

Licence Professionnelle IEM

Documentation fonctionnelle et technique

LANNIER Johan - QUILLON Nicolas

The logo for 'PokéCard' is displayed in a bold, yellow, stylized font. The letters are thick and blocky, with a slight slant. The 'P' and 'C' are particularly large and prominent. The text is centered within a red rectangular background that has a thin yellow border.

Table des matières

Introduction.....	3
1. Fonctionnalités.....	3
2. Technique	10
a. API.....	10
b. Application Android	11

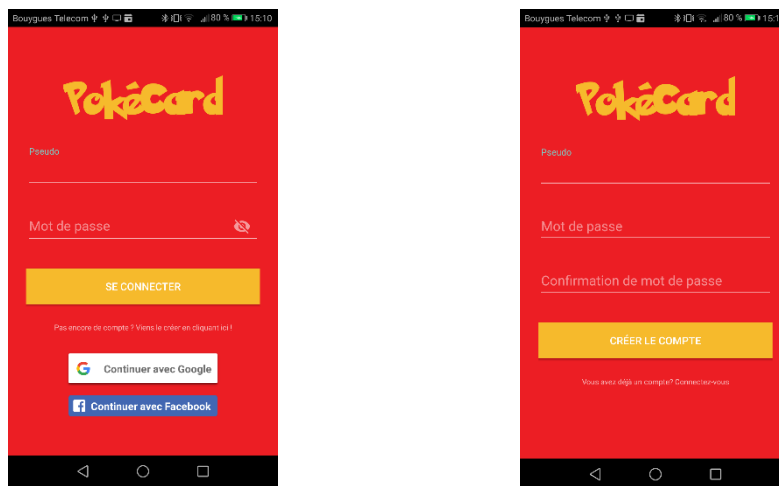
Introduction

Pokecard est un projet d'application mobile où un utilisateur peut gérer une collection de cartes afin de les échanger avec d'autres utilisateurs de l'application. L'application devait interroger une API REST sur un serveur distant que nous devons créer nous-mêmes. Ce document va expliquer les différentes fonctionnalités de l'application ainsi que la partie technique du projet.

1. Fonctionnalités

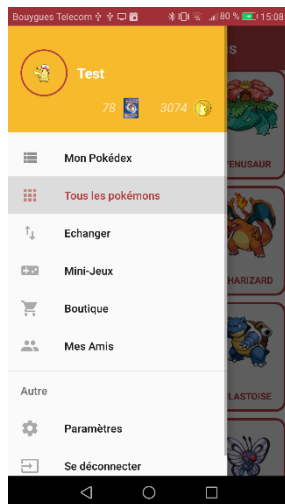
Nous avons implémenté plusieurs fonctionnalités dans notre application :

1.1) *Inscription/connexion via un formulaire ou réseaux sociaux*



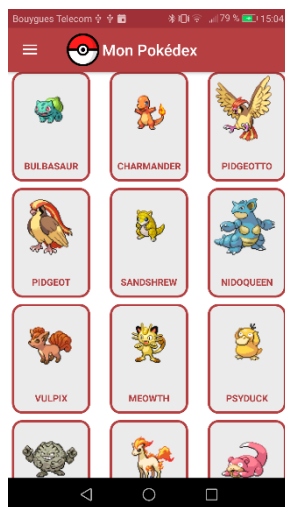
L'utilisateur a la possibilité de s'inscrire sur PokéCard grâce à un formulaire classique (Pseudo, Mot de passe et Confirmation de mot de passe). Il peut ensuite se connecter via le formulaire de connexion grâce à son pseudo et à son mot de passe. L'utilisateur a aussi la possibilité de se connecter via les réseaux Facebook ou Google. Si aucun compte n'est lié, le pseudo sera le nom public et le mot de passe ne sera pas enregistré dans la base de données.

1.2) Gestion d'une monnaie (PokéCoins)



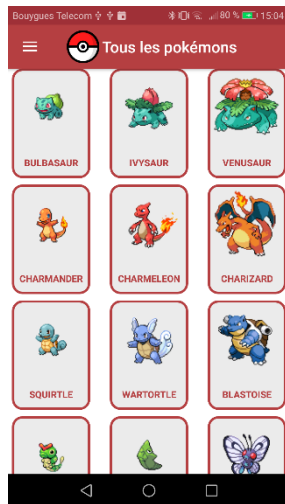
Le PokéCoin est la e-monnaie de l'application PokéCard. Elle permet à l'utilisateur d'acheter des packs de cartes via la boutique. 500 PokéCoins sont obtenus à la création du compte, mais si l'utilisateur en désire plus afin d'accéder à de nouvelles cartes il peut en gagner grâce aux mini-jeux.

