**Tích hợp thanh toán Payment**

1. **Tích hợp với Stripe**

**Tính năng**: Hỗ trợ thanh toán với các thẻ quốc tế như Visa, Mater card thông qua Apple/Android pay

**Package**: [Tipsi-stripe](https://github.com/tipsi/tipsi-stripe)

**Example**: <https://github.com/tipsi/tipsi-stripe/tree/master/example>

**Document**: <https://tipsi.github.io/tipsi-stripe/docs/index.html>

**Hướng dẫn tích hợp**

Chạy lệnh: npm install --save tipsi-stripe để tiến hành cài đặt thư viện

Chạy lệnh: react-native link tipsi-stripe để thực hiện liên kết thư viện với project

Link install detail: <https://tipsi.github.io/tipsi-stripe/docs/installation.html>

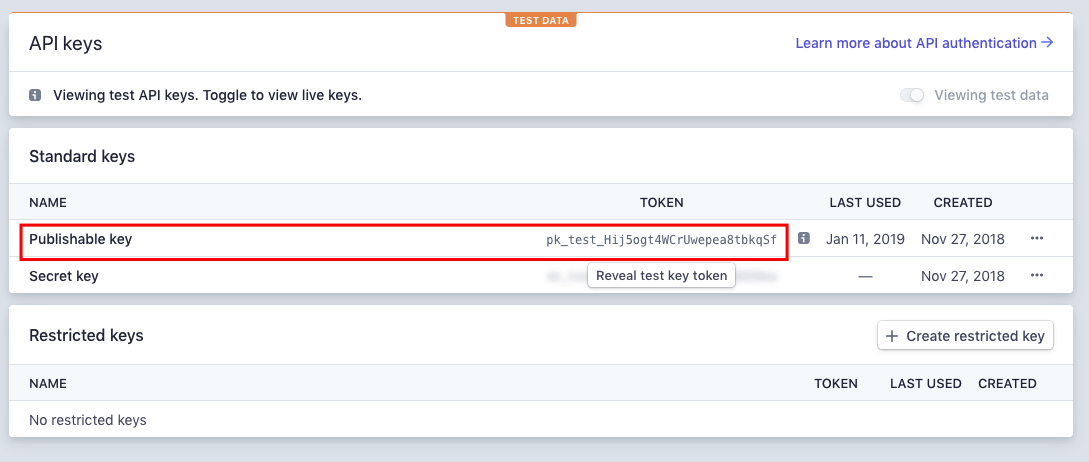
**Usage**

Link: <https://tipsi.github.io/tipsi-stripe/docs/usage.html>

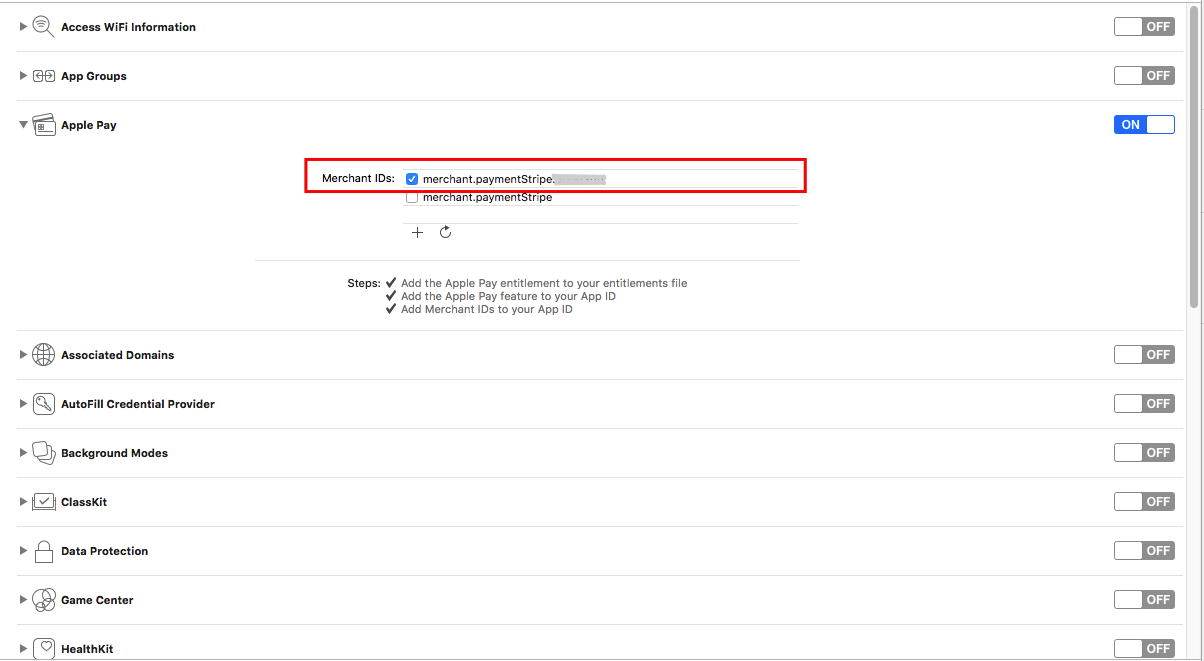
Link get PUBLISHABLE\_KEY & MERCHANT\_ID

