Match My Game

Groupe 16 :

Charlery Malcolm

Mahassadi Antoine

Diouf Amadou Makhtar

Kuicheu Berol

Benguigui Avidan

Notre équipe

- Malcolm Charlery : Chef de projet / DevOps
- Avidan Benguigui : Gestion de l'architecture
- Kuicheu Berol: Développeur front-end
- Antoine Mahassadi : Développeur front-end
- Makhtar Diouf : Développeur back-end

Sommaire

- Problématique
- Solution proposée
- Architecture
- Partie front-end
- Partie back-end
- Perspective d'avenir

Notre problématique

Comment former facilement une équipe pour jouer à un jeu vidéo avec des alliés procédant les même statiques de jeu que moi ?

Notre problématique

Comment former facilement une équipe pour jouer à un jeu vidéo avec des alliés procédant les même statiques de jeu que moi ?

Notre solution

MatchMyGame, la plateforme de mise en relation entre des joueurs souhaitant jouer en multijoueur !

Architecture et besoins fonctionnels

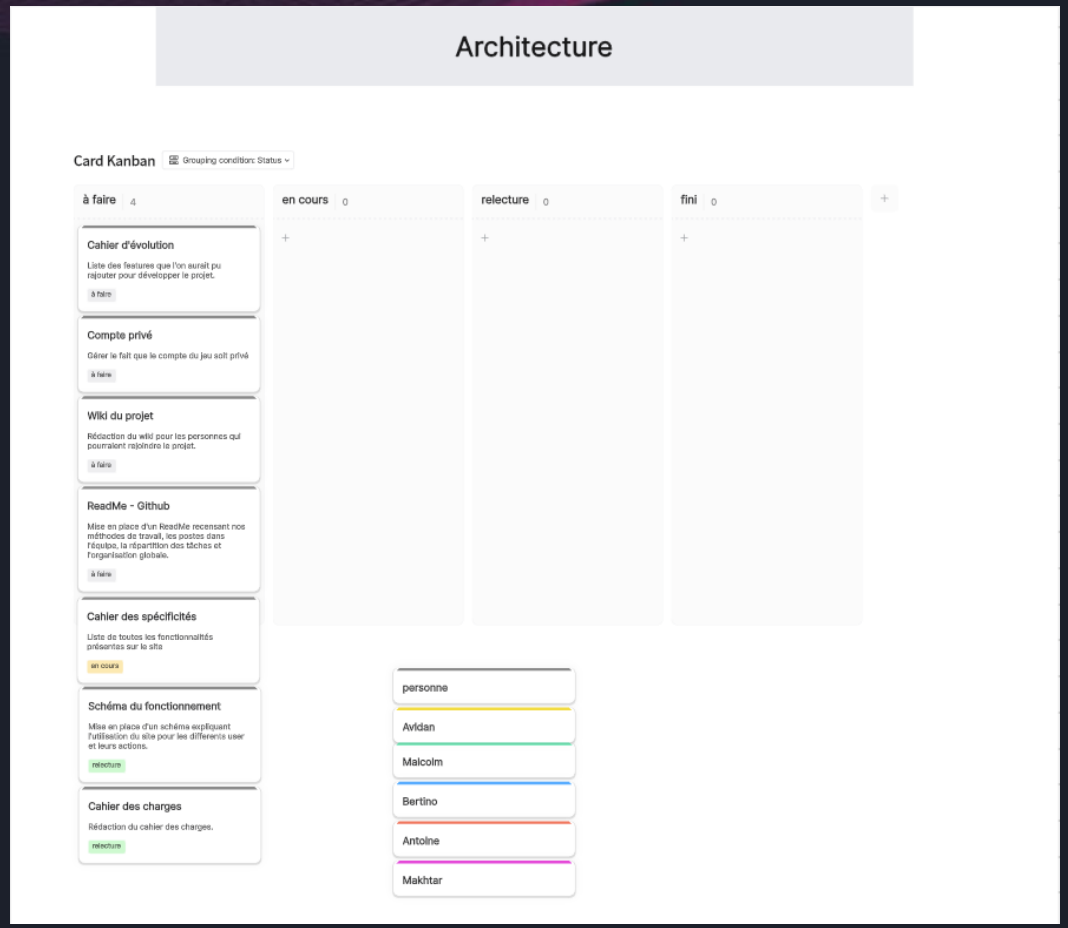
Front-end : VueJS

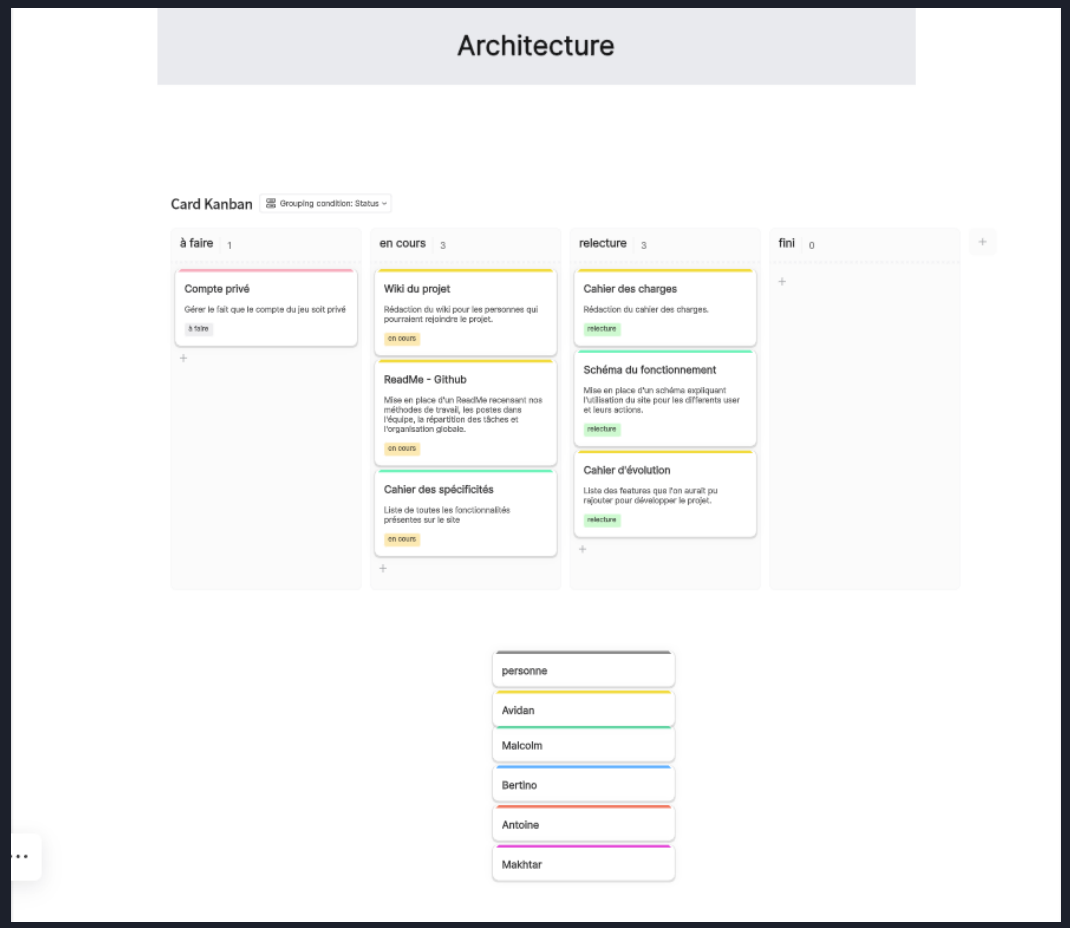
Back-end : Laravel

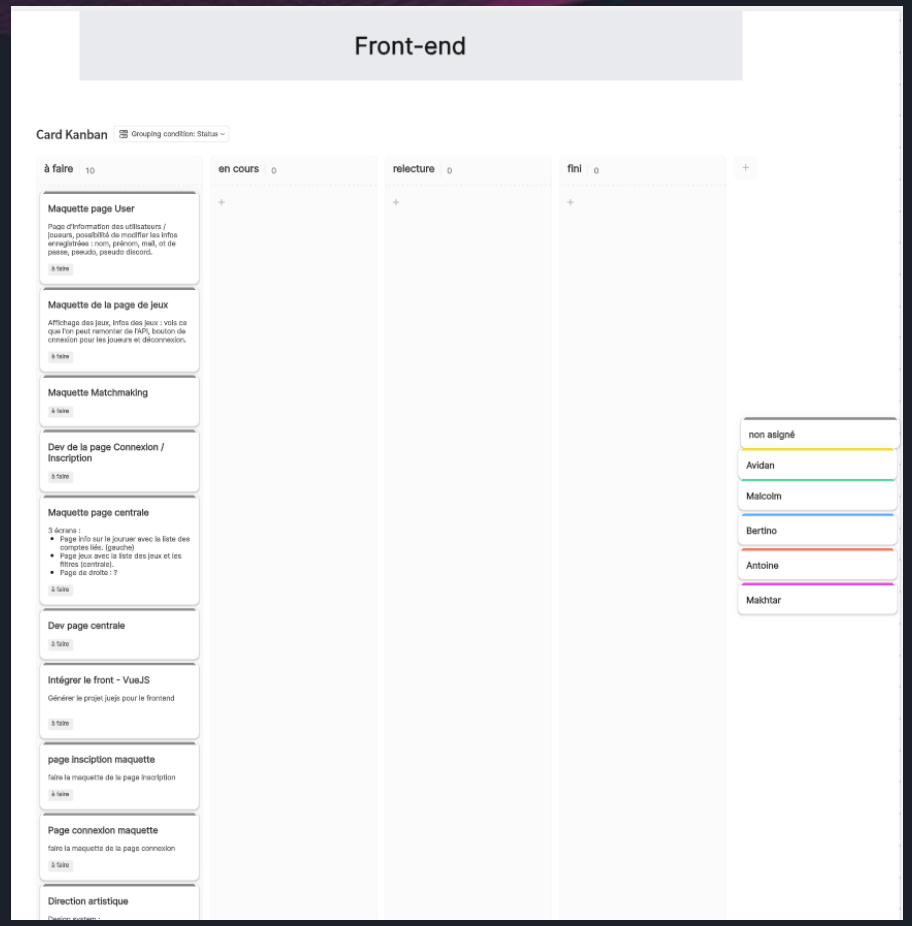
Besoins fonctionnels : Cahier des charges

