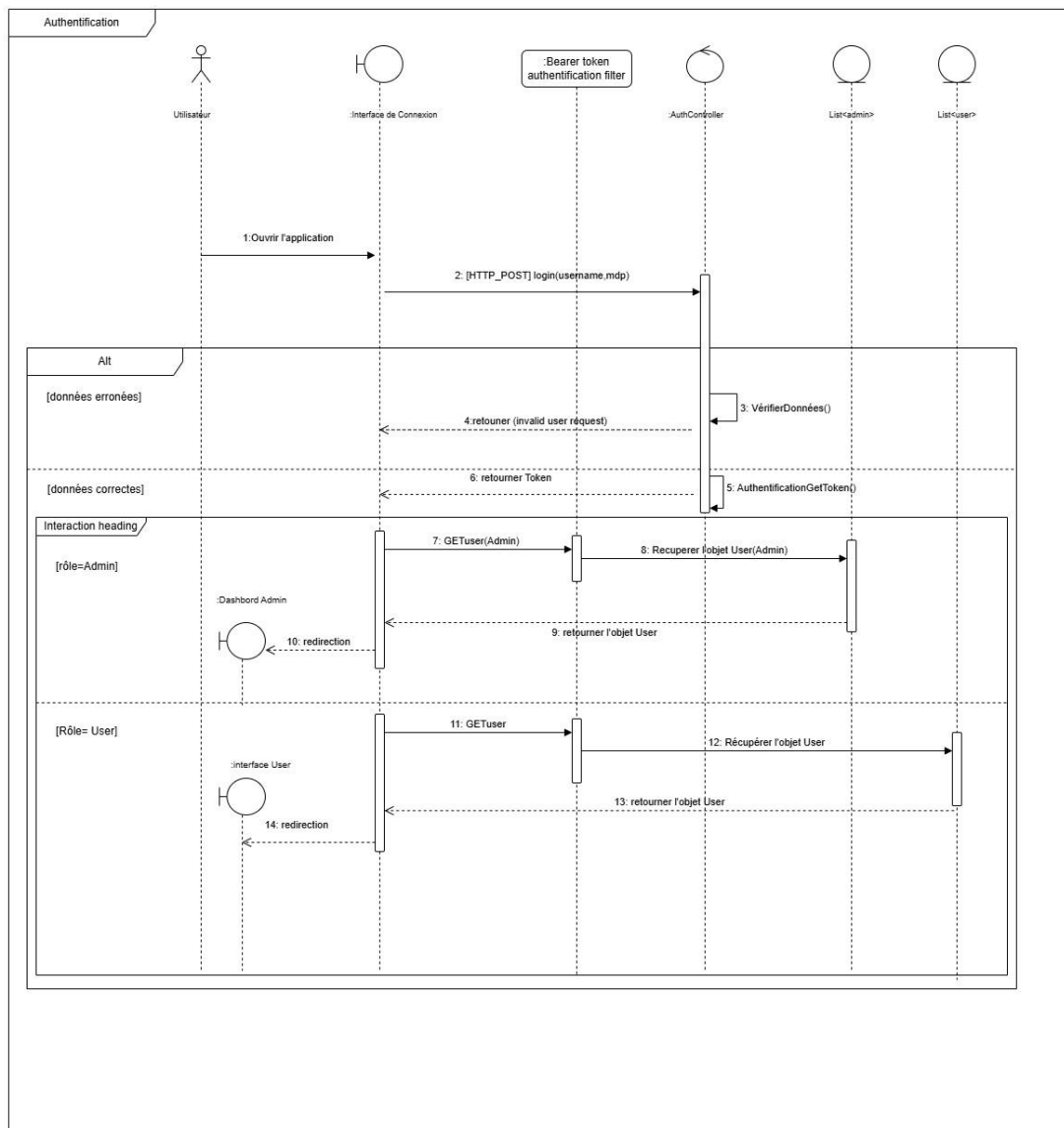
#### **4. Diagramme de séquences**

Le diagramme de séquence permet de décrire la manière dont les éléments du système interagissent entre eux en précisant les différents messages échangés dans un ordre chronologique.

