

# Cahier des charges

Application mobile DASH



# 1. Contexte général du projet

- a. Contexte général et enjeu
- b. Objet de l'application
- c. Public cible
- d. Présentation de l'équipe
- e. Budget
- f. Organisation
- g. Planning

# 2. Les besoins fonctionnels

- a. Les thèmes
- b. Les équipes
- c. Le chronomètre
- d. Le “deck”
- e. Les boutons Valider / Passer ou Swipe
- f. Le compteur de points
- g. L'onglet Règles
- h. Le bouton switch multilinguisme
- i. Le compte joueur
- j. La modale “Notez l'application”
- k. Le contenu payant
- l. Les maquettes

### 3. Conception du site

- a. Eco-conception & Accessibilité
- b. Structure du site

### 4. Contraintes techniques

### 5. Préparation du projet

- a. Front-End
- b. Back-End
- c. Architecture logicielle
- d. Sauvegarde et gestion des versions

### 6. Et après ?

# 1. Contexte général du projet

## a. Contexte général et enjeux.

DASH est une application mobile de jeu d'ambiance et de devinettes. Inspirée du jeu Time's Up, elle ambitionne d'offrir un divertissement toujours renouvelé et un écosystème attractif pour les joueurs les plus assidus.

Les jeux de société ont connu une croissance continue sur les 30 dernières années, avec un taux de croissance à deux chiffres. Cette croissance a touché un large public, allant bien au-delà d'une activité réservée aux enfants.

En 2022, 87% des Français ont déclaré jouer régulièrement à des jeux de société, positionnant la France en tête du marché européen des jeux de société.

Plus de 33 millions de boîtes de jeux de société ont été vendues en France en 2022.

Parmi les différentes catégories de jeux de société, les petits jeux ou jeux "apéritifs" ont connu un succès considérable ces dernières années. Ces jeux sont sociaux, abordables et basés sur des concepts simples, ce qui les rend accessibles à tous, y compris aux joueurs occasionnels. Ils se jouent en quelques minutes et sont parfaits pour animer une soirée entre amis ou en famille.

Nous proposons donc un produit qui n'existe pas à l'heure actuelle. En effet, Asmodée l'entreprise-mère n'a pas développé d'application qualitative de son jeu phare Time's Up. Il y a bien une application pour enfants qui existe uniquement sur l'AppStore mais elle est de piètre qualité, offre peu de contenu et semble à l'abandon, comme l'illustre le commentaire suivant:



Darojoglhgs12, 26/02/2019

**Super mais pas autant que le jeu de cartes**

Toujours mêmes problèmes un an après, y'a t'il des développeurs encore vivants ??

## b. Objet de l'application mobile

Notre objectif est donc de développer une application mobile de jeu d'ambiance et de devinettes. Elle comportera plusieurs parties :

- les thèmes
- les options
- les paramètres
- le compte joueur
- le jeu en lui-même

Il y a une partie monétisation à implémenter. Nous souhaitons faire une application freemium, avec une partie des thèmes gratuite et les autres disponibles en packs payants (thèmes selon les périodes de l'année, les évènements, ex : Thème Jeux Olympiques, Thème Noël etc.)

## c. Public cible

Dash vise un public très divers, c'est un jeu familial mais son cœur de cible est les "millenials". Les 25-45 ans qui sont les plus férus de jeux de société ET d'applications mobiles. Avec son design cartoon et sa facilité d'utilisation, Dash reste adapté à tous les publics.

## d. Présentation de l'équipe

Fondateurs (3 personnes): Les fondateurs sont les créateurs de Dash . Ils ont initié le projet en raison de leur passion pour le jeu Time's Up et leur désir de revitaliser l'expérience de jeu en application mobile.

Directeur Général (CEO) : Le Directeur Général est chargé de la gestion globale de l'entreprise. Il/elle fixe la vision stratégique et les objectifs à long terme de Dash. Il/elle est le/la capitaine de l'entreprise et donne le cap aux autres membres.

Responsable du Développement de la Plateforme (Tech Lead) : Ce rôle est chargé de la conception, du développement et de la maintenance continue de l'application Dash, en veillant à ce qu'elle réponde aux besoins des utilisateurs. Il doit être compétent dans le développement tant front-end que back-end. Il est le chef technique et c'est lui qui a le dernier mot pour tout ce qui concerne le code. Il gère l'équipe de développement.

**Développeur Front-End** : Ce membre de l'équipe est spécialisé dans le développement de l'interface utilisateur (UI) de l'app. Il s'assure que l'application soit conviviale, réactive et esthétiquement agréable.

**Développeur Back-End** : Ce membre de l'équipe est chargé de la mise en place de la logique serveur, de la gestion de la base de données et de l'infrastructure technique du site. Il veille à ce que toutes les fonctionnalités du site fonctionnent correctement du côté serveur. Les développeurs qu'ils soient Front-End ou Back-End sont en étroite collaboration, pour s'assurer que l'interface utilisateur et la logique serveur fonctionnent de manière optimale.

**Responsable Marketing** : Le Responsable Marketing est responsable de la promotion de la plateforme, de l'acquisition d'utilisateurs et de la fidélisation de la communauté Dash. Sa vision, basée sur le besoin client est précieuse pour les développeurs qui manquent parfois de connaissances commerciales.

**Ingénieur en Sécurité Informatique** : Cet employé est responsable de la sécurité des données des utilisateurs et de la plateforme dans son ensemble, en mettant en place des mesures pour prévenir les atteintes à la sécurité. Son rôle est à la fois préventif et correctif. Les mesures qu'il préconise sont essentielles à la bonne marche de l'application.

**Responsable de l'Expérience Utilisateur (UX)** : Ce rôle se concentre sur l'amélioration de l'expérience globale des utilisateurs sur la plateforme, en s'assurant qu'elle soit conviviale et intuitive. Ses suggestions sont ensuite transmises à l'équipe de développement Front-End.

**Responsable du Service Client** : Le responsable du Service Client est chargé de répondre aux questions des utilisateurs, de résoudre les problèmes et de fournir une assistance technique en cas de besoin. Il fait remonter les bugs et suggestions à l'équipe de développement.

**Responsable des Ressources Humaines (RH)** : Le responsable des RH gère les ressources humaines de l'entreprise, y compris le recrutement, la formation et le développement professionnel des employés. Il est en contact avec tous les employés et responsables de l'entreprise.

