

DOSSIER PROJET



Projet : Action ou vérité

Rédacteur : Anthony Nosibor

Date : Le 30 décembre 2021



TABLE DES MATIERES

1. LISTE DES COMPETENCES DU REFERENTIEL COUVERTES PAR LE PROJET 6

1.1 Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité.....	6
1.1.1 Maquetter une application	6
1.1.2 Développer une interface utilisateurs desktop	6
1.1.3 Développer des composants d'accès aux données	6
1.1.4 Développer la partie front-end d'une interface utilisateur web	6
1.1.5 Développer la partie back-end d'une interface utilisateur web	6
1.2 Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	7
1.2.1 Concevoir une base de données	7
1.2.2 Mettre en place une base de données	7
1.2.3 Développer des composants dans le langage d'une base de données	7
1.3 Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	7
1.3.1 Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement.....	7
1.3.2 Concevoir une application.....	7
1.3.3 Développer les composants métier	8
1.3.4 Construire une application organisée en couches	8
1.3.5 Développer une application mobile	8
1.3.6 Préparer et exécuter les plans de tests d'une application	8
1.3.7 Préparer et exécuter le déploiement d'une application	8

2. RESUMES DU PROJET 9

3. CAHIER DES CHARGES 10

3.1	Présentation d'ensemble du projet	10
3.1.1	Présentation du client	10
3.1.2	Description de l'existant	10
3.1.3	Objectifs du projet.....	10
3.1.4	Intervenants sur le projet	11
3.1.5	Cible adressée par le projet	11
3.1.6	Exigences de performances et de volumétrie	12
3.1.7	Multilinguisme & adaptations pour un public spécifique	12
3.1.8	Reprise de l'existant	12
3.2	Description graphique et ergonomique	12
3.2.1	Composants de la charte graphique	12
3.2.2	Maquettes	13
3.3	Besoins fonctionnels « métier »	18
3.3.1	Utilisateurs du projet	18
3.3.2	Processus utilisateur impacté	18
3.3.3	Informations relatives aux contenus	18
3.3.4	Inventaire des besoins fonctionnels	18
3.4	Prestations attendues	20
3.4.1	Livrables et prestations	20
3.4.2	Budget	20
3.4.3	Planning et principaux jalons	21

4. GESTION DE PROJET 22

4.1	Méthodologie	22
4.2	Outils	22
4.3	Planning et suivi	23
4.4	Environnement technique et humain	24

5. SPECIFICATIONS FONCTIONNELLES 25

5.1.	Périmètre fonctionnel du projet	25
5.2	Arborescence..	26
5.2	Fonctionnalités	27
5.3	Fonctionnalités générales	29

6. SPECIFICATIONS TECHNIQUES 31

6.1.	Référencement	31
6.2.	Performances et Volumétrie	31

6.3. Navigation	31
6.4. Accessibilité	33
6.5 Services tiers	33
6.6 Sécurité	33
6.7 Gestion du contenu, des données	34
6.8.3. Modèle Physique de données	34
6.8.3. Conception base de données	35

7. REALISATIONS 36

7.1. Ma définition de redux	36
7.1.1. Extrait(s) de code	37
7.1.2. Argumentation	37
7.2 Utilisation de redux dans l'application.....	37
7.3 Distribution des questions dans l'application	38

8. PRESENTATION DU JEU D'ESSAI DE LA FONCTIONNALITE LA PLUS

REPRESENTATIVE 39

8.1. Fonctionnalité testée	39
8.2. Description des scenarios et résultat des tests	39
8.3. Conclusion	39

9. DESCRIPTION DE LA VEILLE SUR LES VULNERABILITES DE SECURITE 40

10. DESCRIPTION D'UNE SITUATION DE TRAVAIL AYANT NECESSITE UNE	
RECHERCHE SUR SITE ANGLOPHONE OU FRANCOPHONE	41
10.1 Description du besoin	41
10.2 Description de la recherche	41
10.2.1. Moteur de recherche	41
10.2.2. Mots clés	41
10.2.3. Résultat de la recherche (liste Sites)	41
10.2.4. Critères de sélection et choix du(es) site(s)	41
10.2.5. Extrait du(es) site(s)	42
10.2.6. Traduction de l'extrait	43

1. LISTE DES COMPETENCES DU REFERENTIEL COUVERTES PAR LE PROJET

1.1. Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

1.1.1. Maquetter une application

J'ai maqueté l'application Action ou vérité en utilisant le logiciel Figma, pour le design des interfaces desktop et mobile, en respectant les directives du cahier des charges, notamment de la charte graphique (police d'écriture et couleurs).

1.1.2. Développer une interface utilisateur desktop

J'ai pour projet de développer une interface utilisateur desktop grâce au framework *electron* qui permettra à l'administrateur de se créer un compte sur l'application, de pouvoir se rendre sur son profil, d'ajouter une photo de profil, de modifier ses informations de connexion, ainsi que de pouvoir consulter les parties en cours, grâce à une interface ergonomique avec des couleurs rappelant le thème sombre.

1.1.3. Développer des composants d'accès aux données

Les données sont stockées sur la base de données Firebase (Cloud firestore) en "NoSql", et les images y sont reliées depuis l'interface de stockage de Firebase. L'accès et la récupération de données fonctionne comme une API en fonction des requêtes du front via Node Js.

1.1.4. Développer la partie front-end d'une interface utilisateur web

La partie front a été développée en React-native. J'ai aussi utilisé des composants grâce au Node Package Manager (NPM), pour faciliter l'utilisation d'icônes par exemple. Enfin, concernant la sauvegarde des données et le versionning, j'ai utilisé GitHub.

1.1.5. Développer des composants d'accès aux données

J'ai utilisé Node Js pour développer la partie back-end de mon projet, pour d'une part, la performance asynchrone offerte concernant le traitement d'un gros volume de requêtes simultanément de façon efficace, ce qui va se produire sur l'application car les utilisateurs vont utiliser l'application en ligne en même temps. D'autre part pour la disponibilité des packages de NPM qui permettent de développer plus rapidement des composants de base.

1.2. Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

1.2.1. Concevoir une base de données

L'étude du Cahier des charges et des spécifications fonctionnelles m'ont permis de créer un modèle conceptuel de donnée, afin de schématiser et visualiser les données qui vont être utilisées dans le système d'information.

1.2.2 Mettre en place une base de données

Pour la mise en place de la base de données j'ai d'une part installée Firebase sur mon projet (grâce à npm) et d'autre part créé mon projet sur l'interface web afin d'avoir accès à la console, où j'ai pu d'une part créer la base de données en ligne et initialiser le « storage » pour stocker les images. J'ai ensuite pu coder la liaison entre mon projet et Firebase grâce à Node JS et à l'aide de la configuration récupérée sur l'interface de Firebase. La création des collections et des documents se fait à l'exécution du code et des commandes en front.

1.2.3. Développer des composants dans le langage d'une base de données

J'ai développé les requêtes en langage No SQL, ce qui permet de traiter rapidement et avec flexibilité un volume important de données sans contraintes de champs, de façon plus optimale qu'une base de données traditionnelle.

1.3. Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

1.3.1. Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

J'ai suivi la méthode Agile pour la gestion du projet informatique, avec des sprints d'environ 15 jours, ce qui me donne le temps de développer plusieurs fonctionnalités, ce qui me permet par la suite de tester la fonctionnalité de valider son utilité et de réitérer soit pour développer une autre fonctionnalité dans le même volet, soit pour passer au volet suivant, afin de respecter les contraintes de temps.

1.3.2. Concevoir une application

J'ai réalisé une étude de marché en interrogeant plusieurs catégories de personnes concernant leur utilité de l'application. J'ai pu déterminer leurs attentes et définir les principales catégories de l'application qui permettraient principalement de faciliter l'utilisation de l'application en optimisant les fonctionnalités simples ainsi que l'ergonomie. Après cette réflexion, j'ai pu schématiser le design et le système relationnel de l'application en prenant en compte les besoins de chaque utilisateur. J'ai ensuite, à partir des cas d'utilisation, pu établir les UML et diagrammes d'activités afin de pouvoir définir les fonctionnalités à développer, en front et en back.

1.3.3. Développer des composants métier

J'ai utilisé l'interface Firebase ainsi que Node JS pour la création des comptes et la connexion des utilisateurs, avec courriel et mot de passe ou via Facebook et Google.

1.3.4. Construire une application organisée en couches

L'architecture de mon projet répond à l'organisation intuitive de MVC (Modèle, View Controller), où le modèle est sous forme de données de la base et relié au Controller, la View est représentée par le JSX et le constructeur sous forme de classes.

1.3.5. Développer une application mobile

L'application mobile action ou vérité est développée en React Native pour le front et Node JS/Firebase pour le back, le but étant pour les utilisateurs une utilisation plus facile et instantanée de l'application.

1.3.6. Préparer et exécuter les plans de tests d'une application

Action ou vérité va se soumettre à deux types de tests :

Les tests unitaires : afin de vérifier la bonne exécution par rapport à un résultat attendu. La rédaction du cas de test comprend :

- Le type de test à exécuter
- Le scénario d'exécution
- La description du résultat attendu
- La comparaison avec le résultat actuel.

Les tests d'acceptance des utilisateurs : Ce test a pour but de vérifier la bonne adhérence et manipulation des utilisateurs sur l'application, afin de s'assurer que l'UI et l'UX sont optimales.

1.3.7. Préparer et exécuter le déploiement d'une application

Etant donné qu'action ou vérité n'est pas un projet d'entreprise, il n'a pas la possibilité être déployé sur un serveur de test. Cependant, une publicité est mise en place dès sa mise en production, ce qui laisse une marge pour apporter des correctifs avant les premiers utilisateurs. Enfin, les mises à jour et correctifs ultérieurs seront développées et testées dans un environnement local avant la mise en production.

2. RESUMES DU PROJET

English summary :

Action ou Vérité is a mobile application that aims to help users pass the time in a fun way.

In order to design this project in a structured way, I first developed specifications that include the analyzed needs and the technologies necessary for the design of the application. I continued this project initialization phase with the development of a model.

Users of the application, after authentication, go to a game initialization pages, from where they define the number of players and their names.

