

Pan'E-Sport

**DOCUMENTATION DE
MAINTENANCE/ADMINISTRATION**
12/05/2023

ÉQUIPE :

Mathieu DA MOTA LONGO
Ambre SAINTOBERT
Johanna TARENTO

CODING FACTORY

BY ESIEE-IT

L3 CERGY

SOMMAIRE

NOTE DE CADRAGE	1
ARCHITECTURE DE L'APPLICATION	3
MODULES/BIBLIOTHÈQUES	4
OUTILS	13
STOCKAGE DES DONNÉES	14
VERSIONS ET SYSTÈMES	16
MODÈLE DE RAPPORT D'INCIDENT	19
ANNEXES	20

NOTE DE CADRAGE

Bien que l'e-sport soit un domaine en plein essor, il n'existe pas encore d'application permettant la collection de cartes uniques d'e-sportifs.

C'est ainsi qu'est né Pan'E-Sport, application mobile permettant la collection de cartes de joueurs e-sport.

Il est possible d'acheter des cartes aléatoires qui sont générées avec un numéro unique. Les cartes achetées sont visibles dans la collection utilisateur sur la page profil.

FEATURES DISPONIBLES:

- Incription : il est possible de créer un compte utilisateur en renseignant les informations suivantes dans le formulaire: adresse mail, pseudo, mot de passe (d'au moins 8 caractères) et confirmation de mot de passe.
- Connexion : Il est possible de se connecter en entrant son pseudo et son mot de passe.
- Accueil : Affiche les dernières nouveautés ainsi que les matchs passés et à venir.
- Classement : Affiche les meilleurs joueurs du moment.
- Carte : Affiche les cartes existantes. Il est possible d'effectuer une recherche par pseudo ainsi qu'un filtrage par favoris.
- Détail carte : Lors d'un appuie sur une carte les statistiques du joueur sont visibles. Il est possible depuis cette vue d'ajouter la carte aux favoris ou de l'y supprimer.
- Achats: Sur la vue achats, une carte peut être générée aléatoirement. Celle-ci contient un numéro unique. La carte est générée après que le paiement soit effectué et validé avec Stripe.
- Profil : Sur la vue profil, la collection de cartes de l'utilisateur est visible. Un accès aux paramètres utilisateurs y est disponible.
- Paramètres utilisateur : Depuis cette page, les informations personnelles sont modifiables. Il est aussi possible de se déconnecter et de supprimer le compte utilisateur.

