

Cahier des charges

Projet : Join A Match

Demandeur: Pedro BRIGNON

Priorité : NC

| N° Version | Date | Commentaire | Modifié par |
|------------|----------|-------------|----------------------------|
| 1.0 | 17/02/23 | | Vincent, Julia, Nicolas |
| 1.5 | 17/03/23 | | Vincent, Julia, Nicolas |
| | | | |

Table des matières

| 1. | Contexte du projet | | |
|----|--------------------|--|--|
| | | | |
| | I. Client ciblé | | |

| | II. Analyse et concurrences | . 3 |
|----|--------------------------------|-----|
| | III. Plus-value | . 3 |
| | IV. FFOM | . 3 |
| | A. Forces : | 3 |
| | B. Faiblesses : | 3 |
| | C. Opportunités : | 3 |
| | D. Menaces : | 4 |
| 2. | Réalisation | .4 |
| | | _ |
| | I. But du projet | |
| | II. Équipe | |
| | III. Temps de réalisation | . 4 |
| 3. | Spécifications techniques | .5 |
| | I. Front-End | _ |
| | | |
| | A. Architecture: | |
| | II. Back-End | |
| | A. Architecture: | |
| | III. Sécurité | |
| | IV. Plateformes de déploiement | |
| | V. Charte graphique | . 7 |
| | A. Couleurs | 7 |
| | B. Polices d'écriture | |
| | VI. Maquette Navigation Flow | . 8 |
| | A. Page « lobby » (Accueil) | 10 |
| | B. Page d'un jeu | 12 |
| | C. Page profil utilisateur | 15 |
| | D. Page communauté | 18 |
| | E. Page de recherche | 20 |

1. Contexte du projet

I. Client ciblé

Nos clients cibles dans ce projet sont des joueurs de jeu compétitif (FPS, MOBA, etc...) Dans une tranche d'âge de 16 ans et plus.

II. Analyse et concurrences

Notre seul concurrent sur le marché européen est un site web français, ledit site est peu intuitif avec un design moyen, le site n'est plus maintenu et la communauté est inactive depuis l'été 2021.

III. Plus-value

Regroupement de solutions (Twitch, Discord, etc...) pour faciliter la recherche de coéquipier et/ou de communauté de joueurs.

IV. FFOM

A. Forces:

Interface intuitive avec une navigation simple et claire, nous pouvons retrouver l'ensemble des informations d'un joueur sur une seule et même page (jeux, statistiques, etc...)

B. Faiblesses:

L'application est dépendante des différents serveurs externes utilisés (ex : Riot, Twitch, etc...)

C. Opportunités :

Le marché dans ce secteur est très peu exploité.



D. Menaces:

Aucune menace(s) reconnue, le seul risque est une qu'une plateforme comme Twitch fasse un concept similaire directement intégré à leur application.

2. Réalisation

I. But du projet

Le but du projet est de créer une plateforme regroupant les joueurs sur des jeux compétitifs.

De cette manière, les joueurs ont accès à leurs statistiques de jeu et en même temps la possibilité de trouver les coéquipiers en recherchant via leur pseudo d'inscription à la plateforme ou via son pseudo en jeu.

II. Équipe

En charge de ce projet, l'équipe est composée de 3 développeurs full stack :

- Julia JORGE
- Nicolas BAUER
- Vincent BECKER

III. Temps de réalisation



Une deadline est fixée à Aout 2023 (7 mois).

3. Spécifications techniques

Une Dockerisation de la partie Front-end ainsi que les micro-services sera effectuée pour une question de simplicité au moment du déploiement de l'application et une meilleure homogénéité du versioning tout au long de la période de développement.

I. Front-End

La partie front-end (client-side) de l'application est écrite avec Vue 3 (composition), nous avons choisi ce Framework car il est performant, avec une bonne architecture et rapide, le fait qu'une personne du groupe connaisse bien ce Framework est un plus.

Nous ajoutons Pinia au sein du projet pour la gestion des données pour un fonctionnement plus fluide et réactif. Cela aidera aussi à réduire les requêtes au serveur pour une question « éco responsable ».

En ce qui concerne le style, nous allons utiliser la bibliothèque Vuetify (v3) pour les composants tels que les boutons, les alertes etc...

Notre choix s'est porté sur cette bibliothèque car elle est nouvelle pour Vue 3 et ses components sont modernes, dynamiques et plus dans l'air du temps que Bootstrap par exemple.

A. Architecture:

Squelette d'un projet sous Vue 3 en utilisant un système de « Stores » (Pinia), cette technologie a été préféré pour sa rapidité d'exécution ainsi que la maintenabilité de l'application dans le temps (architecture moderne).

