

ENSIIE

RAPPORT DE STAGE DE 1A

**Prokli**

Anne Pottier, Vivien Seignoux, Nicolas Menahem



Encadré par le C-19 et l'IMT Starter

1 juin 2024 - 31 juillet 2024

# Remerciements

Nous tenons à remercier toutes les personnes qui ont fait partie de cette aventure, qui nous ont accompagnés dans notre stage et dans la création de notre projet :

- Hugues Randriatsoa
- Florence Etheve
- Théo Cabrero
- Nassim Khider
- le groupe TrackX

Ainsi que les intervenants de la Summer School et les équipes de la Summer School.

# Contexte

Dans le contexte de la Summer School, nous avons tous les trois travaillé pendant deux mois sur un projet entrepreneurial : Prokli. Durant ces deux mois, et sous la tutelle des deux incubateurs de l'ENSIIE, de Télécom SudParis et de l'Institut Mines-Télécom Business School ; le C-19 et l'IMT Starter, nous avons créé et lancé notre projet : Prokli (anciennement appelé BuddyBattle). Ce projet nous a demandé à la fois d'utiliser et d'approfondir nos compétences de développement mais aussi de découvrir le monde de l'entrepreneuriat : composer un business model, un plan de financement, contacter des prospects, gérer le marketing et travailler en équipe.

Nous avons donc décidé de développer une application mobile innovante destinée aux jeunes adultes âgés de 18 à 25 ans. Cette application a pour objectif de proposer des défis variés et stimulants pour rendre leurs vacances plus dynamiques et divertissantes.

# Contents

<b>Remerciements</b>	<b>1</b>
<b>Contexte</b>	<b>2</b>
<b>1 Présentation du projet entrepreneurial</b>	<b>5</b>
1.1 Origine de l'idée . . . . .	5
1.2 Marchés et positionnement de notre produit par rapport à la concurrence .	5
1.2.1 Marché des applications mobiles . . . . .	6
1.2.2 Marché de la gestion de réseaux sociaux . . . . .	6
1.3 Réponses à notre produit . . . . .	7
1.4 Business model . . . . .	8
1.4.1 Pack Challenger - 19,99 € / mois . . . . .	8
1.4.2 Pack Influencer - 99,99 € / mois . . . . .	8
1.4.3 Pack Influencer Zen - 199,99 € / mois . . . . .	8
1.5 Plan et projections . . . . .	9
1.5.1 Marketing . . . . .	9
1.5.2 Recrutements et R&D . . . . .	9
1.5.3 Développement de l'application . . . . .	9
1.5.4 Besoins financiers . . . . .	9
1.5.5 Projection financière . . . . .	10
<b>2 Création technique de l'application</b>	<b>11</b>
2.1 Objectifs . . . . .	11
2.2 Périmètre du Projet . . . . .	11
2.2.1 Fonctionnalités principales . . . . .	11
2.3 Spécifications Fonctionnelles . . . . .	12
2.3.1 Gestion des utilisateurs . . . . .	12
2.3.2 Système de défis . . . . .	12
2.3.3 Navigation et Interface Utilisateur . . . . .	12
2.3.4 Intégration sociale . . . . .	13
2.4 Spécifications Techniques . . . . .	13
2.4.1 Technologies utilisées . . . . .	13
2.4.2 Architecture . . . . .	13
2.4.3 Sécurité . . . . .	13
2.5 Exigences de Performance . . . . .	14
2.5.1 Temps de chargement rapide . . . . .	14

2.5.2	Scalabilité . . . . .	14
2.5.3	Fiabilité . . . . .	14
2.6	Planification et Gestion de Projet . . . . .	14
2.6.1	Étapes du projet . . . . .	14
2.6.2	Ressources . . . . .	15
2.7	Risques et Gestion des Risques . . . . .	15
2.7.1	Identification des risques . . . . .	15
2.7.2	Plans de mitigation . . . . .	16
2.8	Annexes . . . . .	17
2.8.1	Diagrammes . . . . .	17
2.8.2	Références . . . . .	17
2.8.3	Git . . . . .	17
2.9	Développement . . . . .	18
2.9.1	Initialisation du Projet . . . . .	18
2.9.2	Structure de l'application . . . . .	18
2.9.3	Développement des Fonctionnalités . . . . .	19
2.9.4	Intégration Backend avec Appwrite . . . . .	20
2.9.5	Interface Utilisateur et Expérience Utilisateur . . . . .	21
2.9.6	Optimisation . . . . .	21
2.9.7	Déploiement et Maintenance . . . . .	22
2.9.8	Difficultés Rencontrées et Solutions Apportées . . . . .	23
2.10	Conclusion . . . . .	24
<b>3</b>	<b>Conclusion</b>	<b>25</b>
3.1	Vivien Seignoux . . . . .	25
3.2	Anne Pottier . . . . .	25
3.3	Nicolas Menahem . . . . .	25

# Présentation du projet entrepreneurial

## 1.1 Origine de l'idée

Au cours de vacances entre amis, nous nous sommes rendus compte que l'ennui était un problème majeur, gâchant d'une part notre expérience et les souvenirs de nos vacances. Nous nous sommes alors demandé si d'autres personnes de notre génération ressentaient la même problématique.

Lors d'une étude menée sur un échantillon de 100 personnes (en très grande partie des étudiants de 15 à 30 ans avec une majorité de 18-23 ans), nous avons constaté deux problématiques récurrentes lors de vacances entre amis : l'ennui et le manque de souvenirs. En effet, **70%** admettent ne pas avoir suffisamment de souvenirs photos/vidéos de vacances et **90%** s'ennuient lors de vacances entre amis, dont **40%** sur une journée entière ou plus. Au lieu de chercher une activité, plus de **70%** admettent rester chez eux dormir ou regarder les réseaux sociaux.

