TP 2 : Création de l'Interface Mobile React Native

Objectif

Créer l'interface utilisateur complète de l'application mobile de gestion de produits (sans connexion au backend pour le moment).

Ce que nous allons construire

- Écran de liste des produits
- Écran de détail d'un produit
- Formulaire d'ajout/modification
- Navigation entre les écrans
- Design moderne et responsive

Aperçu des écrans

Écran principal - Liste des produits

```
┌─────────────────────────┐
│  Gestion Produits    ➕ │
├─────────────────────────┤
│ 🔍 Rechercher...        │
├─────────────────────────┤
│ ┌─────────────────────┐ │
│ │ 📱 iPhone 13        │ │
│ │  899€   Stock: 15   │ │
│ └─────────────────────┘ │
│ ┌─────────────────────┐ │
│ │ 💻 MacBook Pro      │ │
│ │  1999€  Stock: 5    │ │
│ └─────────────────────┘ │
│ ┌─────────────────────┐ │
│ │ 🎧 AirPods Pro      │ │
│ │  279€   Stock: 25   │ │
│ └──────────────────────────┘ │
└──────────────────────────────┘
```

Formulaire produit

```
┌──────────────────────────────┐
│ ← Nouveau Produit            │
├──────────────────────────────┤
│                              │
│ Nom du produit               │
│ ┌──────────────────────────┐ │
│ │                          │ │
│ └──────────────────────────┘ │
│                              │
│ Prix (€)      Stock          │
│ ┌───────────┐ ┌───────────┐  │
│ │           │ │           │  │
│ └───────────┘ └───────────┘  │
│                              │
│ Description                  │
│ ┌──────────────────────────┐ │
│ │                          │ │
│ │                          │ │
│ └──────────────────────────┘ │
│                              │
│   [ Enregistrer ]            │
└──────────────────────────────┘
```

Étape 1 : Création du projet React Native

1.1 Initialiser le projet

```
npx react-native init GestionProduitsMobile
```

```
cd GestionProduitsMobile
```

1.2 Installer les dépendances nécessaires

```
# Navigation
npm install @react-navigation/native @react-navigation/stack
npm install react-native-screens react-native-safe-area-context
npm install react-native-gesture-handler

# UI et icônes
npm install react-native-vector-icons
npm install react-native-elements

# Pour iOS uniquement
cd ios && pod install && cd ..
```

1.3 Configuration des icônes (Android)

Ajoutez dans android/app/build.gradle :

```
apply from: "../../node_modules/react-native-vector-icons/fonts.gradle"
```

Étape 2 : Structure du projet

```
src/
 ├── components/
 │   ├── ProductCard.js
 │   ├── SearchBar.js
 │   └── EmptyList.js
 ├── screens/
 │   ├── ProductListScreen.js
 │   ├── ProductDetailScreen.js
 │   └── ProductFormScreen.js
 ├── navigation/
 │   └── AppNavigator.js
```

```
├── data/
│   └── mockData.js
└── styles/
    └── theme.js
```

Étape 3 : Thème et styles (src/styles/theme.js)

```
export const colors = {
  primary: '#3498db',
  secondary: '#2ecc71',
  danger: '#e74c3c',
  warning: '#f39c12',
  dark: '#2c3e50',
  gray: '#95a5a6',
  lightGray: '#ecf0f1',
  white: '#ffffff',
  background: '#f8f9fa'
};

export const fonts = {
  small: 12,
  regular: 14,
  medium: 16,
  large: 18,
  xlarge: 24,
  xxlarge: 32
};

export const spacing = {
  xs: 5,
  sm: 10,
  md: 15,
  lg: 20,
  xl: 30
};
```

Étape 4 : Données de test (src/data/mockData.js)

```
export const mockProducts = [
 {
  id: 1,
  nom: 'iPhone 13 Pro',
  description: 'Le dernier smartphone Apple avec système de caméra pro',
  prix: 999,
  stock: 15,
  image: 'https://via.placeholder.com/300x300/3498db/ffffff?text=iPhone+13'
 },
 {
  id: 2,
  nom: 'MacBook Pro M1',
  description: 'Ordinateur portable ultra-puissant avec puce Apple M1',
  prix: 1999,
  stock: 8,
  image: 'https://via.placeholder.com/300x300/2c3e50/ffffff?text=MacBook+Pro'
 },
 {
  id: 3,
  nom: 'iPad Air',
  description: 'Tablette légère et polyvalente',
  prix: 699,
  stock: 12,
  image: 'https://via.placeholder.com/300x300/9b59b6/ffffff?text=iPad+Air'
 },
 {
  id: 4,
  nom: 'AirPods Pro',
  description: 'Écouteurs sans fil avec réduction de bruit active',
  prix: 279,
  stock: 25,
  image: 'https://via.placeholder.com/300x300/e74c3c/ffffff?text=AirPods+Pro'
 },
 {
  id: 5,
  nom: 'Apple Watch Series 7',
  description: 'Montre connectée avec suivi santé avancé',
  prix: 429,
  stock: 18,
  image: 'https://via.placeholder.com/300x300/f39c12/ffffff?text=Apple+Watch'
 }
];
```