Dans ce qui suit, nous allons détailler les trois scénarios : l'authentification et valider un projet

##### **4.1 Diagramme de séquence de cas d'utilisation « l'authentification »**

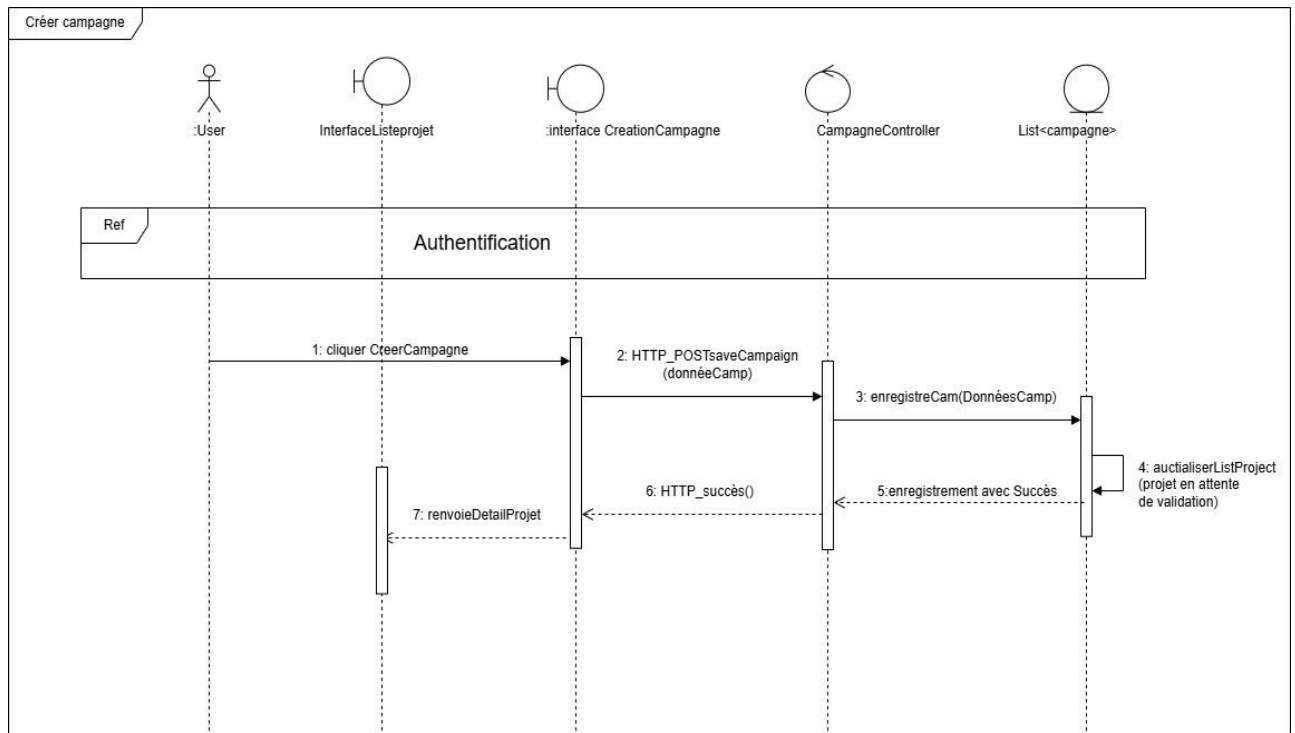
Pour s'authentifier, il faut que l'utilisateur ai déjà activé son compte, il doit remplir soigneusement les deux champs Email et mot de passe, le système vérifie s'il existe vraiment un utilisateur avec l'email et le mot de passe saisis. Si le compte est activé et les informations sont correctes, un jeton d'authentification « Token » lui sera attribué et sera redirigé vers l'interface principale. Sinon, un message d'erreur lui sera affiché.