C'est ainsi que **Prokli** a été créé. Prokli est une application de défis, offrant une gamme de thèmes variés afin que chaque utilisateur puisse y trouver son bonheur. Nous créons également des gammes de défis spécifiques à nos partenaires, tels que des accrobranches, des zoos ou des musées. Nos utilisateurs réalisent ces défis liés à nos partenaires et se prennent en photos pour les valider. Ces photos peuvent ensuite être postées sur les réseaux sociaux de nos partenaires de manière automatisée, directement depuis notre application. Les partenaires peuvent choisir de modérer eux-mêmes ce qui sera posté ou de nous laisser faire cela pour eux.

## 1.2 Marchés et positionnement de notre produit par rapport à la concurrence

Lors du lancement du projet Prokli, nous avons décidé de nous positionner sur deux marchés simultanément :

- **Le marché des applications mobiles**, en l'occurrence liées au divertissement, puisque notre application est dédiée aux défis.
- **Le marché de la gestion des réseaux sociaux d'entreprises**, car notre ap-

l'application propose à nos partenaires d'automatiser la publication de posts sur leurs réseaux sociaux en lien avec leur activité.

### 1.2.1 Marché des applications mobiles

Le marché des applications mobiles est en pleine expansion, et cette dynamique ne semble pas prête à changer dans les années à venir. En effet, **90%** du temps passé sur nos téléphones est consacré à des applications mobiles, pourcentage qui ne devrait pas diminuer avec les récents progrès en *IA*, *5G*, et *Réalité Augmentée*.

De plus, les applications mobiles jouent un rôle important dans le choix des vacances des jeunes adultes. En effet, la moitié d'entre eux utilisent ces applications pour trouver des idées d'activités à faire, ce qui souligne le potentiel de notre application.

#### Concurrents dans les applications de défis

Nous avons plusieurs concurrents dans le secteur du divertissement lié à des défis :

- “**TOZ**”, avec plus de **500k** téléchargements, se concentre sur des défis ou des jeux d'alcool autour des soirées.
- “**Action ou vérité**”, avec **10M** téléchargements, cible également l'univers des soirées.

Ces deux applications mettent l'accent sur des parties courtes jouées lors de soirées. Nous nous démarquons avec **Prokli**, qui permet de s'amuser à tout moment, de jour comme de nuit. De plus, notre grande diversité de défis permet à l'utilisateur de jouer peu importe le lieu et les envies, sans contraintes spécifiques.

De plus, après avoir lu les commentaires d'application similaires à la nôtre, nous nous sommes rendus compte que les critiques les plus récurrentes sont les suivantes : la présence de **publicités non désirées** sur l'application, et la présence d'un **système d'abonnement freemium**.

En effet, “Action ou vérité” contient un système de freemium, à la différence de “TOZ” qui en possédait un avant mais l'a supprimé, suite aux nombreuses remarques des utilisateurs. De plus, aucune de ces deux applications ne contient de publicités.

Nous avons donc décidé de ne mettre aucune publicité ni abonnement freemium pour le moment, excepté des publicités non obligatoires en échange desquelles les utilisateurs peuvent obtenir des avantages. Ainsi, nous offrons dès les premiers mois une expérience utilisateur de qualité le temps de se faire une place dans le marché.

### 1.2.2 Marché de la gestion de réseaux sociaux

Le marché de la gestion des réseaux sociaux est en forte croissance, estimée à **25%** annuellement au cours des 5 prochaines années. En comparaison, le marché de la publicité croît de **10 à 15%** et celui des logiciels de gestion d'entreprises de **7 à 10%**.

**91%** des entreprises utilisent les réseaux sociaux pour le marketing et allouent en moyenne **12%** de leur budget marketing total à ces réseaux, représentant 300 à 600 euros par mois pour nos partenaires tels que les accrobranches.

En résumé, le marché de la gestion des réseaux sociaux est l'un des plus dynamiques actuellement, avec une croissance plus rapide que de nombreux autres marchés majeurs. Nous avons choisi d'aborder ce besoin sous un angle différent des agences de communication classiques, en offrant un service via une application avec un contenu original et collaboratif.

### Concurrents dans la gestion des réseaux sociaux

**Digitaleo** et **Solocal** sont deux entreprises de gestion des réseaux sociaux :

- **Digitaleo** avec un chiffre d'affaires de **10M €**.
- **Solocal** avec un chiffre d'affaires de **240M €** en 2023 (leader français).

Les services compris dans leur pack de base sont les suivants : Création des réseaux sociaux, publications promotionnelles 1/3 fois par semaine, stratégie marketing non personnalisée, gestion des commentaires et rapports standards mensuels. Ces packs débutent à **250 €** par mois et peuvent aller jusqu'à **500 €**.

Nous nous différencions par une **stratégie marketing personnalisée**, des publications originales et créatives qui reflètent l'ambiance du parc, le tout à un tarif inférieur !

## 1.3 Réponses à notre produit

Selon les réponses à nos questionnaires et nos propres idées, nous avons contacté plusieurs activités pour proposer nos services et valider notre business plan basé sur les partenariats :

- Pour les **25%** intéressés par des activités sportives ou naturelles, nous avons contacté des accrobranches, des raftings, des via ferrata, des kartings, des paintballs.
- Pour les **15%** intéressés par des visites culturelles, nous avons contacté des zoos, des musées, des châteaux, des océariums...
- Pour les **35%** restants intéressés par des jeux à l'intérieur, nous avons contacté des laser games, des escape games, et prévoyons une gamme de défis à faire à la maison.