🧩 Étape 5 : Composants

5.1 Carte produit (src/components/ProductCard.js)

```
import React from 'react';
import {
 View,
 Text,
 StyleSheet,
 TouchableOpacity,
  Image
} from 'react-native';
import Icon from 'react-native-vector-icons/MaterialIcons';
import { colors, fonts, spacing } from '../styles/theme';

const ProductCard = ({ product, onPress }) => {
  const stockColor = product.stock > 10 ? colors.secondary :
                    product.stock > 0 ? colors.warning : colors.danger;

  return (
    <TouchableOpacity
      style={styles.container}
      onPress={onPress}
      activeOpacity={0.8}
    >
      <Image
        source={{ uri: product.image }}
        style={styles.image}
      />
      <View style={styles.content}>
        <Text style={styles.name} numberOfLines={1}>
          {product.nom}
        </Text>
        <Text style={styles.description} numberOfLines={2}>
          {product.description}
        </Text>
        <View style={styles.footer}>
          <Text style={styles.price}>{product.prix}€</Text>
```

```
 },

      <View style={[styles.stockBadge, { backgroundColor: stockColor + '20' }]}>
       <Icon name="inventory" size={14} color={stockColor} />
       <Text style={[styles.stockText, { color: stockColor }]}>
        {product.stock}
       </Text>
      </View>
     </View>
    </View>
   </TouchableOpacity>
  );
};

const styles = StyleSheet.create({
 container: {
  backgroundColor: colors.white,
  borderRadius: 12,
  marginHorizontal: spacing.md,
  marginVertical: spacing.sm,
  flexDirection: 'row',
  padding: spacing.md,
  elevation: 3,
  shadowColor: '#000',
  shadowOffset: { width: 0, height: 2 },
  shadowOpacity: 0.1,
  shadowRadius: 4,
 },
 image: {
  width: 80,
  height: 80,
  borderRadius: 8,
  backgroundColor: colors.lightGray,
 },
 content: {
  flex: 1,
  marginLeft: spacing.md,
  justifyContent: 'space-between',
 },
 name: {
  fontSize: fonts.medium,
  fontWeight: 'bold',
  color: colors.dark,
 },
 description: {
  fontSize: fonts.small,
  color: colors.gray,
  marginTop: 4,
 },
 footer: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  alignItems: 'center',
  marginTop: 8,
 },
 price: {
  fontSize: fonts.large,
  fontWeight: 'bold',
    color: colors.primary,
  },
  stockBadge: {
    flexDirection: 'row',
    alignItems: 'center',
    paddingHorizontal: 8,
    paddingVertical: 4,
    borderRadius: 12,
  },
  stockText: {
    fontSize: fonts.small,
    fontWeight: '600',
    marginLeft: 4,
  },
});

export default ProductCard;
```

5.2 Barre de recherche (src/components/SearchBar.js)

```
import React from 'react';
import {
  View,
  TextInput,
  StyleSheet,
  TouchableOpacity
} from 'react-native';
import Icon from 'react-native-vector-icons/MaterialIcons';
import { colors, fonts, spacing } from '../styles/theme';
```

```
const SearchBar = ({ value, onChangeText, placeholder = "Rechercher..." }) => {
  return (
    <View style={styles.container}>
      <Icon name="search" size={24} color={colors.gray} />
      <TextInput
        style={styles.input}
        value={value}
        onChangeText={onChangeText}
        placeholder={placeholder}
        placeholderTextColor={colors.gray}
      />
      {value.length > 0 && (
        <TouchableOpacity onPress={() => onChangeText('')}>
          <Icon name="close" size={20} color={colors.gray} />
        </TouchableOpacity>
      )}
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flexDirection: 'row',
    alignItems: 'center',
    backgroundColor: colors.white,
    marginHorizontal: spacing.md,
    marginVertical: spacing.sm,
    paddingHorizontal: spacing.md,
    borderRadius: 25,
    elevation: 2,
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 1 },
    shadowOpacity: 0.1,
    shadowRadius: 2,
  },
  input: {
    flex: 1,
    paddingVertical: 12,
    paddingHorizontal: spacing.sm,
    fontSize: fonts.medium,
    color: colors.dark,
  },
});

export default SearchBar;
```

