

## **Séance 13 :** Introduction au développement mobile : → avec **React Native**

---

INF27507 – Technologies du commerce électronique

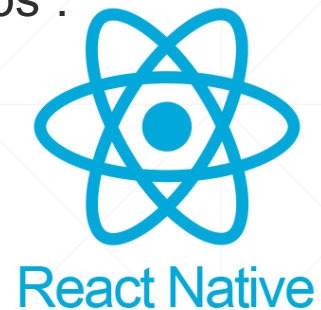
Prof. Yacine YADDADEN, Ph. D.

# Plan

1. Qu'est-ce que React Native ?
2. Comment fonctionne-t-il ?
3. Les composants de base
4. Navigation et gestion des états
5. Les tests unitaires
6. Environnement de développement
7. Les Framework UI et icônes
8. Premier projet – Liste des courses :
  - a. Architecture et Composants
  - b. Système de Navigation
  - c. Création de l'interface UI
  - d. Événement et accès aux champs
  - e. Intégration avec API REST
  - f. Renommer l'application et icône
  - g. Génération de la *release*
9. Questions & Discussion

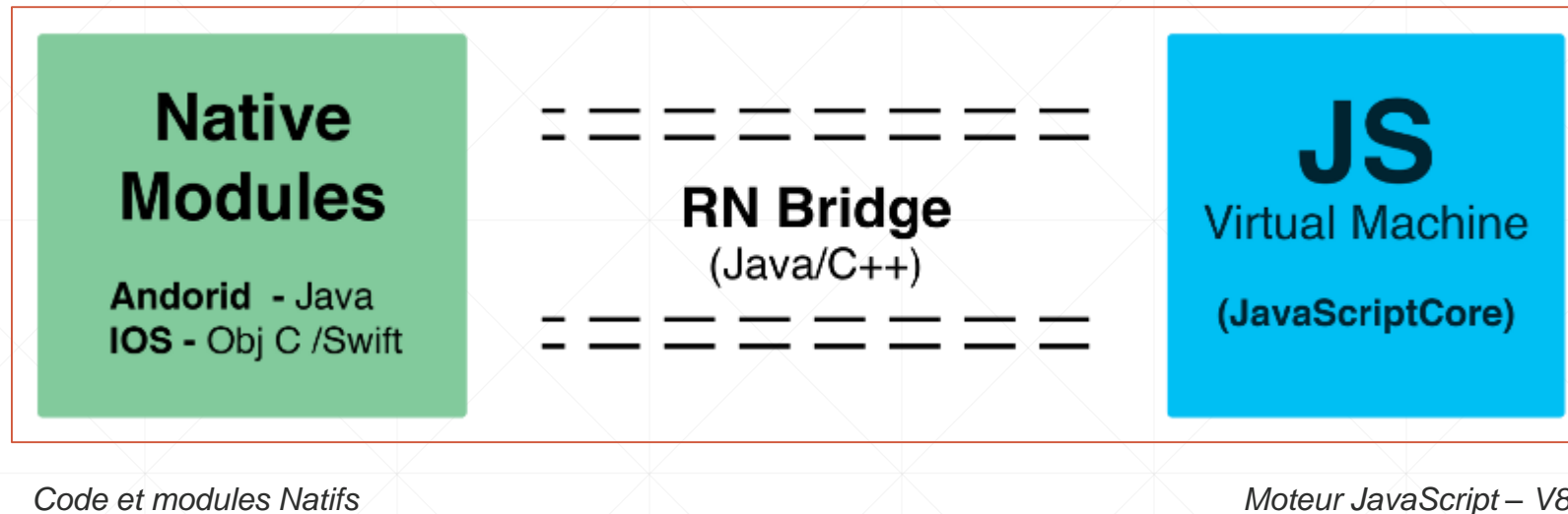
# Qu'est-ce que React Native ?

- **Définition :** « *C'est un Framework destiné à la création d'applications mobiles (iOS et Android) hybrides. Il est basé sur le JavaScript et React.js et ces différents composants.* »,
- Caractéristiques :
  - **Type d'applications :** *Hybride (multiplateformes),*
  - **Langage de programmation :** *JavaScript,*
  - **Licence :** *Open Source.*
- Il est basé sur React.js et par conséquent, on fait d'une pierre deux coups :
  - Développement Web avec les **SPA** (Single Page Applications),
  - Développement mobile.
- On bénéficie également des différents packages disponibles.