Nous avons envoyé environ **600 mails**, appelé **70 villes et villages** et obtenu **6 entretiens** pour **263 prospects** au total. Avec **4 réponses positives** en attente de



signature pour une phase de test et **7 autres commerces** demandant un entretien à la fin de la saison, nous avons confirmé l'intérêt des activités pour notre produit et ajusté notre business model en conséquence.

La gestion de nos réseaux sociaux (Tiktok et Instagram) a également renforcé notre confiance dans l'intérêt du public pour notre produit, avec **32 000 vues** sur Tiktok (le nombre de vues par vidéo a quadruplé en seulement deux mois) et **600 abonnés** sur Instagram, renforçant notre crédibilité auprès des investisseurs.

### 1.4 Business model

Notre business model repose sur un système d'abonnement pour nos partenaires, qui paient chaque mois pour bénéficier de nos défis, de la gestion de leurs réseaux sociaux, et de la visibilité apportée par notre application.

Pour les premiers mois, nous ne prévoyons pas de mettre en place un abonnement payant dans l'application, préférant plutôt un lancement en 2026 si nécessaire. Nous proposons trois abonnements différents, dont les prix ont été déterminés en fonction du temps de travail et de la concurrence (toujours en dessous des prix des agences concurrentes).

#### 1.4.1 Pack Challenger - 19,99 € / mois

Ce pack comprend la création d'une gamme de défis spécifiques à l'activité du partenaire. De plus, l'entreprise partenaire sera visible par tous les utilisateurs de Prokli, ce qui peut accroître sa visibilité.

#### 1.4.2 Pack Influencer - 99,99 € / mois

Ce pack inclut tous les avantages du Pack Challenger, avec en plus l'automatisation des publications sur les réseaux sociaux. Les légendes des posts sont générées par IA et les photos sont prises par des clients au sein de l'entreprise, sans intervention de la part du partenaire. Ils doivent toutefois modérer les photos à publier, s'ils souhaitent une modération.

#### 1.4.3 Pack Influencer Zen - 199,99 € / mois

Le Pack Influencer Zen comprend la création d'une gamme de défis liés à l'activité du partenaire, l'automatisation des publications sur les réseaux sociaux, et la modération des posts selon leurs critères. Les partenaires n'ont plus aucune action à faire concernant la publication des posts.

## 1.5 Plan et projections

### 1.5.1 Marketing

Nous nous concentrons sur deux réseaux sociaux pour maintenir un lien avec nos utilisateurs et partenaires :

- **Tiktok** : Nous sert de lieu de communication avec nos futurs utilisateurs. Avec un public plutôt de jeunes adultes, nous nous efforçons de suivre les trends actuelles, toujours dans le cadre de notre start-up, en jouant sur notre quotidien d'entrepreneurs avec deux ou trois vidéos par semaines. Cette stratégie a porté ses fruits puisque nous cumulons maintenant 32 000 vues et que notre nombre de vues a été multipliée par 4 entre nos 6 premières vidéos et nos 6 dernières.
- **Instagram** : Nous sert de plateforme de visibilité pour nos partenaires, c'est ce réseau social qui est relié à notre site web et c'est toujours vers lui que nous renvoyons nos partenaires s'ils veulent voir nos réseaux sociaux. Avec son aspect plus professionnel et des explications sur qui nous sommes, ce que nous proposons dans l'ambiance visuelle de l'application, ce réseau social qui comptabilise plus de 600 abonnés nous donne plus de crédit au près de nos correspondants

### 1.5.2 Recrutements et R&D

Pour assurer un développement stable de l'application et une bonne mise en vente et attractivité nous prévoyons de recruter dès fin 2024 un **développeur** et un **consultant en marketing**.

En 2026 également, pour suivre nos besoin de nouveautés sur l'application et continuer à assurer un service fluide de la gestion des réseaux sociaux des entreprises via notre application et via une modération par IA nous prévoyons le recrutement de **deux ingénieurs informatiques** supplémentaires.

### 1.5.3 Développement de l'application

Pour assurer un bon dynamisme de l'application et éviter une perte d'intérêt des utilisateurs, les banques de défis seront régulièrement mises à jour et de nouveaux modes de jeux devraient arriver chaque année, avec par exemple le mode de parties entre amis qui devrait apparaître en 2026 et faire partie de l'abonnement payant ou encore la possibilité d'élire le meilleur défi du mois ou de posséder un chat d'équipe.

### 1.5.4 Besoins financiers

Pour couvrir les dépenses en marketing, recherche, et salaires des nouveaux associés, nous prévoyons plusieurs levées de fonds :

- En 2024, environ **10 000 euros** venant de nos proches.

- En 2025, environ **130 000 euros** de bourses, prêts d'honneur ou crowdfunding.
- En 2027, environ **200 000 euros** de business angels.

### 1.5.5 Projection financière

Nous prévoyons un chiffre d'affaire plus élevé de **mai à août** en raison de notre abonnement mensuel ciblant les activités de vacances et les festivals. Nos dépenses seront plus importantes en 2025 et 2027, mais les levées de fonds nous permettront de rester rentables et de nous établir durablement pour 2028.