Users will play a Creole version of the game. Finally, as the project administrator I have the right to view and delete users, via the firebase administration interface.

This initial project is to be improved through new functionalities currently under study. In fact, in a future version of the application, premium users will be able to send and view messages, via a profile page.

In addition, I planned to develop a desktop administration interface allowing me to create an account on the application, to be able to go to the saved profiles, to add a profile picture, to modify the connection information, as well as being able to consult the choices, thanks to an ergonomic interface with colors reminiscent of the dark theme.

On the other hand, I will have the right to inspect the messages.

The project is made, for the front in React JS. The database was made with Firebase (Firestore) and is linked to the back via an API.

Résumé en Français :

Action ou vérité est un jeu mobile qui a pour objectif d'aider les utilisateurs à passer le temps de manière ludique.

Afin de concevoir de manière structurée ce projet, j'ai dans un premier temps élaboré un cahier des charges qui reprend les besoins analysés et les technologies nécessaires à la conception de l'application. J'ai poursuivi cette phase d'initialisation de projet avec l'élaboration d'une maquette.

Les utilisateurs de l'application, après s'être authentifiés, accèdent aux pages d'initialisation de la partie, d'où ils définissent le nombre de joueurs ainsi que leurs noms.

Les utilisateurs joueront à une version créole du jeu.

Enfin, en tant qu'administrateur du projet j'ai un droit de regard et de suppression des utilisateurs, via l'interface d'administration de firebase.

Ce projet initial est amené à être amélioré par l'intermédiaire de nouvelles fonctionnalités actuellement en cours d'étude. En effet, les utilisateurs premium pourront dans une prochaine version de l'application, envoyer et consulter des messages, via une page profile. Par ailleurs, j'ai prévu de développer une interface d'administration desktop permettant me permettant de créer un compte sur l'application, de pouvoir me rendre sur les profils enregistrés, d'ajouter une photo de profil, de modifier les informations de connexion, ainsi que de pouvoir consulter les choix, grâce à une interface ergonomique avec des couleurs rappelant le thème sombre. D'autres part j'aurai un droit de regard sur les messages.

Action ou vérité est fait, pour le front, en React-native. La base de données est réalisée avec Firebase (dont la base de données Firestore).

3. CAHIER DES CHARGES

3.1 Présentation d'ensemble du projet

3.1.1 Présentation du client

La volonté de créer l'application Action ou vérité est venue d'un exercice fait en formation.

Il s'agit d'une application complètement gratuite pour ses utilisateurs, le but étant comme bon nombre d'application de jeux sur le marché, de permettre aux utilisateurs de s'ouvrir aux autres de façon ludique et amusante. Cependant, il n'est pas exclu d'ajouter, dans un second temps de la publicité liée à cette application, afin, de générer des entrées d'argent. D'autres part j'ai prévu d'ajouter dans une prochaine version, une fonctionnalité payante, c'est à dire la possibilité de converser et jouer avec les utilisateurs ayant payé pour cette option.

Les principaux concurrents d'Action ou vérité sont :

- Action ou vérité ? jeu soirée
- Action ou vérité – Hot
- Action ou vérité - jeu de la bouteille

3.1.2 Description de l'existant

Action ou vérité est un PMV (Produit Minimum Viable). Ce projet ne s'inscrit pas dans une stratégie d'entreprise car c'est le seul produit proposé pour l'instant, il est au cœur de ma stratégie de développement professionnel.

Il ne s'agit pas d'un produit nouveau et innovant, en effet il existe une dizaine de jeu similaire sur le play store.

3.1.3 Objectifs du projet

L'objectif du projet est d'aider les utilisateurs à s'ouvrir aux autres d'une manière insolites, loin des freins à la communication inhérents aux interactions en face à face. Pour cela, une étude des forces, faiblesses, opportunités et menaces est mise en œuvre au préalable, pour tester la viabilité de l'idée de l'application. (Cf : Matrice SWOT ci-dessous)

Forces	Faiblesses
--------	------------

<ul style="list-style-type: none"> • Application gratuite • Solution complète à terme • Design ergonomique (UI) 	<ul style="list-style-type: none"> • Application encore inconnue pas de publicité (possibilité de désintérêt pour l'application)
<i>Opportunités</i>	<i>Menaces</i>
<ul style="list-style-type: none"> • Croissance de la demande d'application ludique 	<ul style="list-style-type: none"> • Concurrence forte des agences de conception d'application • Applications similaire multiples

3.1.4 Intervenants sur le projet

Il y a une seule personne sur le projet, qui a pour fonction :

- **Webdesigner** (création du logo, planification de l'agencement des pages, choix des images)
- **Développeur** (front en ReactJs ?, back en NodeJS et base de données sur Firebase)
- **Chef de projet** (recensement des besoin, définition des éléments à ajouter au projet, responsable de la deadline)

3.1.5 Cible adressée par le projet

La clientèle qui est susceptible de se rendre sur l'application sont :

- Les groupes d'amis souhaitant organiser une session + 18.
L'application est destinée principalement aux particuliers (B2C).

Il s'agit globalement d'utilisateurs jeune car plus enclin à pratiquer ce genre de jeu. Leur centre d'intérêt est de bien sûr s'amuser, découvrir de nouvelles personnes ayant les mêmes centres d'intérêt loin des applications de rencontre basique.

Dans l'idéal, l'application doit être bien référencée. Pour cela, l'application a :

- Un nom unique, qui n'existe pas sous une autre forme sur Google ou avec une écriture proche d'une grande marque
- Des boutons de partage vers des réseaux sociaux (Facebook, Instagram,)
- Des mots clés pour optimiser la recherche Google
- Des descriptions sur les images (le « alt » rempli) pour éviter les confusions avec Google
- Du contenu de texte unique
- Une bonne architecture interne

3.1.6 Exigences de performances et de volumétrie

- Nombre de visiteurs escomptés par mois : 100 000 utilisateurs.
- Nombre de nouveaux utilisateurs par mois : 10 000 de façon exponentielle
- Volume de vente et CA mensuel : 0, l'application est gratuite, possibilité de revenus grâce à de la publicité.

3.1.7 Multilinguisme et adaptations pour un public spécifique

Dans un premier temps (première version), l'application mobile est en créole afin d'adhérer à un public, antillais puis déclinée en français afin d'accéder au public plus francophone.

Le logo est quant à lui identique et n'est pas traduit.

3.1.8 Reprise de l'existant

L'application est totalement nouvelle, et aucune version précédente n'est reprise. Il n'y a aucun patrimoine documentaire disponible.

3.2 Description graphique et ergonomique

3.2.1 Composants de la charte graphique

L'identité visuelle a été défini grâce au logiciel [Logoaster](#) :

Logo :



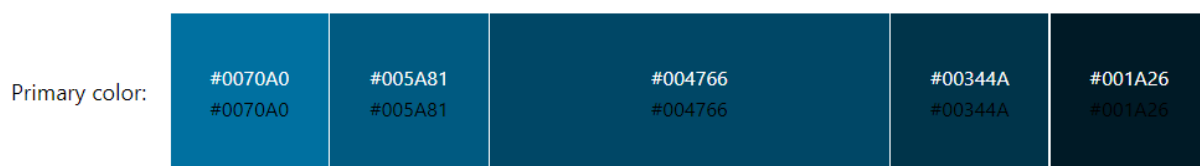
Favicon :



Police et taille de caractères :

- Sans-serif : Quickstand, 14px
- Serif : Roboto, ○ h1 : 2.5rem ○ h2 : 2rem ○ h3 : 1.6rem

Palettes de couleurs :



3.2.2 Maquettes

Après avoir dessiné sur papier les principales fonctionnalités de l'application ainsi que les liens entre les interfaces, des maquettes sont établis :

Mobile :



9:27

Inscription

ou alors

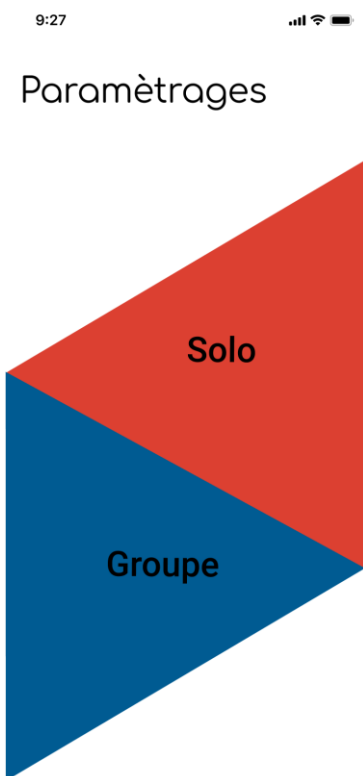
Déjà inscrit ? Connectez-vous.

A standard iOS-style mobile keyboard is shown at the bottom of the screen, with keys for letters, numbers, space, and return.

9:27

Connexion

ou alors

A standard iOS-style mobile keyboard is shown at the bottom of the screen, with keys for letters, numbers, space, and return.

9:27

Entrez le nombre de joueurs :

suivant

9:27

Entrez les nom de joueurs :

suivant

9:27



Page de jeu

SALUT ANTHO !!!!!!!

Action

Vérité




Action

Embrasse la personne à ta droite !

