

## **Compte rendu d'activité KekeliPay**

# **SOMMAIRE**

[CONTEXTE](#)

[OUTILS](#)

[INSTALLATION](#)

[STRUCTURE](#)

[FONCTION UTILISATEUR](#)

[FONCTION ADMIN](#)

[BILAN](#)

# CONTEXTE

Dans le cadre de l'extension des fonctionnalités de sa plateforme d'éducation ([kekelidu.net](http://kekelidu.net)),



l'entreprise Lorden Software, m'a confié la tâche de réaliser une application mobile sur la plateforme Android. La plateforme kekelidunet est une plateforme éducative permettant a des étudiants du niveau primaire jusqu'au lycée de pouvoir s'entraîner a travers des quizzes sur leurs connaissances dans les matieres du programme d'enseignement. Afin d'encourager l'excellence, un système de points denomme « keko » permet aux etudiants de gagner ces points et de pouvoir les echanges contre des produits disponibles chez des entreprises ou boutiques partenaires. Le système d'échanges de points ou credits (entre etudiants et partenaires) etant manuel, l'application mobile aura pour vocation de permettre au etudiants et aux partenaires de :

- Transférer le credit de l'étudiant vers le partenaire
- Générer une carte de credit pour les etudiants baser sur un code QR
- Gérer (pour l'administrateur) les cartes de credits (créer, désactiver), et les utilisateurs de l'application (identifiants, mots de passes, etc...)
- Envoyer aux administrateurs des requetes de retraits (des partenaires) qui voudrait convertir leurs credits « keko » en monnaie courante

- Scanner une carte de credit afin de pouvoir proceder a un transfert de credit.

# OUTILS

## LANGAGES :

- Javascript
- SQL

## FRAMEWORK :

- React Native (FrontEnd)
- Node.JS (BackEnd)

## API :

- REST

## CIBLE :

- ANDROID

## IDE :

- Visual Studio Code

# INSTALLATION

La partie Frontend est fait avec la framework React-Native, l'installation de l'environnement de développement se fait ainsi :

- Installation de Node.js
- Installation du JDK
- Installation d'Android Studio
- Creation d'un appareil virtuel sur Android Studio
- Installation de Expo Go
- Creation du projet

`npx create-expo-app KekeliPay`

`cd KekeliPay`

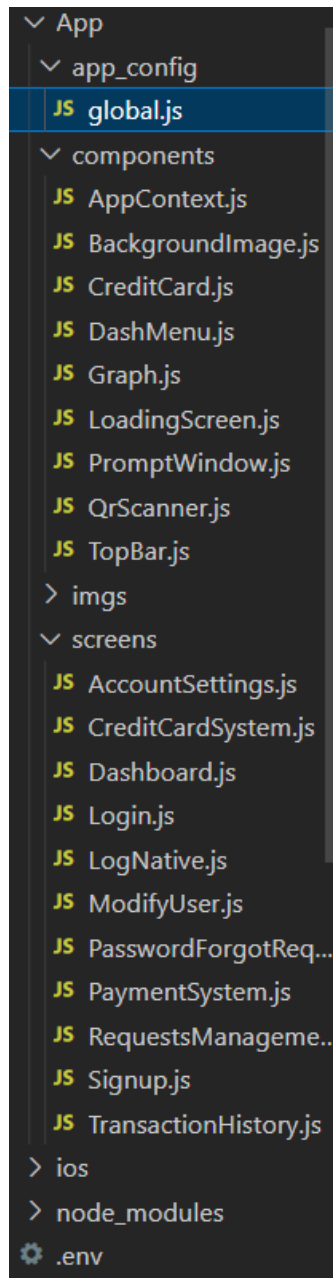
`npm start`

- Executer l'application dans l'emulateur

Dans le dossier du projet (dans l'invite de commande), executer la ligne :