	<b>2025</b>	<b>2026</b>	<b>2027</b>
<b>Nombre moyen de partenaires</b>	22,92	34,92	44,5
<b>Entrées d'argent total des partenaires</b>	29 425 €	83 800 €	106 400 €
<b>Entrées d'argent total des ventes d'abonnements</b>	0 €	35 000 €	40 000 €
<b>Total Entrées</b>	<b>29 425 €</b>	<b>118 800 €</b>	<b>146 400 €</b>
<b>Salaires</b>	36 000 €	36 000 €	108 000 €
<b>Taxes, frais techniques, frais généraux...</b>	48 470 €	63 670 €	117 178 €
<b>Total Sorties</b>	<b>84 470 €</b>	<b>99 670 €</b>	<b>225 178 €</b>
<b>Levées de fonds</b>	140 000 €	0 €	200 000 €
<b>Total</b>	<b>84 955 €</b>	<b>19 130 €</b>	<b>121 222 €</b>

Table 1.1: Projections financières de 2025 à 2027

# Création technique de l'application

## 2.1 Objectifs

Le développement de l'application a impliqué plusieurs phases techniques clés, allant de la conception initiale de l'architecture à la mise en œuvre des fonctionnalités, en passant par les tests et l'optimisation. Le choix des technologies, les méthodes de développement et les pratiques de gestion de projet ont été cruciaux pour assurer le succès de l'application.

## 2.2 Périmètre du Projet

### 2.2.1 Fonctionnalités principales

a) Le projet inclut un système d'authentification qui permet l'inscription et la connexion sécurisées des utilisateurs. Chaque utilisateur pourra ainsi créer un compte personnel et se connecter en toute sécurité pour accéder aux fonctionnalités de l'application.

b) La gestion des profils utilisateurs permettra à chacun d'afficher les défis qu'il a réalisés ainsi que les photos associées. De plus, les utilisateurs auront la possibilité de personnaliser leur photo de profil, rendant leur expérience plus personnelle et unique.

c) Pour le suivi des défis disponibles, l'application affichera les défis avec une catégorie spécifique pour chaque partenaire. Cela facilitera la navigation et la sélection des défis par les utilisateurs en fonction de leurs intérêts et préférences.

d) En ce qui concerne la publication de défis, les utilisateurs pourront soumettre des défis accompagnés de photos ou de vidéos. Cette fonctionnalité encourage la créativité et l'engagement des utilisateurs en leur permettant de proposer de nouveaux défis à la communauté.

e) Un système de vote binaire sera mis en place pour permettre aux utilisateurs de valider ou non les défis soumis. Ainsi, la communauté pourra évaluer la pertinence et l'intérêt des défis proposés.

f) Une barre de recherche sera disponible pour permettre aux utilisateurs de visualiser facilement les photos et vidéos liées à des défis spécifiques. Cette fonctionnalité améliorera l'accessibilité et la convivialité de l'application.

g) Le classement des utilisateurs sera mis à jour en temps réel, basé sur les points obtenus lors de la réalisation des défis. Cette fonctionnalité introduira un élément de compétition amicale, motivant les utilisateurs à participer activement aux défis.

h) Le système d'équipe assignera aléatoirement une équipe à chaque utilisateur lors de sa première arrivée sur la page de classement. Le classement des équipes sera déterminé par la somme des points des utilisateurs dans chaque équipe, favorisant ainsi le travail d'équipe et la coopération.

i) Enfin, un système de signalement permettra aux utilisateurs de signaler des photos qui seraient contraires au règlement. Cette fonctionnalité aidera à maintenir un environnement respectueux et sécurisé pour tous les utilisateurs.

## 2.3 Spécifications Fonctionnelles

### 2.3.1 Gestion des utilisateurs

L'inscription et la connexion des utilisateurs seront gérées par Appwrite, un service qui assurera non seulement l'inscription et la connexion, mais aussi la déconnexion des utilisateurs. Ce système garantira une gestion fluide et sécurisée des comptes utilisateurs.

Chaque profil utilisateur inclura des éléments personnalisables tels qu'un avatar et une bio. Les utilisateurs pourront également consulter leurs statistiques de défis ainsi que le nombre de points qu'ils ont accumulés. Cette personnalisation renforcera l'engagement et la satisfaction des utilisateurs en leur offrant une expérience plus riche.

### 2.3.2 Système de défis

Le système de défis proposera un catalogue de défis classés par catégorie ou par partenaire. Les utilisateurs auront également la possibilité de créer des défis personnalisés, élargissant ainsi l'offre de défis disponibles et permettant une participation active de la communauté.

Le suivi des défis sera effectué à travers la soumission de preuves sous forme de photos ou de vidéos. Ces preuves seront soumises à une validation par vote binaire, où les autres utilisateurs pourront approuver ou refuser la réussite du défi. Ce processus de validation participative renforce la transparence et l'implication de la communauté.

### 2.3.3 Navigation et Interface Utilisateur

L'interface utilisateur comportera plusieurs onglets principaux, notamment Accueil, Classement, Défis, Profil, et Recherche. Ces onglets faciliteront la navigation au sein de l'application, permettant aux utilisateurs d'accéder rapidement aux différentes fonctionnalités.

L'interface sera conçue pour être intuitive et responsive, offrant une expérience utilisateur optimale sur différents appareils. Pour le style, TailwindCSS sera partiellement utilisé, garantissant une présentation moderne et cohérente de l'application.

### 2.3.4 Intégration sociale

Le partage des défis accomplis, en particulier ceux liés aux partenaires, sera automatisé grâce à Zapier. Cela permettra aux utilisateurs de diffuser leurs réussites sur les réseaux sociaux, renforçant ainsi la visibilité de l'application et l'engagement des utilisateurs.

Des notifications seront mises en place pour rappeler aux utilisateurs les défis en cours et les tenir informés des mises à jour en temps réel. Ces notifications contribueront à maintenir une interaction continue avec l'application, incitant les utilisateurs à rester actifs.

## 2.4 Spécifications Techniques

### 2.4.1 Technologies utilisées

Le frontend de l'application sera développé en utilisant React Native, avec le framework Expo pour faciliter le déploiement multiplateforme. Le style de l'interface utilisateur sera partiellement géré par TailwindCSS, offrant une solution moderne et flexible pour la conception.