JOUEUR
SUIVANT



Desktop :




Dashboard

Connexion au Dashboard

Entrez vos email et mot de passe ci-dessous

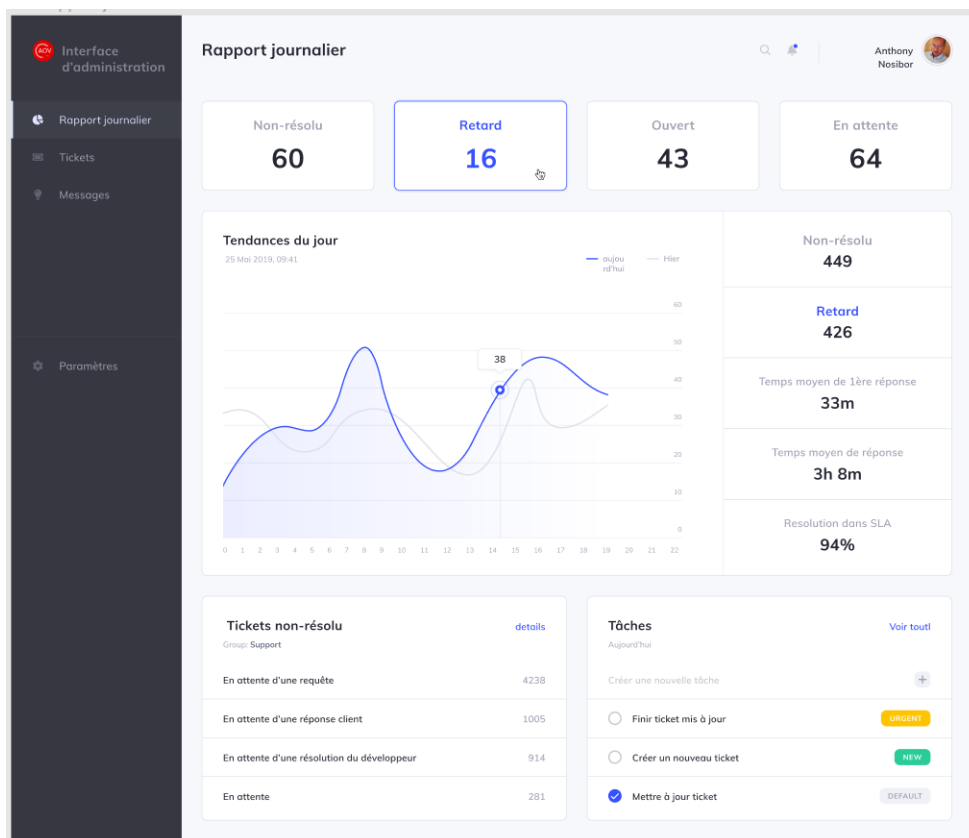
EMAIL

MOT DE PASSE [Mot de passe oublié?](#)



[Connexion](#)

[Vous n'avez pas de compte?](#) [Inscrivez-vous](#)



Interface d'administration

Rapport journalier

Tickets

Messages

Paramètres

Tickets

Anthony Nosibor

Tout les tickets

TF Sort Filter

Ticket details	Nom des clients	Date	Priorité
<div>Contact Email not Linked</div> <div>Updated 1 day ago</div>	<div>Tom Cruise</div> <div>on 24.05.2019</div>	<div>May 26, 2019</div> <div>6:30 PM</div>	<div>HAUTE</div>
<div>Adding Images to Featured Posts</div> <div>Updated 1 day ago</div>	<div>Matt Damon</div> <div>on 24.05.2019</div>	<div>May 26, 2019</div> <div>8:00 AM</div>	<div>BASSE</div>
<div>When will I be charged this month?</div> <div>Updated 1 day ago</div>	<div>Robert Downey</div> <div>on 24.05.2019</div>	<div>May 26, 2019</div> <div>7:30 PM</div>	<div>HAUTE</div>
<div>Payment not going through</div> <div>Updated 2 days ago</div>	<div>Christian Bale</div> <div>on 24.05.2019</div>	<div>May 25, 2019</div> <div>5:00 PM</div>	<div>NORMALE</div>
<div>Unable to add replies</div> <div>Updated 2 days ago</div>	<div>Henry Cavill</div> <div>on 24.05.2019</div>	<div>May 25, 2019</div> <div>4:00 PM</div>	<div>HAUTE</div>
<div>Downtime since last week</div> <div>Updated 3 days ago</div>	<div>Chris Evans</div> <div>on 23.05.2019</div>	<div>May 25, 2019</div> <div>2:00 PM</div>	<div>NORMALE</div>
<div>Referral Bonus</div> <div>Updated 4 day ago</div>	<div>Sam Smith</div> <div>on 22.05.2019</div>	<div>May 25, 2019</div> <div>11:30 AM</div>	<div>BASSE</div>
<div>How do I change my password?</div> <div>Updated 6 days ago</div>	<div>Steve Rogers</div> <div>on 21.05.2019</div>	<div>May 24, 2019</div> <div>1:00 PM</div>	<div>NORMALE</div>

Lignes / page: 8 1-8 of 1240

3.3 Besoins fonctionnels

3.1.1 utilisateurs du projet

Les utilisateurs se déclinent en deux catégories :

- **Joueurs basiques** : Qui utilisent l'application pour le jeu uniquement.
- **Joueurs premium** : Qui souhaitent échanger avec les autres joueurs.

3.1.2 Processus utilisateur impacté

Ce projet n'a aucun impact sur la gestion d'une entreprise car c'est un projet innovant et totalement à part.

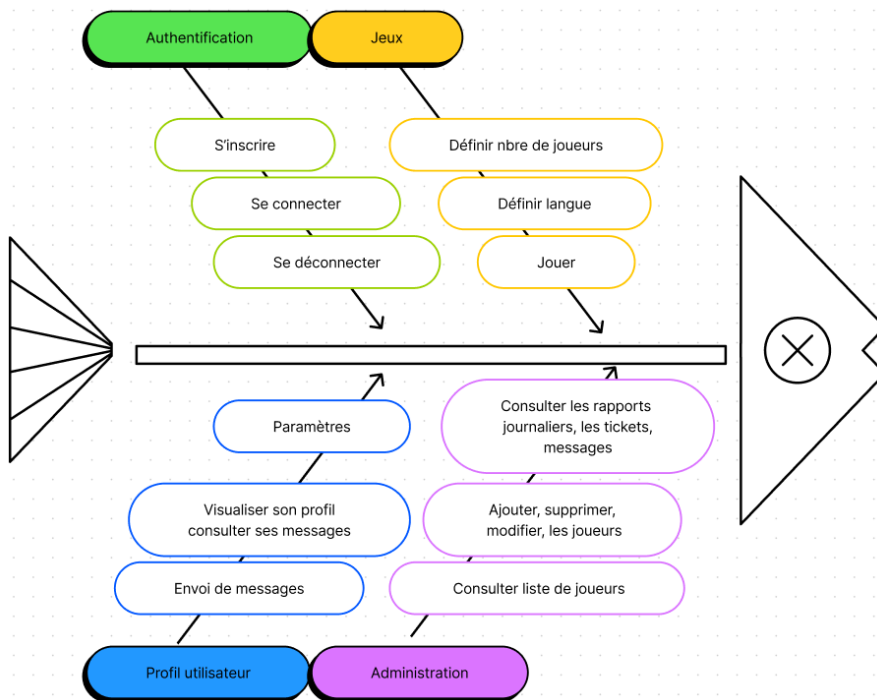
3.1.3 Informations relatives aux contenus

Le type de contenu qui va être présenté sur l'application sont uniquement des images et du texte.

Concernant le droit à l'image, les utilisateurs sont tenus responsables des photos qu'ils publient. Pour les photos de « décoration » (page d'accueil, login...) ce sont des images libres de droits d'auteur.

3.1.4 Inventaire des besoins fonctionnels

Diagramme FISHBONE des fonctionnalités *l'application* :



Versions de Action ou vérité :

Les principaux axes de développement pour en faire une application complète va se décliner en plusieurs versions avec des ajouts de fonctionnalités à chaque fois :

- Création d'un profil utilisateur (photo de profil)
- Possibilité de mettre en favoris des joueurs.
- Améliorations à la suite des retours des utilisateurs concernant l'application.
- Gestion du multilingue (créole ou français).

3.4 Prestations attendues

3.4.1 Livrables et prestations

Prestations attendues :

- Rédaction du cahier des charges
- Rédaction des spécifications fonctionnelles & techniques
- Maquettage & design
- Développement
- Intégration
- Création de la base de données
- Déploiement et gestion de l'hébergement avec Firebase
- Maintenance et mises à jour

3.4.2 Budget

Budget financier :

- **Firebase** : gratuit ou 25€ paiement selon l'utilisation par mois (environ 21€)

Budget temps :

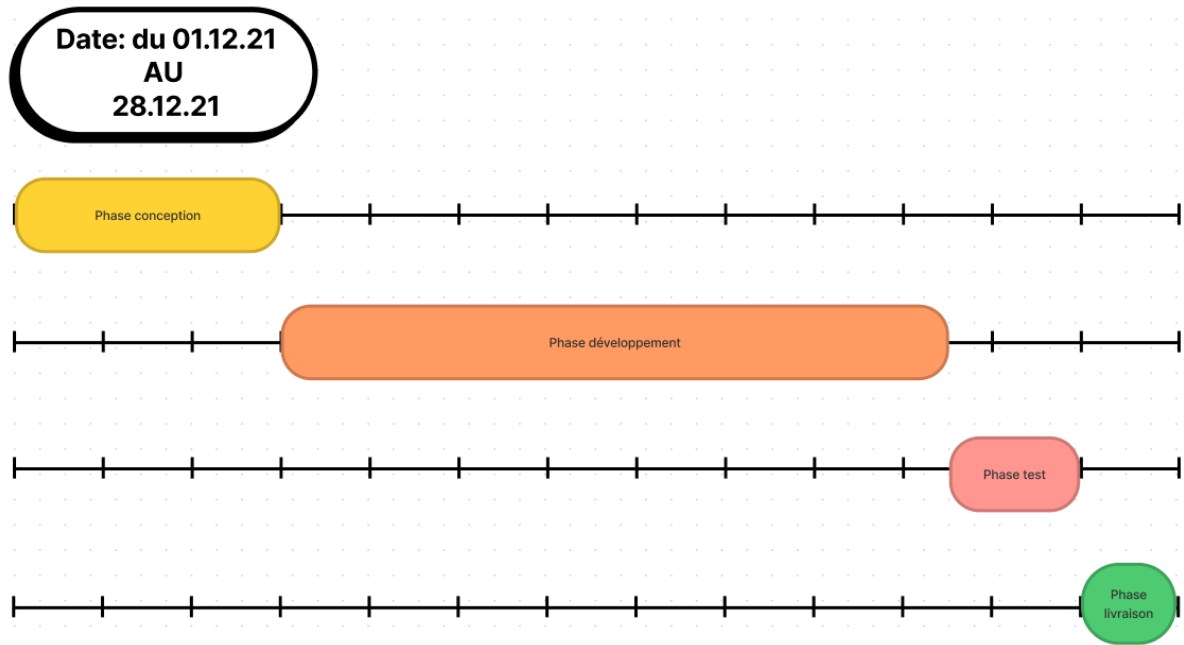
- Le projet est développé sur une période de 1 mois, de décembre à janvier, en 4 phases :
 - Phase conception (10 jours).
 - Phase développement (environ 15 jours).
 - Phase test : 2 jours.
 - Phase livraison : 1 jours.