**Responsable Financier** : Le responsable Financier supervise les finances de Dash, y compris la gestion du budget, la comptabilité et les prévisions financières. De sa

budgetisation dépend la bonne santé de l'entreprise et les possibilités d'embauche et d'évolution.

Responsable de la Gestion des Données : Ce rôle est responsable de la gestion et de l'analyse des données utilisateur pour obtenir des informations précieuses sur le comportement des utilisateurs. Ses rapports et statistiques sont précieux pour l'équipe de développement Back-End qui peut ensuite améliorer et faire évoluer la logique serveur et l'utilisation des données.

## e. Budget

### **Frais de développement technique :**

Salaires et honoraires : trois développeurs juniors au taux horaire de 30€, pour 6 mois de travail, à raison de 25 jours par mois, 7h par jour : 95,040€.

### **Frais opérationnels :**

Frais de bureau : 12,000€/an

Infrastructure technique et hébergement : 800€/an (appelé à augmenter lorsque le trafic utilisateur augmentera également)

**Total première année : 107,640€**

**Marge d'erreur : 15% => 124,000€**

## f. Organisation

L'équipe utilisera la méthode agile pour mieux planifier la conception du site.

Le travail de développement sera découpé en sprints d'une à 4 semaines. Un **daily** meeting aura lieu chaque jour, pour s'assurer de la bonne direction prise par les développeurs. Nous sommes une équipe de trois et prendrons à tour de rôle les postes de Product Owner pour ne pas perdre de vue les besoins clients qui priment sur le reste, et de Scrum Master pour gérer l'organisation quotidienne de l'équipe et les problématiques qui en découlent.

De plus, un serveur Discord sera créé dans le but d'améliorer la communication interne et externe.

Il comprendra des canaux privés pour l'entreprise, des canaux restreints en cas de besoin de consultation externe, et éventuellement un canal public pour permettre au public de contribuer au développement et partager ses idées.

Un tableau Trello pour structurer les tâches individuelles, assurer une organisation transparente et avoir une vue d'ensemble sur les réalisations ainsi que les tâches restantes.

Le versioning du code se fera sur Github. Il existera trois types de branches :

- une branche **main** sur laquelle se feront les “merge” validés par l'entièreté de l'équipe de développement.
- une branche **develop** sur laquelle est envoyée le code qu'un développeur juge avoir terminé afin qu'il soit examiné et validé par ses pairs.
- des branches **feature** (exemple : feature/podium) sur lesquelles seront développées les nouvelles fonctionnalités, ce qui permettra de ne pas impacter la **develop** et de garder la **main** fonctionnelle en permanence.

## g. Planning



# GANTT CHART RETRO PLANNING DASH



MARIANNE



JASON



FAYSSAL

## TASKS

### PLANIFICATION

- Identifier les fonctionnalités clés
- Définir les objectifs et les responsabilités
- Schémas UML : classes, séquences, MCD

### CONCEPTION

- Maquettes et wireframes
- Définir stack technique et hébergement
- Définir architecture globale de l'app

### DEVELOPMENT

- Développement des fonctionnalités front et back
- Tests unitaires et d'intégration

# GANTT CHART RETRO PLANNING DASH



MARIANNE



JASON



FAYSSAL

## TASKS

### COMPANY

- Gestion aspects légaux (CGU, propriété intellectuelle)
- Création d'entreprise
- Mise à jour du budget

### LAUNCHING

- Ecrire la documentation
- Lancement de l'app sur les stores
- Communication et promotion

### DEVELOPMENT

- Développement des fonctionnalités front et back
- Phase de tests
- Corrections bugs / ajout de nouvelles fonctionnalités
- Tests unitaires et d'intégration
- Analyse des retours utilisateurs et mises à jour

### Phase de Planification (Mois 1) :

- Définir les objectifs du projet.
- Identifier les fonctionnalités clés de l'application.
- Élaborer une ébauche du cahier des charges.
- Répartir les rôles et les responsabilités au sein de l'équipe.

### Phase de Conception (Mois 2-3) :

- Créer des wireframes et des maquettes de l'interface utilisateur.
- Définir l'architecture globale de l'application.
- Choisir les technologies à utiliser.

### Phase de Développement (Mois 4-10) :

- Développer les fonctionnalités de base de l'application.
- Effectuer des tests unitaires et d'intégration tout au long du processus.

### Phase de Tests (Mois 11) :

- Effectuer des tests de validation et de performance.
- Corriger les bugs et les problèmes identifiés.

### Phase de Gestion administrative et juridique (Mois 11-12) :

- CGU, propriété intellectuelle, dépôt de la marque.
- Création d'entreprise.
- Réajustement du budget si besoin.

### Phase de Lancement (Mois 12) :

- Préparer la documentation de l'application.
- Soumettre l'application aux boutiques d'applications (App Store, Google Play, etc.).
- Communiquer sur le lancement de l'application.

### Phase de Maintenance et d'Améliorations (après le lancement) :

- Suivre les retours des utilisateurs et les évaluations de l'application.
- Corriger les bugs et apporter des mises à jour régulières.
- Ajouter de nouvelles fonctionnalités en fonction des besoins et des retours.

## 2. Les besoins fonctionnels

Sans rentrer dans le technique, listons les besoins fonctionnels de l'application :

### a. Les thèmes

Notre base de données doit être conséquente afin d'assurer une vraie rejouabilité à l'application.

Le choix des thèmes se fera avant le début du jeu. Les joueurs pourront sélectionner de un à 20 thèmes, chacun contenant 80 cartes (éléments à deviner).

### b. Les équipes

Dans les options, les joueurs définissent obligatoirement le nombre d'équipes, le nom des équipes et le nombre de joueurs par équipe. Il peut y avoir de 2 à 6 équipes et de 2 à 6 joueurs par équipe.

### c. Le chronomètre

Un chronomètre est implémenté. Trois temps sont disponibles : 30, 45, 60 secondes. Un son sera diffusé à la fin du chronomètre pour éviter toute triche.

### d. Le "deck"

Le deck ou jeu de cartes est composé de 42 cartes tirées aléatoirement dans les thèmes sélectionnés par les joueurs. Ce deck sera mélangé et réutilisé dans les manches 2 et 3.