Le backend reposera sur Appwrite, qui assurera la gestion des utilisateurs, des bases de données, et du stockage. Cette technologie fournira une infrastructure robuste et sécurisée pour les fonctionnalités essentielles de l'application.

Pour garantir la qualité du code, Jest sera utilisé pour les tests unitaires et d'intégration. Cela permettra d'identifier et de corriger les éventuelles erreurs tout en s'assurant que les différentes parties de l'application fonctionnent bien ensemble.

### 2.4.2 Architecture

L'architecture de l'application sera modulaire, avec une séparation claire entre les composants de l'interface utilisateur (UI) et la logique métier. Cette approche facilitera la maintenance et l'évolution de l'application en rendant les différentes parties du code plus indépendantes et réutilisables.

La gestion de l'état global de l'application sera centralisée à l'aide de la Context API de React. En utilisant `React.createContext` et `useContext`, l'état global pourra être partagé entre les différents composants sans avoir à passer des props de manière excessive. Cette méthodologie favorisera une architecture plus propre et plus maintenable, en évitant les complications liées au passage de props inutiles à travers l'arborescence des composants.

### 2.4.3 Sécurité

La sécurité de l'application sera assurée par Appwrite, qui gérera l'authentification et l'autorisation via un système de permissions. Ce système garantira que seules les personnes autorisées peuvent accéder aux fonctionnalités et aux données sensibles de l'application.

De plus, des mesures seront prises pour empêcher la réutilisation d'anciens mots de passe et l'utilisation d'informations personnelles dans les mots de passe, renforçant ainsi la sécurité des comptes utilisateurs.

Le stockage des données utilisateurs et des contenus sera sécurisé sur les serveurs d'Appwrite. Cette infrastructure garantira la confidentialité et l'intégrité des informations stockées, protégeant les utilisateurs contre les risques de perte ou de vol de données.

## 2.5 Exigences de Performance

### 2.5.1 Temps de chargement rapide

L'application sera optimisée pour offrir des temps de chargement rapides, ce qui est essentiel pour une expérience utilisateur fluide. Pour ce faire, une optimisation des bundles et des assets sera mise en place. Les bundles, qui sont les fichiers compilés contenant tout le code nécessaire (JavaScript, CSS, images, etc.) pour exécuter l'application, seront allégés et optimisés afin de réduire leur taille et d'accélérer leur chargement.

### 2.5.2 Scalabilité

L'architecture backend de l'application sera conçue pour être hautement scalable, en utilisant Appwrite pour gérer un grand nombre de requêtes simultanées.

La scalabilité horizontale consistera à ajouter plus d'instances d'un service pour répartir la charge. Par exemple, en cas de trafic élevé sur l'application, il sera possible d'ajouter davantage de serveurs pour gérer l'augmentation du nombre de requêtes. Cette approche permettra d'améliorer la tolérance aux pannes et la disponibilité du service.

La scalabilité verticale, quant à elle, impliquera d'ajouter davantage de ressources (CPU, RAM, stockage) à une seule instance de service. Cette méthode permettra à un serveur unique de gérer une charge plus importante, garantissant ainsi la performance de l'application même sous des conditions de forte demande.

### 2.5.3 Fiabilité

Pour assurer la fiabilité et la stabilité de l'application, des tests approfondis seront réalisés sur le code. Ces tests incluront des tests unitaires rigoureux effectués à l'aide de l'API Jest. Grâce à ces tests, les développeurs pourront identifier et corriger les erreurs potentielles avant la mise en production, garantissant ainsi une application stable et fiable pour les utilisateurs.

## 2.6 Planification et Gestion de Projet

### 2.6.1 Étapes du projet

- Conception initiale

- Définition des fonctionnalités, création des maquettes, élaboration de l'architecture technique.
- Développement des composants frontend. (03/06/2024) - (01/07/2024)
- Intégration avec Appwrite. (10/06/2024) - (22/07/2024)
- Tests et débogage. (25/07/2024) - (27/07/2024)
- Phase de bêta-test. (27/07/2024) - (05/08/2024)
- Déploiement sur les stores. (05/08/2024)
- Maintenance et mises à jour.

### 2.6.2 Ressources

- **Équipe de développement**
  - Développement des composants frontend | Anne Pottier / Vivien Seignoux
  - Intégration avec Appwrite | Nicolas Menahem
  - Intégration avec Zapier | Vivien Seignoux / Nicolas Menahem
  - Building de l'application | Anne Pottier / Nicolas Menahem
- Outils et technologies nécessaires.
  - Langages de Programmation | JavaScript/TypeScript
  - Frameworks et Bibliothèques | React Native et Expo
  - Environnements de Développement Intégré (IDE) | Visual Studio Code et Expo Go
  - Outils de Gestion de Versions | Git et GitLab
  - Outils de Conception | Gimp et Figma
  - Outils de Test | Jest
  - Backend et Bases de Données | Appwrite
  - Architecture et état global | React Context API

## 2.7 Risques et Gestion des Risques

### 2.7.1 Identification des risques

Les risques associés au projet peuvent être classés en deux grandes catégories : techniques et commerciales.



Les risques techniques incluent la possibilité de rencontrer des bugs, des problèmes de performance, et des difficultés lors des builds. Ces aspects peuvent affecter la stabilité, la rapidité, et la fiabilité de l'application, compromettant ainsi l'expérience utilisateur.

Sur le plan commercial, il existe des risques liés à l'adoption de l'application sur les plateformes PlayStore et AppStore, ainsi qu'à son acceptation par les utilisateurs. L'échec à obtenir une bonne visibilité sur ces plateformes ou à convaincre les utilisateurs de télécharger et d'utiliser l'application pourrait nuire à son succès commercial.