3.4.3 Planning et principaux jalons

- Début du projet : 1^{er} décembre 2021
- Fin de la phase de conception et début de la phase de développement : 10 décembre 2021
- Date du début des tests utilisateurs : 25 décembre 2021
- Date de mise en service : 28 décembre 2021

Diagramme présentant une vue globale du projet :

Timeline



4. GESTION DE PROJET

4.1 Méthodologie

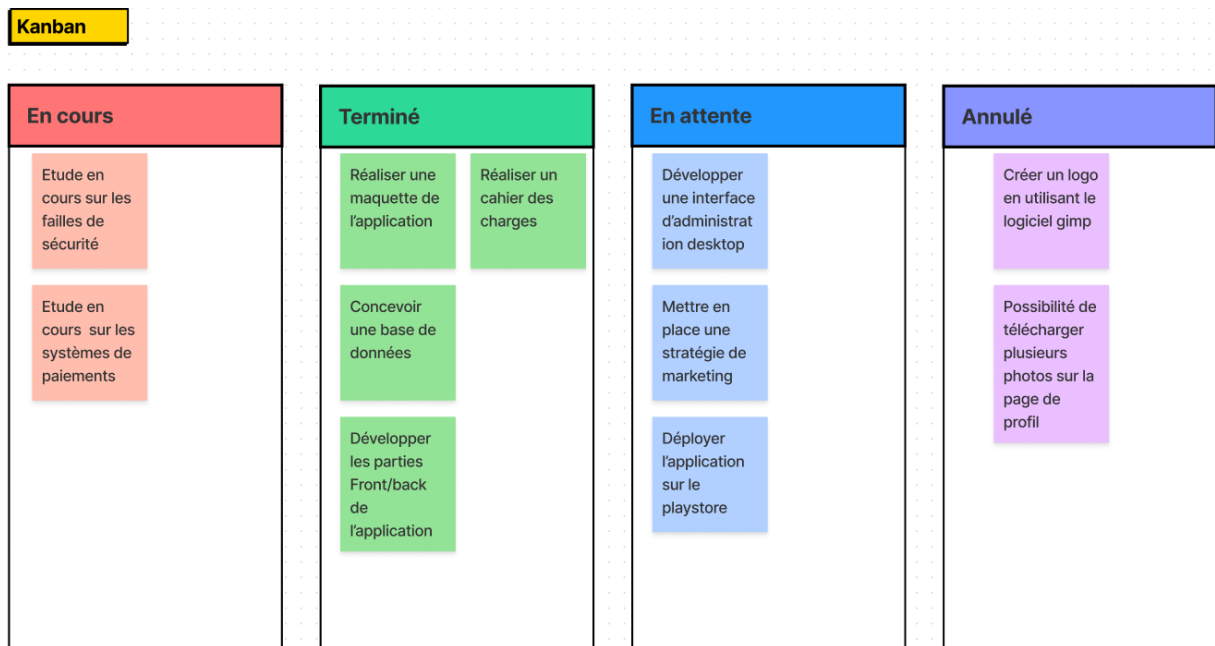
Conformément à la méthode agile, j'ai décomposé mon projet en une suite de petits objectifs atteignables :

- Phase de conception
- Phase de développement
- Phase de test
- Phase de livraison

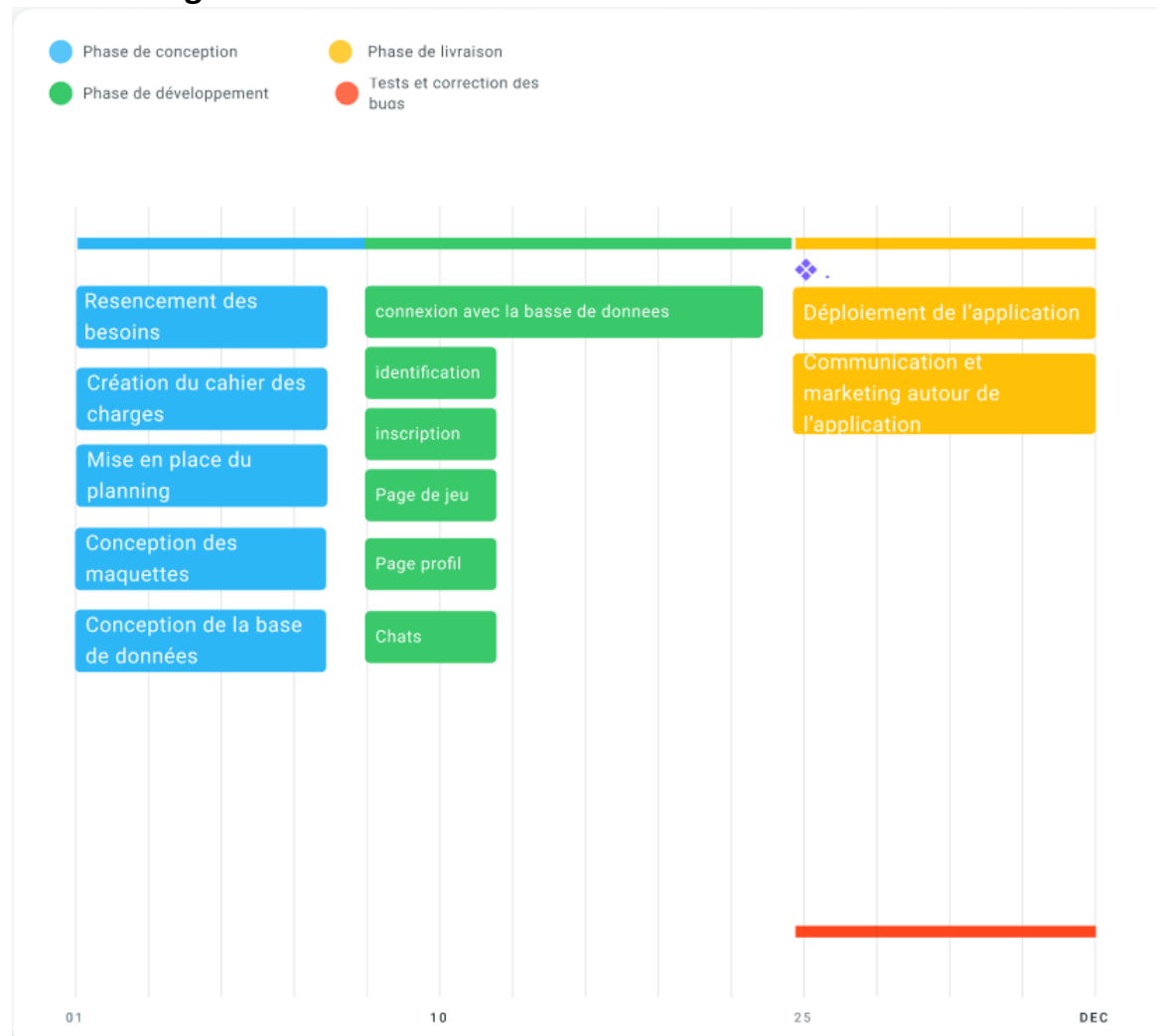
4.2 Outils

Afin de mesurer le temps total de conception du projet, j'ai d'abord réalisé un diagramme de Gantt pour estimer le temps pour exécuter chaque tâche, de la conception à la livraison du produit, en prenant en compte une marge afin de résoudre les bugs et problèmes qui pourraient apparaître. Par la suite, j'ai réalisé un tableau Kanban afin de visualiser plus précisément les tâches qui me restaient à effectuer (Cf *tableau Kanban, diagramme de Gantt ci-dessous*).

Tableau d'avancement du projet (kanban) :



4.3 Planning et suivi



4.4 Environnement technique et humain

Environnement technique :

- Utilisation d'un diagramme de Gantt pour suivre l'avancée du projet et corriger les retards éventuels

Environnement humain :

- Réunion discord avec mon formateur et mes collègues stagiaires pour leurs faire par des avancées de l'application, tester l'application et recueillir leurs retours pour d'éventuelles améliorations.