# Comment fonctionne-t-il ?

Il est responsable de la **communication** entre les **threads natifs** et les **threads JavaScript**





# Les composants de base

- **React Native** vient avec différents *composants prédéfinis*, parmi lesquels :

`<View>`

- **View** qui est l'élément fondamentale pour chaque composant ou vue.

`<Text>`

- **Text** permettant d'afficher du texte.

`<Image>`

- **Image** permettant d'afficher une image.

`<TextInput>`

- **TextInput** permettant d'insérer du texte à partir du clavier.

`<ScrollView>`

- **ScrollView** permettant de conteneur scrollable avec différents composants.

`<StyleSheet>`

- **StyleSheet** permettant d'ajouter une couche d'abstraction pour le style CSS.

- Il y a également d'autres composants, on peut les retrouver ici :

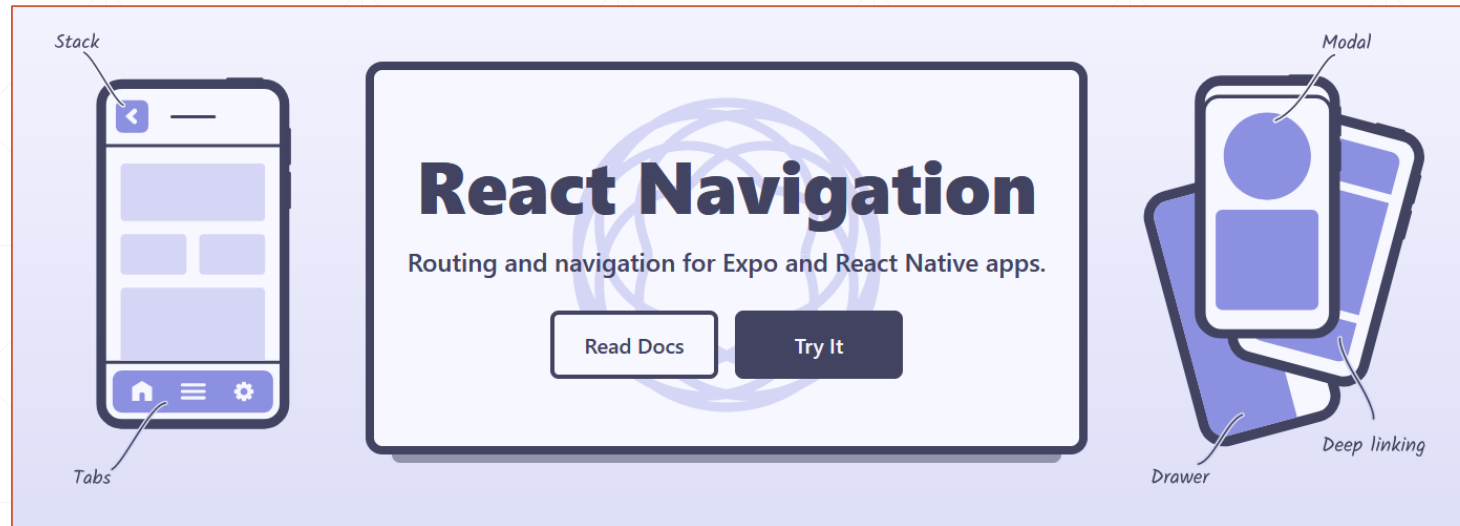
- Lien : <https://reactnative.dev/docs/components-and-apis>

# Plan

1. Qu'est-ce que React Native ?
2. Comment fonctionne-t-il ?
3. Les composants de base
4. **Navigation et gestion des états**
5. Les tests unitaires
6. Environnement de développement
7. Les Framework UI et icônes
8. Premier projet – Liste des courses :
  - a. Architecture et Composants
  - b. Système de Navigation
  - c. Création de l'interface UI
  - d. Événement et accès aux champs
  - e. Intégration avec API REST
  - f. Renommer l'application et icône
  - g. Génération de la *release*
9. Questions & Discussion