### e. Les boutons Valider / Passer ou Swiper

D'un point de vue ergonomique, il est important de se poser la question de la méthode de validation ou d'omission d'une carte. Nous offrons la possibilité de paramétrer cette fonctionnalité : soit un gros bouton vert et un gros bouton rouge, soit un système de Swipe, on glisse vers la droite pour valider, on glisse vers la gauche pour passer.



#### f. Le compteur de points

Le compteur de points se remplit à chaque fois qu'une carte est validée. Il y a un score par manche et un affichage des scores intermédiaires en fin de manche 1 et 2. Enfin, une page podium qui récapitule tous les points gagnés par équipe.

#### g. Les règles

Essentiel que les règles du jeu soient disponibles sur toutes les pages de l'application, c'est pourquoi elles sont accessibles à partir du header.

#### h. Le bouton switch multilinguisme

L'application sera disponible en français et en anglais. Un bouton switch situé dans les paramètres permettra de configurer la langue.

#### i. Le compte joueur

Le joueur pourra créer un compte DASH sur l'application. Ce qui lui donnera accès à du contenu premium (possibilité de créer un thème personnalisé en local, d'avoir des statistiques comme un classement par taux de victoire, d'ajouter ses amis).

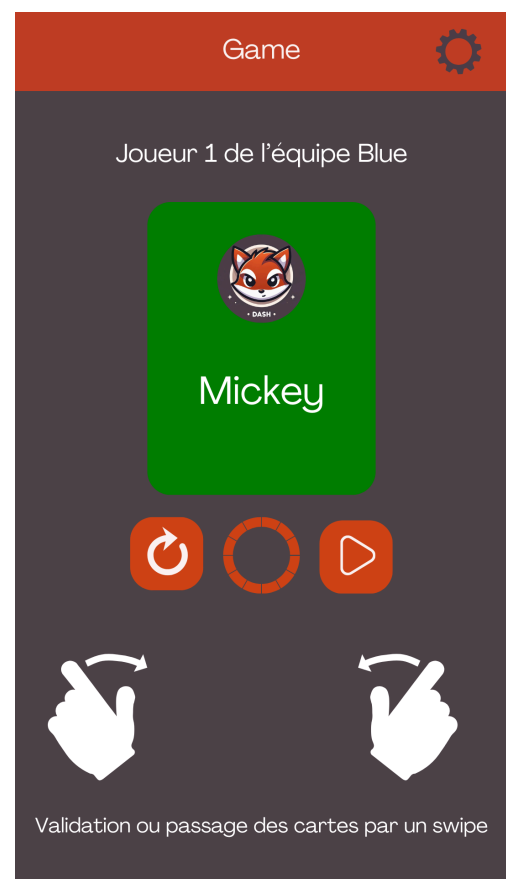
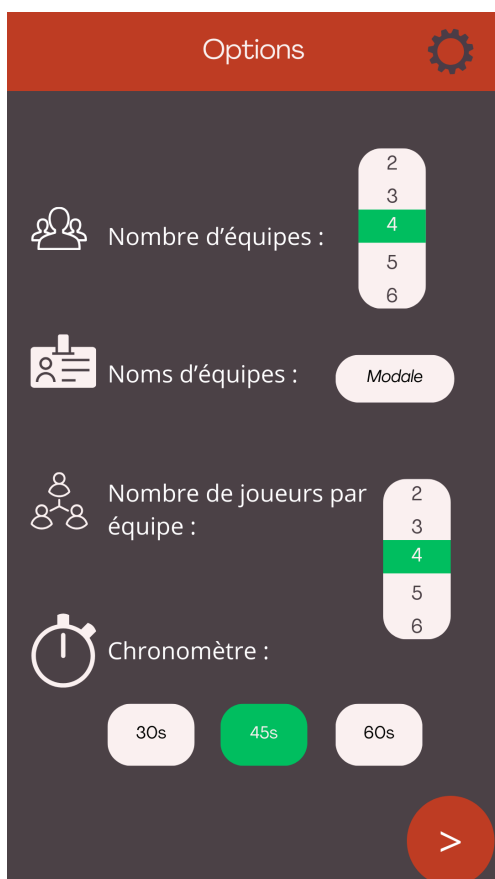
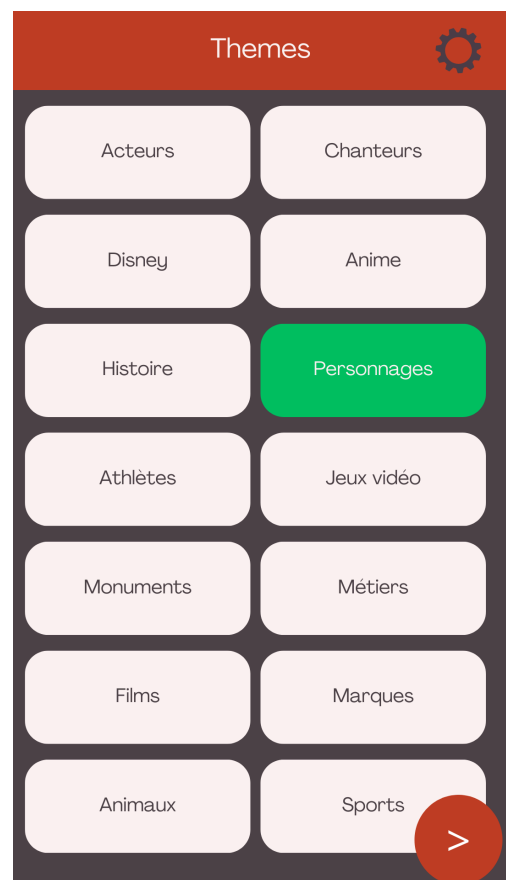
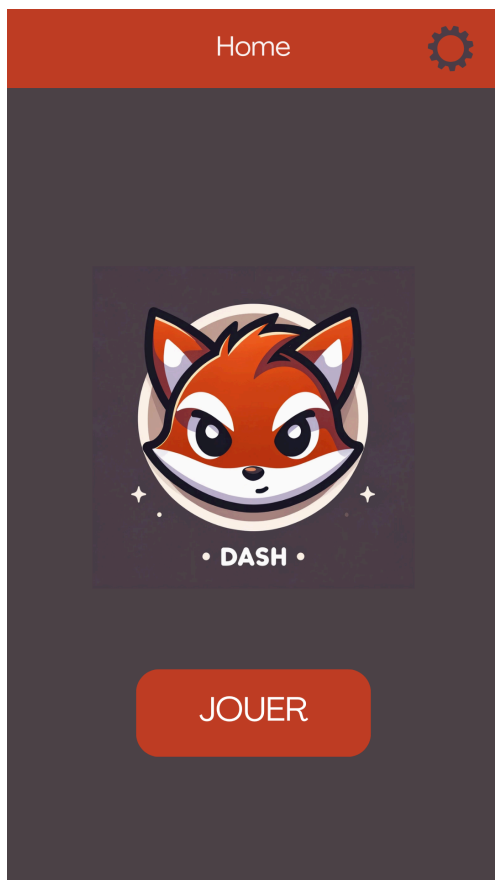
#### j. La modale "Notez l'application"