<https://dashboard.stripe.com/account/apikeys>

* Publish key



* Merchant id



Hàm xử lý thanh toán gọi Apple Pay

try {

**const** token = await stripe.paymentRequestWithNativePay({

shippingMethods: [],

countryCode: 'SG',

currencyCode: 'SGD'

},

[{

label: `Payment for order #${invoice} on the Shop APP.`,

amount: `${cart.totalPrice}`,

}])

await stripe.completeNativePayRequest()

await stripe.cancelNativePayRequest()

*// Call api update checkout success*

if(token && token.tokenId){

this.callCheckoutSuccess(invoice, token.tokenId)

}

} catch (error) {

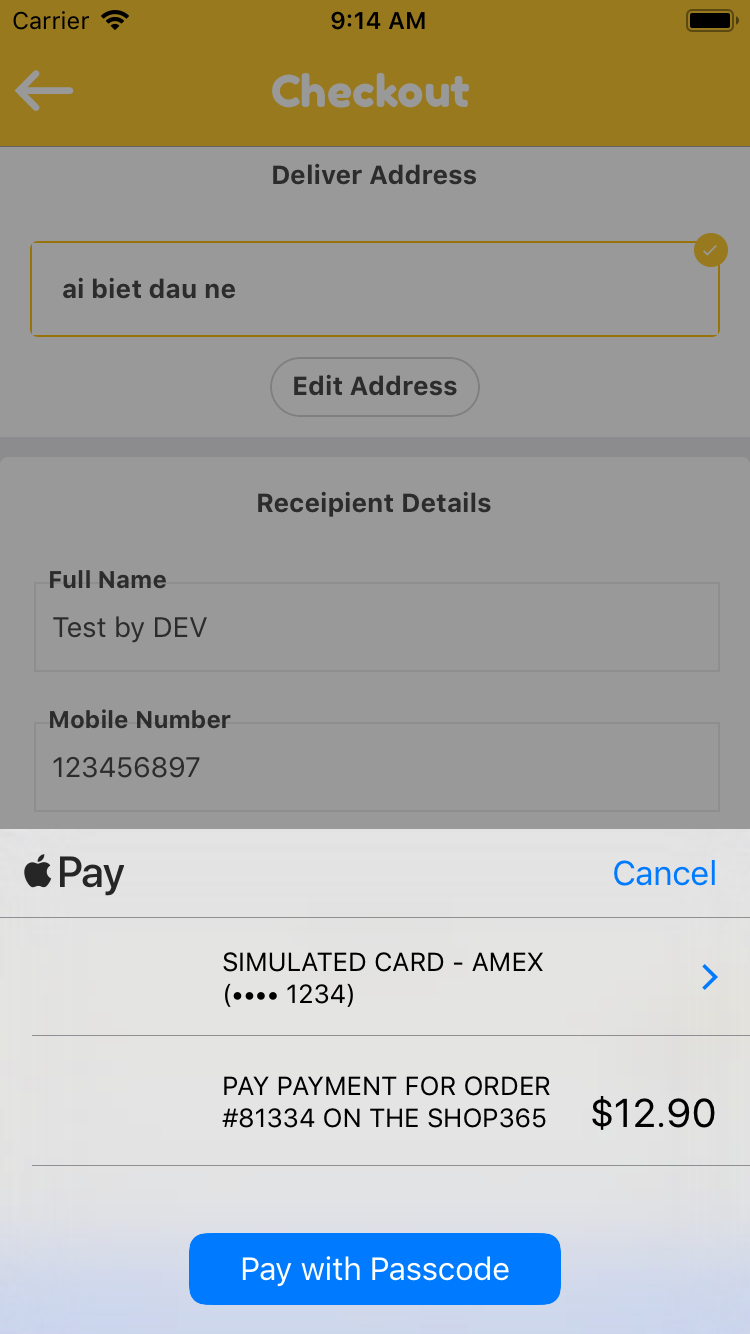
showToast({

text: `Error: ${error.message}`,

duration: 2000

});