### 2.7.2 Plans de mitigation

Pour gérer ces risques, plusieurs stratégies de mitigation seront mises en place.

Sur le plan technique, des tests unitaires seront effectués avec Jest pour identifier et résoudre les bugs dès les premières phases de développement. Un bêta testing sera également réalisé pour obtenir des retours d'utilisateurs réels et identifier les problèmes potentiels avant le déploiement final. De plus, les services d'Expo seront utilisés pour assurer la création de builds fonctionnels, minimisant ainsi les risques de problèmes lors du processus de build.

Concernant les risques commerciaux, l'application sera optimisée pour un déploiement adapté grâce à Expo, garantissant une compatibilité et une performance optimales sur diverses plateformes. Le déploiement sera effectué simultanément sur iOS et Android, maximisant ainsi la portée de l'application dès son lancement et augmentant ses chances d'adoption par un large public.

## 2.8 Annexes

### 2.8.1 Diagrammes

- Interfaces utilisateur



Figure 2.1: Mockup

### 2.8.2 Références

- À l'arrivée sur l'application, inscrivez-vous pour créer un compte personnel.
- Une fois connecté, choisissez une catégorie de défis que vous souhaitez réaliser en appuyant sur les carrés blancs représentant les différentes options disponibles.
- Prenez ensuite votre photo ou vidéo pour documenter la réalisation de votre défi, puis postez-la sur la plateforme.
- Attendez les votes des autres utilisateurs pour savoir si votre défi est validé ou non.
- Si votre défi est validé, vous pourrez voir votre classement s'améliorer, ce qui vous encouragera à participer à d'autres défis pour continuer à progresser.

### 2.8.3 Git

- Lien vers le git de l'application, le code se trouve dans le dossier BuddyBattle (ancien nom du projet) :
  - <https://github.com/Saccharoze/Prokli/tree/main/reactnact/BuddyBattle>

## 2.9 Développement

### 2.9.1 Initialisation du Projet

La première étape du développement consiste à configurer l'environnement de travail. Pour ce faire, le projet React Native sera initialisé à l'aide d'Expo. Ensuite, les dépendances nécessaires, telles que React Native, Expo, TailwindCSS, et Jest, seront installées via npm (Node Package Manager). Cette configuration de base est essentielle pour préparer le terrain en vue du développement de l'application.

### 2.9.2 Structure de l'application

L'application sera organisée de manière claire et modulaire pour faciliter le développement et la maintenance.

- Le dossier **app** contiendra les différentes pages et composants principaux, chacun géré par des membres de l'équipe :
  - **auth** (géré par Nicolas Menahem) : Responsable de la gestion de l'authentification.
  - **part** (géré par Vivien Seignoux et Anne Pottier) : Gère les éléments liés aux partenaires, en utilisant FlatList pour afficher les défis associés et en intégrant des bibliothèques comme expo-linear-gradient pour les arrière-plans et react-native-reanimated pour les animations.
  - **saved** (géré par Vivien Seignoux et Anne Pottier) : Gère les éléments sauvegardés par les utilisateurs, avec une structure similaire à celle du dossier "part".
  - **tabs** (géré par Vivien Seignoux et Anne Pottier) : Contendra les composants pour les onglets de navigation.
- Le dossier **assets** regroupera les ressources statiques, telles que les images et les polices.
- Le dossier **components** (géré par Anne Pottier et Vivien Seignoux) rassemblera les composants réutilisables comme CategorieCard, Create, CustomButton, DefiCard, et bien d'autres, qui seront utilisés à travers l'application.

Voici la liste de l'ensemble des composants utilisés et leur utilité :

- CategorieCard : Composant pour afficher les catégories de défis
- Create : Composant qui ouvre l'appareil photo
- CustomButton : Composant pour customiser un bouton
- DefiCard : Composant pour afficher les défis
- EmptyState : Composant à afficher si la FlatList est vide
- FormField : Formulaire lors de l'inscription ou connexion
- InfoBox : Composant affichant l'avatar, les points et le nombre de post dans la page profile
- PartenaireCard : Composant pour afficher les logos des différents partenaires

- Poll : Composant pour afficher les deux choix des votes
- SearchInput : Barre de recherche
- UserCard : Composant pour afficher les différents utilisateurs
- VideoCard : Composant pour afficher les photos et vidéos des défis postés
- Le dossier `constants` définira les constantes utilisées dans l'application.
- Le dossier `context` (géré par Nicolas Menahem) abritera le Context API pour la gestion de l'état global, notamment dans `'context/GlobalProvider.js'`.
- Le dossier `lib` inclura des bibliothèques et utilitaires, comme `'appwrite.js'` pour la gestion des interactions avec Appwrite
- Enfin, le dossier `node_modules` contiendra les dépendances du projet.

### 2.9.3 Développement des Fonctionnalités

- **Authentification (Dossier auth)**

- **Authentification** : Gérée avec Appwrite et l'API Context de React, cette fonctionnalité inclura l'inscription, la connexion, la déconnexion des utilisateurs, etc., à travers des fonctions définies dans `'lib/appwrite.js'`.  
(cf : 9.4)

- **Navigation (Dossier tabs)**

- **Navigation** : La navigation sera gérée par React Navigation, avec des écrans spécifiques développés par différents membres de l'équipe, tels que :
  - \* `_layout.tsx` (Nicolas Menahem) : Définit le layout de base pour les onglets, permettant la navigation et le changement dynamique de la couleur des icônes.
  - \* `home.tsx` (Vivien Seignoux / Anne Pottier) : Liste les défis postés par d'autres utilisateurs, intégrant des gradients de couleurs en fond d'écran et un système de vote.
  - \* `bookmark.tsx` (Vivien Seignoux / Anne Pottier) : Liste les utilisateurs et leurs informations, en utilisant divers composants et bibliothèques pour un affichage optimisé.
  - \* `defi.tsx` (Vivien Seignoux / Anne Pottier) : Liste les différentes catégories de défis, avec un design enrichi par des gradients de couleurs.
  - \* `profile.tsx` (Vivien Seignoux / Anne Pottier) : Affiche les défis réalisés par l'utilisateur, ainsi que ses points, avatar, et un bouton de déconnexion.