**Figure 15 : Diagramme de séquence de cas d'utilisation « l'authentification »**

#### 4.2 Diagramme de séquence de cas d'utilisation « Créer une campagne »

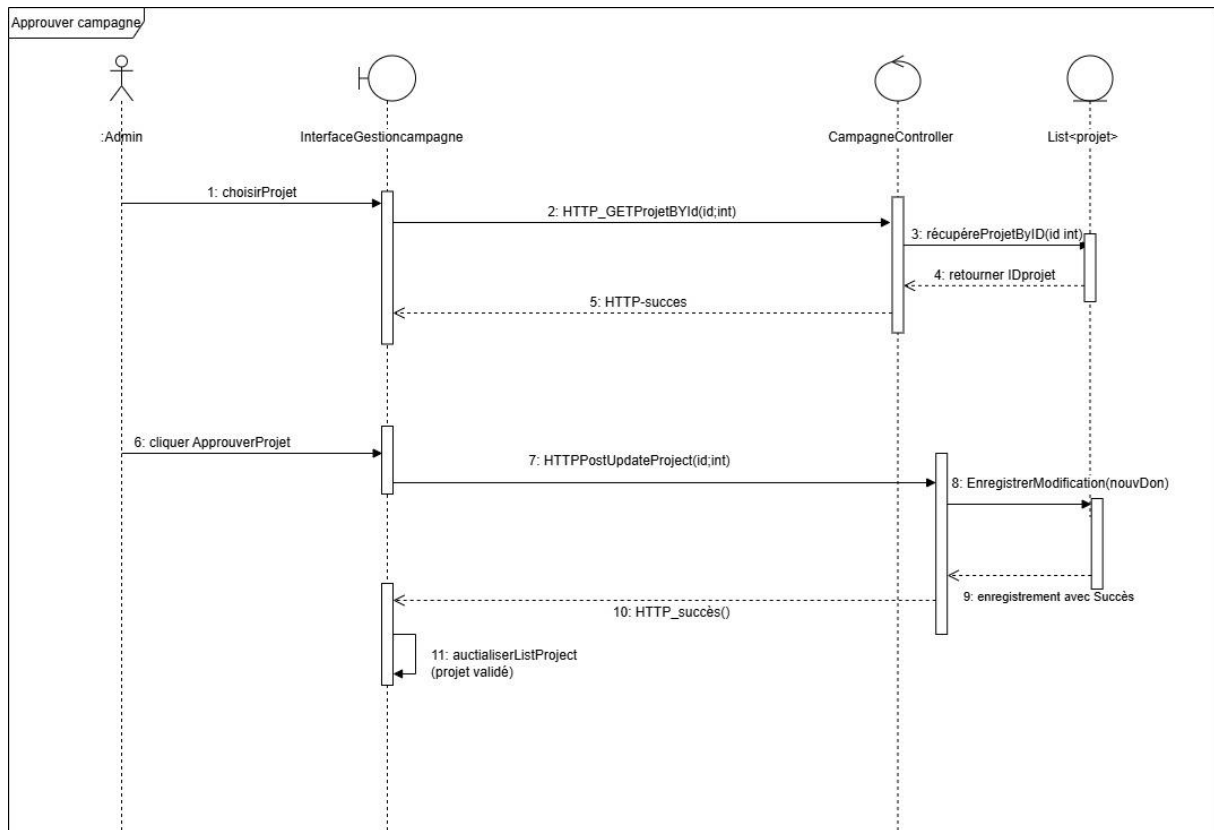
Ce diagramme de séquence illustre les interactions entre le porteur de projet et le système lors de la création d'une nouvelle campagne. Il présente le déroulement du processus depuis la soumission du formulaire jusqu'à l'enregistrement des données dans la base et la confirmation de la création.



**Figure 16 : Diagramme de séquence de cas d'utilisation « Créer une campagne »**

#### 4.3 Diagramme de séquence de cas d'utilisation « Approuver une campagne »

L'administrateur, déjà authentifié, sélectionne un projet en attente dans la liste. Il clique sur « **Approuver** » ce qui déclenche une requête POST avec le nouvel état. Le serveur enregistre la modification, met à jour l'état du projet (validé) et renvoie un code HTTP. Enfin, l'interface affiche une notification de succès et rafraîchit la liste pour retirer le projet traité.



**Figure 17 : Diagramme de séquence de cas d'utilisation « Approuver une campagne »**

## Conclusion

Dans de ce chapitre, nous avons défini l'aspect statique et l'aspect dynamique de notre système grâce au diagramme de classes, une description détaillée des différents cas d'utilisation et les diagrammes de séquences. Dans le chapitre suivant, nous allons procéder à l'étape de réalisation de notre solution.

## **Chapitre 3 : Réalisation**

### **Introduction**

Après avoir réalisé la conception de notre application, nous passons dans ce chapitre à la phase de réalisation qui est considérée comme étant la concrétisation finale de toute la méthode de conception. Nous commençons par spécifier l'environnement matériel et logiciel que nous avons utilisé pour réaliser notre application. Par la suite, nous présentons l'implémentation de quelques parties de notre application. Enfin, nous illustrons certaines interfaces réalisées pour démontrer le bon fonctionnement du système.

#### **1. Environnement de développement**

Dans cette partie, nous présentons les environnements matériels et logiciels utilisés dans le cadre de notre projet.

##### **1.1 Environnement logiciel**

- **Visual Studio Code (VS Code) :**

**Vs Code** est un éditeur de code open source développé par Microsoft. Polyvalent et léger, il prend en charge de nombreux langages de programmation grâce à ses extensions, disponibles sur sa marketplace. Doté d'une interface intuitive, il offre des fonctionnalités avancées comme le débogage, le contrôle Git intégré et la complétion intelligente du code.