Étape 6 : Écrans principaux

6.1 Liste des produits (src/screens/ProductListScreen.js)

```
import React, { useState } from 'react';
import {
  View,
  FlatList,
  StyleSheet,
  Text,
  TouchableOpacity,
  StatusBar
} from 'react-native';
```

```
import Icon from 'react-native-vector-icons/MaterialIcons';
import ProductCard from '../components/ProductCard';
import SearchBar from '../components/SearchBar';
import { mockProducts } from '../data/mockData';
import { colors, fonts, spacing } from '../styles/theme';

const ProductListScreen = ({ navigation }) => {
 const [products] = useState(mockProducts);
 const [searchQuery, setSearchQuery] = useState('');

 // Filtrer les produits
 const filteredProducts = products.filter(product =>
  product.nom.toLowerCase().includes(searchQuery.toLowerCase())
 );

 const renderProduct = ({ item }) => (
  <ProductCard
   product={item}
   onPress={() => navigation.navigate('ProductDetail', { product: item })}
  />
 );

 const renderHeader = () => (
  <>
   <View style={styles.header}>
    <Text style={styles.title}>Gestion Produits</Text>
    <TouchableOpacity
     onPress={() => navigation.navigate('ProductForm')}
     style={styles.addButton}
    >
     <Icon name="add" size={28} color={colors.white} />
    </TouchableOpacity>
   </View>
   <SearchBar
    value={searchQuery}
    onChangeText={setSearchQuery}
   />
  </>
 );

 const renderEmpty = () => (
  <View style={styles.emptyContainer}>
   <Icon name="inventory-2" size={64} color={colors.lightGray} />
   <Text style={styles.emptyText}>
    {searchQuery ? 'Aucun produit trouvé' : 'Aucun produit disponible'}
   </Text>
  </View>
 );

 return (
  <View style={styles.container}>
   <StatusBar barStyle="dark-content" backgroundColor={colors.background} />
   <FlatList
    data={filteredProducts}
    renderItem={renderProduct}
    keyExtractor={item => item.id.toString()}
```

```
        ListHeaderComponent={renderHeader}
        ListEmptyComponent={renderEmpty}
        contentContainerStyle={styles.listContent}
        showsVerticalScrollIndicator={false}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: colors.background,
  },
  header: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    paddingHorizontal: spacing.md,
    paddingTop: spacing.xl,
    paddingBottom: spacing.md,
  },
  title: {
    fontSize: fonts.xxlarge,
    fontWeight: 'bold',
    color: colors.dark,
  },
  addButton: {
  },

    width: 48,
    height: 48,
    borderRadius: 24,
    backgroundColor: colors.primary,
    justifyContent: 'center',
    alignItems: 'center',
    elevation: 3,
    shadowColor: colors.primary,
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.3,
    shadowRadius: 4,
  },
  listContent: {
    paddingBottom: spacing.xl,
  },
  emptyContainer: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    paddingTop: 100,
  },
  emptyText: {
    fontSize: fonts.medium,
    color: colors.gray,
    marginTop: spacing.md,
  },
});
```

```
export default ProductListScreen;
```

