



Programmation et nouvelles technologies

Pré-dossier projet smart city

Higginson Julien, Leseure Enzo, Schoofs Léo

1. Présentation du projet.....	3
2. Fonctionnalités.....	4
Fonctionnalités de base.....	4
Visiteur.....	4
Utilisateur.....	4
Fonctionnalités Bonus.....	4
Utilisateur.....	4
Use Case.....	5
Assignation individuelle	5
3. Design.....	6
Maquettes.....	6
Page Création compte.....	6
Page Accueil.....	6
Page Evenement.....	7
Page Carte.....	7
Page Profil.....	8
Lien figma.....	8
4. Structure.....	9
Fonctionnalités minimales requise.....	9
Description packages utilisés.....	9

1. Présentation du projet

Le projet consiste à développer une plateforme de gestion d'évènements en présentiel destinée aux joueurs de jeux vidéo.

L'objectif est de faire se rencontrer les joueurs et de les connecter entre eux autour de leurs jeux favoris et de faciliter l'organisation d'événements physiques comme des tournois, présentations, rendez-vous dans un bar, jeux de rôles, conventions ou juste un rassemblement de fans.

Ce projet s'inscrit dans le concept de Smart City, en utilisant la technologie pour connecter les habitants d'une même ville et/ou région et encourager les échanges réels autour d'une passion commune : les jeux vidéo.

L'application mobile permettra aux utilisateurs de gérer leur compte, participer à des événements, consulter les informations sur les jeux et interagir avec la communauté.

2. Fonctionnalités

Fonctionnalités de base

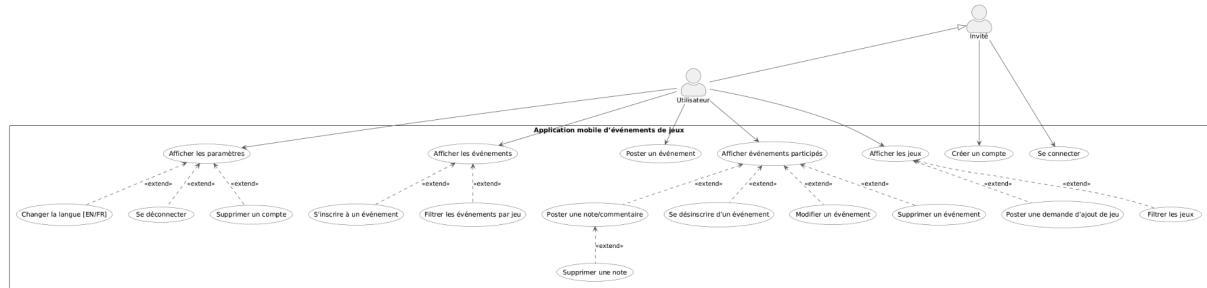
Visiteur

- Se connecter : Accéder à un compte existant
- Créer un compte : Inscription via email

Utilisateur

- Supprimer son compte : Effacement complet des données personnelles
- Se déconnecter : Terminer la session
- Changer la langue : Passer de EN à FR ou inversement
- Modifier le profil (photo de profil, bio, pseudo, mot de passe)
- Poster une note/commentaire : Pour les événements passés auxquels il a participé (avec possibilité d'ajouter des photos provenant de la galerie)
- Supprimer une note : Suppression d'une note qu'il a créé
- Poster un event : Ajouter un événement avec photos (provenant de la galerie)
- Supprimer un event : Supprimer un événement qu'il a créé
- Afficher les événements disponibles : Liste avec filtres par jeu et date
- Afficher les événements disponibles à proximité : Carte utilisant la position géographique
- Historique des événements : Voir ceux déjà participés
- Filtre sur l'historique des événements (sur le jeu et sur la date)
- Participer / se désinscrire d'un événement : Gestion de la participation
- Demande d'ajout de jeu : Proposer un nouveau jeu pour approbation
- Afficher les jeux disponibles : Liste avec filtres par tags et plateformes
- Afficher les infos d'un jeu : nom, studio, date de sortie, plateformes, tags, description, images.

Use Case



Attribution individuelle :

Étudiant	Fonctionnalité assignée
Enzo	création/suppression de notes
Julien	création/suppression de jeux
Léo	Création de comptes + suppression

3. Design

Maquettes

Page Création compte

Création de compte

Nom d'utilisateur
Entrez un nom d'utilisateur

Date de Naissance
Entrez une date

Email
Entrez un email valide

Mot de passe
Entrez un mot de passe

Confirmer mot de passe
Confirmez le mot de passe

Créer le compte

Page Accueil

Bonjour Joey !