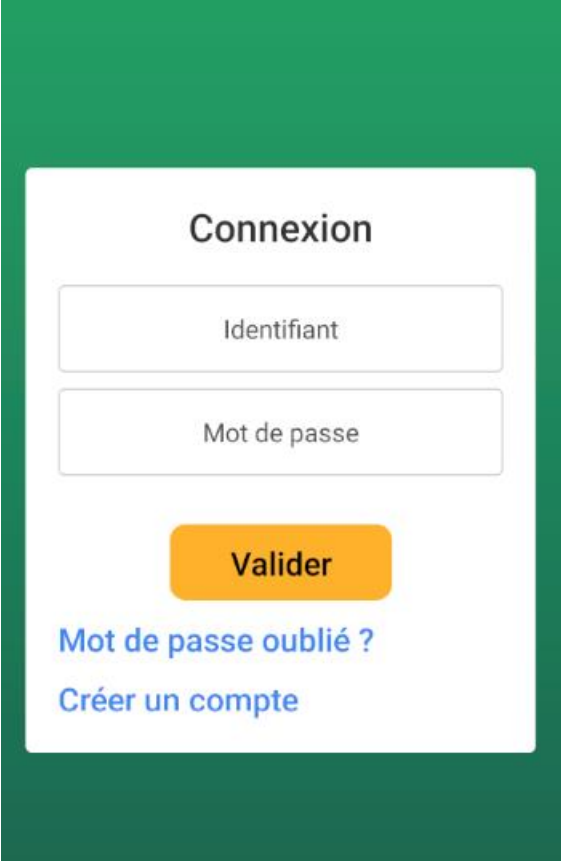
« `npx react-native run-android` »

# STRUCTURE



L'application est composée de « screens » qui sont les pages de l'application (page « login » pour la page de connexion par exemple), les composants sont en quelques sortes des modules permettant de mieux structurer le code (le composant « DashMenu » par exemple, est le système qui gère les boutons des actions principales pour les utilisateurs et les admins dans la page « Dashboard »).

La page initiale est la page du login demande un nom d'utilisateur/un mot de passe. Elle permet aussi d'aller aux pages de création de compte ou de réinitialisation de mot de passe :



**Connexion**

Identifiant

Mot de passe

**Valider**

[Mot de passe oublié ?](#)

[Créer un compte](#)



Quelle est votre categorie :