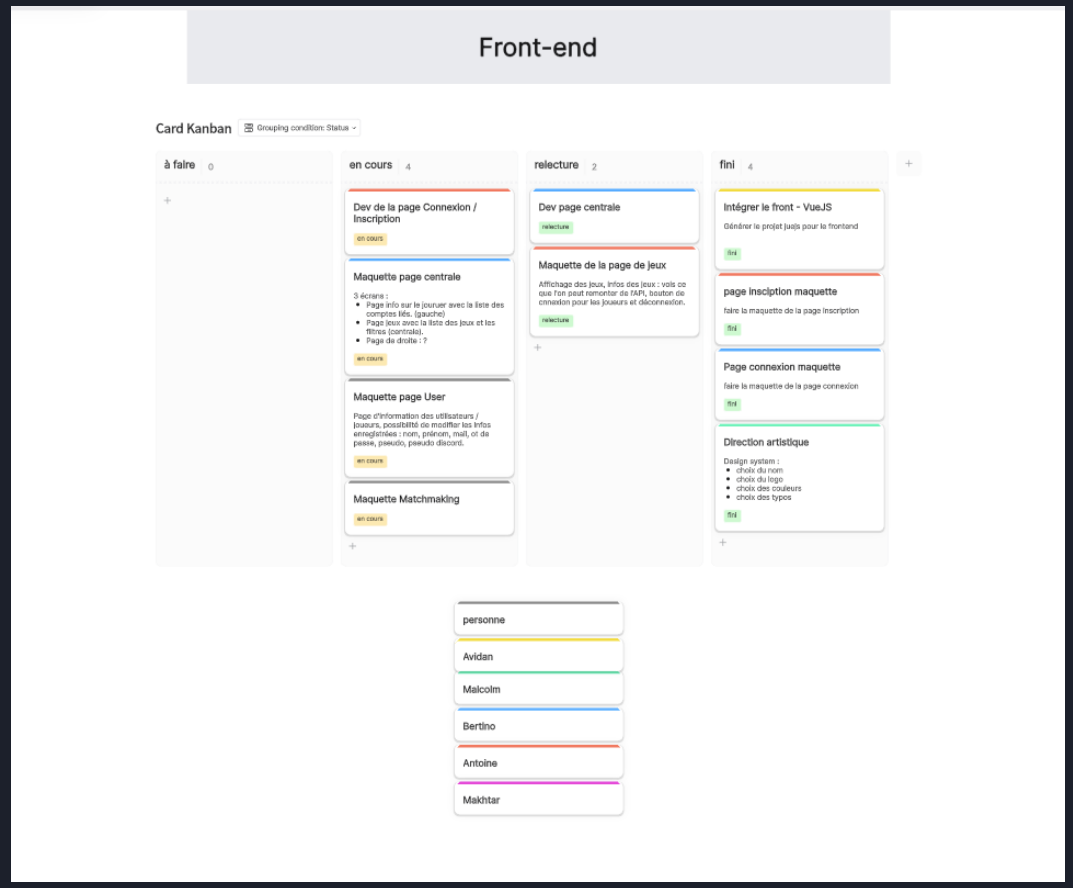
Notre organisation : README

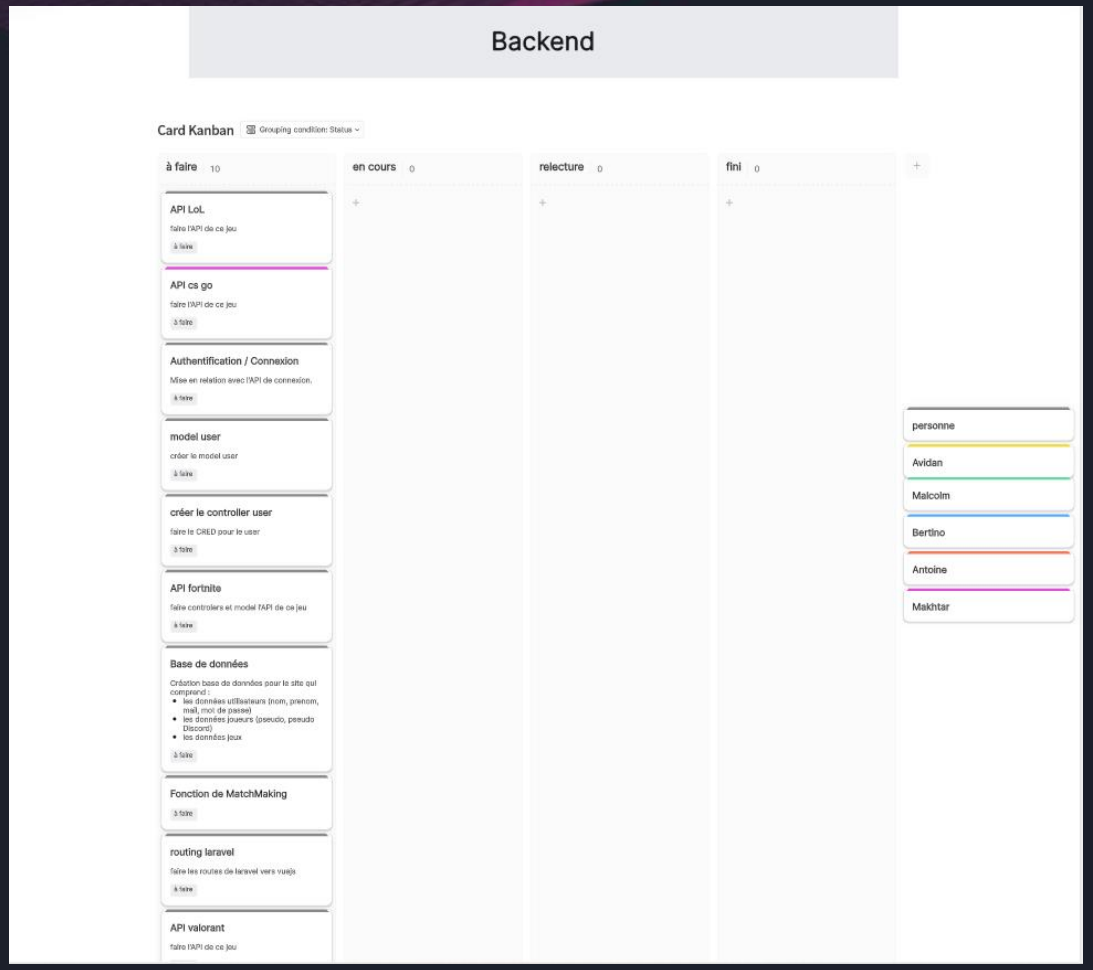
Nos tâches


Architecture
Card Kanban
Grouping condition: Status
à faire 4
en cours 0
relecture 0
fini 0
Cahier d'évolution
Liste des features que l'on aurait pu rajouter pour développer le projet.
à faire
Compte privé
Gérer le fait que le compte du jeu soit privé
à faire
Wiki du projet
Rédaction du wiki pour les personnes qui pourraient rejoindre le projet.
à faire
ReadMe - Github
Mise en place d'un ReadMe recensant nos méthodes de travail, les postes dans l'équipe, la répartition des tâches et l'organisation globale.
à faire
Cahier des spécificités
Liste de toutes les fonctionnalités présentes sur le site
en cours
Schéma du fonctionnement
Mise en place d'un schéma expliquant l'utilisation du site pour les différents user et leurs actions.
relecture
Cahier des charges
Rédaction du cahier des charges.
relecture
personne
Avidan
Malcolm
Bertino
Antoine
Makhtar


Architecture
Card Kanban
Grouping condition: Status
à faire 1
en cours 3
relecture 3
fini 0
Compte privé
Gérer le fait que le compte du jeu soit privé
à faire
Wiki du projet
Rédaction du wiki pour les personnes qui pourraient rejoindre le projet.
en cours
ReadMe - Github