}



Hàm xử lý thanh toán gọi Android Pay

try {

**const** token = await stripe.paymentRequestWithNativePay({

total\_price: `${cart.totalPrice}`,

currency\_code: 'USD',

shipping\_address\_required: false,

phone\_number\_required: false,

shipping\_countries: [

'VN', 'SG'

],

line\_items: cart.listitem

})

this.setState({ loading: false })

*// Call api update checkout success*

this.callCheckoutSuccess(invoice, token)

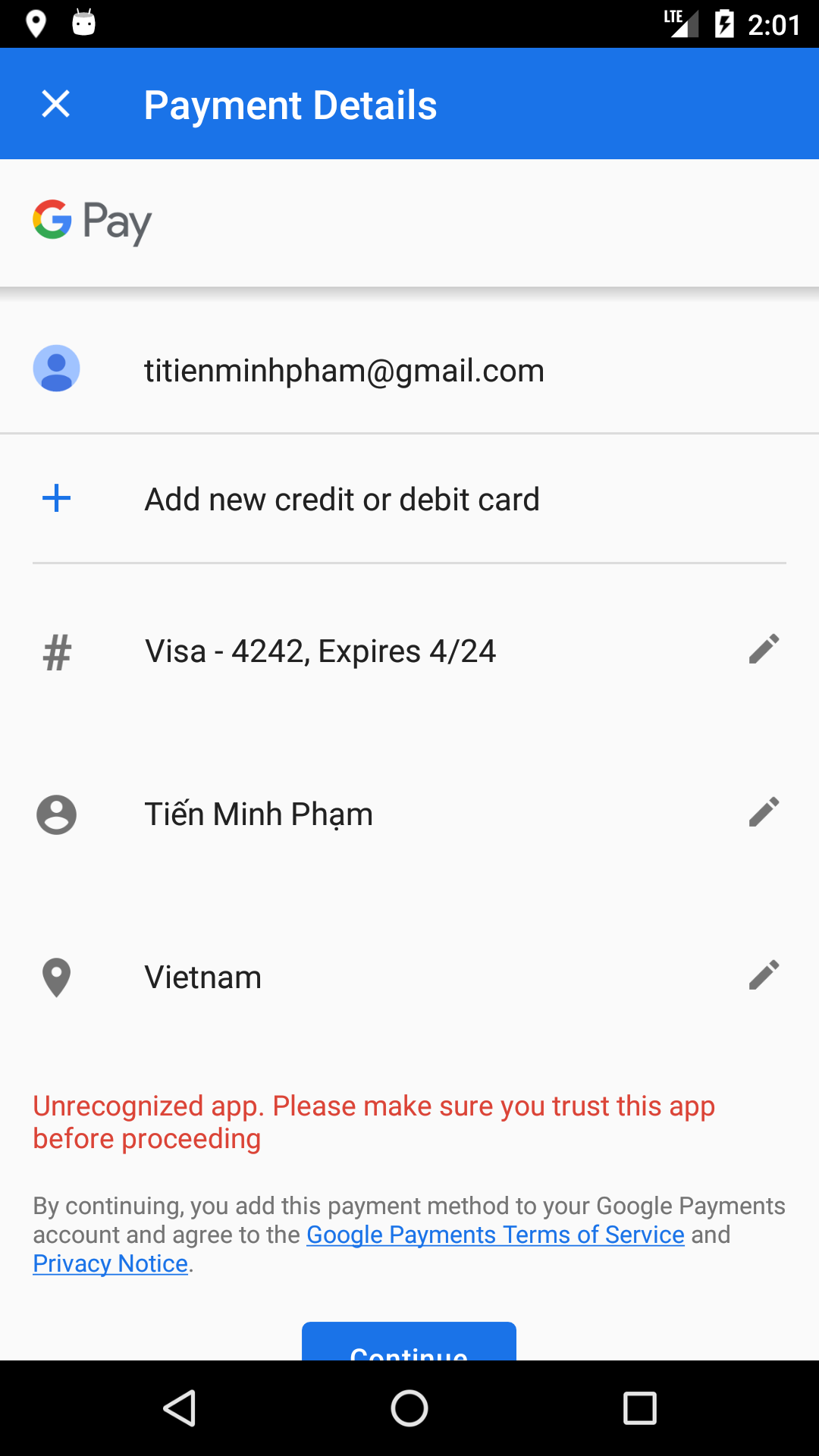
} catch (error) {

console.log('Payment credit error: ', error);

this.setState({ loading: false })