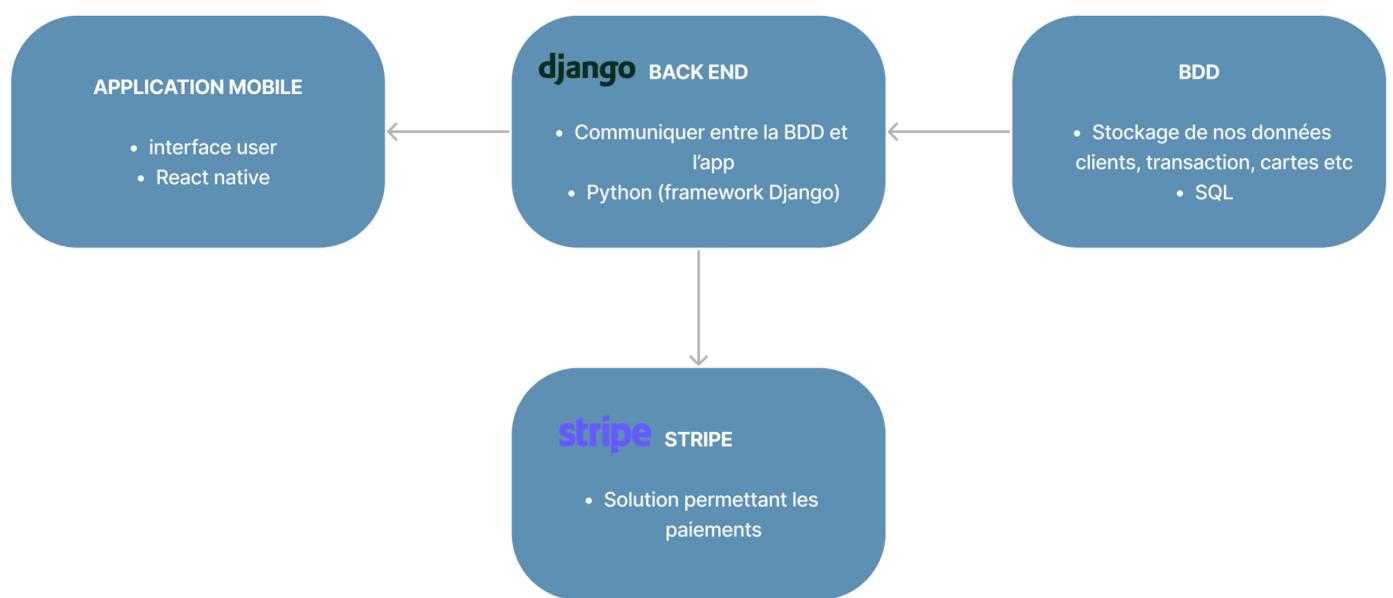
NOTE DE CADRAGE

Certaines features ne sont pas disponibles car non prévues. Cependant il y en a aussi certaines qui ne sont pas encore implémentées mais sont à venir.

FEATURES NON DISPONIBLES:

- Incription : La confirmation par e-mail n'est pas encore accessible.
- Connexion : En cas de mot de passe oublié, il n'est pas encore possible d'accéder à un lien sécurisé permettant de le modifier depuis la page Connexion
- Accueil : Les cards affichant les nouveautés et matchs à venir ne sont pas cliquables.
- Classement : Il n'est pas possible d'appuyer sur les cartes des joueurs afin d'ouvrir celles-ci.
- Carte : Il n'est pas encore possible d'effectuer une recherche par nom/prénom ou par équipe
- Marketplace : La marketplace n'est pas encore disponible, il n'est donc pas encore possible d'acheter des cartes précises à d'autres utilisateurs ni d'en vendre.
- Achats: L'achat de decks contenant plusieurs cartes n'est pas encore disponible.
- Paramètres utilisateur : La double authentification n'est pas encore implémentée.

ARCHITECTURE DE L'APPLICATION



MODULES/BIBLIOTHÈQUES

APP - Pan'E-Sport

Tech Stack

- React-Native
- WebStorm
- JWT
- ModifSVG

Colors

- TabBar : 1D1F24
- Fond : 2b2d31
- Color icon de base : 676D75
- Color icon select : FFFFFF

Install Dependencies

Configuration de l'environnement :

```
> npm install  
> npm update  
> npx react-native pod-install ios
```

Lancer l'application :

```
> npx react-native start  
> npx react-native run-ios
```

Pour run l'iphone que vous voulez :

```
> npx react-native run-ios --simulator="iPhone 14"
```

!? Error

Si erreur de ruby faire les commandes suivantes :

```
> brew install ruby-build  
> brew install rbenv  
> rbenv install 2.7.6  
> rbenv local 2.7.6
```

API - Pan'E-Sport

Tech Stack

- Django
- REST-framework
- IDE - WebStorm
- JWT

Colors APP

- TabBar : 1D1F24
- Fond : 2b2d31
- Color icon de base : 676D75
- Color icon select : FFFFFF

Install Dependencies

Configuration de l'environnement :

```
> pip3 install -r requirements.txt
```

Lancer l'API :

```
> python3 manage.py runserver
```

SQL

Migrer la base de données :

```
> python3 manage.py makemigrations
```

```
> python3 manage.py migrate
```

Modèles en introspectant une base de données existante :

```
> python manage.py inspectdb
```

```
> python manage.py inspectdb > models.py
```