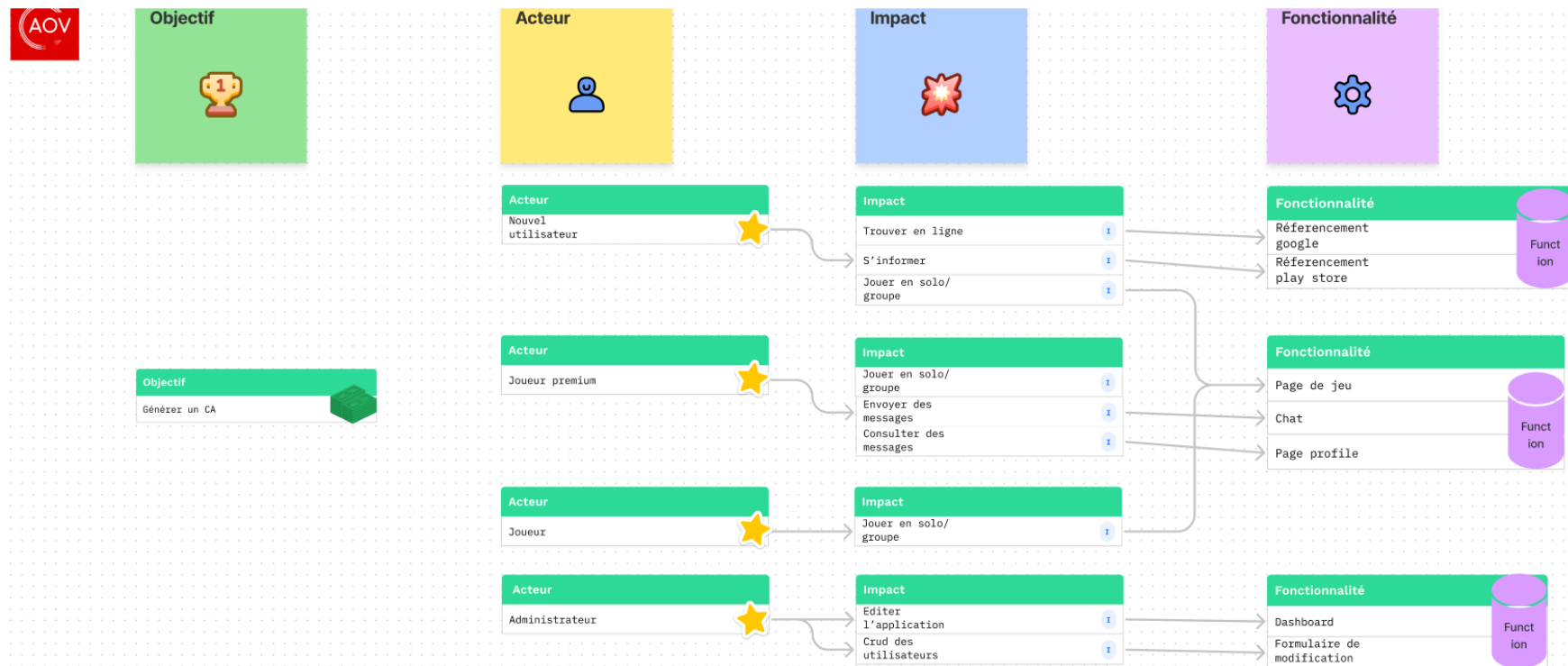
5. SPECIFICATIONS FONCTIONNELLES

5.1 Périmètre fonctionnel du projet

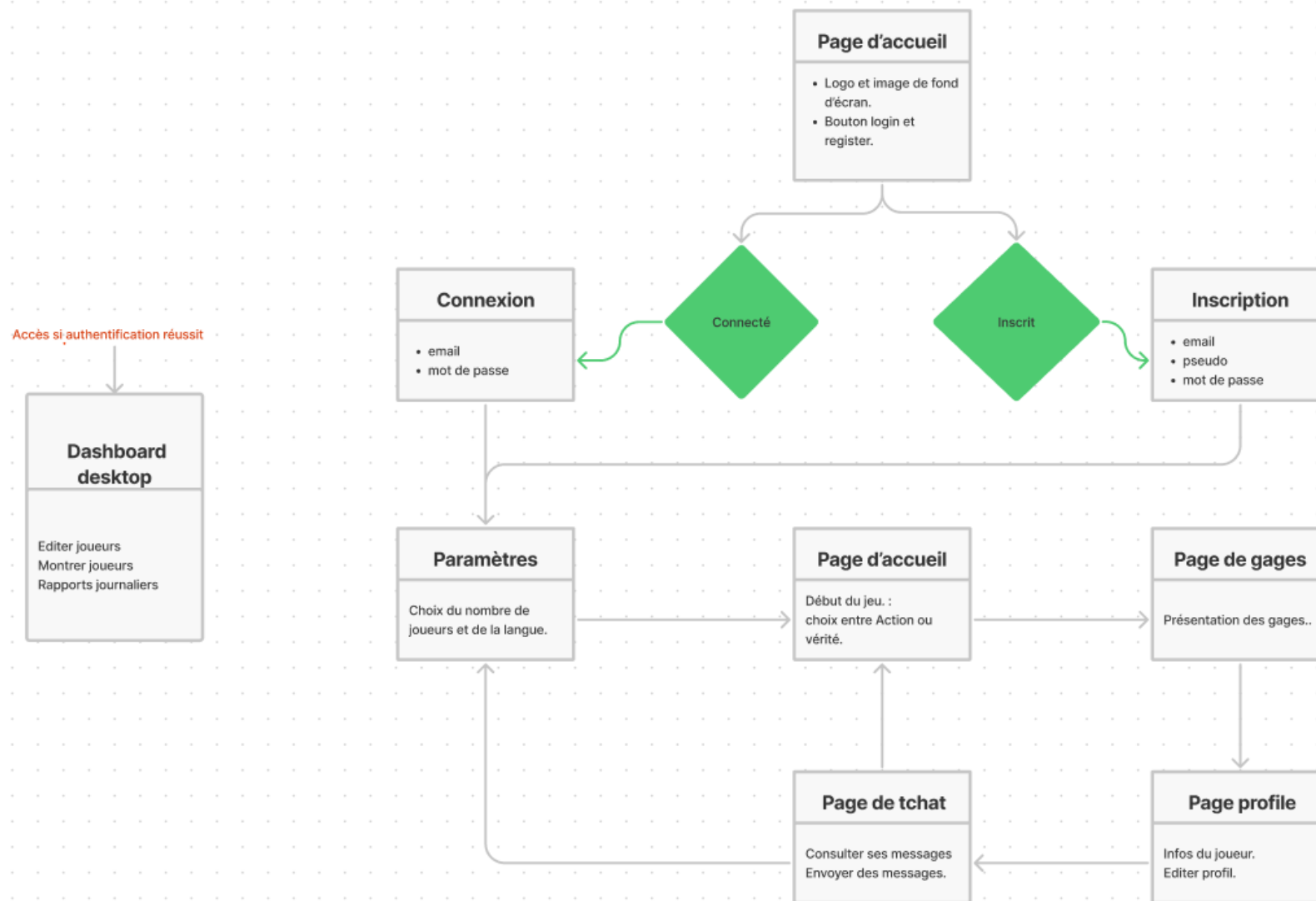
Le site sera entièrement rédigé en français, donc à destination d'un public francophone. Cependant il sera possible de lancer une partie en créole, donc l'application s'adresse aussi à un public créolophone.

Impact mapping :

L'élaboration d'un impact mapping ma permis de préparer une bonne vision de l'application que je désire réaliser



5.2 Arborescence du projet



5.3 Fonctionnalités

5.3.1 Utilisateurs

Acteurs	Joueur basique	Joueur premium	Administrateur
Fonctionnalité			
Authentification			
Se connecter/déconnecter		x	x
Changer son mot de passe		x	x
Envoyer un courriel pour changer de mot de passe			x
Supprimer un utilisateur			x
Navigation			
Page de lancement (splash screen)	x	x	x
Page d'authentification	x	x	x
Page de profil/réglages	x	x	x
Page d'accueil	x	x	x
Page d'administrateur			x
Jeu			

Définir nombre de joueurs	x	x	x
Définir la langue	x	x	x
Lancer une partie	x	x	x
Commenter/envoyer des messages		x	x
<i>Profil</i>			
Changer sa photo de profil		x	x
Changer son nom d'utilisateur		x	x
Changer sa bio		x	x
Supprimer son compte		x	x
Supprimer plusieurs comptes			x

5.4 Fonctionnalités générales

5.3.2.1 Autorisations

Les autorisations se font sur une base de rôles :

- L'utilisateur connecté basique a accès à un ensemble de routes protégées, telles que la page de jeu et la page profil sans pouvoir ni envoyer ni recevoir de messages.
- L'utilisateur connecté premium a accès à un ensemble de routes protégées, telles que la page de jeu, la page de profil et la consultation ou envoi de messages.
- L'administrateur quant à lui est défini en tant que tel via la base de données, ce qui lui donne, via les routes protégées l'accès à une interface d'administration desktop d'où il peut piloter l'application.

L'autorisation se fait lors de l'authentification.

5.3.2.2 Envoi de mail

L'application utilise des fonctions de mailing pour renvoyer un lien à l'utilisateur s'il a perdu son mot de passe, de deux façons distinctes :

- Soit l'utilisateur connaît son adresse email et l'application envoie le mail immédiatement lors de la saisie de l'email de l'utilisateur
- Soit l'utilisateur envoie un courriel à l'administrateur qui, depuis sa page admin peut, en cliquant sur un bouton, envoyer un courriel avec un lien à l'utilisateur pour réinitialiser son mot de passe.

Enfin, le courriel est envoyé depuis "noreply@actionouverité.com" et l'administrateur de l'application n'est pas en copie de ces courriels.

5.3.2.3 Gestion des langues

La gestion des langues est prévue dans la version 2 d'action ou vérité. En effet, dès la page de paramètres, l'utilisateur a le choix de lancer une partie en français, ou en créole. Le passage d'une langue à l'autre se fera depuis l'interface de réglages du compte utilisateur, par le biais d'un switch :



5.3.2.4 Workflow de l'application

L'application et en particulier la phase de jeu en elle-même, ne repose sur aucun workflow global de validation, c'est à dire que la page de jeu sera disponible et visualisable dès que l'utilisateur valide son formulaire. Cependant, d'autres utilisateurs peuvent signaler à l'administrateur (par courriel puis dans une version ultérieure grâce à un simple bouton) un message ne respectant pas les conditions d'utilisation de l'application. A partir de là, l'administrateur peut visualiser, depuis son interface, la publication concernée et peut décider de la supprimer ou non. Enfin, au bout de quelques publications ne respectant pas les conditions d'utilisation de l'application, l'utilisateur pourrait être supprimé définitivement de l'application.

6. SPECIFICATIONS TECHNIQUES

6.1 Référencement

Comme indiqué dans le cahier des charges, il serait pertinent pour Action ou vérité d'être bien référencé. Pour cela, en plus d'avoir un nom unique, l'application applique au mieux les recommandations de google, à savoir :

- L'ajout d'une balise "alt" sur les images, avec des mots clés pertinents sur l'activité de l'application
- Un design simplifié et ergonomique
- Le code HTML est conforme aux standards HTML 5 et les balises titres (<title>, <h1>, <h2>...) sont étudiées pour fournir des mots clés stratégiques sur le contenu du site, grâce à une liste de mots clés que les internautes seraient susceptibles de taper sur google pour arriver sur le site.

6.2 Performances et Volumétrie

L'application est réalisée en front avec React-native avec Node Js en back et Firebase en base de données via la base de données Firestore.