1.3) Affichage du pokédex (Pokémons possédés par l'utilisateur)



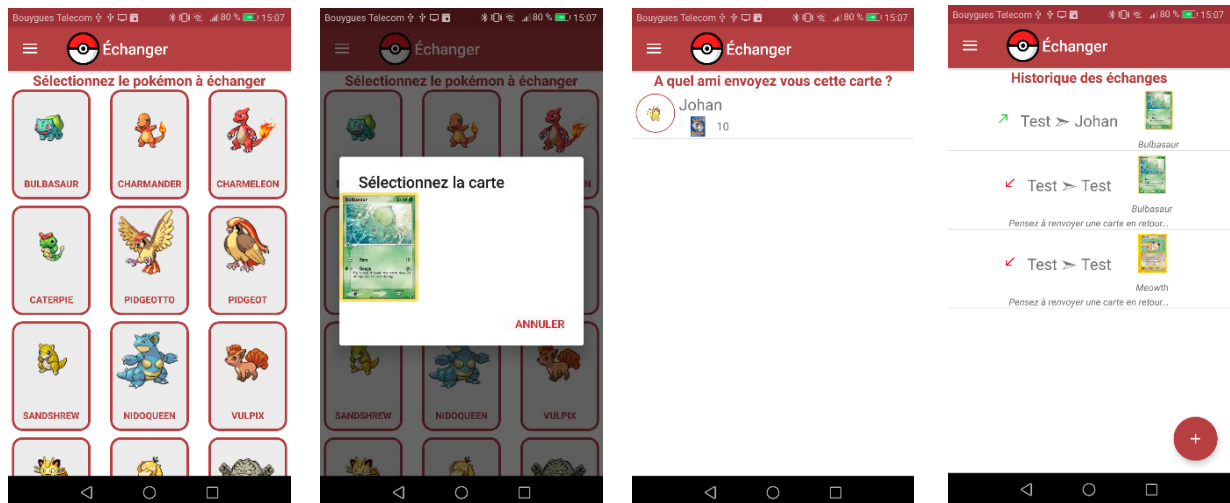
Le pokédex affiche les pokémons sur lequel le joueur possède au moins une carte. Les cartes possédées en plusieurs exemplaires sont indiquées par un logo bleu avec le nombre de cartes. Lors d'un clic sur un pokémon, les détails de celui-ci s'affichent (nom, poids, taille, type...) ainsi que les cartes liées à ce pokémon.

1.4) Affichage de tous les Pokémons existants



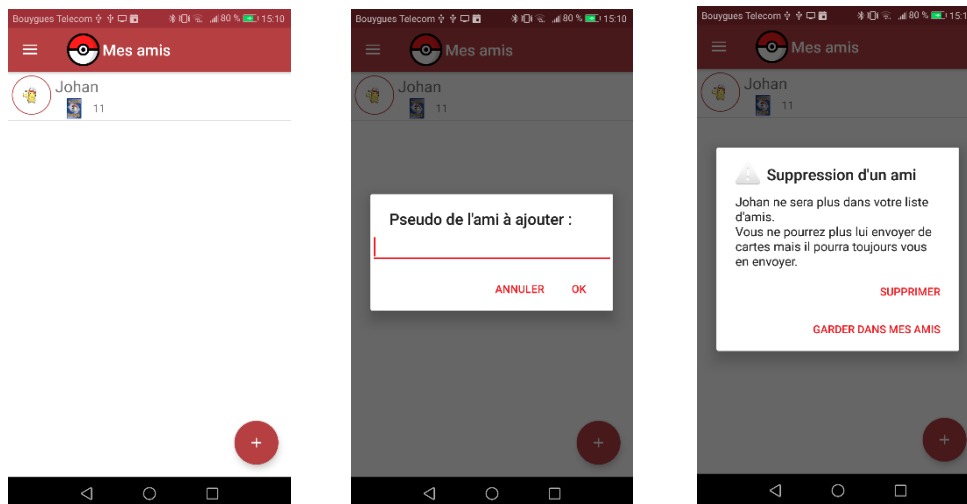
L'une des pages de l'application permet d'accéder à tout les pokémons et à toutes les cartes existantes via les détails.