}

Lưu ý cần đăng nhập tài khoản Google trước khi thực hiện thanh toán với Android Pay



Đối với thanh toán với Stripe, người dung không thể thực hiện thanh toán ảo. Nhưng có thể check log [để kiểm tra trạng thái thanh toán](https://dashboard.stripe.com/test/logs?method%5b%5d=post&method%5b%5d=delete&direction%5b%5d=connect_in&direction%5b%5d=self).

**Thực hiện Checkout Stripe không thông qua Apple/Google pay**

Với cách thức này, ta không cần phải gọi đến phương thức thanh toán của Apple/Google pay. Thay vào đó ta sẽ dung hàm Charge được sử dụng dước Backend để thực hiện giao dịch.

**Qui trình thực hiện** : *Khi call hàm* ***paymentRequestNativePay*** *-> Popup nhập thông tin thẻ hiện lên, sau khi xác nhận thông tin thẻ thành công với Stripe. Phía Stripe sẽ trả về token tương ứng của thẻ. -> Tiếp tục call api với Backend để gửi thông tin token + thông tin đơn hàng (giá, số lượng, invoice) để thanh toán với Stripe. Cuối cùng lưu thông tin transaction vào database.*

Sử dụng hàm paymentRequestNativePay:

handleStripePayment = **async** (invoice, quantity, price, fullName) **=>** {

**const** user = userData.auth;

try {

**const** token = await stripe.paymentRequestWithCardForm({

*// Only iOS support this options*

smsAutofillDisabled: true,

*// requiredBillingAddressFields: 'full', // full \ zip \ disable*

prefilledInformation: {

email: '',

phone: '',

billingAddress: {},

},

})