Le choix s'est porté vers ces outils car :

- **React-native** : est la librairie javascript idéale pour construire des applications mobiles, avec des composants prêts à l'emploi téléchargeables. La particularité de cette librairie flexible est de pouvoir développer via une gestion d'état globale grâce à Redux, ce qui offre une liaison de données unidirectionnelle où les données en back sont disponibles depuis n'importe où dans l'application. Enfin, React permet de réduire les interactions avec le DOM, ce qui augmente l'optimisation et la vitesse des mises à jour, ce qui est un avantage pour les bases de données importantes.
- **Node JS**: est un framework back-end de javascript léger, efficace, permettant d'une part de gérer un important volume de données et d'autre part d'accélérer les applications en général, grâce à son système de requêtes asynchrones (sans bloquer le serveur dans l'attente d'une réponse).

- **Firebase** : est une plateforme de stockage de données backend pour les applications web et mobiles. Firebase est l'outil idéal pour lier l'application à une plateforme en back où les utilisateurs peuvent récupérer leurs données. La base de données utilisée par Action ou vérité est en NoSql, ce qui rend la base de données plus flexible. De plus, la base de données Firebase sauvegarde les activités de l'utilisateur dans le cache, avant de les renvoyer sur le serveur dès que la connexion est rétablie, ce qui est l'idéal pour une utilisation dans des endroits reculés et possiblement sans réseau. Enfin, l'authentification via firebase est simplifiée et plus sécurisée pour protéger les données des utilisateurs, et accessible depuis plusieurs terminaux (web et mobiles).

- **6.3 Navigation**

Action ou vérité est disponible sur mon téléphone personnel. Il est prévu d'héberger mon application sur firebase.

6.4 Accessibilité

La version 2 de l'application mobile sera hébergée sous firebase. Il y a un courriel associé à l'application :

- noreply@actionouverité.com pour les envois de lien pour les mots de passes perdus

Action ou vérité est compatible avec tous les mobiles, car React-native est cross-platform, donc supporté par *android* et *ios*.

6.5 Services tiers

Les services tiers intégrés à Action ou vérité sont :

- Des modules firebase (emailing en cas de mot de passe perdu)
- Google Analytics (pour surveiller les visites sur l'application)
- Google captcha (lors de l'inscription, pour éviter un remplissage de la base de données par un robot)
- Lien google et Facebook pour l'authentification

6.6 Sécurité

La sécurité est gérée sur plusieurs plans :

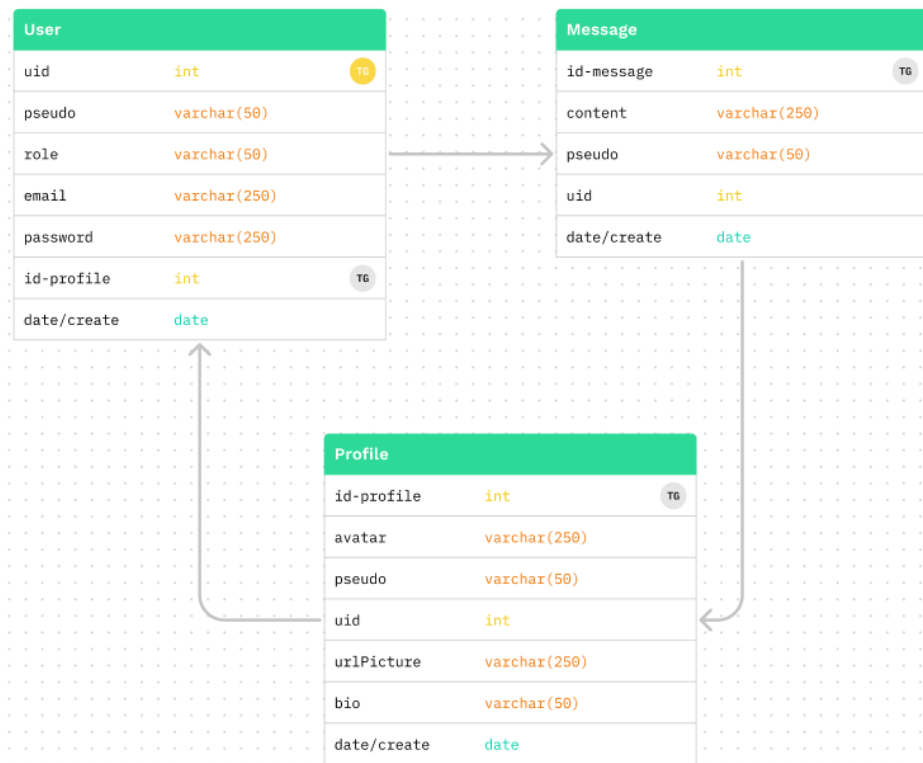
- Obligation pour un utilisateur de s'authentifier pour pouvoir utiliser l'application (et donc traçabilité des publications)
- Gestion des droits en fonction des rôles (utilisateur/administrateur)

L'application est sauvegardée deux fois :

- Sur le serveur d'hébergement de Firebase
- En local sur ordinateur
- Le stockage est uniquement sauvegardé sur firebase dans la partie stockage de l'application.
- Le versionning est indiqué dans le package.json

6.7 Gestion du contenu, des données

6.7.1 Modèle physique de données



La base de données d'Action ou vérité est en NoSQL, il n'y a donc pas de clés primaires et étrangères. Afin de pouvoir sauvegarder des données sur Firestore, il va falloir les structurer au modèle **NoSQL orienté Documents**.

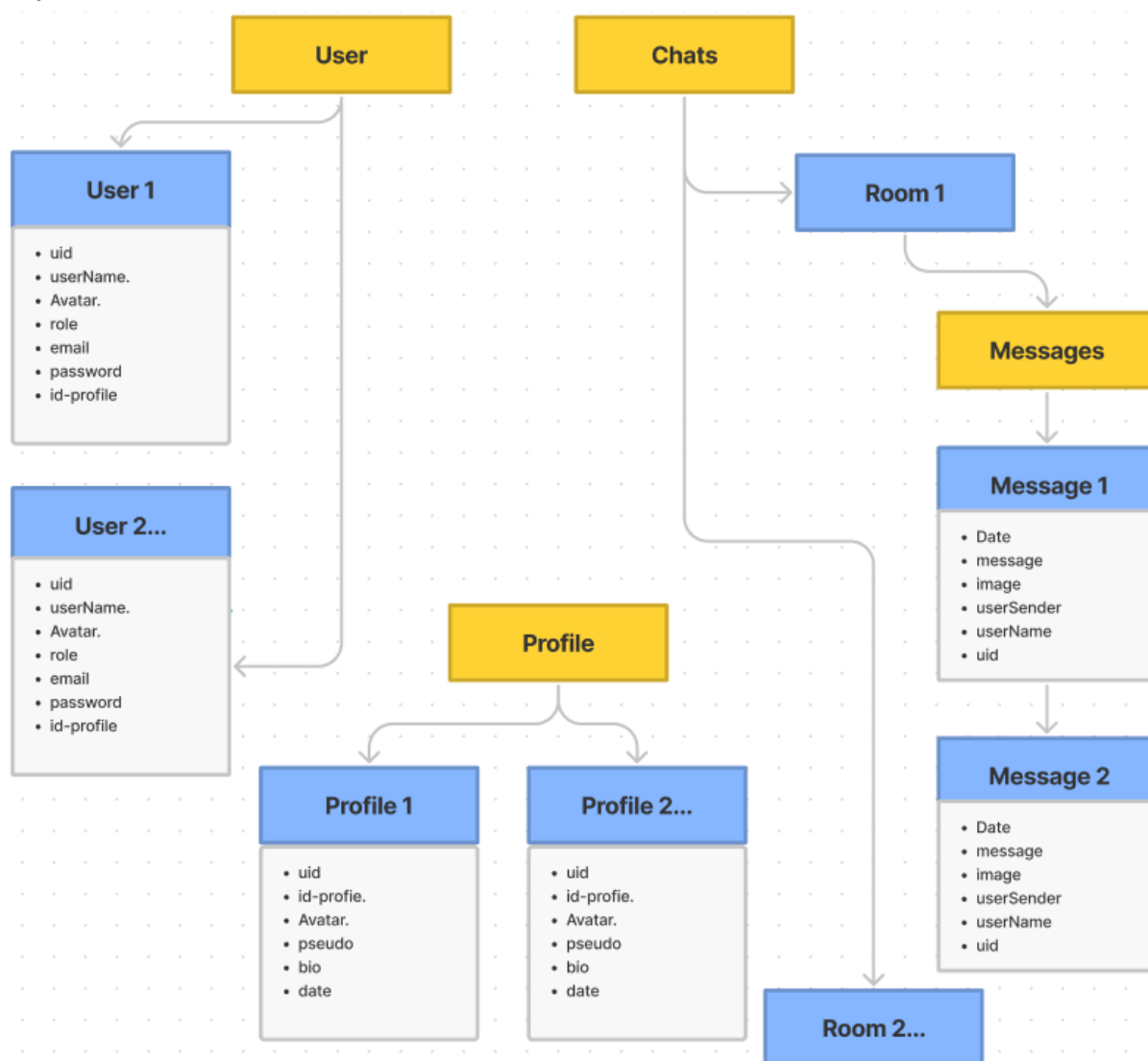
6.7.2 Conception base de données Firestore

Cloud Firestore, permet de stocker les données dans des documents. Ces documents sont stockés dans des collections, qui sont des conteneurs pour mes documents. Les documents prennent en charge de nombreux types de données, à partir de simples chaînes et des nombres, à des objets complexes, imbriqués.

On peut également créer des sous-collections dans des documents et créer des structures de données hiérarchiques qui évoluent au fur et à mesure que la base de données se développe

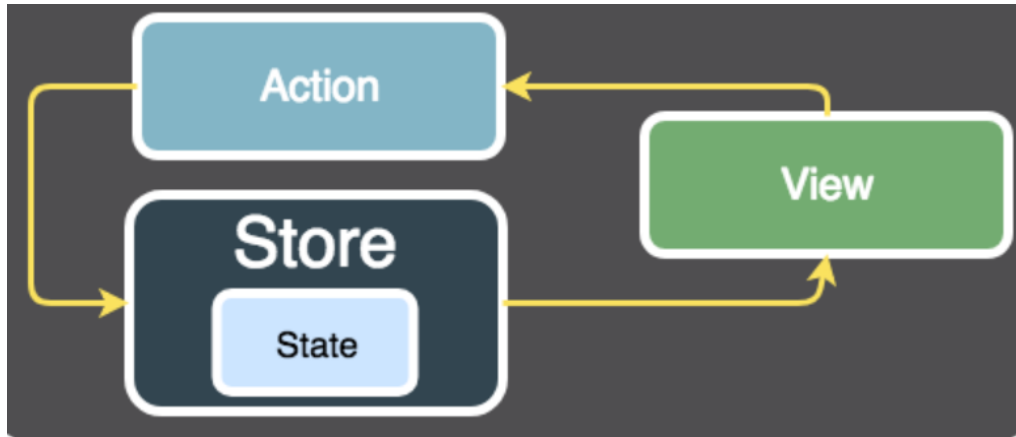
Ainsi, pour mon projet de chats, voici la manière dont j'ai choisi d'organiser ma base de données Firestore :

En jaune les **collections** et en bleu les **documents**.