6.2 Détail produit (src/screens/ProductDetailScreen.js)

```
import React from 'react';
import {
  View,
  Text,
  StyleSheet,
  ScrollView,
  Image,
  TouchableOpacity,
  Alert
} from 'react-native';
import Icon from 'react-native-vector-icons/MaterialIcons';
import { colors, fonts, spacing } from '../styles/theme';

const ProductDetailScreen = ({ route, navigation }) => {
  const { product } = route.params;

  const handleEdit = () => {
    navigation.navigate('ProductForm', { product });
  };

  const handleDelete = () => {
    Alert.alert(
      'Supprimer le produit',
      ` Êtes-vous sûr de vouloir supprimer "${product.nom}" ?` ,
      [
        { text: 'Annuler', style: 'cancel' },
   {
    text: 'Supprimer',
    onPress: () => {
     // Logique de suppression
     navigation.goBack();
    },
    style: 'destructive'
   }
  ]
 );
};

const stockColor = product.stock > 10 ? colors.secondary :
          product.stock > 0 ? colors.warning : colors.danger;

return (
 <ScrollView style={styles.container} showsVerticalScrollIndicator={false}>
  <Image source={{ uri: product.image }} style={styles.image} />

  <View style={styles.content}>
   <View style={styles.header}>
    <Text style={styles.name}>{product.nom}</Text>
    <Text style={styles.price}>{product.prix}€</Text>
   </View>

   <View style={styles.stockContainer}>
    <Icon name="inventory" size={20} color={stockColor} />
    <Text style={[styles.stockText, { color: stockColor }]}>
     {product.stock} unités en stock
    </Text>
   </View>

   <View style={styles.section}>
    <Text style={styles.sectionTitle}>Description</Text>
    <Text style={styles.description}>{product.description}</Text>
   </View>

   <View style={styles.actions}>
    <TouchableOpacity
     style={[styles.button, styles.editButton]}
     onPress={handleEdit}
    >
     <Icon name="edit" size={20} color={colors.white} />
     <Text style={styles.buttonText}>Modifier</Text>
    </TouchableOpacity>

    <TouchableOpacity
     style={[styles.button, styles.deleteButton]}
     onPress={handleDelete}
    >
     <Icon name="delete" size={20} color={colors.white} />
     <Text style={styles.buttonText}>Supprimer</Text>
    </TouchableOpacity>
   </View>
  </View>
 </ScrollView>
```

```
 );
};

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: colors.background,
 },
 image: {
  width: '100%',
  height: 300,
  backgroundColor: colors.lightGray,
 },
 content: {
  padding: spacing.lg,
 },
 header: {
  marginBottom: spacing.md,
 },
 name: {
  fontSize: fonts.xlarge,
  fontWeight: 'bold',
  color: colors.dark,
  marginBottom: spacing.xs,
 },
 price: {
  fontSize: fonts.xxlarge,
  fontWeight: 'bold',
  color: colors.primary,
 },
 stockContainer: {
  flexDirection: 'row',
  alignItems: 'center',
  backgroundColor: colors.white,
  padding: spacing.md,
  borderRadius: 8,
  marginBottom: spacing.lg,
 },
 stockText: {
  fontSize: fonts.medium,
  fontWeight: '600',
  marginLeft: spacing.sm,
 },
 section: {
  marginBottom: spacing.lg,
 },
 sectionTitle: {
  fontSize: fonts.large,
  fontWeight: '600',
  color: colors.dark,
  marginBottom: spacing.sm,
 },
 description: {
  fontSize: fonts.medium,
  color: colors.gray,
  lineHeight: 24,
  },
  actions: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    marginTop: spacing.xl,
  },
  button: {
    flex: 1,
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'center',
    padding: spacing.md,
    borderRadius: 8,
    marginHorizontal: spacing.xs,
  },
  editButton: {
    backgroundColor: colors.primary,
  },
  deleteButton: {
    backgroundColor: colors.danger,
  },
  buttonText: {
    color: colors.white,
    fontSize: fonts.medium,
    fontWeight: '600',
    marginLeft: spacing.sm,
  },
});
```

```
export default ProductDetailScreen;
```