# Navigation et gestion des états

Afin d'avoir un **système de navigation** sur l'application mobile avec **React Native**, il est recommandé d'utiliser :



Lien : <https://reactnavigation.org/>

# Navigation et **gestion des états**

La gestion des **états** (*données*) au niveau d'une application mobile avec **React Native** peut s'effectuer avec un des moyens ci-dessous :

1. Gestion **locale** au niveau de *chacun des composants* :
  - *Il est recommandé dans ce cas d'utiliser le système des **Hooks**<sup>1</sup> inclus directement dans **React**.*
2. Gestion **globale** ou centralisée :
  - *Il est recommandé quand l'application contient beaucoup de composants. On peut faire appel à des bibliothèques externes telles que : MobX, **Redux**<sup>2</sup> ou Recoil.*

<sup>1</sup> <https://fr.reactjs.org/docs/hooks-intro.html>

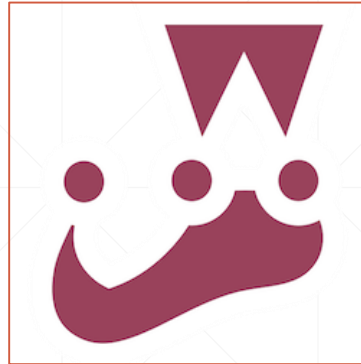
<sup>2</sup> <https://redux.js.org/>



# Les tests unitaires

Comme pour les *application Web* avec **React**, il est recommandé d'utiliser :

- La *bibliothèque* **JEST** dédiée aux *tests unitaires* avec **React** :



- Lien : <https://jestjs.io/fr/>

# Environnement de développement

- Afin de mettre en place l'environnement de développement, il faut installer :

- ✓ **Node.js** (LTS) : <https://nodejs.org/en/>

- ✓ **OpenJDK 11** : <https://openjdk.java.net/>

- ✓ **Interpréteur Python 3.9** : <https://www.python.org/>

- ✓ **Android Studio** : <https://developer.android.com/studio>

- Il faut également installer les paquets suivants :

*C'est un outil permettant de créer des projets avec l'ensemble des fichiers nécessaire pour React Native.*

```
npm install -g expo  
npm install -g expo-cli
```

*Outil de diagnostic permettant de vérifier que l'ensemble des prérequis sont bien installés.*

```
npm install -g react-native-cli
```

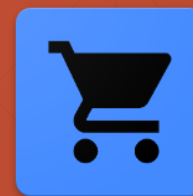
*C'est une autre alternative avec des fonctionnalités supplémentaires très utiles.*

```
npx @react-native-community/cli doctor
```