Particulier (Secteur informel) ☒

Entreprise (Secteur formel) ☐

**Suivant**

### Creation de compte

Nom

Prénom

Nom d'utilisateur

Téléphone

Email

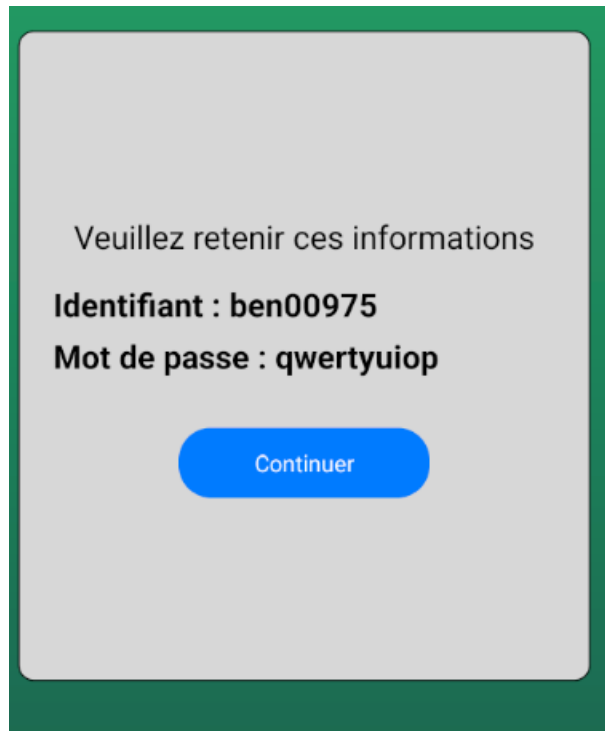
Suivant

### Entrez un mot de passe

Mot de passe

Valider





La création d'un compte envoie une requête POST à l'API :

```
app.post('/signup', function(req,res) {
  // console.log(req.body)
  db.query(
    'SELECT * FROM kekelipay_users WHERE username = ? OR mail = ? or contact = ?',
    [req.body.username, req.body.mail, req.body.contact], function(error, results, field) {
      if(error){
        res.json({ message: '0' });
      }
      else if (results.length > 0){
        res.json({ message: '3' });
      }
      else {
        db.query(
          'INSERT INTO kekelipay_users (username,password,prenom,nom,contact,mail,balance) VALUES (?, ?, ?, ?, ?, ?, 0)',
          [req.body.username, hashPassword(req.body.password), req.body.prenom, req.body.nom, req.body.contact, req.body.mail, req.body.balance],
          function(error, results) {
            if(error){
              res.json({ message: '0' });
            }
            else if (results.insertId != 0){
              res.json({ message: '1' });
            }
            else {
              res.json({ message: '2' });
            }
          }
        )
      }
    }
  )
});
```

Le mot de passe est hashé par une fonction utilisant le module « bcrypt »

Le login utilise aussi une fonction du module bcrypt qui permet de comparer le mot de passe crypté enregistré dans la base de données et le mot de passe entre :

```

app.post('/login', async function(req,res) {
  db.query(
    'SELECT * FROM kekelipay_users WHERE username = ?',
    [req.body.username], async function(error,results,field) {
      if(error){
        res.json({ message: error });
      }
      else if (results.length > 0){
        // req.session.username =
        // res.redirect('/dashboard');
        const passCheck = await passwordCheck(results[0]["password"],req.body.password);
        if (passCheck == true){
          res.json({ message: 'userLoggedIn', data: results });
          console.log(`User : ${req.body.username} logged in`);
          db.query(
            `UPDATE kekelipay_users SET kekelipay_users.nb_connexion = kekelipay_users.nb_connexion + 1
            [req.body.username], async function(error,results,field) {
              }
            )
          }
        else {
          res.json({ message: '0' }); //wrong username or password
        }
      }
    }
  )
}

```

```

const passwordCheck = async (storedPass,inputPass) => {
  try {
    return await bcrypt.compare(inputPass, storedPass)
  } catch (error){
    return error;
  }
};

```

La récupération de mot de passe demande le mail de l'utilisateur, sur lequel elle envoie un code à entrer :

kekedunetsuppserver@gmail.com  
to sass ▼

## Réinitialiser votre mot de passe

Bonjour,

Vous avez fait une demande de réinitialisation de votre mot de passe. Veuillez utiliser ce code :

**PZtl4**

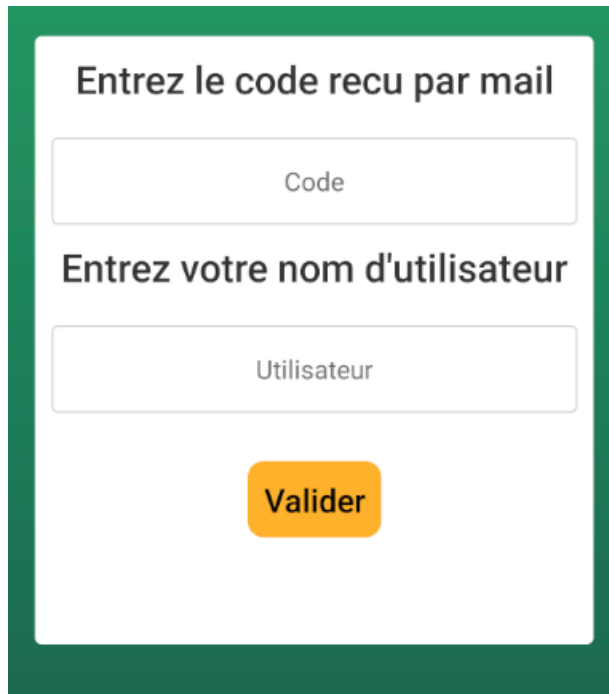
Ce code expire dans 30 minutes. Si vous n'avez pas fait cette demande, ignorez ce mail.