**Figure 18 : logo Visual Studio Code**



- **Postman :**

Postman est un outil collaboratif utilisé pour tester les API. Il permet d'effectuer des requêtes HTTP (GET, POST, PUT, DELETE...) de manière intuitive et d'inspecter les réponses du serveur. Il est particulièrement utile pour valider le bon fonctionnement du backend, tester les routes Express.js et simuler des scénarios utilisateurs.



**Figure 19 : logo Postman**

- **Axios :**

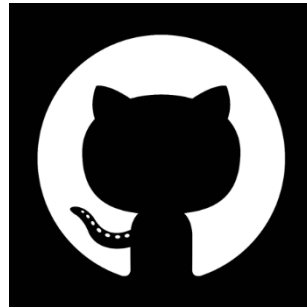
Axios est une bibliothèque JavaScript basée sur les Promises, utilisée pour effectuer des requêtes HTTP côté client et côté serveur. Elle simplifie la communication avec les API en fournissant une syntaxe claire pour envoyer des requêtes GET, POST, PUT, DELETE, etc. Dans notre projet, Axios a été utilisé pour gérer les échanges entre le frontend développé avec React et le backend Express, notamment pour l'envoi de formulaires, la récupération des données des campagnes, et l'authentification.



**Figure 20 : logo Axios**

- **GitHub :**

**GitHub** est une plateforme de gestion de version basée sur Git. Elle nous a permis de suivre l'évolution du code source, de collaborer efficacement, de gérer les versions du projet et d'assurer un historique clair du développement. Chaque fonctionnalité a été développée dans une branche dédiée, puis fusionnée avec la branche principale via des pull requests.



**Figure 21 : logo Github**

- **MongoDB :**

**MongoDB** est une base de données NoSQL orientée documents, qui stocke les données au format BSON (similaire au JSON). Elle est particulièrement adaptée aux applications modernes comme TuniFund, en raison de sa flexibilité, sa scalabilité et sa simplicité de manipulation. Elle a été utilisée pour stocker les informations des utilisateurs, des campagnes, des transactions, etc.



**Figure 22 : logo MongoDB**

## 1.2 Frameworks utilisés

- **Node.js :**

**Node.js** est une plateforme open source basée sur le moteur JavaScript V8 de Chrome, permettant d'exécuter du code JavaScript côté serveur. Connu pour sa rapidité et sa gestion asynchrone, il est idéal pour construire des applications web évolutives, comme des API ou des systèmes en temps réel. Avec son gestionnaire de paquets NPM, il offre une vaste bibliothèque de modules pour simplifier le développement.



Figure 23 : logo Node.js

**Express.js :**

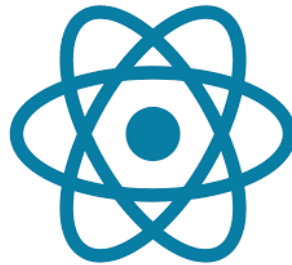
**Express.js** est un Framework minimaliste pour Node.js qui facilite la création d'API RESTful. Il permet de structurer le backend en organisant les routes, les middlewares et les contrôleurs de manière claire. Express.js a été utilisé dans notre projet pour gérer les différentes routes backend (authentification, campagnes, paiements...).



Figure 24 : logo Express.js

- **React :**

**React** est une bibliothèque JavaScript open source, développée par Facebook, utilisée pour créer des interfaces utilisateur interactives. Basée sur un concept de composants réutilisables, elle facilite le développement d'applications web dynamiques et performantes. Grâce à son Virtual DOM, React optimise les mises à jour de l'interface pour une expérience utilisateur fluide.



**Figure 25 : logo React**

- **StarUML :**

**StarUML** est un outil d'ingénierie logicielle destiné à prendre en charge une modélisation flexible et claire à l'aide du langage de modélisation unifié, ainsi que du langage de modélisation des systèmes et des notations de modélisation classiques.



**Figure 26 : logo StarUML**

- **draw.io :**

**diagrammes.net** est un logiciel de dessin graphique multiplateforme développé en HTML5 et JavaScript. Son interface peut être utilisée pour créer des diagrammes tels que des organigrammes, des wireframes, des diagrammes UML, des organigrammes et des diagrammes de réseau.



**Figure 27 : logo draw.io**

### **1.3 Langage de programmation**

#### **JavaScript (JS) :**

**JavaScript (JS)** est un langage de programmation léger et polyvalent principalement utilisé pour le développement web. Il permet de rendre les pages web interactives en manipulant le DOM, en gérant des événements ou en effectuant des requêtes HTTP. Compatible avec tous les navigateurs modernes, JavaScript est également utilisé côté serveur grâce à des plateformes comme Node.js.