Server Docker

Nom: Docker/Docker-compose

Version Docker: 23.0.4, build f480fb1

Version Docker-Compose: 2.3.3

Chemin du fichier de configuration: docker-compose.yml

Tech Stack

Docker

Docker-docs

Docker-compose

configuration

Pour accéder au fichier de configuration :

```
> sudo nano docker-compose.yml
```

Démarrage d'un conteneur

Pour démarrer un conteneur :

```
> sudo docker compose run -d (name container)
```

```
> sudo docker compose up --build
```

Voir tous les conteneurs démarrés

Pour afficher tous les conteneurs démarrés, exécutez l'une des commandes suivantes :

```
> sudo docker compose ps
```

```
> sudo docker ps -a
```

Server Docker

Arrêt d'un conteneur

Pour arrêter un conteneur, exéutez les commandes suivantes :

```
> sudo docker compose stop  
> sudo docker compose rm -f  
> sudo docker stop (id)  
> sudo docker compose rm -f
```

Accès à un shell dans un conteneur

Pour accéder à un shell dans un conteneur :

```
> sudo docker exec -it (container name ou id) bash
```

Logs du conteneur

Pour afficher les logs d'un conteneur :

```
> sudo docker compose logs (container name ou id)
```

Base de données

- **Nom:** MariaDB
- **Accès manuel:** SequelPro / DataGrip
- **Version:** 15.1 Distrib 10.6.12-MariaDB
- **Port:** 3306
- **Utilisateur:**
- **Mot de passe:**

Tech Stack

- Mysql
- Mariadb

Configuration :

```
> sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Démarrage de la base de données :

```
> sudo mariadb
```

Commandes importantes pour MySQL

Créer un utilisateur de la base de données :

```
CREATE      USER      'MACHIN'@'localhost'      IDENTIFIED      BY  
'PASSWORDMACHIN';
```

Donner les priviléges à l'utilisateur pour une base de données spécifique :

```
GRANT ALL PRIVILEGES ON BASEDEDONNEE.* TO 'USERMACHIN'@'%'  
IDENTIFIED by "PASSWORDMACHIN";
```

Donner tout les priviléges à l'utilisateur pour toutes les bases de données :

```
GRANT ALL PRIVILEGES ON *.* TO 'USERMACHIN'@'%' IDENTIFIED by  
"PASSWORDMACHIN";
```

Base de données

Donner les privilèges de lecture à l'utilisateur pour toutes les bases de données :

```
GRANT SELECT ON *.* TO 'USERMACHIN'@'%' IDENTIFIED by  
"PASSWORDMACHIN";
```

Pour que les changements de privilèges soient pris en compte :

```
FLUSH PRIVILEGES;
```

Redémarrer la base de données :

```
> sudo service mysql restart # Si vous utilisez MySQL
```

```
> sudo service mariadb restart # Si vous utilisez MariaDB
```

Note : Les commandes MySQL doivent être exécutées dans l'interface de ligne de commande de MySQL.

Pour y accéder, exéutez simplement la commande suivante :

```
> mysql -u NOM_UTILISATEUR -p
```

Remplacez "NOM_UTILISATEUR" par le nom d'utilisateur que vous souhaitez utiliser pour accéder à la base de données.

Commandes Unix d'administration système

*Ces commandes sont valables sur la plupart des distributions Linux. Il est possible que vous deviez ajouter **sudo** devant chacune des commandes afin d'avoir les autorisations pour les exécuter.*

Groupes

Créer un nouveau groupe nommé testgroup :

```
> sudo groupadd testgroup
```

Ajouter l'utilisateur existant wwwuser au groupe testgroup :

```
> sudo usermod -a -G testgroup wwwuser
```

Utilisateurs

Créer un nouvel utilisateur nommé projectuser :

```
> sudo useradd projectuser
```

Super admin :

```
> su root
```

Permissions

Ajouter une permission d'exécution à un fichier script.sh :

```
> sudo chmod +x script.sh
```

Définir wwwuser comme l'utilisateur propriétaire du fichier script.sh :