En fin de partie après le podium. Une modale invitant à noter l'application sur les stores.

#### k. Le contenu payant

Avatars et packs de thèmes événements.

## I. Les maquettes



## 3. Conception de l'application

### a. Eco-Conception et accessibilité

Conception et développement équilibrés :

Nous adoptons une approche équilibrée dans la conception de notre application mobile. Bien que nous utilisions des scripts pour améliorer l'expérience utilisateur, nous veillons à ne pas surcharger la plateforme avec des éléments inutiles qui pourraient augmenter la consommation d'énergie.

Optimisation des médias et des scripts :

Nous nous engageons à optimiser à la fois les médias, tels que les images, et les scripts utilisés sur notre plateforme. Cela garantit que les fichiers multimédias sont légers et que les scripts sont efficaces, réduisant ainsi la demande d'énergie pour leur exécution.

Responsive

Notre approche du responsive design est également centrée sur l'efficacité énergétique et l'accessibilité. En optimisant notre application pour qu'elle s'adapte harmonieusement à différentes tailles d'écran (mobile et tablette), nous réduisons la nécessité de charger des ressources supplémentaires ou inadaptées. Cela se traduit par une consommation d'énergie moindre, tant du côté des appareils des utilisateurs que de nos serveurs. De plus, un design responsive améliore l'accessibilité de notre application, en garantissant que les utilisateurs bénéficient d'une expérience fluide et intuitive, quelle que soit la taille de leur écran.

Hébergement responsable :

Lors du choix de notre hébergeur web, nous privilégions ceux qui utilisent des sources d'énergie renouvelable ou qui mettent en place des pratiques de réduction de leur empreinte carbone. Cela contribue à minimiser l'impact environnemental de notre infrastructure, même lorsque nous utilisons des scripts.

#### Mise en cache réfléchie :

Nous mettons en place des mécanismes de mise en cache intelligents pour réduire la quantité de données transférées entre notre serveur et les utilisateurs. Cela permet d'optimiser les performances tout en minimisant la consommation d'énergie.

#### Accessibilité universelle maintenue :

Nous maintenons notre engagement envers l'accessibilité universelle en veillant à ce que l'utilisation de scripts ne soit pas discriminatoire envers les utilisateurs ayant des besoins spécifiques en matière d'accessibilité. Nous nous assurons que les balises sémantiques sont utilisées correctement et que les attributs liés à l'accessibilité spécifique sont mis en place. DASH s'engage à garantir une accessibilité universelle. L'application sera conforme aux directives WCAG<sup>1</sup> (Web Content Accessibility Guidelines) pour assurer une expérience optimale aux utilisateurs en situation de handicap, notamment en proposant des alternatives textuelles pour les médias, une structure logique et une bonne lisibilité.

#### Surveillance de l'impact environnemental :

Nous continuons à évaluer régulièrement l'empreinte carbone de notre plateforme, y compris celle liée à l'utilisation de scripts. Ces évaluations nous aident à prendre des mesures pour optimiser notre impact environnemental global. DASH s'efforce de concevoir et de développer son application de manière responsable, en tenant compte de l'utilisation de scripts pour améliorer l'expérience des utilisateurs tout en minimisant l'impact environnemental.

### b. Structure de l'application

Les pages principales de l'application sont au nombre de 10 (voir annexes). Certaines ne sont accessibles qu'après avoir rempli la page précédente ou après s'être connecté.

#### Accueil

La page d'accueil, avec le logo, le nom de l'application, un bouton jouer, la roue dentée pour les paramètres et l'icône de joueur pour se connecter à son compte.

---

<sup>1</sup> <https://www.w3.org/WAI/standards-guidelines/wcag/fr>

## Thèmes

Les thèmes gratuits sont au nombre de 20. Ils sont affichés sous forme d'icône cliquables.  
Les thèmes payants apparaissent en bandeau, en dessous du header.

## Options

Page où l'on configure le nombre d'équipes, leur nom, le nombre de joueurs par équipe et le chronomètre. Le joueur qui possède un compte peut indiquer à quelle équipe il appartient.

## Jeu

Page qui affiche quel joueur de quelle équipe est le narrateur. Affiche la carte à faire deviner, les boutons ou le swipe, le chronomètre, un bouton lecture ou pause pour celui-ci.

## Score intermédiaire

Page qui affiche en fin de manche 1 et 2 les scores des équipes.

## Podium

Page qui affiche tous les scores des équipes et annonce l'équipe gagnante.

## Règles

Page qui répertorie les règles du jeu, toujours accessible via le header.

## Paramètres

Page qui permet de configurer la langue, le choix entre boutons ou swipe, et qui contiendra une page "à propos" avec les mentions légales, les conditions générales de vente et la politique de confidentialité..

## Compte joueur

Page qui permet de voir et de changer son avatar, de consulter ses statistiques, d'ajouter un ami, de consulter sa liste d'amis, de créer un thème personnalisé de 80 éléments.

## Inscription / Connexion compte joueur

Formulaire pour s'inscrire ou se connecter à son compte.

In fine, l'application doit être intuitive et conviviale.



## 4. Contraintes techniques

### Sécurité des données

Il nous faut assurer la sécurité des données des utilisateurs, y compris leurs informations personnelles. Il est essentiel de mettre en place des protocoles de sécurité robustes et de prévenir les attaques potentielles. La sécurité des données et des utilisateurs doit être une priorité.