Merci,

L'équipe KekeliPay

Le code est généré pas une fonction assez simple :

```
const generateRandomString = (myLength) => {  
  const chars =  
    "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz1234567890";  
  const randomArray = Array.from(  
    { length: myLength },  
    (v, k) => chars[Math.floor(Math.random() * chars.length)]  
  );  
  const randomString = randomArray.join("");  
  return randomString;  
};
```



Entrez le code reçu par mail

Entrez votre nom d'utilisateur

Valider

Avec le bon code et nom d'utilisateur, la réinitialisation du mot de passe est alors possible.

Après une connexion réussie, l'utilisateur se retrouve sur cette page :

# FONCTION UTILISATEUR

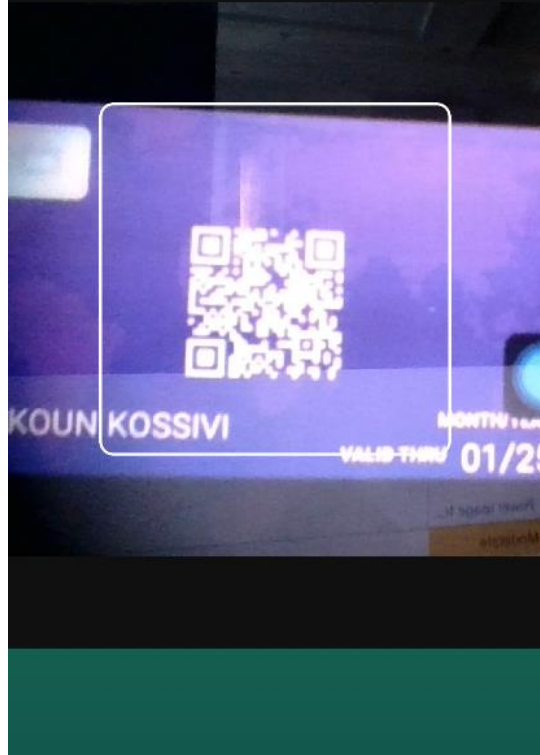


Le graph affiche les transactions que l'utilisateur a eu à faire au cours des jours des semaines.

La liste sur la gauche affiche les 5 dernières transactions et la liste de droite affiche les 5 étudiants ayant fait le plus de transaction avec l'utilisateur.

Scanner un code QR permet d'effectuer une transaction :

## Scanner une carte de Credit Kekeli



Carte de crédit  
authentifiée

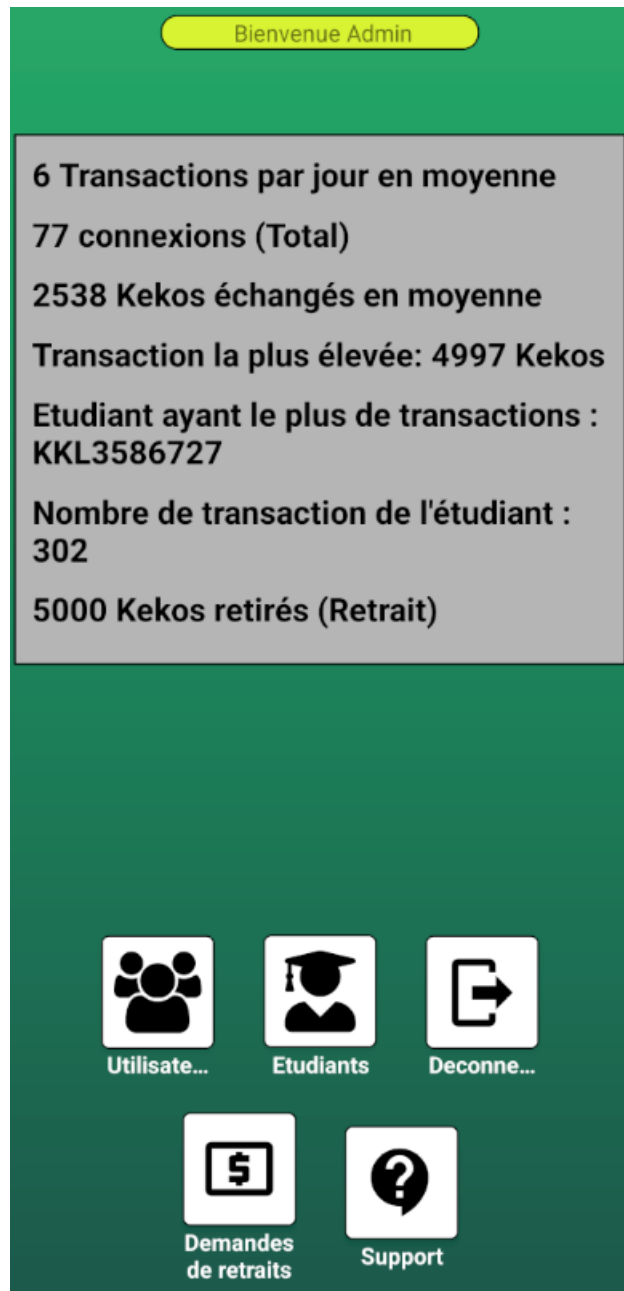
Montant à prélever :