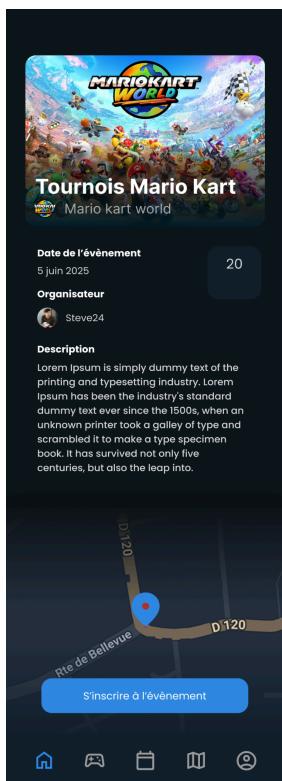
Évènements à venir

Tournoi Mario-kart
5 juin 2025 à 2 km
Mario kart world

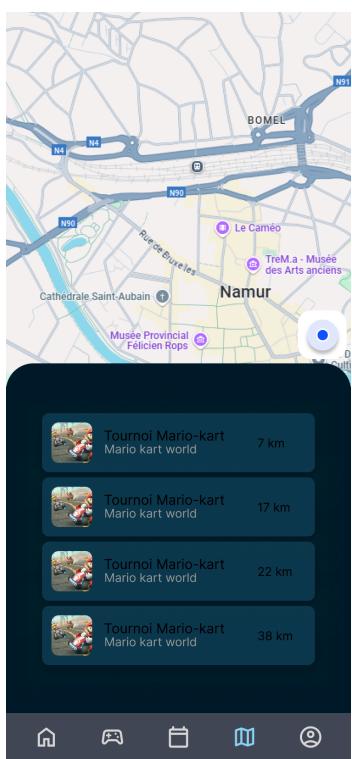
Évènements proches de vous

Navigation icons: Home, Games, Calendar, Map, Profile

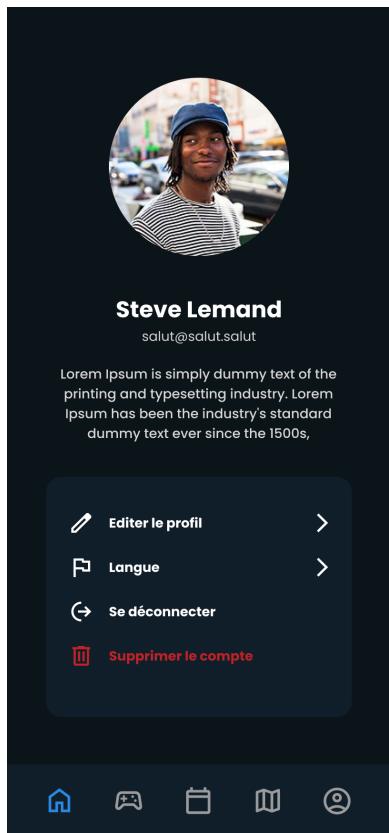
Page Evenement



Page Carte



Page Profil



Lien figma

https://www.figma.com/design/ISEWmQcy4MyAAulsTSpntr/Untitled?node_id=0-1&m=dev&t=IS9nDC0GoeJEbtRf-1

4. Structure

Description structure utilisé

- **index.js**: point d'entrée.
- **App.js**: définit la base de l'application.
- **App.json**: contient la configuration de l'application.
- **./src/**: contient le code source de l'app.
- **./assets/**: contient la petite icône de l'application.
- **./config/**: contient les 2 URLs pour accéder à l'API.
- **./constants/**: contient les constantes (traduction et couleurs) utilisées à travers l'application.
- **./navigation/**: contient le gestionnaire de navigation pour les déplacements entre les pages.
- **./screens/**: contient toutes les pages de l'application.
- **./services/**: contient le fichier api.js .
- **./store/**: contient les différentes données persistantes de l'application.
- **./styles/**: contient une feuille de style globale partagées parmi la majorité des pages.

Description packages utilisés

- **expo**: Framework pour déployer et tester des applications mobiles.
- **react et react-native**: librairies utilisées pour construire la base de l'application mobile.
- **react-navigation/native-stack et react-navigation/bottom-tabs**: librairies permettant de gérer la navigation entre les différentes pages de l'application et bottom-tabs permet d'avoir une barre de navigation en bas de l'écran
- **reduxjs/toolkit et react-redux**: permet de pouvoir gérer l'état global de l'application et react-redux permet de pouvoir faire le lien entre react et redux.
- **axios**: librairie HTTP pour pouvoir se connecter et communiquer avec le backend.
- **react-native-async-storage/async-storage**: Système de stockage persistant pour pouvoir sauvegarder localement les différentes informations de l'utilisateur (son token et sa langue dans notre cas).
- **react-native-community/datetimepicker**: Composant permettant de récupérer l'heure/date du système directement.
- **expo/vector-icons**: librairie d'icône vectorielle (svg), utilisée pour avoir accès à des icônes gratuites.
- **expo-build-properties**: configuration d'expo pour les builds natifs.
- **expo-image-picker**: permet de pouvoir accéder à la galerie du téléphone et en récupérer les images.
- **expo-location**: donne accès à la localisation de l'appareil
- **react-native-maps**: donne accès à Google Maps ou Apple Maps facilement pour avoir des intégrations de maps dans l'appli mobile.

- **react-native-safe-area-context**: force l'affichage dans les zones “sûres” de l'écran (pas dans les encoches/barres de status/...)