

Spécification Technique pour la page qrCodePage.tsx

1. Introduction

La page `qrCodePage.tsx` est une page React Native dédiée à la lecture de QR codes. Il permet aux utilisateurs de scanner des QR codes et d'envoyer les informations scannées à un serveur via une connexion WebSocket. Ce document décrit en détail les spécifications techniques de la page `qrCodePage.tsx`.

2. Objectifs

- Scanner des QR codes via l'appareil mobile.
- Envoyer les informations scannées (valeur du QR code) et le montant total de la transaction à un serveur via WebSocket.
- Afficher un message une fois le QR code scanné.
- Réinitialiser le montant total et naviguer vers la page d'accueil lors du retour.

3. Dépendances

- `react-native`
- `react-native-vision-camera`
- `react-router-native`
- `@react-native-material/core`
- Contexte `TotalAmountContext` pour gérer le montant total de la transaction.
- Contexte `SocketContext` pour gérer la connexion WebSocket.

4. Interface

4.1. Props

- `clientId`: Identifiant unique du client. Utilisé pour l'envoi des données de paiement.

4.2. État Local

- `scanned`: État local pour indiquer si un QR code a été scanné. Initialisé à `false`.

5. Fonctionnalités

5.1. Initialisation de la Permission de la Caméra

- `useEffect` pour demander la permission d'utiliser la caméra à la montée du composant si elle n'est pas encore accordée.

```
useEffect(() => {  
  if (!hasPermission) {  
    requestPermission();  
  }  
}, [hasPermission]);
```

5.2. Scanner de QR Code

- `codeScanner`: Hook `useCodeScanner` pour scanner des QR codes.

```
const codeScanner = useCodeScanner({  
  codeTypes: ['qr', 'ean-13'],  
  onCodeScanned: codes => {  
    const codeValue = codes[0]?.value ?? '';  
    console.log(`Scanned ${codes.length} codes ! | `, codeValue);  
    sendPayment(codeValue); // Envoyer le paiement via le socket  
    setScanned(true); // Mettre à jour l'état pour indiquer qu'un  
    code a été scanné  
  },  
});
```

5.3. Envoi des Informations de Paiement

- `sendPayment`: Fonction pour envoyer les données de paiement via WebSocket.

```
const sendPayment = (codeValue: string) => {  
  if (socket) {  
    console.log("client id qrcode :" + clientID);  
    socket.emit('payement', { codeValue, totalAmount, clientID });  
  }  
};
```

5.4. Retour à la Page d'Accueil

- `handleBack`: Fonction pour réinitialiser le montant total et naviguer vers la page d'accueil.

```
const handleBack = () => {  
  resetTotalAmount(); // Réinitialiser le montant total  
  navigate('/'); // Naviguer vers la page d'accueil  
};
```

6. Interface Utilisateur

- Titre: "Lecture de QR Code".
- Affichage du flux de la caméra pour scanner un QR code (affiché uniquement si `scanned` est `false`).
- Affichage d'un message une fois le QR code scanné.
- Lien "Retour" pour retourner à la page d'accueil et réinitialiser le montant total.

```
return (  
  <Box style={{ height: '100%', width: '100%' }}>  
    <Box style={{ height: '90%', width: '100%', backgroundColor:  
'white', justifyContent: 'center', alignItems: 'center' }}>  
      {device && !scanned && (  
        <Camera style={{ width: '100%', height: '100%' }}  
device={device} isActive={true} codeScanner={codeScanner}  
enableZoomGesture />  
      )}  
      {scanned && (  
        <Text style={{ fontSize: 24, color: 'green' }}>QR code  
scanné</Text>  
      )}  
    </Box>  
    <Text onPress={handleBack} style={{ position: 'absolute',  
bottom: 20, left: 20, fontSize: 24, color: 'blue' }}>Retour</Text>  
  </Box>  
);
```

7. Styles

Styles définis pour le conteneur principal, le conteneur de la caméra et le lien "Retour".

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#fff',
  },
  cameraContainer: {
    height: '90%',
    width: '100%',
    backgroundColor: 'white',
    justifyContent: 'center',
    alignItems: 'center',
  },
  scannedText: {
    fontSize: 24,
    color: 'green',
  },
  backLink: {
    position: 'absolute',
    bottom: 20,
    left: 20,
    fontSize: 24,
    color: 'blue',
  },
});
```

8. Conclusion

La page `qrCodePage.tsx` permet de scanner des QR codes et d'envoyer les informations scannées à un serveur via une connexion WebSocket. Elle affiche un message une fois le QR code scanné et permet de réinitialiser le montant total et de retourner à la page d'accueil.

Cette spécification technique fournit une vue d'ensemble complète de la structure et des fonctionnalités du composant, facilitant ainsi sa compréhension et son utilisation.