# Premier projet

---

Liste des courses – **Go Shop !**

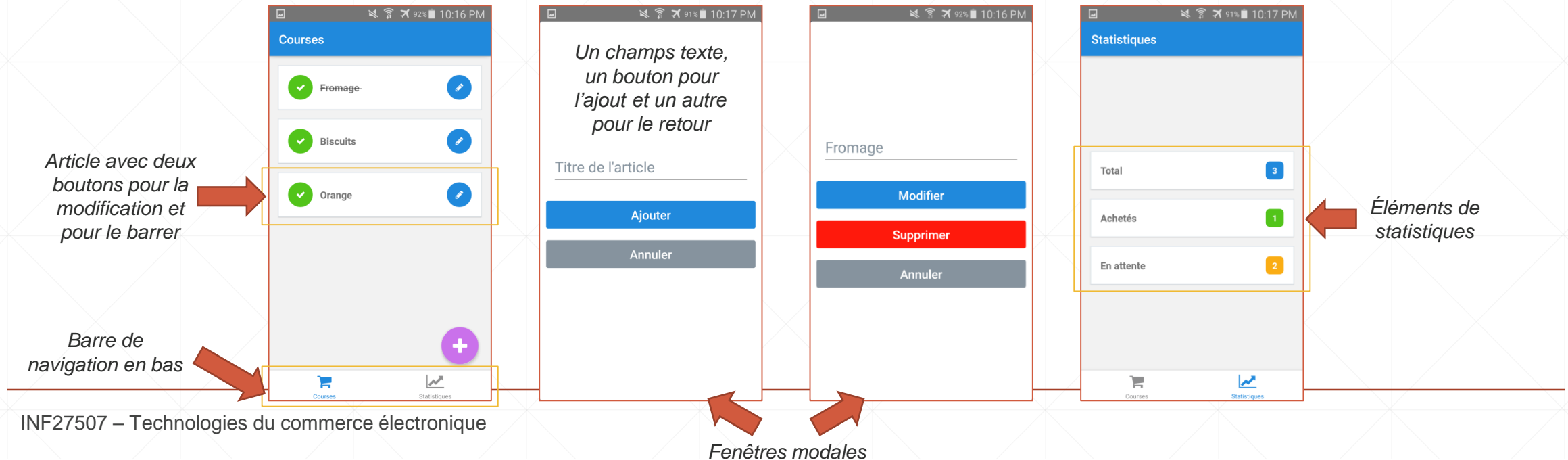


# Plan

1. Qu'est-ce que React Native ?
2. Comment fonctionne-t-il ?
3. Les composants de base
4. Navigation et gestion des états
5. Les tests unitaires
6. Environnement de développement
7. Les Framework UI et icônes
8. **Premier projet – Liste des courses :**
  - a. Architecture et Composants
  - b. Système de Navigation
  - c. Création de l'interface UI
  - d. Événement et accès aux champs
  - e. Intégration avec API REST
  - f. Renommer l'application et icône
  - g. Génération de la *release*
9. Questions & Discussion

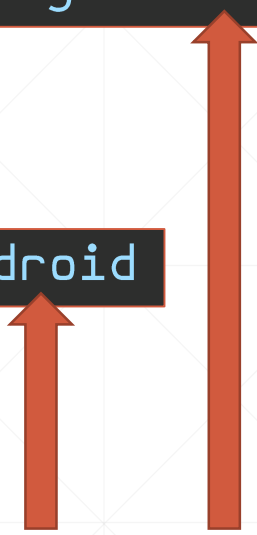
# Architecture et Composants

- L'application à créer permettra de gérer une liste des courses quand on fait son épicerie. On pourra ajouter des articles, les supprimer, les modifier, les barrer et aussi avoir des statistiques.
- Voici un aperçu des composants de l'application :

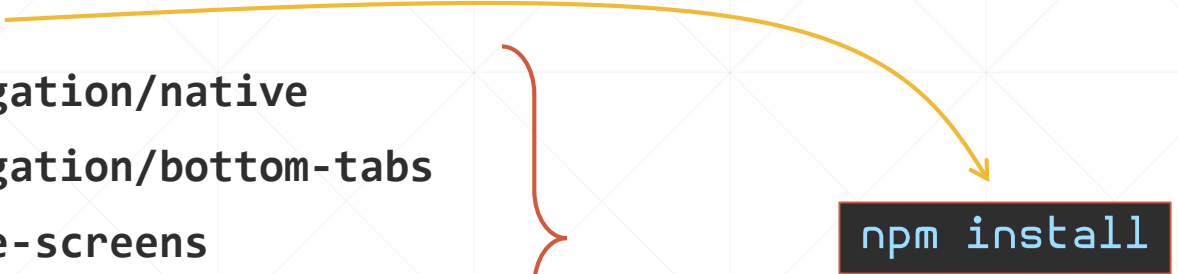





# Création du projet

- Afin de créer un nouveau projet :
    - Première méthode : `npx react-native init projectName`
    - Deuxième méthode : `expo init project-name`
  - Pour lancer le projet :
    - Première méthode : `npx react-native run-android`
    - Deuxième méthode : `npm start`
  - Pour la démonstration qui suivra, j'utiliserai la **première méthode**.
- 

# Système de Navigation

- Pour avoir le système de navigation choisi, il faut installer les paquets suivants avec la commande :
  - `@react-navigation/native`
  - `@react-navigation/bottom-tabs`
  - `react-native-screens`
  - `@react-navigation/elements`
  - `react-native-safe-area-context`

```
npm install
```
- Afin d'avoir accès aux icônes, on doit également installer le paquet suivant :
  - `react-native-vector-icons`  `npx react-native link react-native-vector-icons`