**Figure 28 : logo JavaScript**

### **1.4 Autres technologies web**

- **HTML (HyperText Markup Language) :**

**HTML** est le langage standard utilisé pour structurer les pages web. Dans notre projet, il est généré à travers les composants React sous forme de JSX. Il permet de définir les éléments de la page comme les formulaires, titres, boutons, images, etc.



**Figure 29 : logo HTML**

**CSS (Cascading Style Sheets) :**

CSS est un langage de style utilisé pour définir l'apparence des éléments HTML. Il permet de gérer la mise en page, les couleurs, les polices, et l'adaptabilité responsive. Nous avons utilisé du CSS externe pour assurer une séparation claire entre la logique (JavaScript) et le style.



**Figure 30 : logo CSS**

## **2. Démonstration**

Dans cette section, nous illustrons le fonctionnement et l'application pratique des concepts abordés précédemment à travers un exemple concret. Cette démonstration vise à montrer étape par étape comment les technologies ou méthodologies présentées peuvent être mises en œuvre dans un contexte réel, en détaillant leur utilisation et les résultats obtenus.

### **2.1 Grille d'écran de la page de « accueil »**

La figure démontre la grille d'écran "page d'accueil"

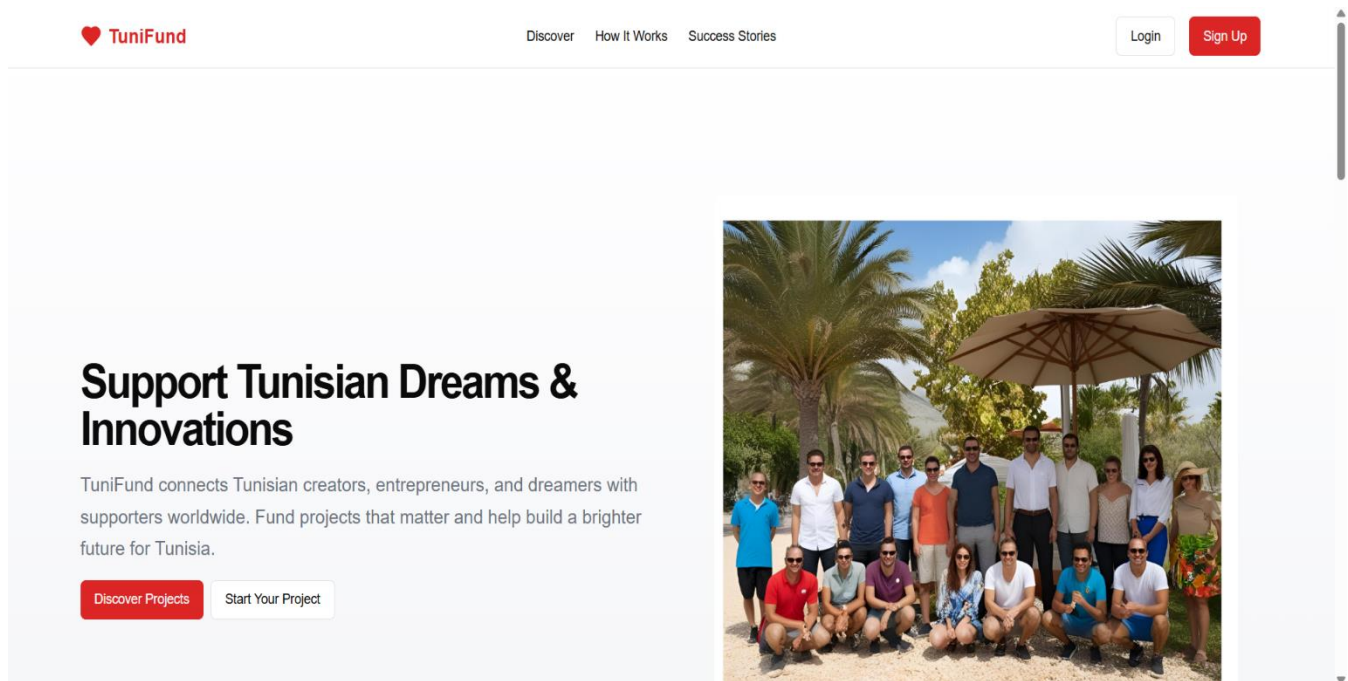


Figure 31 : Interface « page d'accueil »

## 2.2 Grille d'écran de la page d'inscription

La figure suivante illustre l'interface de la page d'inscription. Les utilisateurs peuvent s'inscrire sur la plateforme en saisissant leurs informations personnelles, leur adresse e-mail ainsi que leur mot de passe.