II. Back-End



La partie back-end (server-side) de l'application est rédigée en PHP à l'aide du framework « Symfony » (v6) couplé à API Platform pour gérer la connexion et la communication entre le client et le serveur.

Symfony a été choisi pour construire l'API mère car ce Framework est le plus complet du marché.

Quant aux micro-services, ils seront construits à l'aide du Framework « Lumen », nous avons l'avons choisi car il est bien plus léger que Symfony et plus adapté à l'architecture microservice.

La base de données quant à elle est en SQL sur un serveur MySQL.

Nous l'avons choisi pour ses performances, car il est optimisé pour travailler avec des bases de données relationnels et gérer des grandes quantités de données. Il offre également des fonctionnalités de sécurité robustes pour protéger les données stockées. Un système de « rôle » est également implémenté dans le langage ce qui permet à des utilisateurs d'accéder à des données spécifiques en fonction de leur(s) rôle(s) attribué(s). Nous choisissons le service OAuth en passant par des Token/JWT pour inscrire et authentifier les utilisateurs à l'application.

A. Architecture:

Construction de différentes API sous format de micro-services.

Un micro-service qui se chargera de la partie utilisateur avec Lumen pour une plus grande sécurité ainsi qu'une meilleure maintenabilité.

Un deuxième qui collectera les données des différentes API externes (Riot, Twitch, etc...) lui aussi avec Lumen, ce choix est fait pour garder un contrôle total sur L'application et si un problème survient avec l'api Riot, cela n'impactera pas l'application.

Pour le troisième micro-service, il aura comme mission la gestion du contenu interne de l'application (Symfony) et de récupérer les données des deux microservices pour envoi au Front.

III. Sécurité

Pour garantir la sécurité des données utilisateur, nous utilisons le JSON Web Token pour authentifier les utilisateurs.

IV. Plateformes déploiement

Le format de l'application sera au format web, car notre clientèle cible sont des joueurs PC.

V. Charte graphique

En ce qui concerne le style de base du site, nous allons nous servir d'une palette de couleurs bleu / violet (voir nuancier ci-dessous).

A. Couleurs

Thème clair:

| ADACEC | 4352DB | 6B89DB | FFFFFF |
|---------------------|--------|--------|--------|
| Maximum Blue Purple | | | White |

Thème sombre:



6564DB 232ED1 101D42 0D1317

Slate Blue Persian Blue Oxford Blue Rich Black FOGRA 29

B. Polices d'écriture

- Gros titres : Rubik Black 900 italic

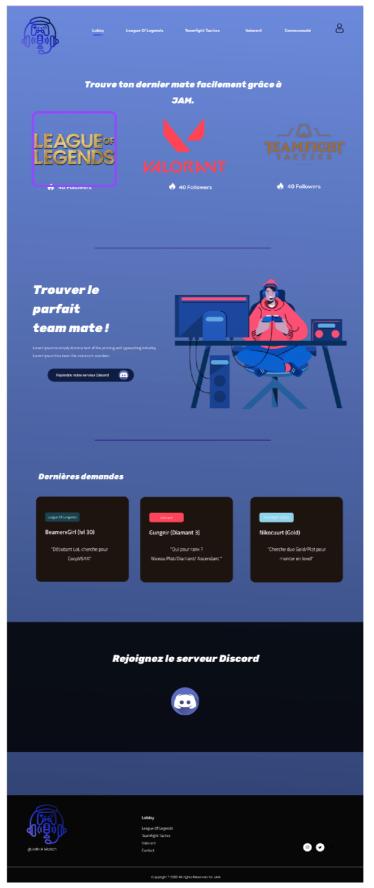
- Sous-titres: Rubik Medium 500

- Nav-Bar: Rubik Regular 400

- Textes : Titillium Regular 400

VI. Maquette Navigation Flow

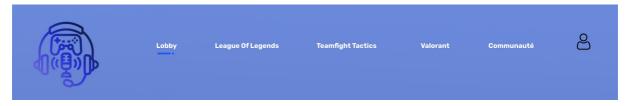






A. Page « lobby » (Accueil)

Voici la barre de navigation :

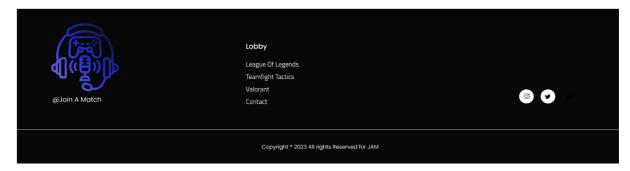


Explication des différents liens :

- Le logo : En cliquant dessus, renvoie sur la page lobby (page d'accueil) Lobby : Permets de se rendre sur la page lobby.
- League Of Legends : Emmène sur la page dédiée au jeu : League Of Legends
- Teamfight Tactics : Emmène sur la page dédiée au jeu : Teamfight Tactics
- Valorant : Emmène sur la page dédiée au jeu : Valorant
- Icône « Utilisateur » : Nous emmène sur la page de compte, si l'utilisateur n'est pas connecté, il sera redirigé automatiquement sur la page de connexion.