Suivant

1	2	3	-
4	5	6	_
7	8	9	⌫
,	0	.	✓



# FONCTION ADMIN



Le menu admin permet de gérer les comptes utilisateurs (Modifier les informations de l'utilisateur, Supprimer l'utilisateur, Transférer du credit du compte utilisateur vers le compte admin), les comptes etudiants (Generer une carte de credit, desactiver une carte de credit) ainsi que les requetes (demande de retrait, demande de support) :

## < Dashboard

Rechercher par ID, Nom ou Prenom

roger94 Solde : 6000  
ID : 1 / Roger.A KIKAL

user2 Solde : 1000  
ID : 19 / M.Sassou togo1

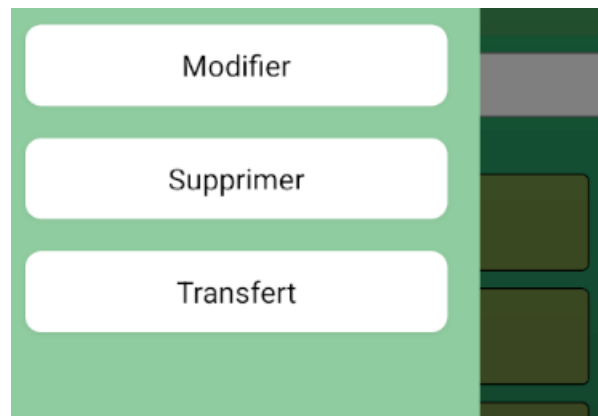
champion\_baguida3 Solde : 4000  
ID : 25 / abj Le Champion

ayoko56 Solde : 0  
ID : 27 / Kpoda ayoko

patk94 Solde : 0  
ID : 29 / ALLAGLO Patrick

Liste utilisateurs





Actions sur un utilisateur

< LogNative

Modifier le nom d'utilisateur  
Actuel : roger94

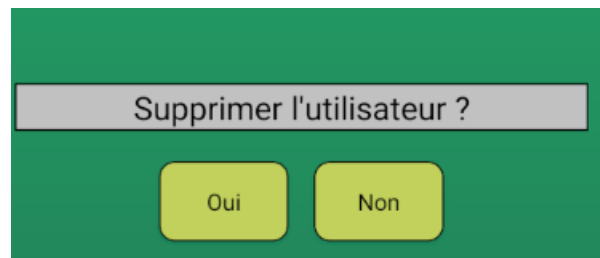
Changer le mail  
Actuel : rogera@gmail.com

Changer le contact  
Actuel : 91887532

Changer le nom et prenom  
Actuel : Roger.A

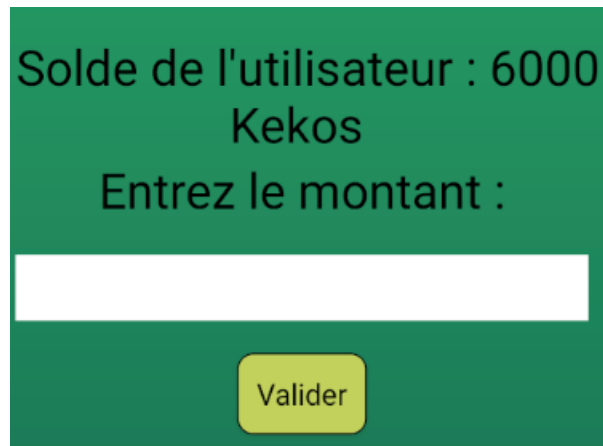
Changer le mot de passe

Modifier un utilisateur



A confirmation dialog box with a green background. At the top, a grey rectangular box contains the text "Supprimer l'utilisateur ?". Below this, there are two yellow buttons with black outlines. The left button is labeled "Oui" and the right button is labeled "Non".