```
> sudo chown wwwuser script.sh
```

Définir testgroup comme le groupe propriétaire du fichier script.sh :

```
> sudo chown :testgroup script.sh
```

Faites bien attention à écrire les deux-points devant le nom du groupe_

Commandes Unix d'administration système

Fichiers & Dossiers

Lister le contenu du répertoire dans lequel on se trouve :

```
> sudo ls -l
```

Créer un nouveau fichier notes.txt :

```
> sudo touch notes.txt
```

Créer un nouveau dossier logs :

```
> sudo mkdir logs
```

Supprimer un fichier log.json :

```
> sudo rm log.json
```

Pare-feu (ufw)

Assurez-vous toujours d'ouvrir le port 22 (SSH) avant d'activer le pare-feu sur un serveur, sinon votre connexion risque d'être coupée et l'accès à votre serveur deviendra très difficile, voire impossible sans avoir à le formater et le réinstaller.

Les règles sont exécutées dans l'ordre de leur entrée, il faudra donc par exemple d'abord refuser toutes les connexions avant d'exécuter des règles qui autorisent les connexions sur un port donné.

Lister les règles actuelles du pare-feu de façon détaillée :

```
> sudo ufw status verbose
```

Activer le pare-feu :

```
> sudo ufw enable
```

Désactiver le pare-feu :

```
> sudo ufw disable
```

Ajouter une règle pour refuser, par défaut, toute connection entrante :

```
> sudo ufw default deny incoming
```

Ajouter une règle pour accepter les connections TCP/UDP sur le port 443 :

```
> sudo ufw allow 443
```

Ajouter une règle pour accepter les connections en TCP only, sur le port 80 :