Le footer, servira à montrer l'identité de la plateforme, on trouvera le logo de la plateforme, son plan pour se rendre dans les différentes pages.

On peut trouver également, le copyright © ainsi que les icônes des réseaux sociaux « Instagram et Twitter »



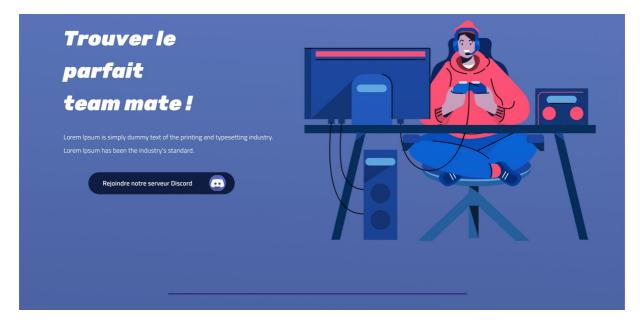
Lorsque que l'utilisateur arrive sur notre plateforme, il atterri sur la page « Lobby » qui représente notre page d'accueil.



À l'arrivée, nous montrons directement les jeux tendances pour trouver rapidement des coéquipiers sur ces derniers.



La section d'après, indique à l'utilisateur qu'il est recommandé de se connecter sur le Discord pour trouver des coéquipiers et pour bénéficier des fonctionnalités de JAM







Avant d'arrivé au pied de page, l'utilisateur aura la possibilité de rejoindre le serveur Discord de l'application en cliquant sur l'icône Discord se trouvant à la suite des dernières demandes de jeu (voir image ci-dessous).



B. Page d'un jeu



En cliquant sur le lien « League Of Legend », nous arrivons sur sa page dédiée comme le montre l'image ci-dessus.

Le thème de la page sera géré en fonction des couleurs du jeu sélectionné, comme dans cet exemple pour la page de LOL.

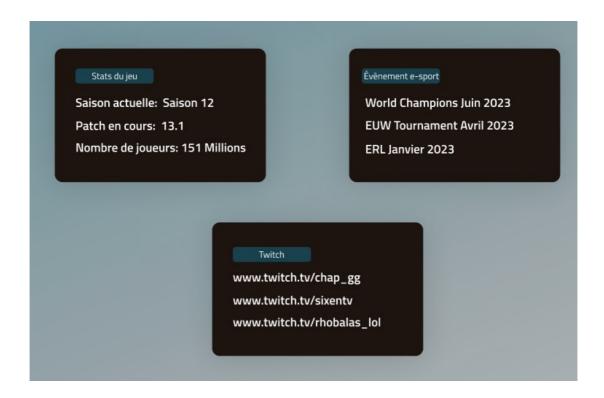
En milieu de page, se trouve un bouton d'accès à notre section de recherche pour trouver des joueurs, en lien avec la page sur laquelle l'utilisateur se trouve, ceci afin d'optimiser la rapidité de recherche.



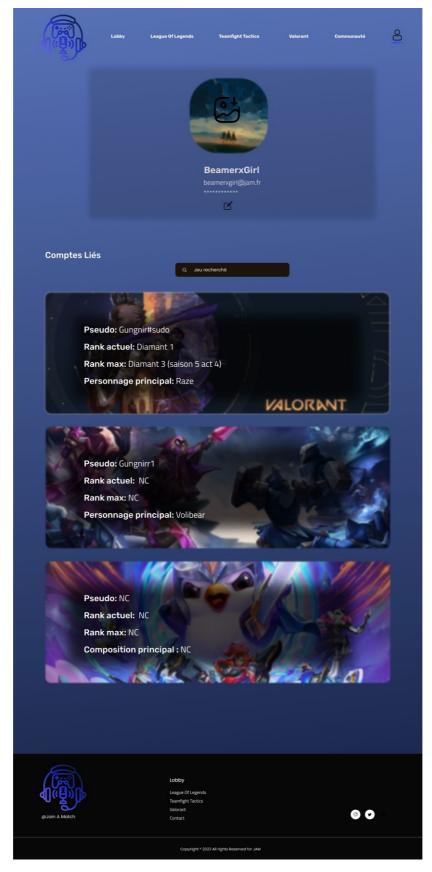
On peut retrouver sur la page quatre différentes cases contenant respectivement :

- Les statistiques du jeu
- Les évènements à venir
- Les chaînes Twitch populaire au jeu