Supprimer un utilisateur

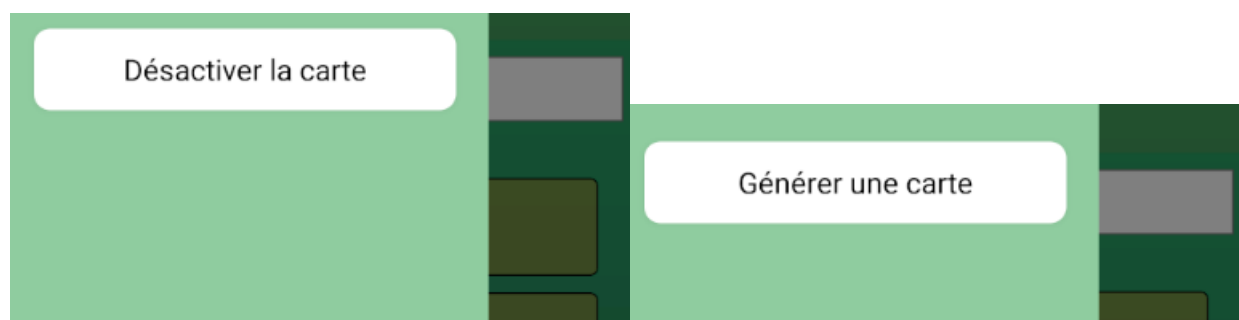


A form with a green background. It displays the text "Solde de l'utilisateur : 6000 Kekos" in a large, bold, black font. Below this, it says "Entrez le montant :". Underneath the text is a white rectangular input field. At the bottom center of the form is a yellow button with a black outline labeled "Valider".

Transférer du compte utilisateur vers le compte admin

< Dashboard	
Rechercher par ID, Nom ou Prenom	
ID : KKL4226921 Solde : 0	GNININVI Maru
ID : KKL4591759 Solde : 4001	DOUKPO Yao
ID : KKL7764631 Solde : 0	ZOEKU Bénédicte
ID : KKL1938766 Solde : 0	BOUAKA Akoua Bernadette Makafui
ID : KKL6595633 Solde : 0	WILSON BAHUN Adjé Sitou
ID : KKL4894956 Solde : 0	GABIAM Dédévi Annie Amenuvevé
ID : KKL2786553 Solde : 0	TOGBLE Yao Mensah
ID : KKL3724269 Solde : 0	KOUVONOU Nadia
ID : KKL3563829 Solde : 0	WOKOUN Kossivi Firmin
ID : KKL6211875 Solde : 0	

Liste des etudiants



Actions sur un étudiant

Lorsque l'admin selectionne un etudiant, une requete est envoyee afin de determiner si l'etudiant en question possede déjà une carte de credit active ou non, si l'etudiant ne possede pas de carte active, une nouvelle carte lui sera generer avec une date de validite de 12 mois ainsi qu'avec le pin que l'etudiant aura rentrer :

Entrez un code PIN :