*Lier les icônes à l'application*
- Le code pour le système de navigation se trouve dans :
  - Chemin dans l'application : `./src/components/Navigations.js`

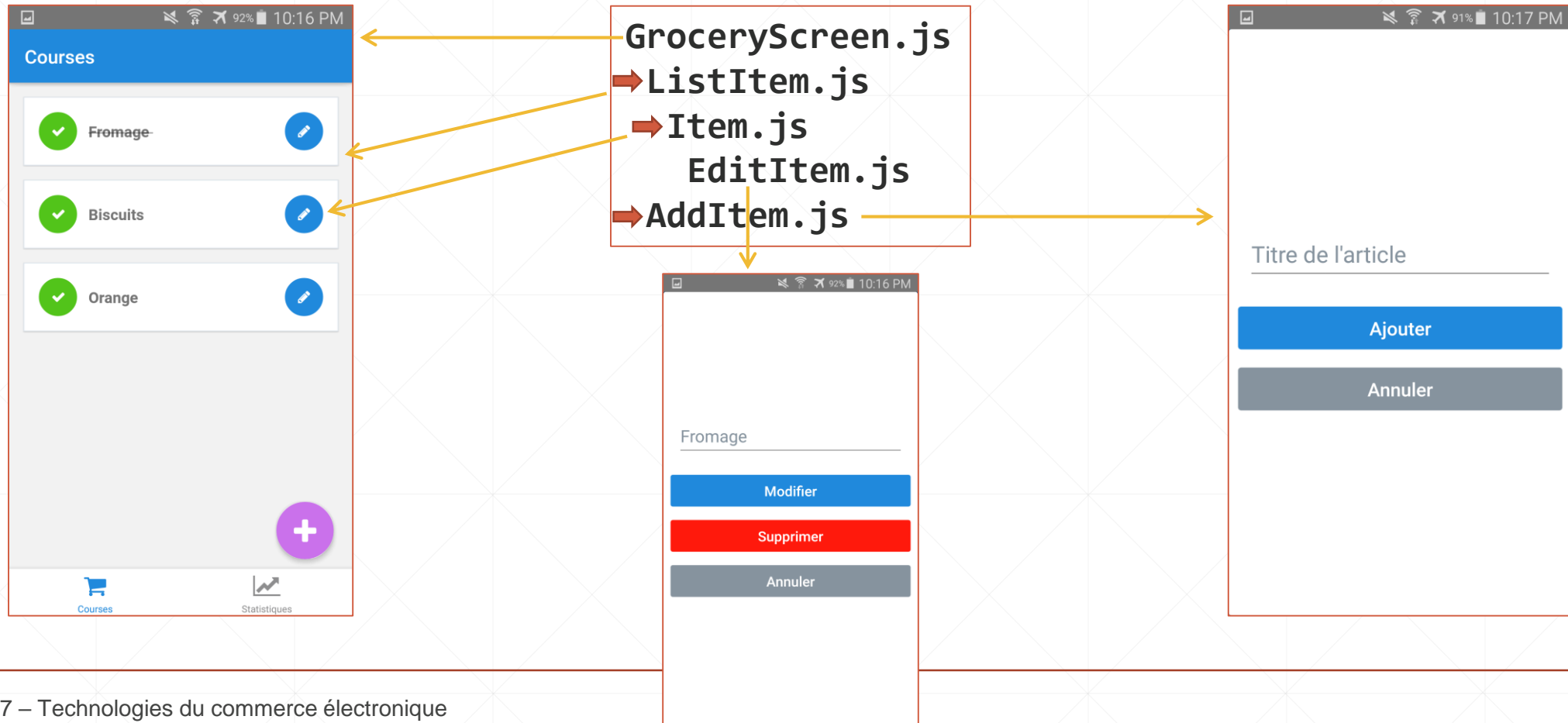
# Création de l'interface UI

- Pour l'interface utilisateur, il est possible d'utiliser différents **Framework UI** tels que :
  - Native Base : <https://nativebase.io/>
  - ➔ ▪ **React Native Elements** : <https://reactnativeelements.com/>
  - React Native Paper : <https://callstack.github.io/react-native-paper/>
- Afin d'utiliser **React Native Elements**, il faut installer :
  - `react-native-elements`
- On aura, ainsi, accès à l'ensemble des composants qui viennent avec,
- On utilisera la documentation fournie pour afficher les différents composants.