Mise en place d'un ReadMe recensant nos méthodes de travail, les postes dans l'équipe, la répartition des tâches et l'organisation globale.
en cours
Cahier des spécificités
Liste de toutes les fonctionnalités présentes sur le site
en cours
Cahier des charges
Rédaction du cahier des charges.
relecture
Schéma du fonctionnement
Mise en place d'un schéma expliquant l'utilisation du site pour les différents user et leurs actions.
relecture
Cahier d'évolution
Liste des features que l'on aurait pu rajouter pour développer le projet.
relecture
personne
Avidan
Malcolm
Bertino
Antoine
Makhtar


Front-end
Card Kanban
Grouping condition: Status
à faire 10
en cours 0
relecture 0
fini 0
Maquette page User
Page d'information des utilisateurs / joueurs, possibilité de modifier les infos enregistrées : nom, prénom, mail, ot de passe, pseudo, pseudo discord.
à faire
Maquette de la page de jeux
Affichage des jeux, infos des jeux : vois ce que l'on peut remonter de l'API, bouton de connexion pour les joueurs et déconnexion.
Maquette Matchmaking
Dev de la page Connexion / Inscription
Maquette page centrale
3 écrans :
Page info sur le jouruer avec la liste des comptes liés. (gauche)
Page jeux avec la liste des jeux et les filtres (centrale).
Page de droite : ?
Dev page centrale
Intégrer le front - VueJS