7. REALISATIONS

7.1 Ma définition de Redux :



Redux m'a permis de gérer un store. Ce dernier représente un objet contenant les données que je souhaite intégrer à mon application, c'est-à-dire **tout mon state global**.

La View crée et envoie une action. Cette action est récupérée par le store. Le store modifie le state de l'application en fonction de l'action reçue. La View détecte les changements du state global et se re-rend.

Afin de réaliser la page « gages », où l'utilisateur peut visualiser ses gages concernant par exemple une action, j'ai donc utilisé cette bibliothèque open-source JavaScript de gestion d'état pour applications.

Dans l'exemple suivant, j'ai déployé mon store à la racine de mon projet dans le fichier index.js de l'application grâce au provider, afin de rendre disponible mes données dans toute l'application :

```
import React from 'react' ;

import {AppRegistry} from 'react-native';
import App from './App';
import {name as appName} from './app.json';

import {Provider} from 'react-redux' ;
import store from './Redux/store' ;

const AppContext = () =>{
  return(
    <Provider store={store}>
      <App />
    </Provider>
  )
}

AppRegistry.registerComponent(appName, () => AppContext);
```

7.1.1 Extrait de code

```

> Connexion      1  const questions = {
> Home           2      verite:[
> Inscription     3      {question: "Dans ce groupe, qui amènerais-tu avec toi sur une île déserte ?", point: 1},
JS index.js      4      {question: "Raconte la plus grande honte que tu as vécue ?", point: 1},
JS style.js      5      {question: "Quel est le rêve le plus étrange que tu aies fait ?", point: 1},
> Participant    6      {question: "Quel est ton plus grand secret ?", point: 1},
JS index.js      7      {question: "Si aujourd'hui était le dernier jour de ta vie, que ferais-tu ?", point: 1},
JS style.js      8      {question: "De 0 à 10. Quelle note donnerais-tu (choisir quelqu'un du groupe) en beauté, sympathie ou intell
> Partie         9      {question: "De 0 à 10. Quelle note te donnerais-tu à toi-même en beauté, sympathie ou intelligence ?", point
JS index.js     10     {question: "Quel est le pire cadeau que tu aies jamais reçu ?", point: 1},
> Questions     11     {question: "Quel est le meilleur cadeau qu'on t'ait jamais fait ?", point: 1},
JS index.js     12     {question: "Si c'était la fin du monde et que tu ne pouvais sauver qu'une seule personne, qui serait-elle ?"
> JS data.js     13     {question: "As-tu déjà nagé nu dans l'eau ?", point: 1},
> SaisieJoueur  14     {question: "Est-ce que quelqu'un t'a déjà trahi ?", point: 1},
JS index.js     15     {question: "Qu'est-ce qui manquera jamais dans ton frigo ?", point: 1},
> SplashScreen  16     {question: "Est-ce que tu pourrais coucher avec quelqu'un pour beaucoup d'argent ?", point: 1},
JS index.js     17     {question: "As-tu déjà envoyé une photo coquine par portable ?", point: 1},
JS splash.js    18     {question: " Si oui, combien ?", point: 1},
JS style.js     19     {question: "As-tu déjà embrassé plusieurs personnes le même jour ?", point: 1},
> constants     20     {question: "De quoi es-tu le/la plus fier.e ?", point: 1},
> FirebaseConte 21     {question: "Te crois-tu attirant.e ?", point: 1},
> images        22     {question: "Si tu pouvais ramener une personne décédée à la vie, qui serait-elle ?", point: 1},
> ios          23     {question: "Si tu pouvais voyager dans le temps, à quelle époque irais-tu ?", point: 1},
> navigations   24     {question: "As-tu menti en jouant à ce jeu ?", point: 1},
JS Navigator.js 25     {question: "Quelle est ta technique de séduction ?", point: 1},
> STRUTURE     26     {question: "Quel le plus grand mensonge que tu aies raconté ?", point: 1},
> SCRIPTS NPM  27     {question: "Quelle est la pire connerie que t'aies jamais faite ?", point: 1},
> ZIP EXPLORER 28     {question: "Quel est ton plus gros regret ?", point: 1},
                29     {question: "Quel est ton plus gros remords ?", point: 1},
                30     {question: "Ton plus gros complexe ?", point: 1},
                31     {question: "T'as déjà été bourré.e ?", point: 1},
                32     {question: " Si oui, raconte-nous ce qu'il s'est passé pendant ta soirée la plus alcoolisée.Tas déjà fumé un
0
L 5, col 1  Espaces: 4  UTF-8  CRLF  {} JavaScript  Go

```

7.1.2 Argumentation

Mes questions sont contenues dans un fichier data.js. Dans une constante (*questions*) j'ai intégré 2 tableaux :

1. Vérité
2. Action

7.2 Utilisation de Redux dans l'application :

- Redux permet de créer un Store qui contient un état, réagissant à des **actions dispatchées**, et auquel on peut souscrire pour être notifié des changements. Ci-dessous une **action** permettant d'afficher les questions. Cette action est un événement qui se produit dans l'application en fonction des entrées de l'utilisateur et qui déclenchent des mises à jour dans l'état :

```

1  import {AFF_QUEST} from './types' ;
2
3  export const affichequestion = (payload) => ({
4      type: AFF_QUEST,
5      payload
6  })
7

```

- **Les réducteurs** sont des fonctions qui prennent la valeur d'un state et une action comme arguments et renvoient un nouveau state résultat. En d'autres termes, (**state, action**) => **newState**. Ci-dessous, le state initial prend comme valeur les données contenues dans data.js, donc toutes les questions action/vérité. La constante « questions » retourne une copie du state initial.

```
import {AFF_QUEST} from '../Actions/types' ;
import data from '../../components/Questions/data' ;

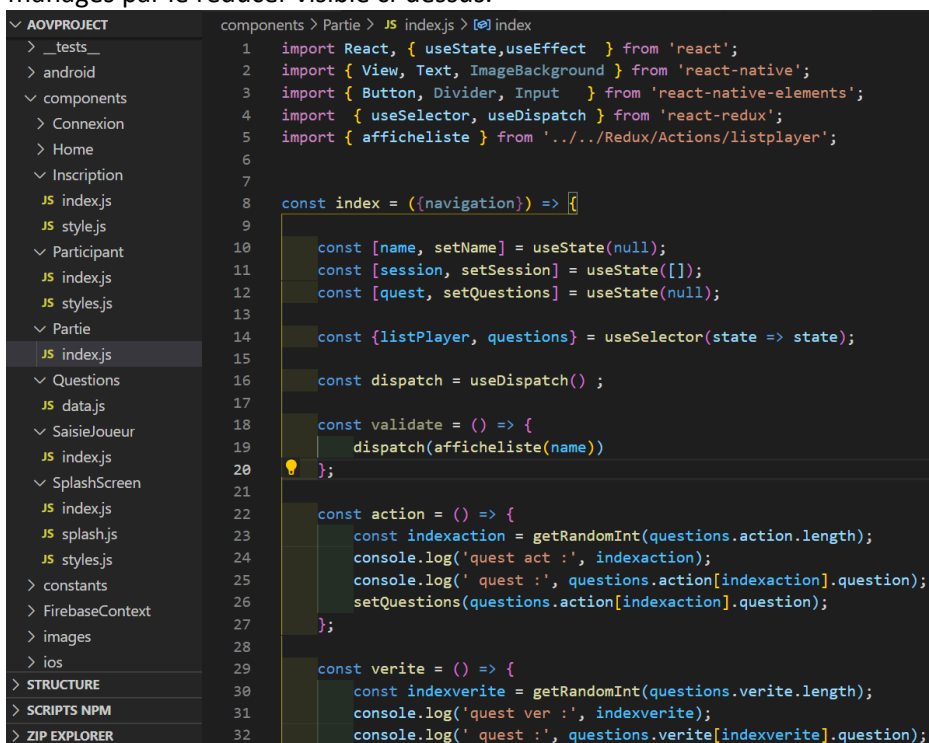
const initStateList = data ;

const questions = (state = initStateList, action) => {
  console.log("action", action)
  return state
}

export default questions ;
```

7.3 Distribution des questions sur la page de jeu :

- Dans un fichier index.js, j'ai importé grâce au hook « *useSelector* », mes questions contenu et managés par le reducer visible ci-dessus.



```
1 import React, { useState,useEffect } from 'react';
2 import { View, Text, ImageBackground } from 'react-native';
3 import { Button, Divider, Input } from 'react-native-elements';
4 import { useSelector, useDispatch } from 'react-redux';
5 import { afficheliste } from '../../Redux/Actions/listplayer';
6
7
8 const index = ({navigation}) => {
9
10   const [name, setName] = useState(null);
11   const [session, setSession] = useState([]);
12   const [quest, setQuestions] = useState(null);
13
14   const {listPlayer, questions} = useSelector(state => state);
15
16   const dispatch = useDispatch() ;
17
18   const validate = () => {
19     dispatch(afficheliste(name))
20   };
21
22   const action = () => {
23     const indexaction = getRandomInt(questions.action.length);
24     console.log('quest act :', indexaction);
25     console.log(' quest :', questions.action[indexaction].question);
26     setQuestions(questions.action[indexaction].question);
27   };
28
29   const verite = () => {
30     const indexverite = getRandomInt(questions.verite.length);
31     console.log('quest ver :', indexverite);
32     console.log(' quest :', questions.verite[indexverite].question);
```

- Dans les constantes action et vérité, j'utilise la fonction getRandomInt défini ci-dessous, ce qui me permet de générer de manière aléatoire les questions :

```
const getRandomInt = (max) => {
  return Math.floor(Math.random() * max);
};
```

8. PRESENTATION DU JEU D'ESSAI DE LA FONCTIONNALITE LA PLUS REPRESENTATIVE

8.1 Fonctionnalité testée

La fonctionnalité testée ajoute puis sauvegarde un utilisateur dans la base de données, et dirige l'utilisateur sur la page de jeu.