### 2.9.4 Intégration Backend avec Appwrite

- **Configuration d'Appwrite (Fichier `appwrite.js`)** | Nicolas Menahem

- L'intégration avec Appwrite sera configurée dans le fichier `appwrite.js` (géré par Nicolas Menahem). Cette configuration comprendra l'initialisation de l'instance Appwrite et la définition des fonctions nécessaires pour interagir avec le backend.

Parmi ces fonctions, on trouvera :

- \* `createUser` : créer un utilisateur
- \* `signUp` : pour s'inscrire
- \* `signIn` : pour se connecter
- \* `signOut` : pour se déconnecter
- \* `getCurrentUser` : obtenir les informations de l'utilisateur actuel
- \* `getAllUsers` : retourne un tableau de tous les utilisateurs de la base de données
- \* `getAllPosts` : retourne un tableau de toutes les photos/vidéos de la base de données
- \* `getAllDefis` : retourne un tableau de tous les défis de la base de données
- \* `getLatestPosts` : retourne un tableau des 10 dernières photos/vidéos postés
- \* `searchPosts` : recherche un défi particulier suivant son nom
- \* `getUserPosts` : retourne un tableau de toutes les photos/vidéos posté par l'utilisateur actuel
- \* `getNotMadeDefi` : retourne un tableau de tous les défis réalisés par l'utilisateur
- \* `getVotedPosts` : retourne un tableau de toutes les photos/vidéos où l'utilisateur a soumis un vote
- \* `getNotVotedPosts` : retourne un tableau de toutes les photos/vidéos où l'utilisateur n'a pas voté
- \* `getFilePreview` ; `getFileInfo` ; `uploadFile` ; `createVideo` : permet d'uploader un média dans la base de données
- \* `updateIntegers` : met à jour le compteur de vote après qu'un utilisateur ait voté
- \* `deleteDefi` : supprimer un défi posté de la base de données
- \* `uploadAvatar` ; `getAvatarPreview` ; `updateAvatar` : upload un media (avatar) dans la base de donnée
- \* `teamChosen` : attribue une équipe (montagne, soleil, océan) à l'utilisateur
- \* `getRank` : obtient un tableau des trois équipes classées en fonction de leurs points
- \* `reportDefi` : ajoute un signalement supplémentaire à un défi posté sur l'application (au bout de 10 signalements le défi est supprimé)

- **Structure de la base de données** | Nicolas Menahem









KEY		TYPE	DEFAULT VALUE
 username	Required	String	-
 email	Required	Email	-
 avatar	Required	Url	-
 points		Integer	0
 photosvoted		Relationship with photosvoted	-
 accountId	Required	String	-
 team		String	-
 reported		Relationship with reported	-

Figure 2.2: Attributs de la classe user

### 2.9.5 Interface Utilisateur et Expérience Utilisateur

L'interface utilisateur (UI) et l'expérience utilisateur (UX) seront optimisées à l'aide de TailwindCSS pour garantir une apparence cohérente et réactive.

- **Utilisation de TailwindCSS** (géré par Anne Pottier et Vivien Seignoux) : TailwindCSS sera configuré dans le fichier tailwind.config.js, où les classes de style personnalisées seront définies.

Ces classes seront ensuite appliquées aux différents composants de l'application pour styliser efficacement l'interface, tout en assurant une cohérence visuelle et une adaptabilité sur différents appareils.

### 2.9.6 Optimisation

L'optimisation de l'application sera une priorité pour garantir une performance fluide et une expérience utilisateur satisfaisante.

- **Optimisation de l'application** (géré par Nicolas Menahem) : Des techniques d'optimisation seront mises en œuvre pour réduire la taille des bundles, ce qui contribuera à diminuer les temps de chargement.

De plus, les requêtes effectuées vers Appwrite seront optimisées pour améliorer la rapidité et l'efficacité des interactions backend, garantissant ainsi une application plus réactive et moins gourmande en ressources.

<input type="checkbox"/>	Document ID	title	photo	valide	refuse	users	done	points	reports
<input type="checkbox"/>	66a6459...cb4b78a	Prenez la tyrolienne...	https://cloud.appwri...	0	0	66a6442b000379be...	false	50	0
<input type="checkbox"/>	66a6449...8dfa74a	freer	https://cloud.appwri...	1	0	66a6442b000379be...	true	100	0
<input type="checkbox"/>	66a6111...371bce2	Fais 10 squats	https://cloud.appwri...	1	0	668cfb60000d6b1c1...	true	10	0
<input type="checkbox"/>	66a39f2...694c207	Prenez vous en photo...	https://cloud.appwri...	1	0	668cfb60000d6b1c1...	true	30	0
<input type="checkbox"/>	669f6ba...0b569de	Faites le quick jump...	https://cloud.appwri...	1	0	668cfb60000d6b1c1...	true	40	0
<input type="checkbox"/>	669f5f4...2679a55	Échappez vous du lab...	https://cloud.appwri...	1	0	668cfb60000d6b1c1...	true	20	0

Figure 2.3: Attributs de la classe photo

KEY		TYPE	DEFAULT VALUE
T	name	Required String	-
#	points	Required Integer	-
T	categorie	Required String	-

Figure 2.4: Attributs de la classe défis

### 2.9.7 Déploiement et Maintenance

Le déploiement de l'application sur les plateformes mobiles ainsi que sa maintenance continue feront l'objet d'une attention particulière pour assurer sa pérennité et son évolution.