*// To do something*

} catch (error) {

showToast({

text: `${error.message}`,

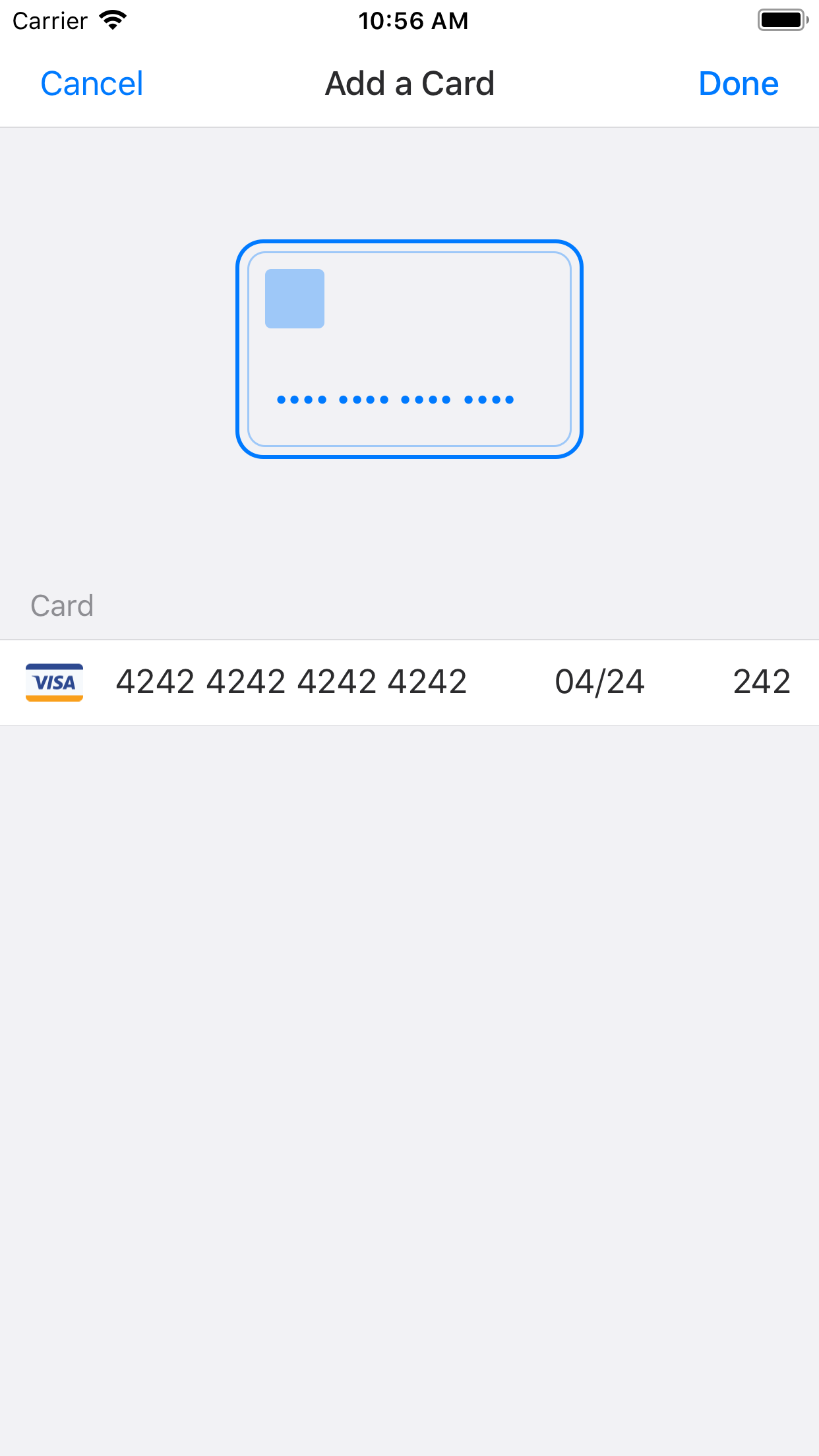
duration: 2000

});

}

}

Popup nhập thông tin thẻ khai báo lấy token với Stripe



Kết quả giao dịch:

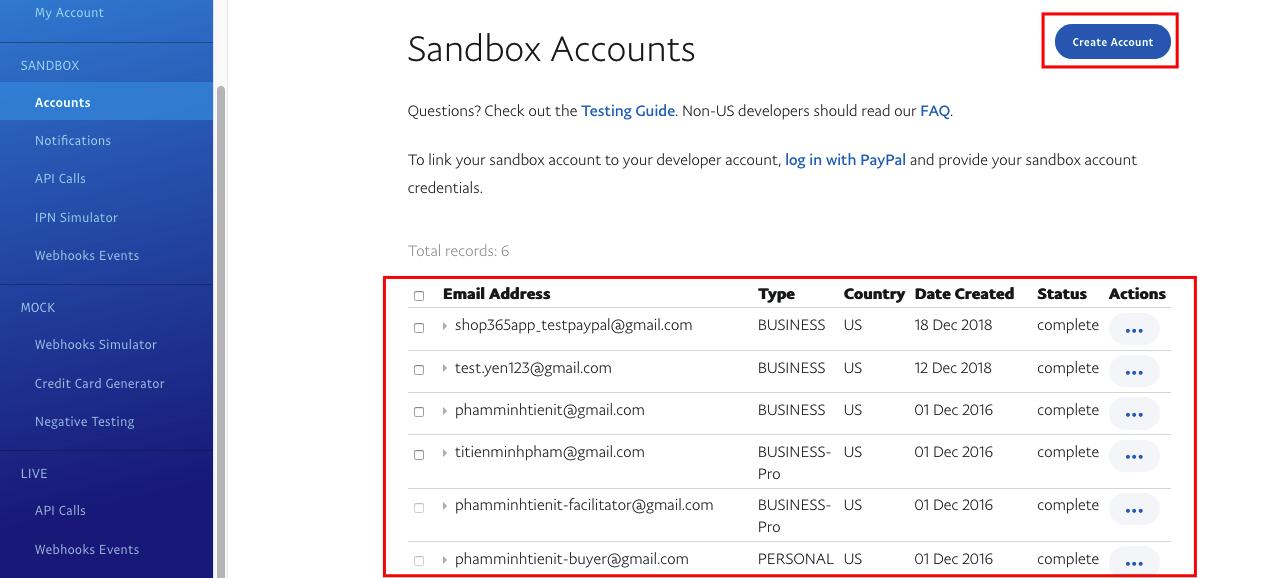


1. **Tích hợp thanh toán với Paypal**

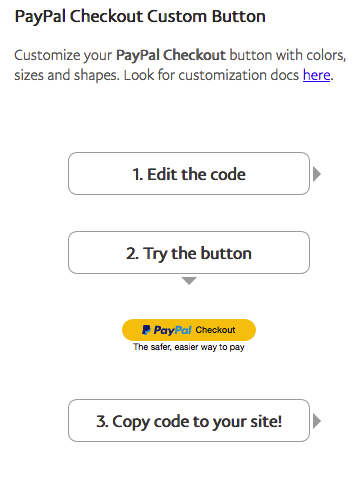
**Tính năng**: Thực hiện thanh toán với Paypal

Khởi tạo tài khoản khoản với Paypay: <https://www.paypal.com>

Sau khi đã có tài khoản, trước khi tiến hành tích hợp ta sẽ khởi tạo tài khoản để thực hiện thanh toán với môi trường [Sanbox](https://developer.paypal.com/developer/accounts/) của Paypal.



Custom button payment with Paypay: <https://developer.paypal.com/demo/checkout/#/pattern/checkout>



Các thuộc tính được sử dụng trong Button: <https://developer.paypal.com/docs/checkout/how-to/customize-button/#customization-example>

Tích hợp trong với React Native code.

Để nhúng button payment của Paypal vào project thì ta dung webview để hiển thị. Tuy nhiên cần phải truyền các thông tin liên quan (id, số lượng, thành tiền) của 1 đơn hàng trước khi gọi thanh toán với button này.

Để làm điều đó ta sử dụng thuộc tính javaScriptEnabled của Webview

Code nhúng button Paypay vào Webview

<WebView

ref={ref **=>** (this.webview = ref)}

style={{

marginTop: (Platform.OS === 'android') ? 0 : 20,

height: deviceHeight - Anim.headerHeight,

}}

source={(Platform.OS === 'android') ? {uri: 'file:///android\_asset/index.html'} : require('../html/index.html')}

onMessage={(event) **=>** this.onMessage(event, invoice)}

injectedJavaScript={Platform.OS == 'android' ? jsCode : patchPostMessageJsCode}

javaScriptEnabledAndroid={true}

javaScriptEnabled={true}

scalesPageToFit={Platform.OS == 'android' ? true : false}

/>

Chỗ source phải khai báo thư mục index.html trong asset vì khi build bản Release của Android, trên thiết bị thật sẽ không gọi được file này dẫn đến button không hiển thị.

Biến injectedJavaScript để thực hiện nhúng thông tin đơn hàng vào button.

Code mẫu:

**const** jsCode = `price = ${price}; ` + `invoice = ${invoice}`;

**const** patchPostMessageFunction = **function**() {

**var** originalPostMessage = window.postMessage;

**var** patchedPostMessage = **function**(message, targetOrigin, transfer) {

originalPostMessage(message, targetOrigin, transfer);

};

patchedPostMessage.toString = **function**() {

return String(Object.hasOwnProperty).replace('hasOwnProperty', 'postMessage');

};

window.postMessage = patchedPostMessage;

};

**const** patchPostMessageJsCode = '(' + String(patchPostMessageFunction) + ')();' + `price = ${price}; ` + `invoice = ${invoice}`;

Hàm nhận kết quả trả về sau khi xử lý thanh toán:

onMessage(event, invoice) {

**var** result = JSON.parse(event.nativeEvent.data) || {};

cartData.invoice = invoice;

console.log('result data ', result)

*//Checkout is completed*

if(Platform.OS == "android"){

if(result && result.id && result.state === 'approved')

{

cartData.paymentToken = result.id;

MainNav.navigate('Checkout', {invoice: invoice, paymentToken: result.id})

return

}

}

else{

if(result && result.paymentID && result.returnUrl)

{

cartData.paymentToken = result.paymentID;

MainNav.navigate('Checkout', {invoice: invoice, paymentToken: result.paymentID})

return

}

}

*// Checkout is cancle*

if(result && result.cancelUrl && result.paymentToken)

{

showToast({

text: `Cancel checkout with Paypal. Please try again`,

duration: 4000

});

return

}

*// Checkout is error*

showToast({

text: `Error checkout with Paypal. Please try again`,

duration: 4000

});

}

Kiểm tra trạng thái thanh toán đơn hàng: <https://developer.paypal.com/developer/notifications/>