Valider

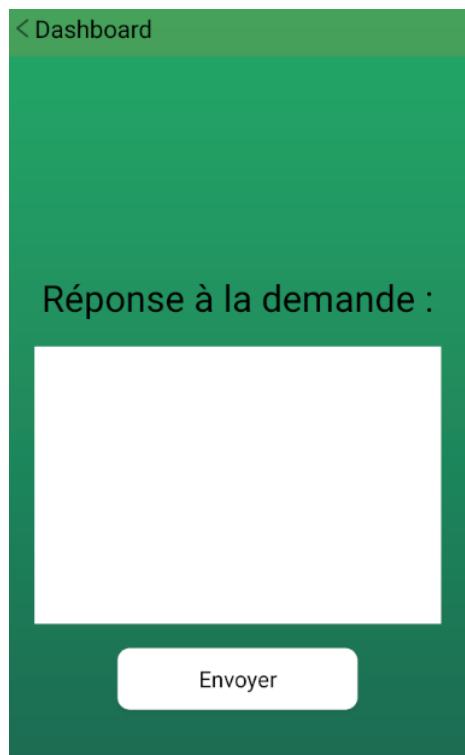
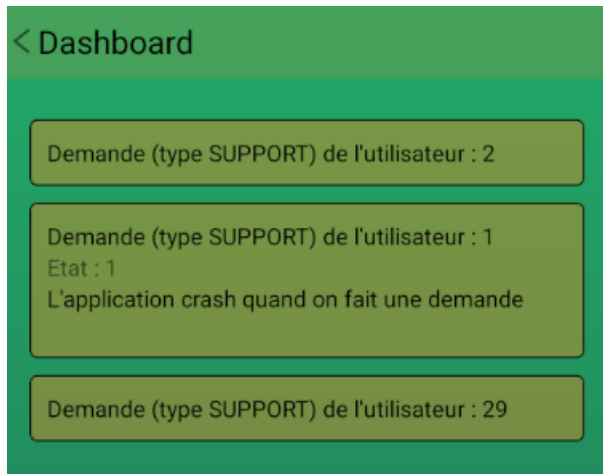
```
app.post('/generatecard', function(req,res) {
  db.query(
    'SELECT * FROM kekelipay_credit_cards WHERE user_id = ?',
    [req.body.user_id],function(error,results,field) {
      if(error){
        res.json({ message: '0' });
      }
      else if (results.length > 0){
        // req.session.username =
        res.json({ message: '2' });
        // res.redirect('/dashboard');
      }
      else {
        var currentDate = new Date();
        var validityDate = new Date(currentDate.setMonth(currentDate.getMonth() + card_validity));
        validityDate = new Date(validityDate.getFullYear(), validityDate.getMonth());
        db.query(
          'INSERT INTO kekelipay_credit_cards (id,user_id,validity,pin,active) VALUES (?,?,,?,1)',
          [uuid.v4(),req.body.user_id,validityDate,req.body.pin_code],function(error,results,field) {
            if(error){
              res.json({ message: error });
            }
            else if (results.length > 0){
              // req.session.username =
              res.json({ message: results });
              // res.redirect('/dashboard');
            }
            else{
              db.query(
                'SELECT * FROM kekelipay_credit_cards WHERE user_id = ?',
                [req.body.user_id],function(error,results,field) {
                  if(error){
                    res.json({ message: error });
                  }
                  else if (results.length > 0){
                    // req.session.username =
                    res.json({ message: '1', results: results });
                    // res.redirect('/dashboard');
                  }
                }
              );
            }
          }
        );
      }
    }
  );
});
```



La carte de crédit possède un code QR générer automatiquement à partir d'un id crypter générer par une fonction issue de la librairie uuid. Cela permettra au système d'identifier la carte lors du scan de la carte.



L'admin peut soit décliner ou accepter la demande de retrait. L'utilisateur qui a fait la demande sera notifier par mail de la suite de sa demande.



L'admin peut répondre à une requête de support d'un utilisateur. L'utilisateur qui a fait la demande sera notifier par mail de la suite de sa demande.

# BILAN

L'expérience acquise au cours de ce stage fut très enrichissante. La découverte d'un nouveau Framework a été tout d'abord un peu difficile mais au fil des jours, je me suis senti de plus en plus à l'aise dans cet environnement.

Professionnellement, ce stage m'a permis de découvrir plusieurs aspects du développement logiciel dans un milieu professionnel.

Personnellement, le fait de devoir s'organiser soi-même pour apprendre un nouvel environnement de développement a été très bénéfique car je me sens plus sûr et prêt dans le cas où je dois (pour des projets futurs) m'adapter et découvrir de nouveaux environnements de programmation.