#### Côté serveur Node.JS et API :

- Authentification JWT : Utilisation de Jetons Web JSON pour sécuriser les échanges entre le client et le serveur. Cela afin de vérifier l'identité de l'utilisateur à chaque requête.
- Validation des entrées : Mise en place d'une validation stricte des données côté serveur pour prévenir les injections SQL et les attaques XSS (bibliothèque express-validator).
- Rate Limiting : Limiter le nombre de requêtes qu'un utilisateur peut faire sur une période donnée pour prévenir les attaques DDoS et par force brute.

#### Côté client React-Native :

- Stockage sécurisé : Utilisation de "react-native-secure-storage" pour stocker les données sensibles comme les tokens JWT.
- Communication sécurisée : Toutes les communications entre l'application et le serveur utilisent HTTPS.
- Mises à jour régulières : Maintenir l'application à jour avec les dernières versions de React Native pour bénéficier des correctifs de sécurité.

#### Gestion des mots de passe :

- Hashage et salage : Stocker les mots de passe en utilisant un hashage robuste (comme bcrypt) avec salage pour prévenir les attaques par dictionnaire et rainbow table.
- Politique de mot de passe forte : Exiger des mots de passe complexes (longueur minimale, utilisation de caractères spéciaux, etc.).

### Protection contre les attaques spécifiques :

- Man-in-the-Middle (MitM) : Utiliser le certificat SSL pour prévenir l'interception des données.
- Injection SQL : Utiliser des requêtes préparées et l'ORM pour interagir avec la base de données.
- Overlay attack : Gérer correctement les permissions pour prévenir l'utilisateur si une autre application tente de se superposer à la nôtre.

### Sensibilisation et transparence envers les utilisateurs :

- Politique de confidentialité claire : Expliquer comment les données sont utilisées, stockées et protégées. Rendre cette politique accessible et facile à comprendre.
- Formation des utilisateurs : Informer les utilisateurs sur les bonnes pratiques de sécurité, comme choisir un mot de passe fort ou reconnaître les emails de phishing.

### Tests et audits de sécurité :

- Tests d'intrusion : Faire régulièrement des tests d'intrusion pour identifier et corriger les vulnérabilités.
- Audit de code : Faire auditer notre code par des experts en sécurité pour détecter les failles potentielles.

## Performance et scalabilité

S'assurer que l'application puisse gérer un grand nombre d'utilisateurs simultanés (donc autant d'appels à notre API pour les thèmes et les parties). Maintenir un code simple et facilement scalable pour les mises à jour qui se doivent d'être régulières sur un produit de ce type.

## Compatibilité des systèmes d'exploitation

Dans un premier temps, à cause des contraintes de temps et techniques, nous développerons cette application uniquement pour le PlayStore Android (en France, 75% des téléphones sont des Android). Néanmoins, il faut préparer le terrain pour la compatibilité iOS également.

## Base de données

Concevoir une base de données robuste pour stocker les thèmes, les informations sur les parties, les comptes joueurs et leurs statistiques. La gestion efficace des bases de données est essentielle pour garantir des temps de réponse rapides.

## Mise à jour et maintenance

Prévoir des procédures de mise à jour régulières pour maintenir l'app, corriger les bugs et ajouter de nouvelles fonctionnalités. L'application doit être facilement maintenable.

## Hébergement

Sélectionner un service d'hébergement web fiable et évolutif pour garantir que l'application reste en ligne et performante même en cas de montée en charge.

## Sauvegarde des données

Mettre en place des procédures de sauvegarde régulières pour éviter la perte de données en cas de problème technique.

## Législation sur la protection de la vie privée

Respecter les lois et réglementations locales et internationales concernant la protection de la vie privée des utilisateurs, telles que le RGPD en Europe.

## Coûts

Évaluer les coûts associés à la création, à l'hébergement et à la maintenance de l'app, en tenant compte du budget disponible.

## Évolutivité

Penser à l'avenir et à la possibilité d'ajouter de nouvelles fonctionnalités à mesure que l'application se développe.

## Expérience d'utilisation

Chercher à offrir une expérience utilisateur fluide et intuitive. Cela inclut une interface conviviale, une navigation facile à comprendre, des temps de chargement rapides, des images optimisées et une ergonomie de jeu adéquate.

## SEO

Notre application étant uniquement mobile, nous n'avons pas à gérer le SEO comme nous le ferions avec une application web en nous intéressant au référencement naturel à travers les balises sémantiques HTML5, la richesse du contenu ou les métadonnées.

Néanmoins, il existe un équivalent au SEO pour le mobile c'est l'ASO<sup>2</sup> (App Store Optimization ou optimisation du référencement dans les magasins applicatifs). Comme le SEO, il répond à des normes strictes :

- **Mots-clés** : les utilisateurs à la recherche d'une application comme la nôtre taperont des mots-clés qu'il nous faut avoir ciblés au préalable. Les mots comme "jeu", "game", "ambiance", "devinettes", "guess", "guessing", "Time's Up", "card" doivent renvoyer à notre app. Comme les mots-clés principaux sont déjà saturés par les applications les plus connues, il faudra prêter attention aux mots-clés secondaires comme "team", "competition", "quizz", "common knowledge" pour attirer plus d'utilisateurs.
- **Choisir le bon nom** : Nous avons choisi DASH comme nom pour notre app, car il est court, évocateur (notion de rapidité, d'agilité), et avec son logo, est facilement mémorisable.
- **Description riche et informative** : La description est l'un des éléments clés que l'algorithme de Google Play Store prend en compte pour classer les applications dans les résultats de recherche. Il est donc important d'y intégrer les mots ou expressions clés sur lesquels nous souhaitons nous positionner. En principe, les mots-clés stratégiques doivent être présents entre 3 et 5 fois dans notre description.
- **Les notes des utilisateurs** : Les notes sont un levier que nous devons exploiter pour améliorer notre référencement. En effet, Google Play Store aime présenter aux internautes des applications qui ont beaucoup de feed-back positifs. Si nous avons de nombreux retours positifs, de plus en plus d'internautes nous feront confiance et téléchargeront notre application. Il est aussi important de répondre aux commentaires des utilisateurs.