1.5) Échanger des cartes



Le but principal de l'application est d'échanger des cartes avec ses amis. Le joueur doit donc sélectionner le pokémon qu'il souhaite envoyer, ensuite la carte correspondante puis l'envoyer à l'ami de son choix. Ensuite cet échange apparait dans son historique des échanges ainsi que dans celui de son ami.

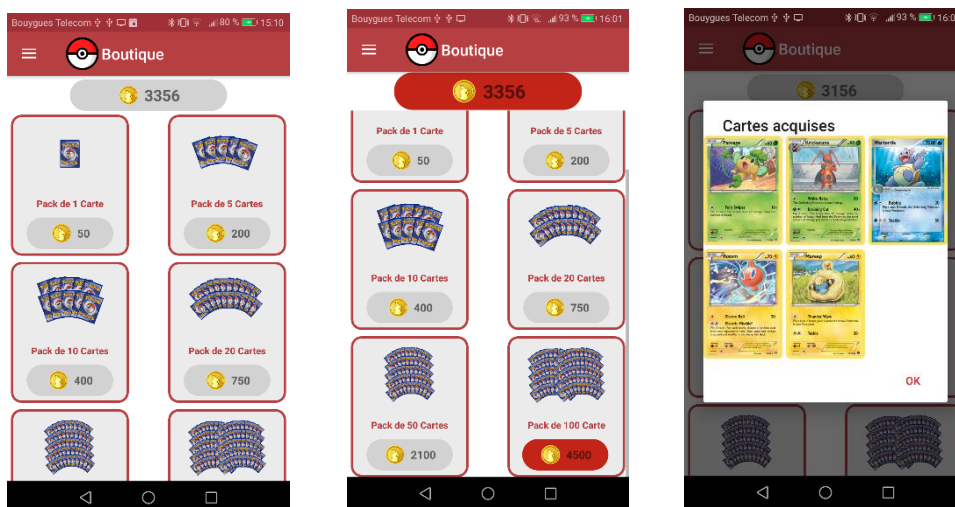
1.6) Gestion des amis



Dans l'application PokéCard il y a la possibilité d'ajouter des joueurs en amis grâce à leur pseudo. Il est nécessaire d'ajouter des utilisateurs à vos amis si vous voulez échanger des cartes avec lui.