Générer le projet juejs pour le frontend
page insciption maquette
faire la maquette de la page inscription
Page connexion maquette
faire la maquette de la page connexion
Direction artistique
non asigné
Avidan
Malcolm
Bertino
Antoine
Makhtar


Front-end
Card Kanban
Grouping condition: Status
à faire 0
en cours 4
relecture 2
fini 4
Dev de la page Connexion / Inscription
en cours
Maquette page centrale
3 écrans :
Page info sur le jouruer avec la liste des comptes liés. (gauche)
Page jeux avec la liste des jeux et les filtres (centrale).
Page de droite : ?
Maquette page User
Page d'information des utilisateurs / joueurs, possibilité de modifier les infos enregistrées : nom, prénom, mail, ot de passe, pseudo, pseudo discord.
Maquette Matchmaking
Dev page centrale
relecture
Maquette de la page de jeux
Affichage des jeux, infos des jeux : vois ce que l'on peut remonter de l'API, bouton de connexion pour les joueurs et déconnexion.
Intégrer le front - VueJS
Générer le projet juejs pour le frontend
fini
page insciption maquette
faire la maquette de la page inscription
Page connexion maquette
faire la maquette de la page connexion
Direction artistique
Design system :
choix du nom
choix du logo
choix des couleurs
choix des typos
personne
Avidan
Malcolm
Bertino
Antoine
Makhtar