---

<sup>2</sup> Article en anglais daté du 12/04/23 sur l'ASO et ses critères :  
<https://www.marinsoftware.com/blog/an-in-depth-guide-to-google-play-aso>

## 5. Préparation du Projet

### a. Front-end

Pour la partie application client, nous avons balayé les différentes options qui s'offrent à nous : Flutter, Ionic, Cordova, React-Native. Notre choix s'est porté sur ce dernier framework pour les raisons suivantes.

#### React Native<sup>3</sup>

React Native est un framework open-source développé par Meta pour la création d'applications mobiles. Il permet aux développeurs de construire des applications pour les principales plateformes mobiles, iOS et Android, en utilisant JavaScript et React. La particularité de React Native est qu'il offre une approche basée sur les composants de React (que l'équipe de développement connaît bien), mais génère des éléments d'interface utilisateur natifs, permettant ainsi aux applications de bénéficier de performances et d'une apparence similaires à celles des applications natives traditionnelles.



React Native s'appuie sur la programmation déclarative et un modèle de composants réutilisables. Cela favorise la rapidité de développement et la facilité de maintenance. Un des avantages majeurs de React Native est le partage de code entre les plateformes, ce qui réduit significativement le temps et les coûts de développement.

Le framework propose également une fonctionnalité appelée "Hot Reloading", qui permet aux développeurs de voir instantanément le résultat des dernières modifications apportées au code sans avoir à redémarrer l'application entière. Cette caractéristique améliore considérablement l'expérience de développement et accélère le processus de test et de débogage.

React Native bénéficie d'une large communauté et d'un écosystème riche, offrant une multitude de bibliothèques et de plugins tiers pour étendre ses fonctionnalités de base. Il est particulièrement apprécié dans les projets visant à réduire les délais et les coûts de développement tout en fournissant une expérience utilisateur de haute qualité sur les deux principales plateformes mobiles.

---

<sup>3</sup> Article en anglais de comparaison entre React Native et Flutter daté du 14/09/23 : <https://www.thedroidsonroids.com/blog/flutter-vs-react-native-comparison>

## JSX

Le JSX (JavaScript XML) dans React Native est un élément syntaxique central qui permet aux développeurs d'écrire la structure de l'interface utilisateur de leurs applications de manière déclarative et intuitive. Il remplace le HTML.

Dans le JSX de React Native, les composants sont utilisés comme des balises XML ou HTML. Ces composants peuvent être des éléments natifs de React Native, tels que `<View>` ou `<Text>`, ou des composants personnalisés créés par les développeurs.

## CSS

En React Native, la mise en forme des composants s'effectue à travers un système de styles qui s'inspire de CSS, mais qui est adaptée aux besoins spécifiques du développement d'applications mobiles. Ce système de styles est principalement géré par l'objet `StyleSheet`, une abstraction fournie par React Native pour déclarer les styles de manière optimisée et performante.

Les styles sont déclarés comme des objets JavaScript, où les propriétés de style suivent la convention `camelCase`, par opposition au `kebab-case` de CSS. Par exemple, `backgroundColor` au lieu de `background-color`.

React Native permet de combiner plusieurs styles en utilisant un tableau. Cela permet de composer des styles de manière dynamique et réactive, en fonction des états ou des props des composants.

## b. Back-end

PHP ou Node.js ? Voici les choix qui s'offrent à nous pour la conception de notre serveur et de son API. Nous avons opté pour Node.js pour les raisons suivantes.

### Node.js<sup>4</sup>

Node.js est un environnement d'exécution open source pour le JavaScript qui permet l'exécution de code JavaScript côté serveur. Il est construit sur le moteur JavaScript V8 de Google Chrome et permet de développer des applications réseau rapidement et efficacement. Node.js fonctionne sur un modèle d'entrées/sorties non bloquant et



---

<sup>4</sup> Article en anglais de comparaison entre Node.js et PHP daté du 13/09/23 : <https://www.netguru.com/blog/node-js-vs-php>

orienté événement, ce qui le rend particulièrement adapté pour des applications nécessitant des performances élevées en matière de traitement des données en temps réel. Node.js est également reconnu pour son écosystème riche en modules, géré via le Node Package Manager (NPM), qui offre une vaste gamme de bibliothèques et d'outils facilitant le développement d'applications.

## Supabase

Type de Service : Supabase est un Backend-as-a-Service (BaaS), fournissant une infrastructure de backend et une suite d'outils pour le développement rapide d'applications.

Base de données : Au cœur de Supabase se trouve PostgreSQL, un système de gestion de base de données relationnelle puissant et open source. Supabase étend PostgreSQL avec une API RESTful et des fonctionnalités de temps réel pour un accès facile et dynamique aux données.

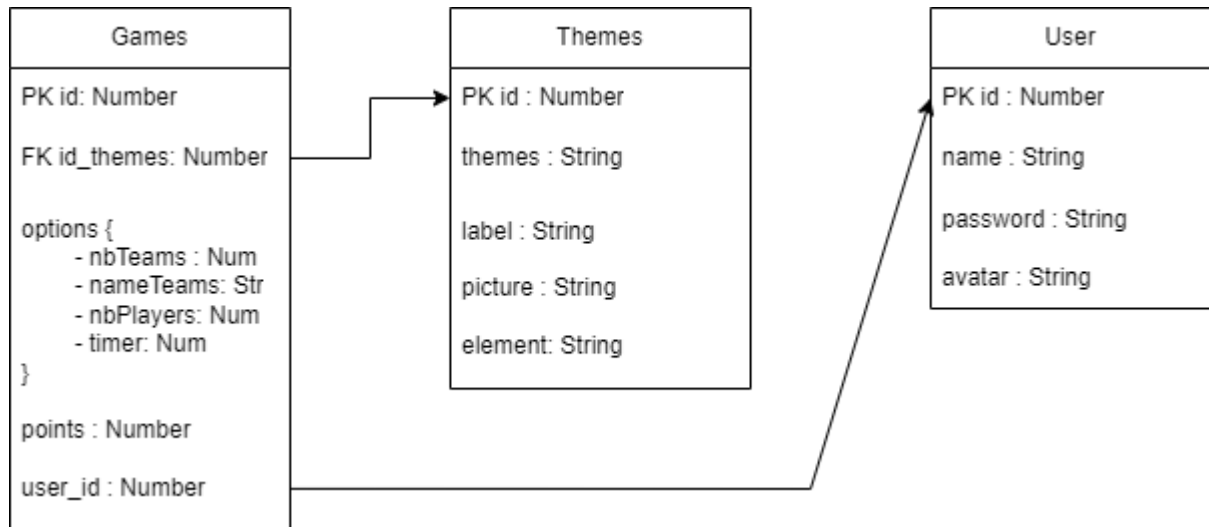
Authentification et autorisation : Supabase offre un système robuste pour gérer l'authentification des utilisateurs et les règles d'autorisation, permettant un contrôle sécurisé et flexible de l'accès aux données.