```
> sudo ufw allow 80/tcp
```

Commandes Unix d'administration système

Services (systemctl)

Lister les services installés :

> sudo systemctl --type service

Démarrer un service :

> sudo systemctl start [application]

Arrêter un service :

> sudo systemctl stop [application]

Redémarrer un service :

> sudo systemctl restart [application]

Activer un service pour qu'il soit exécuté au démarrage de la machine :

> sudo systemctl enable [application]

Désactiver un service pour qu'il ne soit plus exécuté au démarrage de la machine :

> sudo systemctl disable [application]

Standard

Voir le CPU du server :

> sudo htop

Mettez à jour la liste des paquets :

> sudo apt update

Affichez les paquets pour lesquels une mise à jour est disponible :

> sudo apt list --upgradable

Mettez à jour les paquets installés sur le système avec au choix :

> sudo apt upgrade

upgrade : les paquets seront remplacés par des versions plus récentes, mais sans qu'aucun autre paquet ne soit ajouté ou supprimé. Par exemple, une nouvelle version de Firefox sera installée avec apt upgrade.

> sudo apt full-upgrade

full-upgrade : même chose que apt upgrade, mais supprime des paquets si cela est nécessaire pour installer les nouvelles versions des paquets.

Supprimez tous les paquets devenus obsolètes et qui ne sont plus nécessaires :

> sudo apt autoremove

Supprimez le cache des paquets qui ne peuvent plus être téléchargés :

> sudo apt autoclean

OUTILS

L'éditeur de code utilisé par l'équipe est Webstorm by JetBrains. C'est un environnement de développement intégré pour JavaScript et les technologies connexes.