Backend
Card Kanban
Grouping condition: Status
à faire 10
en cours 0
relecture 0
fini 0
API LoL
faire l'API de ce jeu
à faire
API cs go
faire l'API de ce jeu
à faire
Authentification / Connexion
Mise en relation avec l'API de connexion.
à faire
model user
créer le model user
à faire
créer le controller user
faire le CRED pour le user
à faire
API fortnite
faire controlers et model l'API de ce jeu
à faire
Base de données
Création base de données pour le site qui comprend :
les données utilisateurs (nom, prenom, mail, mot de passe)
les données joueurs (pseudo, pseudo Discord)
les données jeux
à faire
Fonction de MatchMaking
à faire
routing laravel
faire les routes de laravel vers vuejs
à faire
API valorant
faire l'API de ce jeu
personne
Avidan
Malcolm
Bertino
Antoine
Makhtar

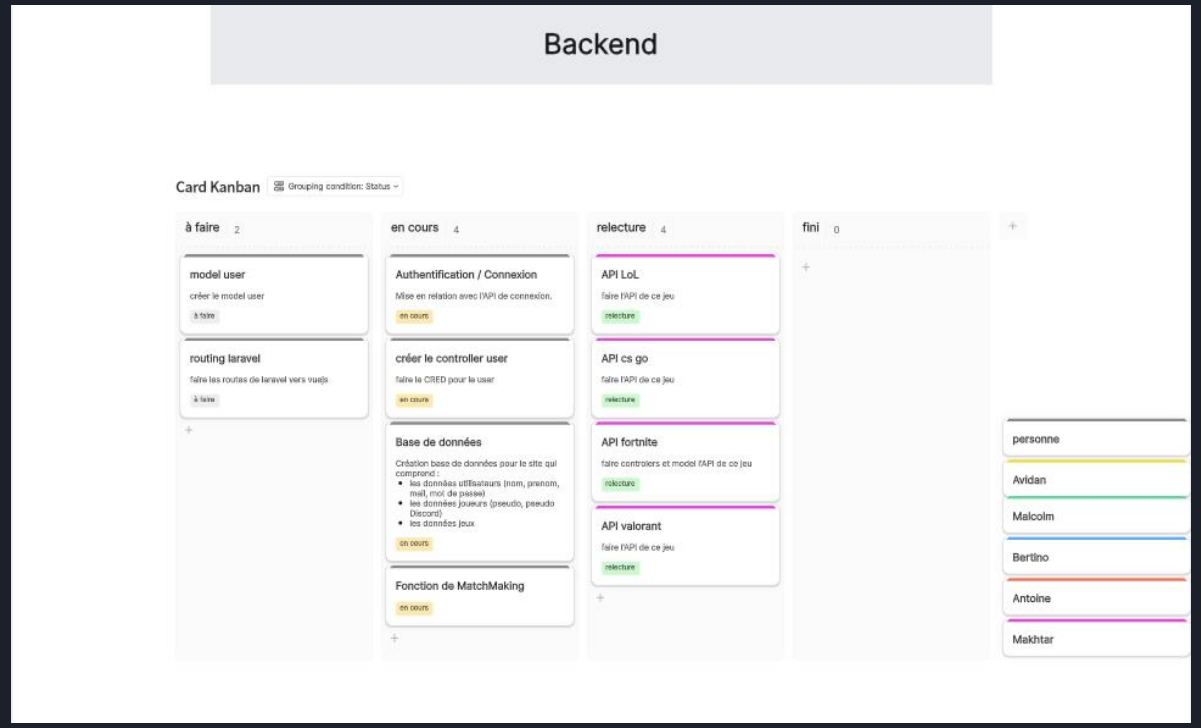
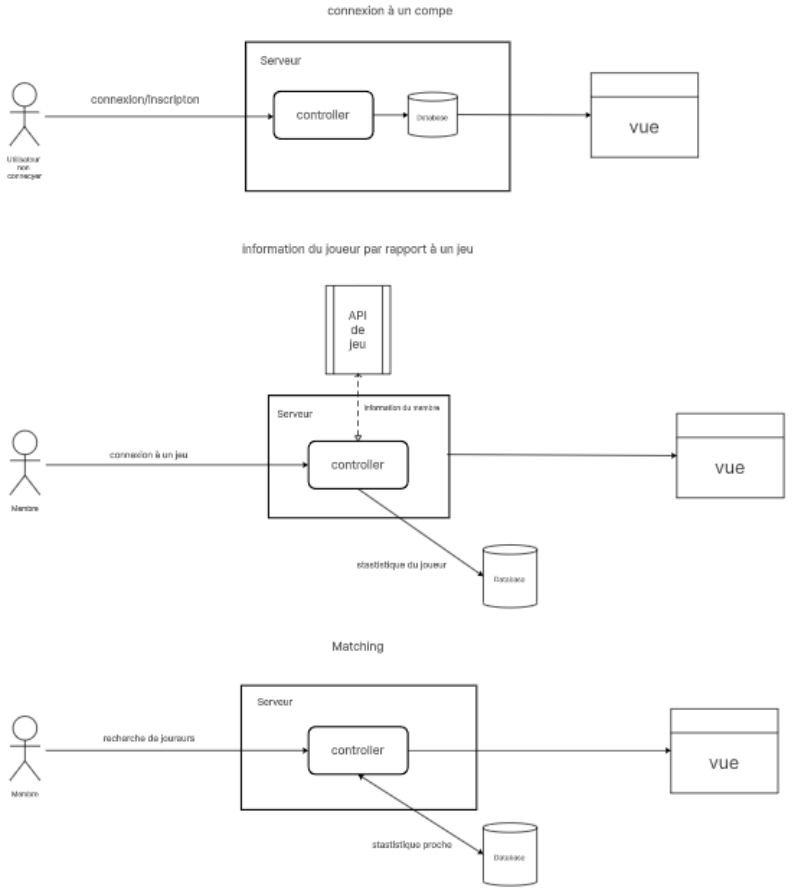