- **Déploiement de l'application** : La préparation pour le déploiement sur les stores, tels que l'App Store et Google Play, impliquera la configuration des builds et des certificats nécessaires pour chaque plateforme. Cette étape est cruciale pour rendre l'application accessible au grand public.
- **Maintenance continue** : Après le déploiement, un suivi régulier des bugs et des retours des utilisateurs sera assuré pour identifier et résoudre les problèmes potentiels. Des mises à jour régulières seront également déployées pour ajouter de nouvelles fonctionnalités et améliorer continuellement l'expérience utilisateur, en réponse aux besoins et aux feedbacks reçus.

### 2.9.8 Difficultés Rencontrées et Solutions Apportées

Lors du développement de l'application, plusieurs défis techniques ont été identifiés et résolus pour assurer un fonctionnement optimal et une expérience utilisateur de qualité.

- **Problèmes avec la base de données**

- **Difficultés :**

- \* Les relations dans la base de données étaient mal conçues, permettant à un utilisateur de soumettre plusieurs votes pour le même défi en se déconnectant puis en se reconnectant. De plus, un utilisateur pouvait poster un défi plusieurs fois en appuyant rapidement sur le bouton "POSTER". Il était également possible de réaliser le même défi plusieurs fois de suite, et les signalements de photos inappropriées étaient limités à une seule par utilisateur.

- **Solutions :**

- \* Les relations dans la base de données ont été corrigées pour enregistrer les IDs des photos et vidéos déjà votées par un utilisateur, empêchant ainsi les votes multiples. De plus, l'utilisateur est désormais redirigé directement vers la page d'accueil après avoir posté un défi, ce qui évite les soumissions multiples. Les défis déjà réalisés par l'utilisateur sont désormais filtrés pour éviter les répétitions. Enfin, la structure de la base de données a été modifiée pour permettre à un utilisateur de signaler plusieurs photos inappropriées.

- **Problèmes avec le code JavaScript dans appwrite.js**

- **Difficultés :**

- \* Certaines fonctions étaient mal conçues, ce qui entraînait des problèmes de comptabilisation des votes utilisateurs, des requêtes de base de données qui ne renvoyaient pas les bons documents, et des erreurs lors du téléchargement des médias.

- **Solutions :**

- \* Le code des fonctions a été corrigé, notamment la fonction `updateInteger()`, et des méthodes comme `filter()`, `include()`, et `some()` ont été utilisées pour filtrer et obtenir les bons documents. La fonction `getFileInfo()` a également été implémentée pour récupérer correctement l'extension des médias uploadés.

- **Problèmes avec l'interface utilisateur**

- **Difficultés :**

- \* L'interface utilisateur était jugée visuellement peu attrayante, avec des couleurs de fond unies, une police d'écriture trop agressive, un manque de reliefs et de mouvements, ainsi qu'un manque de personnalisation des images et logos.



### – Solutions :

- \* L'interface a été améliorée en utilisant la bibliothèque `expo-linear-gradient` pour des couleurs de fond plus harmonieuses. La police a été changée pour "Dinos-Font", plus arrondie et douce. `TailwindCSS`, `StyleSheet`, et `react-native-reanimated` ont été utilisés pour ajouter des animations et des effets de relief aux composants. Enfin, `Gimp` a été utilisé pour ajuster les couleurs et formes des images et logos, les rendant plus adaptés à l'application.

Ces interventions ont permis de surmonter les obstacles techniques tout en améliorant l'efficacité et l'esthétique de l'application.

## 2.10 Conclusion

Ce cahier des charges détaille les différentes étapes nécessaires au développement de l'application Prokli, en insistant sur les aspects techniques et fonctionnels. La structure claire et la planification rigoureuse permettront de mener à bien ce projet ambitieux, en assurant une expérience utilisateur optimale et en répondant aux besoins de notre cible principale : les jeunes adultes de 18 à 25 ans.

# Conclusion

## 3.1 Vivien Seignoux

La Summer School m'a permis de confirmer mon appétence pour l'entrepreneuriat. La dimension de travail d'équipe m'a beaucoup plu mais j'ai réalisé que je ne souhaite pas travailler avec des amis, il vaut mieux avoir une relation purement professionnelle avec ses collègues. Techniquement, je me suis beaucoup amélioré en HTML, CSS et javascript lors de la création du site web. J'ai également mieux compris le fonctionnement des bases de données.

## 3.2 Anne Pottier

La Summer School m'a permis de confirmer mon appétence pour les projets, qu'ils soient informatiques ou plus entrepreneurial, j'ai beaucoup apprécié pouvoir apprendre de nouvelles compétences, que ce soit en JavaScript, guidée par Nicolas et des compétences plus globales sur la préparation d'un plan financier ou la gestion d'une équipe. Je recommencerais avec plaisir dès je retrouve un jour un projet qui me passionne.

## 3.3 Nicolas Menahem

Au cours de la Summer School, j'ai développé une application mobile en React Native, ce qui m'a permis d'approfondir mes connaissances en JavaScript et TypeScript. J'ai également appris à démarcher des prospects lors d'appels téléphoniques, renforçant ainsi mes compétences en communication. Cette expérience a complètement éveillé en moi un intérêt pour l'entrepreneuriat et la réalisation de projets innovants