 The screenshot displays the 'Create an account' registration form on the TuniFund website. The form is titled 'Create an account' with a subtitle 'Enter your information to create an account'. It contains several input fields: 'First name' (with the example 'John'), 'Last name' (with the example 'Doe'), 'Email' (with the example 'name@example.com'), 'Password', and 'Confirm Password'. Below these fields is a prominent red 'Create Account' button. Underneath the button, there is a section labeled 'OR CONTINUE WITH' followed by a 'Google' login option with the Google logo. At the bottom of the form, there is a link that says 'Already have an account? Sign in'. The TuniFund logo is visible in the top left corner of the page.

Figure 32 : Interface de la « page d'inscription »

## 2.3 Grille d'écran de la page de « connexion »

La figure 2.3 illustre la grille d'écran de la page *connexion*. Elle permet aux utilisateurs de se connecter à la plateforme en saisissant leur adresse e-mail et leur mot de passe.

♥ TuniFund

**Welcome back**

Enter your email and password to sign in to your account

Email  
name@example.com

Password [Forgot password?](#)

**Sign In**

OR CONTINUE WITH

Google GitHub

Don't have an account? [Sign up](#)

Figure 33 : Interface de la « page de connexion »

## 2.4 Grille d'écran de la page « Discover »

La figure 2.4 illustre la grille d'écran de la page *Discover*. Elle permet aux utilisateurs de consulter la liste des campagnes disponibles sur la plateforme. Chaque campagne affichée peut être explorée en détail, et l'utilisateur a la possibilité d'effectuer un don directement à partir de cette page.

**Discover Projects**  
Find and support innovative Tunisian projects

Q Search projects... Newest

Categories	Project Name	Description	Progress	Amount Raised	Time Left
All Projects	Olive Oil Co-op	Support local farmers with modern equipment	51% Funded	6743 DT raised	21 days left
Technology	Tech Education Center	Teaching coding to underserved communities	64% Funded	4232 DT raised	34 days left
Agriculture	Artisan Marketplace	Connecting artisans to global markets	91% Funded	2683 DT raised	9 days left
Education					
Arts & Culture					
Healthcare					
Environment					
Community					
Funding Status					
All					
Less than 25%					

Figure 34: Interface de la page « Discover »



## 2.5 Grille d'écran de la page « Créer une campagne »

La figure 2.5 présente la grille d'écran de la page *Créer une campagne*. Cette interface permet aux utilisateurs porteurs de projets de soumettre une nouvelle campagne de financement participatif.

The screenshot shows the 'Create Your Project' form on the TuniFund website. The header includes the TuniFund logo, navigation links (Home, Discover, How It Works, About), and a user profile (Ahmed). The main heading is 'Create Your Project' with the subtitle 'Share your idea with the Tunisian community and get funded'. The form is titled 'Project Details' and includes the following fields:

- Project Title:** A text input field with the placeholder 'Enter a clear, specific title'.
- Category:** A dropdown menu with the placeholder 'Select a category'.
- Short Description:** A text input field with the placeholder 'Provide a brief summary of your project (100-150 characters)'.
- Full Description:** A text input field with the placeholder 'Tell your story. What are you trying to accomplish? Why does it matter? How will you use the funds?'.

Figure 35 : Interface - Partie 1 de la page « Créer une campagne »

The screenshot shows the 'Create Rewards' form on the TuniFund website. The header includes the TuniFund logo, navigation links (Home, Discover, How It Works, About), and a user profile (Ahmed). The main heading is 'Create Rewards' with the subtitle 'Create rewards for your backers'. The form includes the following fields:

- Reward Title:** A text input field with the placeholder 'Early Bird Special'.
- Amount (TND):** A text input field with the placeholder '100'.
- Description:** A text input field with the placeholder 'Describe what backers will receive'.

At the bottom of the form, there are two buttons: 'Save as Draft' and 'Submit for Review'. Below the buttons, a small text line reads: 'By submitting your project, you agree to TuniFund's [Terms of Service](#) and [Privacy Policy](#)'.

Figure 36 : Interface - Partie 2 de la page « Créer une campagne »

## 2.6 Grille d'écran de la page de « Détail de la campagne »

La figure ... illustre la grille d'écran de la page « Détail de la campagne ». Elle permet aux utilisateurs de consulter les informations détaillées d'une campagne (projet) spécifique, ainsi que de contribuer en effectuant un don.