## Création de l'interface UI (Suite)

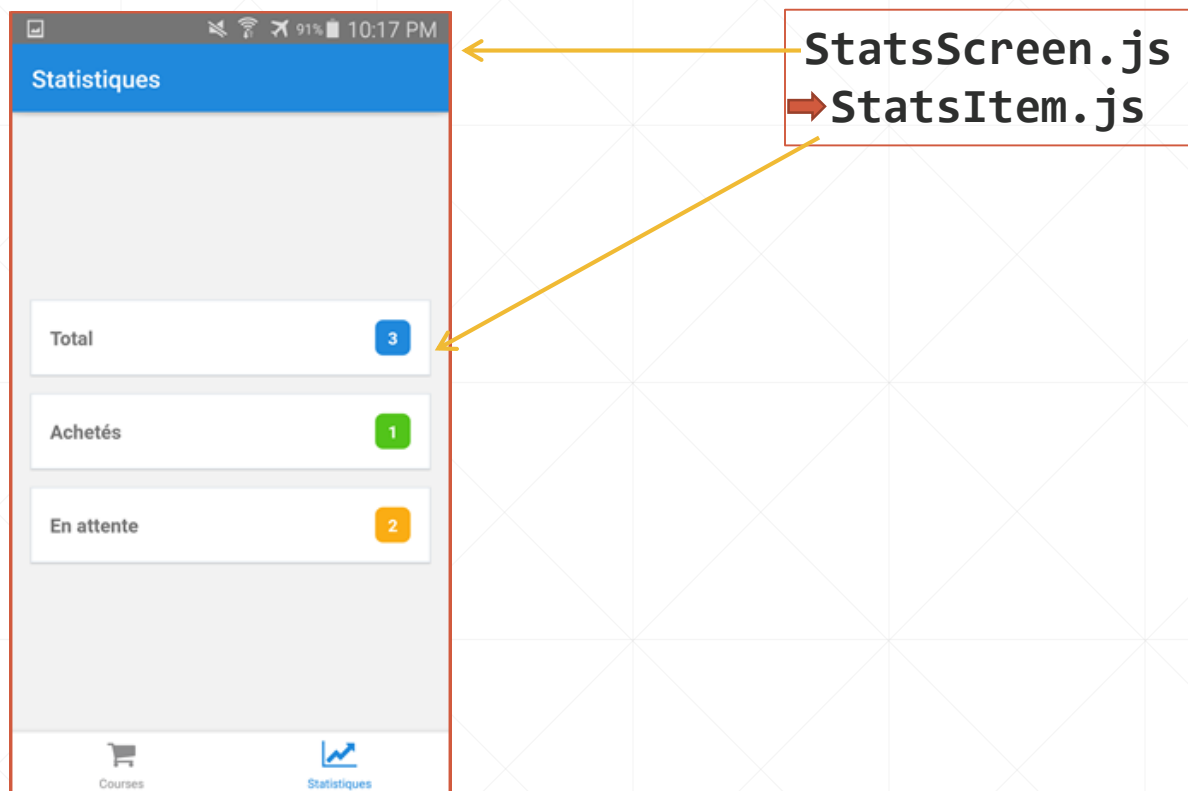
- On utilisera les *composants de base* suivants – **React Native** :
  - **View, SafeAreaView, ScrollView, Text, StyleSheet.**
- On utilisera les *composants UI* suivants – **React Native Elements** :
  - **ThemeContext** : *afin d'accéder aux couleur utilisées dans le thème du Framework UI,*
  - **Button** : *afin d'afficher un bouton,*
  - **Input** : *afin d'afficher un champs texte,*
  - **Overlay** : *afin d'afficher une fenêtre modale,*
  - **Icon** : *afin d'accéder et d'utiliser des icônes (voir <https://oblador.github.io/react-native-vector-icons/>),*
  - **FAB** : *afin d'afficher des boutons flottants,*
  - **Card** : *afin d'afficher des conteneurs,*
  - **Badge** : *afin d'englober certaines valeurs dans des badge avec couleur.*