Stockage : Il intègre une solution de stockage pour gérer les fichiers et les données binaires, avec une intégration facile à la base de données principale.

API en temps réel : Une caractéristique notable de Supabase est sa capacité à offrir des fonctionnalités en temps réel. Les modifications apportées à la base de données peuvent être automatiquement envoyées aux clients abonnés, permettant une mise à jour dynamique et instantanée des interfaces utilisateur.

### c. Architecture logicielle

## Modèle Conceptuel de Données DASH



## Cas d'utilisation de l'application DASH

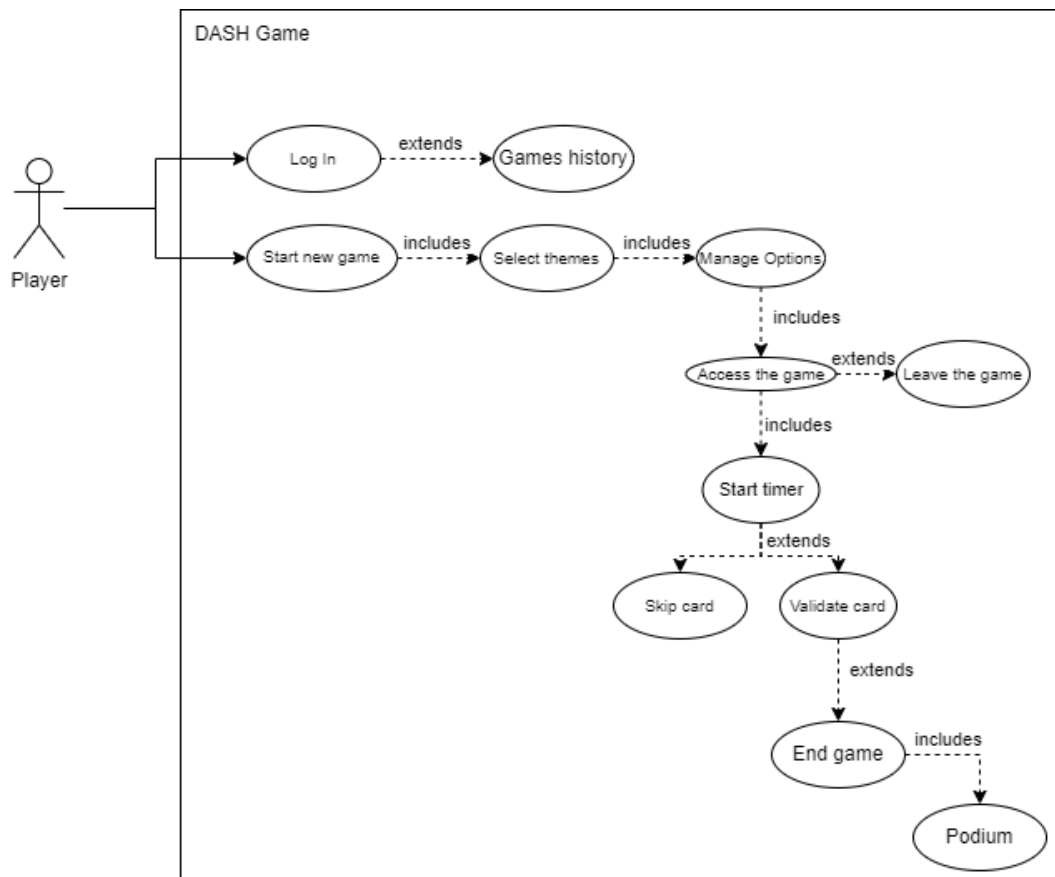
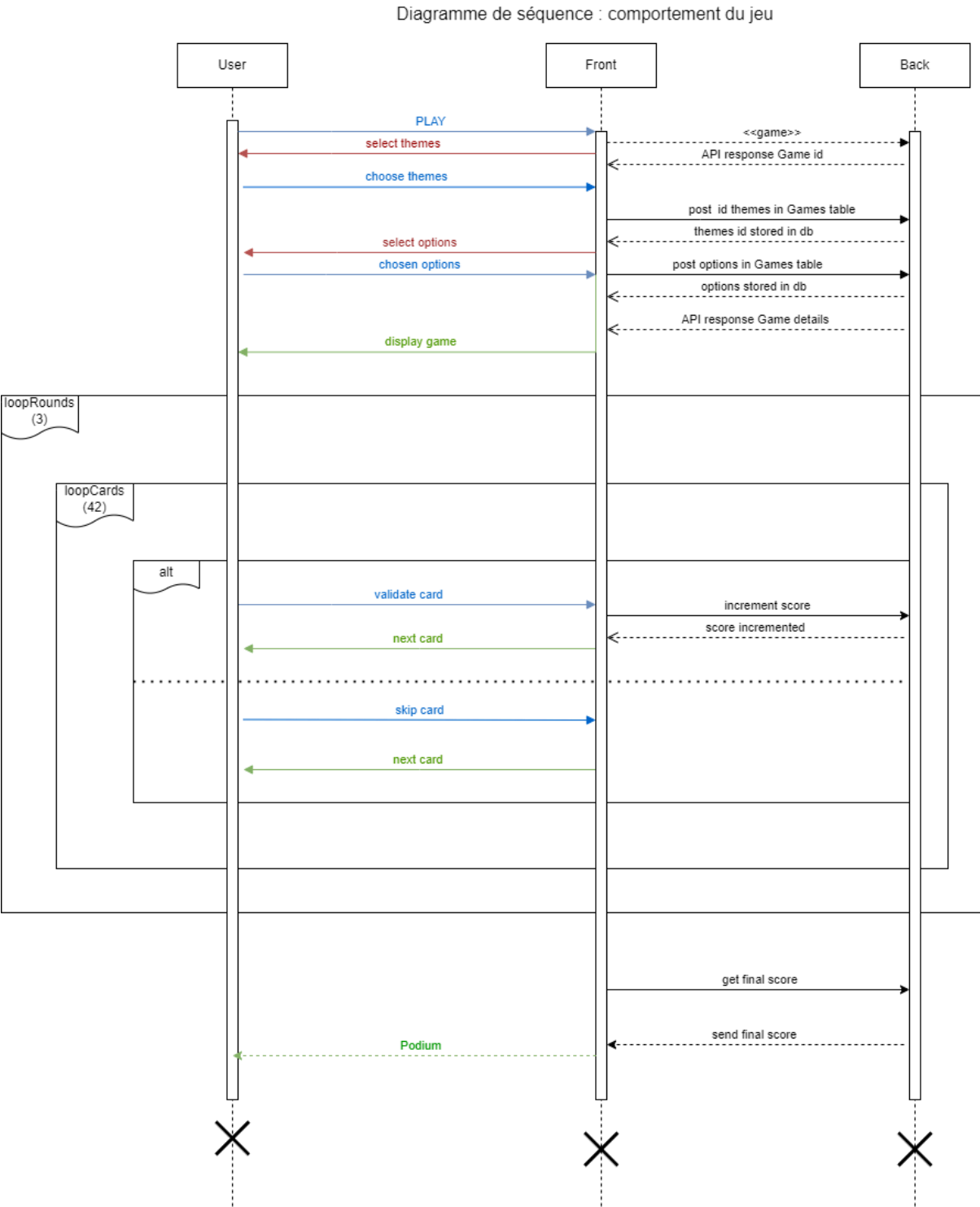