C. Page profil utilisateur

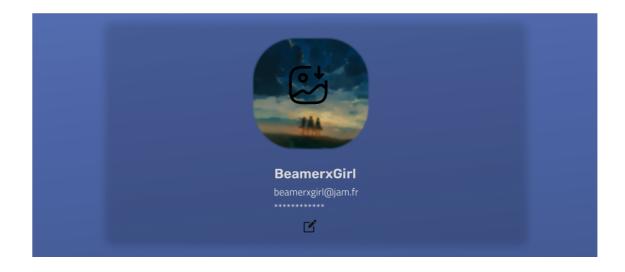


Pour accéder à cette page, l'utilisateur <u>doit</u> être connecté à la plateforme, sinon il doit s'inscrire.

Une fois connecté, l'utilisateur sera redirigé sur la page « lobby » (voir page 8). Pour accéder à son profil, il faudra se rendre sur la barre de navigation et cliquer sur l'icône « profil » (voir page 9)

Sur la première partie de cette page, l'utilisateur à la possibilité de changer son image de profil en cliquant sur l'icône « éditer » se trouvant sur l'image. Il peut voir également ses informations tels que son « mail » et son « mot passe »

(caché). Il a également la possibilité de changer modifier ses informations via l'icône « éditer » se trouvant juste en dessous.

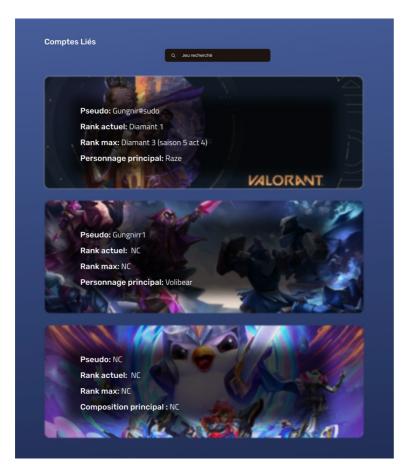


À la suite, l'utilisateur pourra observer ses statistiques de jeu tels que :

- Son pseudo en jeu.
- Son rang actuel.



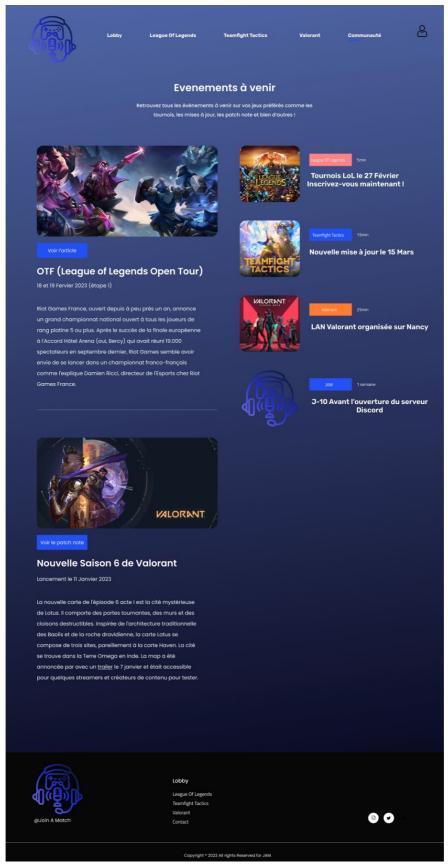
- Son rang maximal atteint durant la saison en cours du jeu en question.
- Et le personnage ou la composition qu'il joue le plus souvent.



Le bouton de recherche sera utilisé pour rechercher un jeu.

D. Page communauté



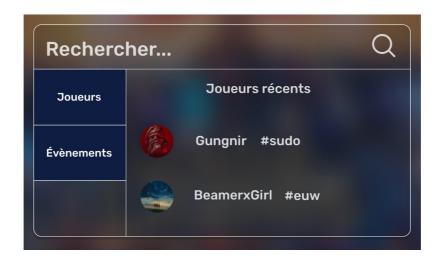


En arrivant sur la page « communauté » via le lien dans la barre de navigation « communauté »



Nous accédons à la page de la communauté de la plateforme, elle sert à montrer les évènements qui s'y présenteront comme des tournois ou simplement des annonces comme des nouvelles saisons arrivant sur les jeux.

E. Composant de recherche



Le « composant de recherche » servira justement à rechercher des joueurs et des évènements.

Le composant s'ouvre n'importe où sur le site en cliquant sur un bouton de recherche.

Par défaut, la recherche affiche les joueurs recherché récemment.

En cliquant sur « Joueurs » on pourra rechercher des joueurs.

En cliquant sur « Évènements » on pourra rechercher des évènements.

Lu et approuvé :

| Le : | À : | Signature : |
|------|-----|-------------|
| | | African |