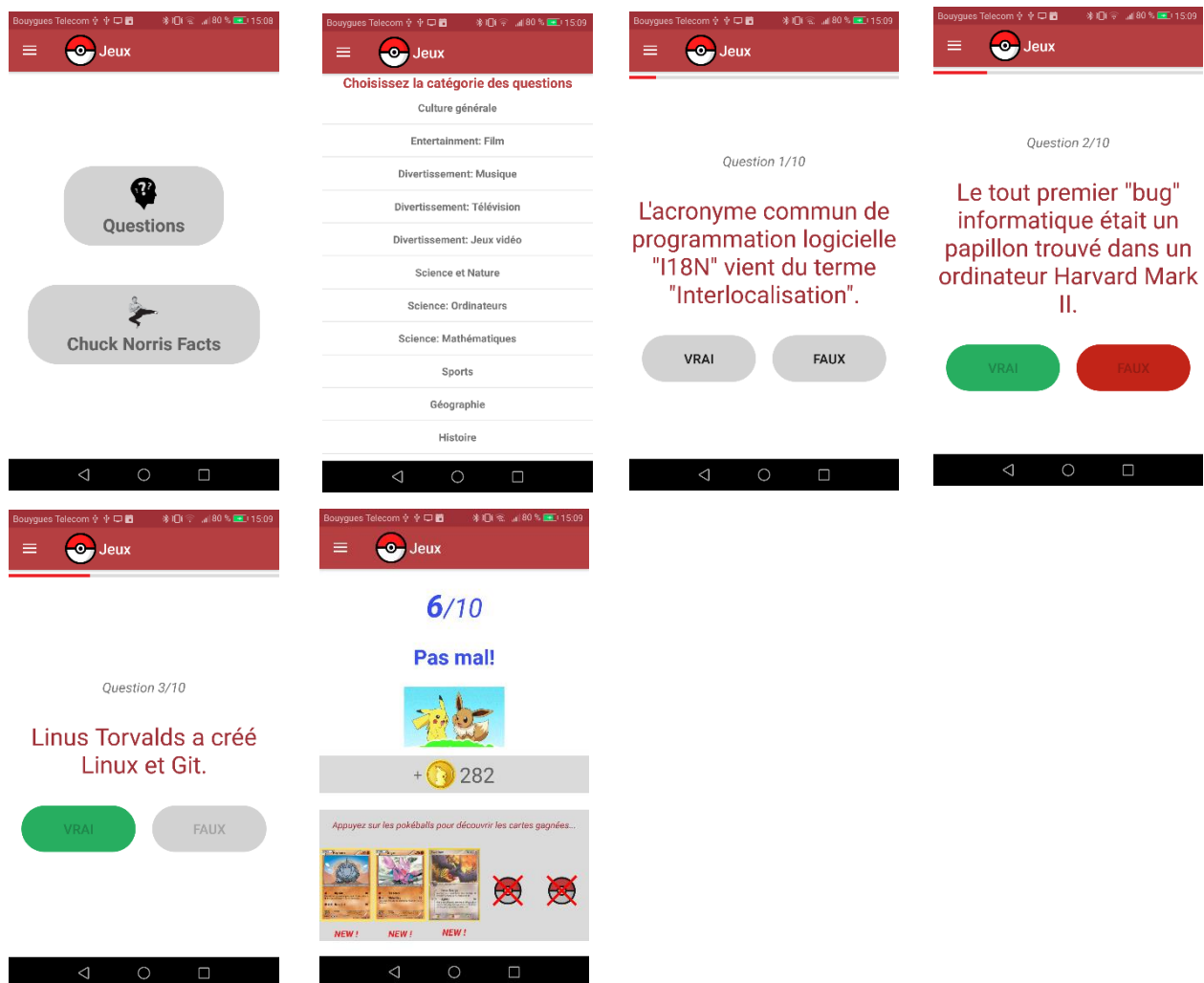
Vous avez, bien entendu, la possibilité de supprimer un joueur de vos amis.

1.7) Acheter des cartes via une boutique



Grâce à ses PokéCoins, le joueur peut acheter des packs afin de gagner des nouvelles cartes. Si le joueur a assez de PokéCoins pour acheter le pack désiré, des cartes sont ajoutées aléatoirement à son pokédex. Il est débité immédiatement du nombre de PokéCoins correspondant au pack acheté.

1.8) Gagner des PokéCoins grâce à des mini-jeux (questions et Chuck Norris Facts)



Afin de gagner des PokeCoins, le joueur peut jouer à des mini-jeux. Le premier mini-jeu consiste à répondre à 10 questions et, en fonction du score obtenu, un nombre de PokéCoins et de nouvelles cartes seront ajoutés au compte du joueur.

Le deuxième mini-jeu consiste à lire des Chuck Norris Facts. A chaque nouveau fait, 20 PokéCoins sont ajoutés au compte.

1.9) Ajouter des cartes via des tags NFC

Nous pouvons imaginer commercialiser des cartes Pokémon équipées d'une puce NFC afin de les ajouter directement dans l'application via un scan NFC.

Une fois le téléphone posé sur la carte, l'application va reconnaître immédiatement de quelle carte il s'agit et l'ajoutera directement au compte du joueur (elle sera préalablement affichée).

Une vidéo correspondante à cette fonctionnalité a été ajoutée dans notre GitHub

(<https://github.com/johanlannier/PokeCard>)

1.10) Modifier la photo de profil du compte

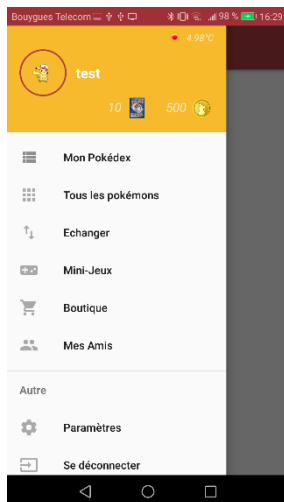


Le joueur a la possibilité, via la page des paramètres de l'application, de changer de photo de profil parmi plusieurs images.