Diagramme de séquence : comportement du jeu DASH



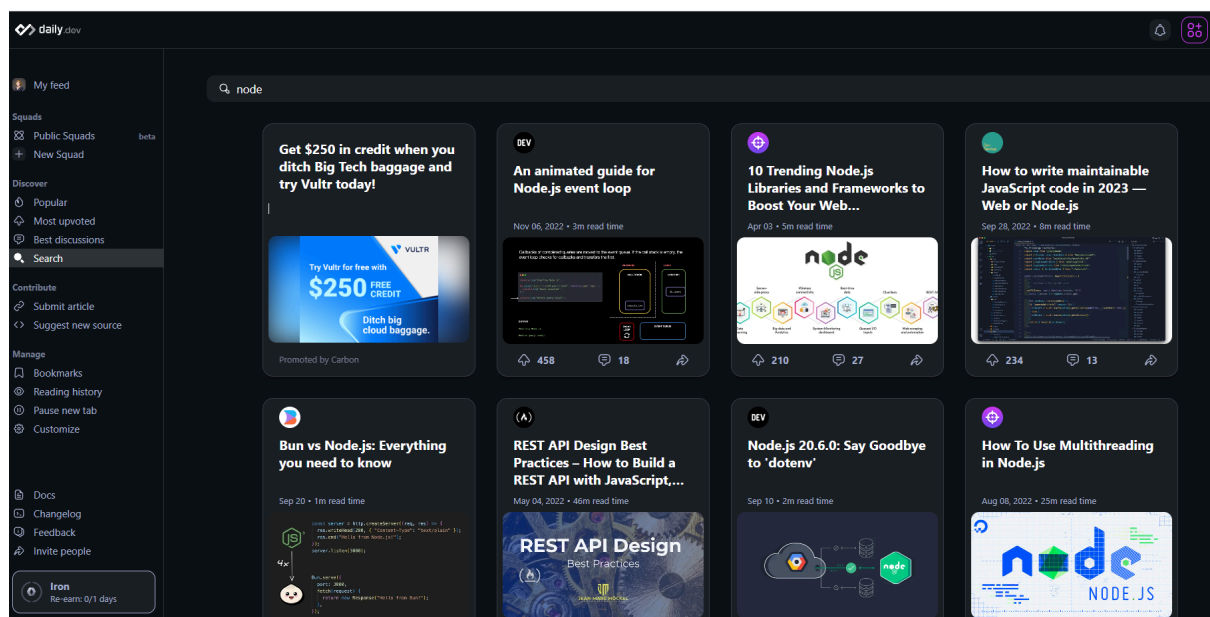
## d. Veille

Le projet s'étalera sur un an, or on le sait, les langages, frameworks, librairies et autres sont en constante évolution. Il est essentiel de faire de la veille quotidiennement pour se tenir informé des changements qui peuvent impacter notre application.

Nous avons la chance de faire partie d'une communauté très portée sur le partage d'informations et les moyens de faire de la veille efficace sont nombreux.

Nous avons sélectionné deux canaux différents :

- [daily.dev](https://daily.dev/)<sup>5</sup> est un agrégateur d'articles sous forme d'extension de navigateur, qui permet, par un système extrêmement bien fait de filtres de proposer un contenu pertinent.



- X (anciennement Twitter), BlueSky et LinkedIn comptent de nombreux développeurs passionnés qui partagent leurs astuces et leurs découvertes. Par exemple, le développeur senior Stéphane Montlouis qui est spécialisé en React-Native et nous a beaucoup aidés par ses tutoriels et ses explications sur l'architecture d'une app React-Native. Il existe une communauté de développeurs sur chaque réseau social et ils sont souvent réactifs et bienveillants pour répondre aux questions.

---

<sup>5</sup> <https://daily.dev/>

## 6. Et après ?

Rejouabilité à l'infini. C'est l'ambition de DASH. L'objectif est de faire de DASH l'application numéro 1 en termes de jeu d'ambiance sur le PlayStore.

Fonctionnalités futures :

- Que chaque utilisateur enregistré puisse visualiser la partie en cours sur son propre téléphone, au lieu de se passer un téléphone comme actuellement.
- Partager son thème personnalisé avec ses amis.
- S'inscrire avec les réseaux sociaux et partager ses scores en ligne

Perspectives futures :

Si l'application rencontre le succès escompté, il serait judicieux pour les fondateurs de créer leur entreprise, SARL ou SAS afin de définir les responsabilités et rôles de chacun de manière formelle et d'embaucher des salariés (première embauche : un(e) UI/UX designer !)