# Création de l'interface UI – Liste des courses





# Création de l'interface UI – Les statistiques



# Événement et accès aux champs

- Généralement, on déclenchera des événement en appuyant sur des boutons :

Le composant  
La propriété à laquelle  
est associée l'action

```
<Button  
  title="Annuler"  
  onPress={props.close}  
/>
```

- Afin de récupérer le contenu d'un champs texte :

- On utilisera un **Hook** : `const [title, setTitle] = useState('');`

- Qu'on associera au composant :

Le composant  
La propriété à laquelle  
est associé le **Hook**

```
<Input  
  placeholder="Titre de l'article"  
  onChangeText={text => setTitle(text)}  
/>
```

Préalable

```
import {useState} from 'react';
```

# Intégration avec API REST

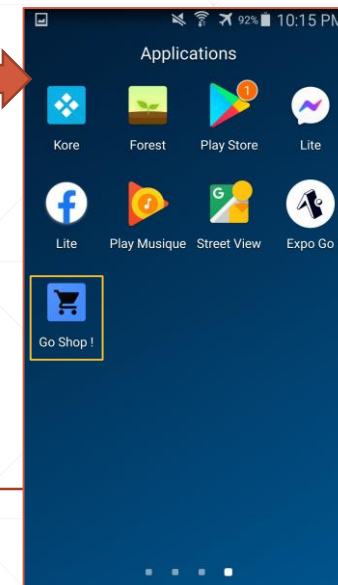
- Pour la communication avec une **API REST**, on utilisera :
  - **redux** : *gestion des données centralisée,*
  - **react-redux** : *intégration de redux avec react,*
  - **redux-thunk** : *add-on pour intégration avec **API REST**,*
  - **axios** : *client HTTP pour communication avec **API REST**.*
- L'**API REST** existe déjà et se trouve sur : <https://to-do-list-rest-api.herokuapp.com/>
  - **/api/swagger** : *accès à la documentation de l'**API REST**,*
  - **/create** : *création d'un **UUID** pour l'utilisation de l'**API REST**.*

# Renommer l'application et icône

- Afin de renommer l'application (*pour **Android***), il faut :
  1. Modifier la valeur de **displayName** dans **app.json**
  2. Modifier la valeur de **app\_name** dans **android/app/src/main/res/values/strings.xml**
  3. Exécuter les commandes : **cd android** et **./gradlew clean** puis **cd ..**
- Pour changer l'icône, il faut la générer avec différentes tailles d'image sur :
  - Lien : <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>
  - Puis il faut suivre les étapes suivantes :
    1. Remplacer les images dans **./android/app/src/main/res** par celles du fichier **.zip** généré
    2. Il y a deux types d'images avec à avoir : **ic\_launcher.png** et **ic\_launcher\_round.png**

# Génération de la *release*

- Pour la génération de la *release* ou de la *version définitive*, il faut utiliser les :
  - Commandes : `npx jetify puis cd android puis ./gradlew clean puis ./gradlew assembleRelease puis ./gradlew bundleRelease puis cd ..`
- Transférer l'application vers le mobile :
  - Commande : `npx react-native run-android --variant=release`
- Voici le résultat final
- Le code source est sur **GitHub**<sup>1</sup>



<sup>1</sup> Lien : <https://github.com/yyaddaden/GroceryShoppingListApp>



# Questions & Discussion

---

# Bibliographie

1. Abbott, D., Djirdeh, H., Accomazzo, A., & Shoemaker, S. (2019). Fullstack react native: The Complete Guide to react native. Fullstack.io.
2. Haverbeke, M. (2014). Eloquent javascript: A modern introduction to programming. No Starch Press.
3. Banks, A., & Porcello, E. (2017). Learning React: functional web development with React and Redux. " O'Reilly Media, Inc."