WebStorm fournit la complétion automatique du code, l'analyse à la volée du code, le support du refactoring et l'intégration de VCS (version control system).

Afin d'avoir un aperçu de notre application, nous utilisons l'émulateur IPhone 14 de Xcode.

Pour la gestion des données, l'équipe utilise Datagrip by JetBrains mais aussi l'API avec un compte administrateur (Les identifiants sont récupérables auprès de l'administrateur).



STOCKAGE DES DONNÉES

Nos données sont stockées dans une base de donnée SQL.

Celle-ci est accessible via une API à l'adresse www.api/panesport.com..

Pour accéder à l'API, il est nécessaire d'être connecté avec un compte administrateur. Ces identifiants sont récupérables auprès de l'administrateur de l'API.

API Pan'E-Sport v 1.5

Log in

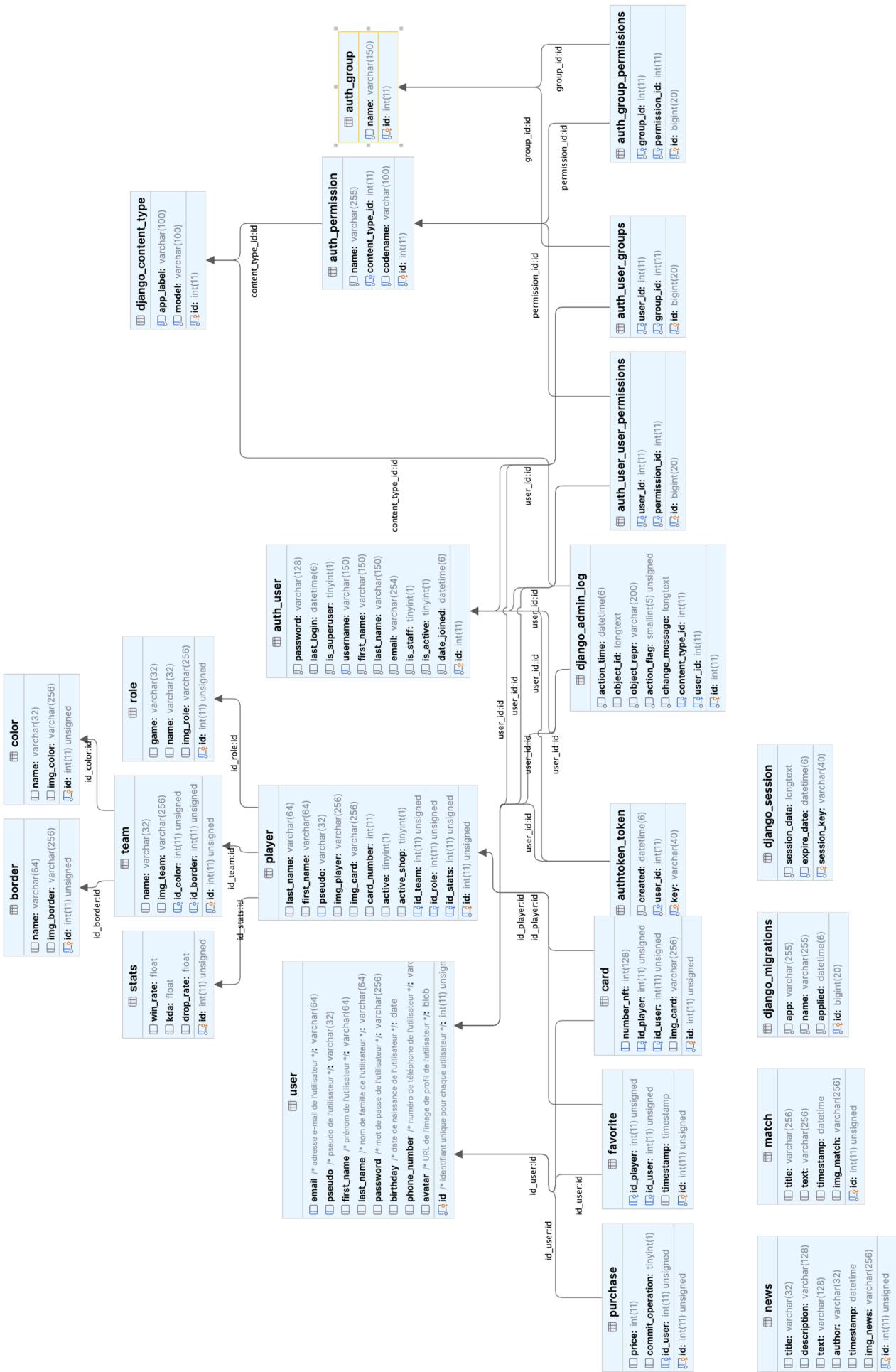
Versions

API Pan'E-Sport Version 1.5

Api available

api.panesport.com/api/cards/
api.panesport.com/api/players/
api.panesport.com/api/purchases/
api.panesport.com/api/roles/
api.panesport.com/api/stats/
api.panesport.com/api/teams/
api.panesport.com/api/users/
api.panesport.com/api/news/
api.panesport.com/api/matchs/
api.panesport.com/api/favorites/
api.panesport.com/api/colors/
api.panesport.com/api/borders/

Copyright © 2023 Pan'E-sport. Tous droits réservés.



VERSIONS ET SYSTÈMES

Server

Tech

- OS: Ubuntu 22.04 LTS (GNU/Linux 5.15.0-27-generic x86_64)
- Modèle: VPS vps2020-value-1-2-40
- Mémoire: 2 Go
- Storage: 40 Go
- Sauvegarde automatique: Non

IP

- IPv4 : 51.254.120.199
- IPv6 : 2001:41d0:401:3000::4b98
- Gateway : 2001:41d0:401:3000::1
- Nom de domaine lié : panesport.com

Tech Stack

- Ubuntu

✓ Connexion

- Login: ubuntu
- Login: root

SSH