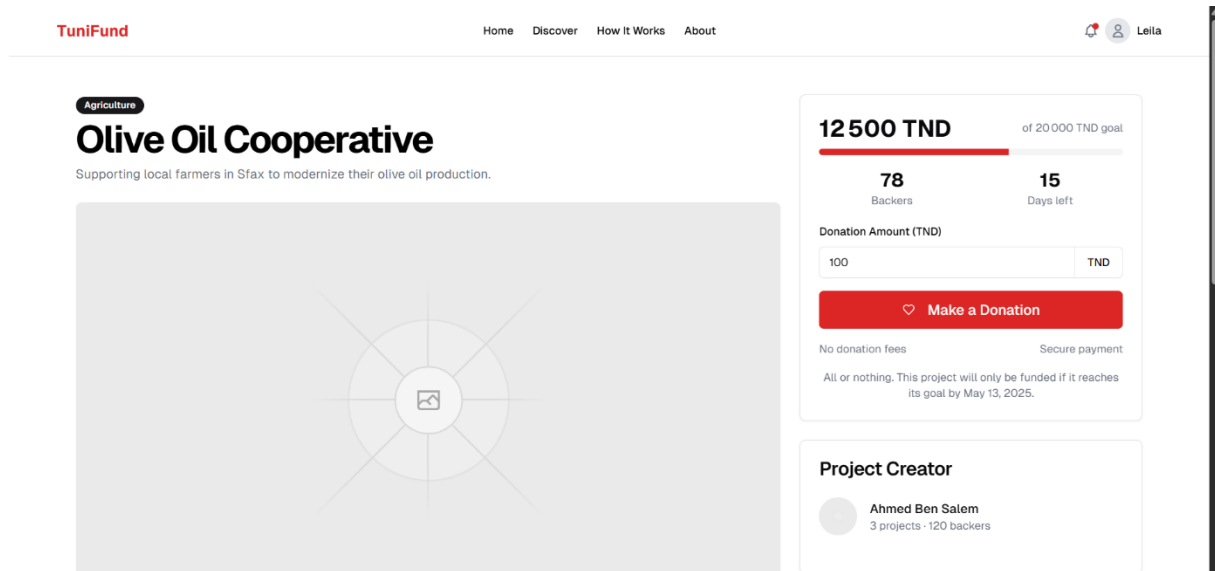


Figure 37 : Interface de la page « Détail de la campagne »

## 2.7 Grille d'écran de la page de « Dashboard Admin »

La figure suivante illustre la grille d'écran du tableau de bord destiné à l'administrateur. Cette interface permet à l'administrateur de gérer efficacement l'ensemble de la plateforme.

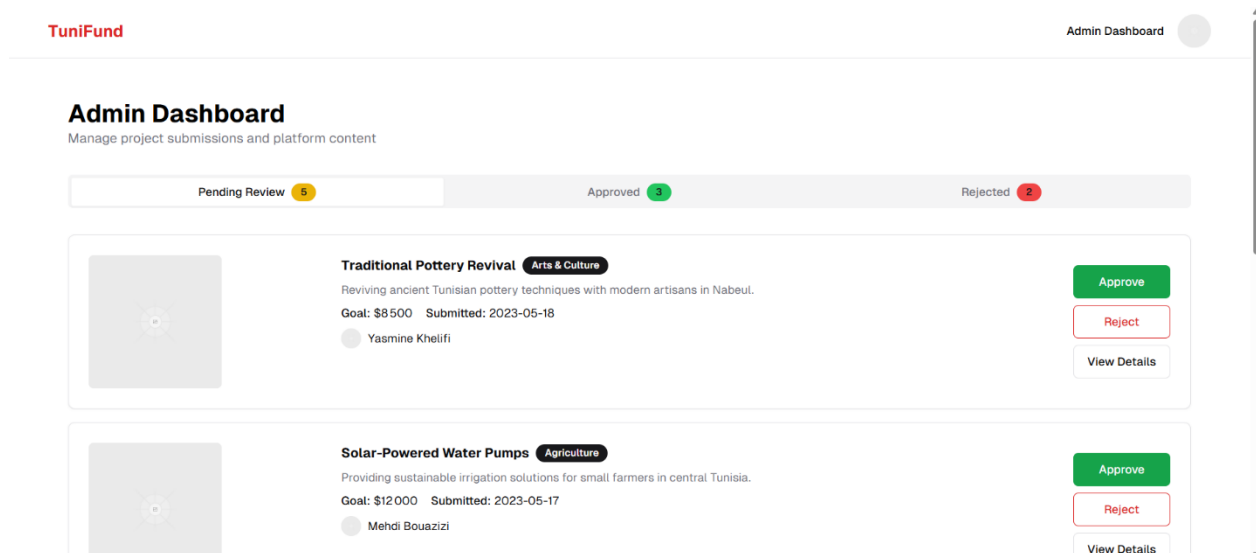


Figure 38 : Interface de la page « Dashboard Admin »

## Conclusion

Dans ce chapitre, nous avons présenté l'environnement matériel et logiciel utilisé lors du développement de notre projet. Ensuite, nous avons réalisé une démonstration pratique pour illustrer le bon fonctionnement de notre application, en mettant en avant ses principales fonctionnalités et son comportement dans des cas réels d'utilisation.