1.11) Modifier le pseudo de l'utilisateur

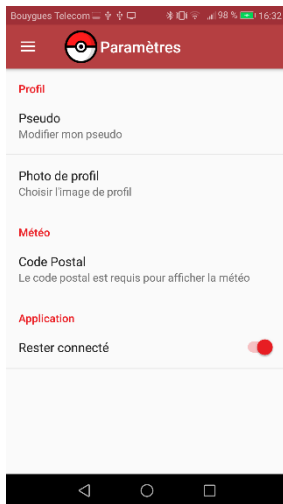
Le joueur a la possibilité, via la page des paramètres de l'application, de changer de pseudo.

1.12) Affichage de la météo



La météo est affichée en haut du navigation drawer. Elle contient une icône affichant le temps ainsi que la température. La météo correspond au code postal entré dans la page des paramètres de l'application.

1.13) Connexion automatique



L'utilisateur a la possibilité de rester connecté à son compte. Si l'item « Rester connecté » est activé, il ne sera plus nécessaire d'entrer ses informations de connexion, il sera directement amené sur la page de son pokédex.

2. Technique

a. API

L'API est un serveur express NodeJS. On a choisi cette technologie pour sa grande flexibilité et sa simplicité d'installation. NodeJS est également une technologie très répandue, ce qui permettait de trouver rapidement des solutions aux problèmes que l'on rencontrait.

L'architecture du serveur est assez simple : les routes sont regroupées suivant plusieurs groupes et chaque groupe possède son propre Controller :

- Auth : ce sont les routes pour gérer l'authentification dans l'application
- User : ce sont les routes qui touchent principalement la base de données, et celle qui affecte directement un utilisateur
- Pokemon : ce sont les routes qui gèrent la récupération des pokémons
- Cards : ce sont les routes qui gèrent toutes les actions avec les cartes
- Exchange : ce sont les routes qui gèrent les échanges de cartes
- Option : ce sont les routes qui gèrent les paramètres
- Quizz : ce sont les routes qui gèrent les quizz
- ChuckNorris : la route récupère les Chuck Norris Facts
- Weather : la route récupère la météo

Il y a également un Manager qui permet de gérer les requêtes vers les API. Voici la liste des API auxquelles notre API communique :

- <https://pokeapi.co/api/v2/pokedex/1> : Cette API retourne le pokédex ainsi que des informations sur les Pokémons
- <https://docs.pokemontcg.io/> : Cette API permet de retourner les cartes des Pokémons
- https://opentdb.com/api_config.php : Cette API propose des quizz suivant certaines catégories (la liste est disponible sur l'API)
- <http://openweathermap.org/> : Cette API permet de récupérer la météo
- <https://api.chucknorris.io/> : Cette API permet de récupérer les Chuck Norris facts

Notre API implémente des bibliothèques sous forme de "node_module". On a choisi d'implémenter :

- Lodash : c'est une bibliothèque implémente un certain nombre de fonctions utiles pour coder, telles que le tri d'un tableau, ou de recherche d'un objet dans un tableau d'objets
- mysql : c'est un module permettant de se connecter à une base de données MySQL
- bcrypt : c'est un module permettant de crypter les mots de passe des utilisateurs
- http : c'est un module permettant de faire des requêtes HTTP
- https : c'est un module permettant de faire des requêtes HTTPS



b. Application Android

L'application PokéCard a été développée en natif. Elle possède un menu appelé Navigation Drawer grâce auquel il est assez facile de se repérer et de se déplacer dans l'application.

Toutes les fonctionnalités citées plus haut dans ce document sont implémentées grâce à l'API, aucun appel vers un quelconque site internet est effectué sur l'application sans passer par l'API. C'est pourquoi il n'existe pas de mode « hors-ligne ».

Afin de développer plus facilement, plusieurs librairies ont été utilisées :

- **Retrofit** : permet de facilement faire des requêtes http vers un serveur.
- **Gson** : permet de créer/lire un fichier Json
- **Picasso** : permet de télécharger une image en asynchrone et gère un cache
- **CircleImageView** : permet d'afficher une image dans un cercle