```
ssh ubuntu@51.254.120.199
```

GITHUB

<https://github.com/Pan-E-Sport>

VERSIONS ET SYSTÈMES

API

asgiref==3.6.0

Django==4.2.1

djangorestframework==3.14.0

mysqlclient==2.1.1

pytz==2023.3

sqlparse==0.4.4

psycopg2==2.9.5

stripe==5.4.0

psycopg2-binary==2.9.6

pillow==9.5.0

gunicorn==20.1.0

djangorestframework-simplejwt==5.2.2

django-cors-headers==3.14.0

requests==2.30.0

VERSIONS ET SYSTÈMES

APP

Dependencies:

```
@react-native-async-storage/async-storage: ^1.18.1
@react-navigation/bottom-tabs: ^6.5.7
@react-navigation/native: ^6.1.6
@react-navigation/native-stack: ^6.9.12
axios: ^1.3.4
jwt-decode: ^3.1.2
react: 18.2.0
react-native: 0.71.7
react-native-datepicker: ^1.7.2
react-native-linear-gradient: ^2.6.2
react-native-modal-datetime-picker: ^15.0.0
react-native-safe-area-context: ^4.5.0
react-native-screens: ^3.20.0
react-native-svg: ^13.8.0
react-native-vector-icons: ^9.2.0
```

MODÈLE DE RAPPORT D'INCIDENT

RAPPORT D'INCIDENT

IDENTITÉ DE L'EMPLOYÉ

NOM:

PRÉNOM:

SERVICE:

POSTE:

DESCRIPTION DE L'INCIDENT

DATE DE L'INCIDENT:

HEURE DE L'INCIDENT:

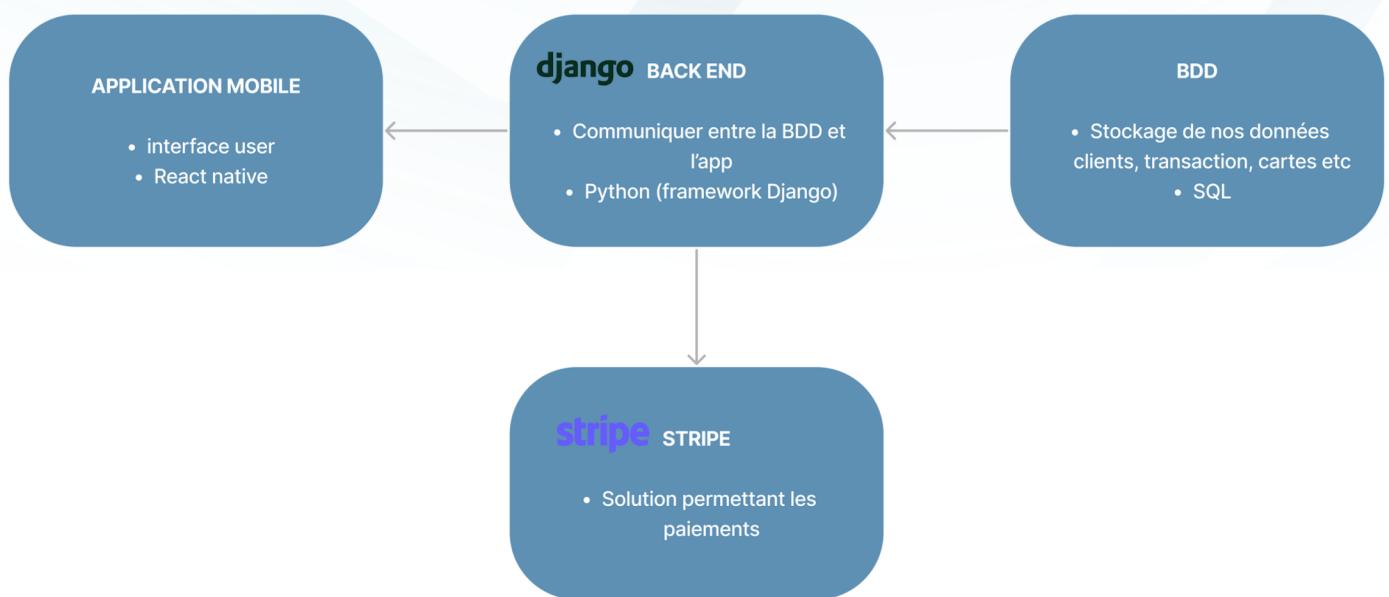
DESCRIPTION DE L'INCIDENT:

DESCRIPTION DES POTENTIELLES CONSÉQUENCES:

DATE:

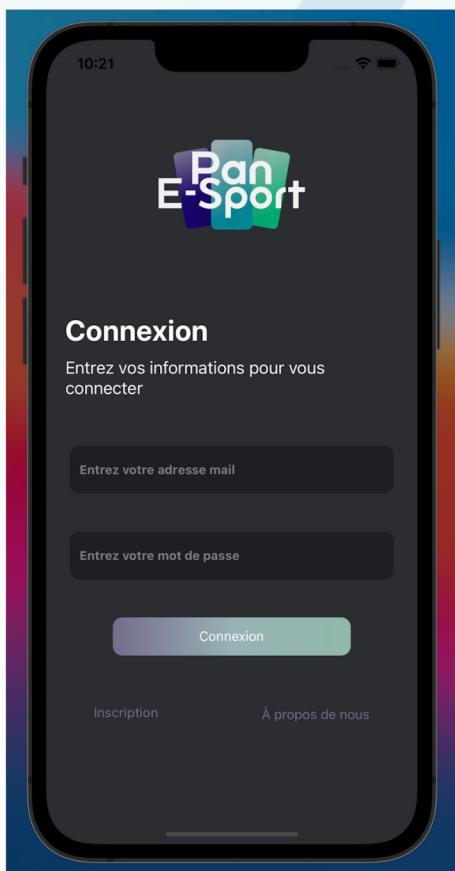
SIGNATURE:

ANNEXE



(1) Architecture détaillée de l'application

ANNEXE



(2) Émulateur iPhone14 de Xcode



(3) Logo de Xcode

ANNEXE

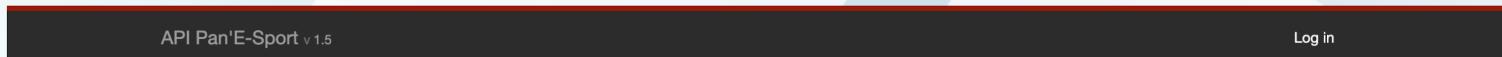


(4) Logo de Webstorm by JetBrains



(5) Logo de DataGrip by JetBrains

ANNEXE



API Pan'E-Sport v 1.5

[Log in](#)

Versions

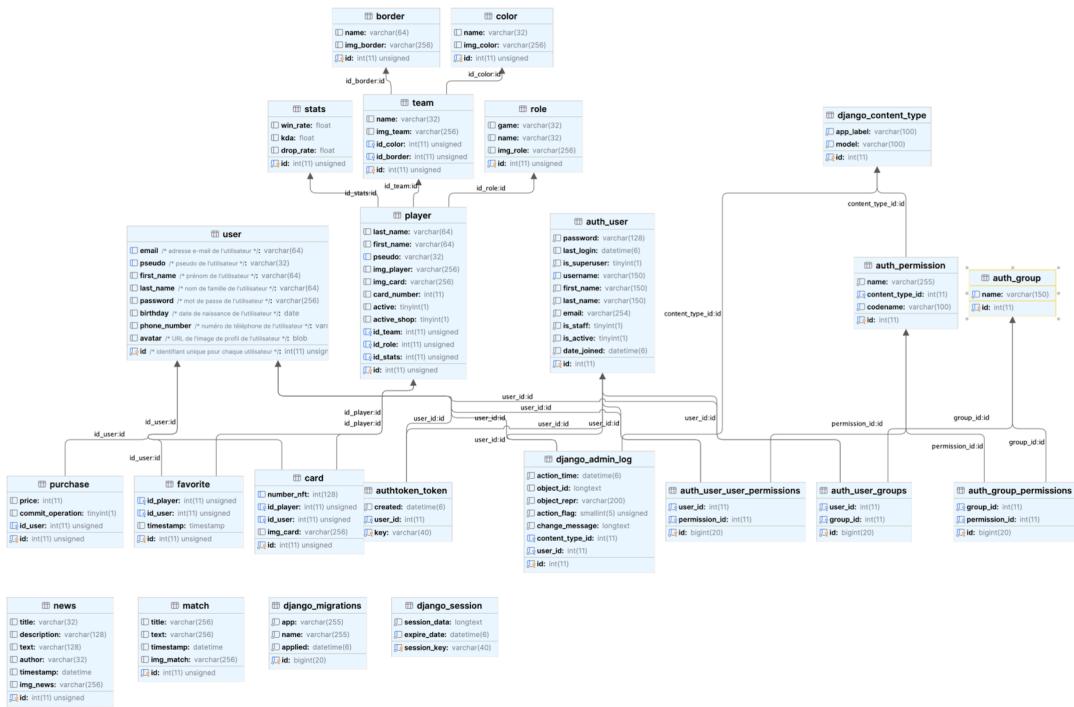
API Pan'E-Sport Version 1.5

Api available

api.panesport.com/api/cards/
api.panesport.com/api/players/
api.panesport.com/api/purchases/
api.panesport.com/api/roles/
api.panesport.com/api/stats/
api.panesport.com/api/teams/
api.panesport.com/api/users/
api.panesport.com/api/news/
api.panesport.com/api/matchs/
api.panesport.com/api/favorites/
api.panesport.com/api/colors/
api.panesport.com/api/borders/

Copyright © 2023 Pan'E-sport. Tous droits réservés.

(6) Accueil de l'API Pan'E-Sport



(7) Base de donnée PAn'E-Sport

ANNEXE

RAPPORT D'INCIDENT

IDENTITÉ DE L'EMPLOYÉ

NOM:

PRÉNOM:

SERVICE:

POSTE:

DESCRIPTION DE L'INCIDENT

DATE DE L'INCIDENT:

HEURE DE L'INCIDENT:

DESCRIPTION DE L'INCIDENT:

DESCRIPTION DES POTENTIELLES CONSÉQUENCES:

DATE:

SIGNATURE:

(8) Rapport d'incident Pan'E-Sport