6.3 Formulaire produit (src/screens/ProductFormScreen.js)

```
import React, { useState } from 'react';
import {
  View,
  Text,
  TextInput,
  StyleSheet,
  ScrollView,
  TouchableOpacity,
  Alert
} from 'react-native';
import Icon from 'react-native-vector-icons/MaterialIcons';
import { colors, fonts, spacing } from '../styles/theme';

const ProductFormScreen = ({ route, navigation }) => {
  const editMode = route.params?.product != null;
  const [formData, setFormData] = useState({
    nom: route.params?.product?.nom || '',
    prix: route.params?.product?.prix?.toString() || '',
    stock: route.params?.product?.stock?.toString() || '',
    description: route.params?.product?.description || ''
  });

  const handleSubmit = () => {
    // Validation
    if (!formData.nom || !formData.prix || !formData.stock) {
```

```
    Alert.alert('Erreur', 'Veuillez remplir tous les champs obligatoires');
    return;
  }

  // Logique de sauvegarde
  Alert.alert(
    'Succès',
    ` Produit ${editMode ? 'modifié' : 'ajouté'} avec succès` ,
    [{ text: 'OK', onPress: () => navigation.goBack() }]
  );
};

return (
  <ScrollView style={styles.container} showsVerticalScrollIndicator={false}>
    <View style={styles.form}>
      <Text style={styles.title}>
        {editMode ? 'Modifier le produit' : 'Nouveau produit'}
      </Text>

      <View style={styles.inputGroup}>
        <Text style={styles.label}>Nom du produit *</Text>
        <TextInput
          style={styles.input}
          value={formData.nom}
          onChangeText={(text) => setFormData({...formData, nom: text})}
          placeholder="Ex: iPhone 13"
          placeholderTextColor={colors.gray}
        />
```

```
</View>

<View style={styles.row}>
 <View style={[styles.inputGroup, styles.halfWidth]}>
  <Text style={styles.label}>Prix (€) *</Text>
  <TextInput
   style={styles.input}
   value={formData.prix}
   onChangeText={(text) => setFormData({...formData, prix: text})}
   placeholder="0"
   placeholderTextColor={colors.gray}
   keyboardType="numeric"
  />
 </View>

 <View style={[styles.inputGroup, styles.halfWidth]}>
  <Text style={styles.label}>Stock *</Text>
  <TextInput
   style={styles.input}
   value={formData.stock}
   onChangeText={(text) => setFormData({...formData, stock: text})}
   placeholder="0"
   placeholderTextColor={colors.gray}
   keyboardType="numeric"
  />
 </View>
</View>
```

```
 <View style={styles.inputGroup}>
  <Text style={styles.label}>Description</Text>
  <TextInput
   style={[styles.input, styles.textArea]}
   value={formData.description}
   onChangeText={(text) => setFormData({...formData, description: text})}
   placeholder="Description du produit"
   placeholderTextColor={colors.gray}
   multiline
   numberOfLines={4}
  />
 </View>

 <TouchableOpacity
  style={styles.submitButton}
  onPress={handleSubmit}
  activeOpacity={0.8}
 >
  <Icon
   name={editMode ? "check" : "add"}
   size={24}
   color={colors.white}
  />
  <Text style={styles.submitText}>
   {editMode ? 'Enregistrer' : 'Ajouter'}
  </Text>
 </TouchableOpacity>
</View>
```

```
  </ScrollView>
 );
};

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: colors.background,
 },
 form: {
  padding: spacing.lg,
 },
 title: {
  fontSize: fonts.xlarge,
  fontWeight: 'bold',
  color: colors.dark,
  marginBottom: spacing.xl,
 },
 inputGroup: {
  marginBottom: spacing.lg,
 },
 row: {
  flexDirection: 'row',
  justifyContent: 'space-between',
 },
 halfWidth: {
  width: '48%',
 },
label: {
  fontSize: fonts.medium,
  fontWeight: '600',
  color: colors.dark,
  marginBottom: spacing.sm,
},
input: {
  backgroundColor: colors.white,
  borderRadius: 8,
  paddingHorizontal: spacing.md,
  paddingVertical: spacing.md,
  fontSize: fonts.medium,
  color: colors.dark,
  borderWidth: 1,
  borderColor: colors.lightGray,
},
textArea: {
  height: 120,
  textAlignVertical: 'top',
},
submitButton: {
  backgroundColor: colors.primary,
  flexDirection: 'row',
  alignItems: 'center',
  justifyContent: 'center',
  padding: spacing.md,
  borderRadius: 8,
  marginTop: spacing.lg,
 },
 submitText: {
  color: colors.white,
  fontSize: fonts.large,
  fontWeight: '600',
  marginLeft: spacing.sm,
 },
});

export default ProductFormScreen;
```

🧭 Étape 7 : Navigation (src/navigation/AppNavigator.js)

```
import React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createStackNavigator } from '@react-navigation/stack';
import ProductListScreen from '../screens/ProductListScreen';
import ProductDetailScreen from '../screens/ProductDetailScreen';
import ProductFormScreen from '../screens/ProductFormScreen';
import { colors } from '../styles/theme';

const Stack = createStackNavigator();

const AppNavigator = () => {
 return (
  <NavigationContainer>
   <Stack.Navigator
    screenOptions={{
     headerStyle: {
      backgroundColor: colors.white,
      elevation: 0,
      shadowOpacity: 0,
    },
    headerTintColor: colors.dark,
    headerTitleStyle: {
      fontWeight: '600',
    },
  }}
>
  <Stack.Screen
    name="ProductList"
    component={ProductListScreen}
    options={{ headerShown: false }}
  />
  <Stack.Screen
    name="ProductDetail"
    component={ProductDetailScreen}
    options={{
      title: 'Détail du produit',
      headerBackTitle: 'Retour'
    }}
  />
  <Stack.Screen
    name="ProductForm"
    component={ProductFormScreen}
    options={{
      title: 'Produit',
      headerBackTitle: 'Annuler'
```

```
          }}
        />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

export default AppNavigator;
```

Étape 8 : Fichier App.js principal

```
import React from 'react';
import AppNavigator from './src/navigation/AppNavigator';

const App = () => {
  return <AppNavigator />;
};

export default App;
```

Lancement de l'application

```
# Android
npx react-native run-android

# iOS
npx react-native run-ios
```