## **Conclusion générale et perspectives**

Le projet *TuniFund* est une plateforme de crowdfunding visant à faciliter la collecte de fonds pour des projets à impact social et économique. Grâce à une architecture technique solide basée sur des technologies modernes comme **Node.js**, **React**, et **MongoDB**, la plateforme offre une expérience utilisateur fluide et sécurisée. Les fonctionnalités de base telles que la création de campagnes, la gestion des dons et l'authentification des utilisateurs sont intégrées de manière efficace et conviviale.

L'interface utilisateur a été pensée pour être simple et intuitive, permettant aux utilisateurs de créer et de suivre des campagnes facilement. Le système d'administration est également conçu pour offrir une gestion complète des campagnes et des utilisateurs, assurant la transparence et l'efficacité de l'ensemble du processus.

Bien que le projet *TuniFund* soit déjà fonctionnel, plusieurs améliorations et évolutions sont envisagées pour l'avenir :

### **1. Optimisation de l'expérience utilisateur :**

L'une des priorités futures est l'amélioration continue de l'interface utilisateur et de l'expérience utilisateur (UX/UI). L'ajout de fonctionnalités plus avancées, comme des recommandations personnalisées de campagnes, pourrait renforcer l'engagement des utilisateurs. Ces améliorations permettront de rendre la navigation encore plus intuitive et agréable.

### **2. Renforcement de la sécurité :**

Des mesures de sécurité supplémentaires, telles que l'authentification multifactorielle (MFA) et des systèmes avancés de détection de fraudes, pourraient être intégrées pour garantir la protection des utilisateurs et des données personnelles. La mise en place de telles solutions renforcera la confiance des utilisateurs dans la plateforme.

### **3. Expansion des fonctionnalités :**

L'ajout de nouvelles fonctionnalités, comme la possibilité de suivre l'impact des projets financés ou des outils permettant de mieux visualiser l'utilisation des fonds, pourrait augmenter la transparence et l'engagement des donateurs. Des options de financement participatif en temps réel ou la possibilité de suivre les projets en fonction de leur évolution pourraient également être envisagées pour améliorer l'interaction avec la plateforme.

4. **Amélioration des capacités d'analyse et de prédiction :**

Une meilleure capacité à analyser les tendances et à prédire le succès des campagnes pourrait être un atout majeur pour l'optimisation des collectes de fonds. L'utilisation d'outils d'analyse avancée pour évaluer les performances des campagnes et ajuster les stratégies pourrait considérablement améliorer les résultats pour les utilisateurs et les administrateurs.

5. En conclusion, *TuniFund* dispose d'un fort potentiel pour devenir une plateforme encore plus robuste et complète, offrant une meilleure expérience à ses utilisateurs et contribuant efficacement à la réussite des projets qu'elle soutient.

## Webographie

1. GoFundMe – Site officiel  
<https://www.gofundme.com> [Consulté le 20 mars 2025]
2. Kickstarter – Plateforme de financement participatif  
<https://www.kickstarter.com> [Consulté le 22 mars 2025]
3. Indiegogo – Crowdfunding platform  
<https://www.indiegogo.com> [Consulté le 25 mars 2025]
4. Tunisia Crowdfunding  
<https://www.tunisiacrowdfunding.com> *(si le site est accessible)* [Consulté le 27 mars 2025]
5. DabaPay – Solution locale de paiement  
<https://www.dabapay.tn> *(vérifié si existant)* [Consulté le 30 mars 2025]
6. React – Documentation officielle  
<https://react.dev> [Consulté le 2 avril 2025]
7. Node.js – Site officiel  
<https://nodejs.org> [Consulté le 3 avril 2025]
8. Express.js – Documentation  
<https://expressjs.com> [Consulté le 5 avril 2025]
9. MongoDB – Site officiel  
<https://www.mongodb.com> [Consulté le 7 avril 2025]
10. Axios – GitHub / documentation  
<https://axios-http.com> [Consulté le 10 avril 2025]
11. Postman – API Platform  
<https://www.postman.com> [Consulté le 12 avril 2025]
12. GitHub – Plateforme de gestion de versions  
<https://github.com> [Consulté le 15 avril 2025]

13. StarUML – Outil de modélisation UML

<http://staruml.io>

[Consulté le 18 avril 2025]

14. Figma – Interface Design Tool

<https://www.figma.com>

[Consulté le 13 mai 2025]