Backend
Card Kanban
Grouping condition: Status
à faire 2
en cours 4
relecture 4
fini 0
model user
créer le model user
à faire
routing laravel
faire les routes de laravel vers vuejs
à faire
Authentification / Connexion
Mise en relation avec l'API de connexion.
en cours
créer le controller user
faire le CRED pour le user
en cours
Base de données
Création base de données pour le site qui comprend :
les données utilisateurs (nom, prenom, mail, mot de passe)
les données joueurs (pseudo, pseudo Discord)
les données jeux
en cours
Fonction de MatchMaking
en cours
API LoL
faire l'API de ce jeu
relecture
API cs go
faire l'API de ce jeu
relecture
API fortnite
faire controlers et model l'API de ce jeu
relecture
API valorant
faire l'API de ce jeu
relecture
personne
Avidan
Malcolm
Bertino
Antoine
Makhtar

Schéma fonctionel



Front-end

- Réalisation d'un figma pour les maquettes.

- Intégration des maquettes en VueJS

Back-end

- Gestion des utilisateurs
- Liaison des comptes de jeu
- Matchmaking en temps réel
- Statistiques de jeu
- Personnalisation du profil
- Gestion des erreurs et des exeptions

MatchMyGame > app > Http > Controllers > MatchmakingController.php

```php
<?php

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\User;

class MatchmakingController extends Controller
{
    public function startSearch(Request $request)
    {
        $user = $request->user();
        $user->update(['is_searching' => true]);

        dispatch(new MatchmakingJob($user));

        return response()->json(['message' => 'Recherche de match commencée'], 200);
    }

    public function stopSearch(Request $request)
    {
        $user = $request->user();
        $user->update(['is_searching' => false]);


        return response()->json(['message' => 'Recherche de match arrêtée'], 200);
    }

    public function matchFound(Request $request, User $matchedUser)
    {
        $user = $request->user();


        return response()->json(['message' => 'Match trouvé avec ' . $matchedUser->username], 200);
    }
}
```

```php
<?php


use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\GameAccount;

class GameAccountController extends Controller
{
    public function link(Request $request, $gameType, $gameAccountId)
    {
        $user = $request->user();

        $existingAccount = GameAccount::where('user_id', $user->id)
            ->where('game_type', $gameType)
            ->where('game_account_id', $gameAccountId)
            ->first();

        if ($existingAccount) {
            return response()->json(['message' => 'Le compte de jeu est déjà lié.'], 422);
        }

        GameAccount::create([
            'user_id' => $user->id,
            'game_type' => $gameType,
            'game_account_id' => $gameAccountId,
        ]);

        return response()->json(['message' => 'Compte de jeu lié avec succès'], 200);
    }

}
```

En conclusion

Perspectives d'avenir

- Mise en relation du back et du front
- Développement et intégration des fonctionnalités prévues
- Réalisation d'un logo
- Implémentation de l'appel vocal

Merci de votre attention