8.2 Description des scénarios et résultat des tests

Jeu d'essai	Description	Résultat attendu	Résultat obtenu
Données d'inscription conforme	L'adresse gmail est valide et identique à celle entré dans le formulaire d'inscription.	Redirection vers page de jeu.	OK
Données d'inscription non-conformes	Le champ adresse gmail n'est pas saisi	Le bouton « valider » ne devrait pas être activé car tous les champs doivent être remplis. Maintien sur la page d'authentification.	OK
Données d'inscriptions FACEBOOK conforme	Le compte facebook est valide et identique à celui entré dans le formulaire d'inscription.	Redirection vers page de jeu.	OK
Données d'inscriptions FACEBOOK non-conforme	Le champ du compte facebook n'est pas saisi	Le bouton « valider » ne devrait pas être activé car tous les champs doivent être remplis. Maintien sur la page d'authentification.	OK
Récupération et affichage des données	Les données sont renvoyées vers le front pour identifier les joueurs sur la page de jeu. Par ailleurs elles seront aussi disponibles pour le back afin de les consulter, les supprimer ou les modifier.	Affichage des données sur la page de jeu de l'utilisateur concerné	OK

8.3 Conclusion

Le plan de test de la fonction d'ajout d'un nouvel utilisateur est un exemple de test réalisé pour chaque fonctionnalité de Action ou vérité. Le système de désactivation du bouton permet de prévoir rapidement les erreurs humaines et empêchent ainsi certains dysfonctionnements.

Les tests ont été concluants lors de la phase de développement ainsi que lors de la phase de test.

9. DESCRIPTION DE LA VEILLE SUR LES VULNERABILITES DE SECURITE

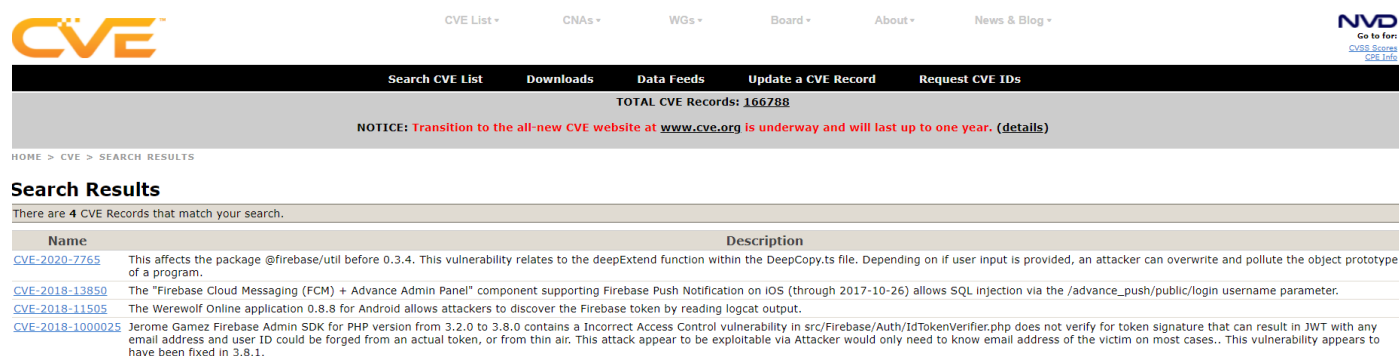
Le CVE common vulnerabilities exposure est une base de données, internationale qui enregistre les découvertes de failles de sécurité informatiques, vulnérabilités et risques chronologiquement.

A travers la fiche CVE de la technologie Firebase ci-dessus, j'ai pu constater une liste de faille de type XSS.

Le XSS ou Cross-Site Scripting est un type de faille de sécurité où un pirate peut par l'intermédiaire de script javascript, prendre le contrôle d'un navigateur. L'objectif de l'attaquant est d'obtenir des données confidentielles et sensible.

Les portes d'entrées à ce type de faille sont les formulaires, les inputs, les url, partout où un utilisateur peut communiquer dans une application.

En 2021, les chercheurs d'Avast ont récemment découvert plus de 19 000 instances ouvertes de Firebase sur Internet. Parmi eux, 10 % étaient ouverts, ce qui signifie que leurs données étaient exposées à des utilisateurs non authentifiés. Toutes les bases de données ne contiennent pas d'informations sensibles ou personnellement identifiables.



The screenshot shows the CVE website interface. At the top, there's a navigation bar with links like 'CVE List', 'CNAs', 'WGs', 'Board', 'About', and 'News & Blog'. Below this is a search bar and a 'Search CVE List' button. A banner indicates 'TOTAL CVE Records: 166788' and a notice about the transition to the new CVE website at www.cve.org. The 'Search Results' section shows 4 records matching the search. The first record, CVE-2020-7765, is highlighted. Its description states: 'This affects the package @firebase/util before 0.3.4. This vulnerability relates to the deepExtend function within the DeepCopy.ts file. Depending on if user input is provided, an attacker can overwrite and pollute the object prototype of a program.'

Name	Description
CVE-2020-7765	This affects the package @firebase/util before 0.3.4. This vulnerability relates to the deepExtend function within the DeepCopy.ts file. Depending on if user input is provided, an attacker can overwrite and pollute the object prototype of a program.
CVE-2018-13850	The "Firebase Cloud Messaging (FCM) + Advance Admin Panel" component supporting Firebase Push Notification on iOS (through 2017-10-26) allows SQL injection via the /advance_push/public/login username parameter.
CVE-2018-11505	The Werewolf Online application 0.8.8 for Android allows attackers to discover the Firebase token by reading logcat output.
CVE-2018-1000025	Jerome Gamez Firebase Admin SDK for PHP version from 3.2.0 to 3.8.0 contains a Incorrect Access Control vulnerability in src/Firebase/Auth/IdTokenVerifier.php does not verify for token signature that can result in JWT with any email address and user ID could be forged from an actual token, or from thin air. This attack appear to be exploitable via Attacker would only need to know email address of the victim on most cases.. This vulnerability appears to have been fixed in 3.8.1.

10. DESCRIPTION D'UNE SITUATION DE TRAVAIL AYANT NECESSITE UNE RECHERCHE SUR UN SITE ANGLOPHONE OU FRANCOPHONE

10.1 Description du besoin

Ajouter Firebase à mon projet.

10.2 Description de la recherche

10.2.1 Moteur de recherche

J'ai utilisé Google Chrome pour faire ma recherche.

10.2.2 Mots clés

« *installer firebase sur un projet react-native* »

10.2.3 Résultat de la recherche (liste sites)

J'ai cliqué sur plusieurs sites qui semblaient correspondre à ma requête sur la première page :

- <https://firebase.google.com/docs/android/setup?authuser=1#create-firebase-project><https://firebase.google.com/docs/storage/web/create-reference>
- <https://openclassrooms.com/fr/courses/4872916-creez-un-backend-scalable-et-performant-sur-firebase/4982766-integrez-firebase-dans-une-application-android>

10.2.4 Critères de sélection et choix du site

J'ai choisi le site en fonction de trois critères :

- La fiabilité de l'information (documentation officielle)
- La possibilité d'appliquer la solution pour le web (et non Android ou Swift)
- La date de la publication de l'article (le plus récent possible)

Le site qui répondait le mieux à ces critères était la documentation officielle de Firebase qui donnait des exemples de codes

10.2.5 Extrait du site

actionverite - Présentation - Con

Add Firebase to your Android pr

Intégrez Firebase dans une applic

+

← → ↺ 🔒

firebase.google.com/docs/android/setup?authuser=1#create-firebase-project

📄 🔍 🗑️ ☆ 🛠️ 🧩 🌐 En pause

🔥

Firebase

Products ▾

Use Cases

Pricing

Docs ▾

Community ▾

Support

🔍

Recherche

English ▾

Go to console

⋮

Fundamentals

Get started with Firebase

▾ Add Firebase to an app

Apple platforms (iOS+)

Android

Web

▸ Add Firebase to a game

Add Firebase to a server

▸ Use Firebase with a framework

Set up projects programmatically

Manage your Firebase projects

Platforms and frameworks

Protect project resources with App Check

Prototype and test with Emulator Suite

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud SQL

Cloud Storage

Cloud Vision

Cloud Video Intelligence

Cloud Web Security

Cloud Workflows

Cloud X

Cloud Z

Cloud Y

Cloud W

Cloud V

Cloud U

Cloud T

Cloud S

Cloud R

Cloud Q

Cloud P

Cloud O

Cloud N

Cloud M

Cloud L

Cloud K

Cloud J

Cloud I

Cloud H

Cloud G

Cloud F

Cloud E

Cloud D

Cloud C

Cloud B

Cloud A

Firestore

Cloud Storage

Cloud Firestore

Cloud Functions

Cloud Messaging

Cloud Scheduler

Cloud Run

Cloud

Conditions préalables

- Installer ou mettre à jour Android Studio est à sa dernière version.
- Assurez-vous que votre projet répond aux exigences suivantes :
 - Cible API niveau 19 (KitKat) ou supérieur
 - Utilise Android 4.4 ou supérieur
- Utilise Jetpack (AndroidX), qui comprend la réalisation de ces exigences de version :
 - `com.android.tools.build:gradle v3.2.1` ou plus tard
 - `compileSdkVersion 28` ou plus tard
- Mettre en place un dispositif physique ou utiliser un émulateur pour exécuter votre application. Notez que Firebase avec une dépendance SDKs sur les services Google Play nécessitent l'appareil ou émulateur pour avoir des services Google Play installés.
- Connectez - vous à Firebase en utilisant votre compte Google.
Si vous ne possédez pas déjà un projet Android et que vous voulez juste essayer un produit Firebase,