

Компоненты для Android и IOS

Цель работы: В этой работе мы познакомимся со специфичными компонентами для каждой из платформ.

Контрольные вопросы 1. Какие основные принципы UI есть одновременно и в Human Interface Design и в Material Design?

Ответ: Material Design и Human Interface Design соблюдают принцип KISS, интерфейс должен быть простым, понятный и очевидный, задачи должны решаться минимальным числом действий, также они интуитивно понятны для пользователя. Соблюдается принцип адаптивности (кроссплатформенность). Дизайн-элементы, от линейных иконок до переключателей начинают выглядеть одинаково. Тренды, соблюдаемые компаниями, всё больше сходятся, например, персонализация на основе обоев и системной темы пользователя, техника LCH (яркость, цвет, оттенок), которая позволяет без особых усилий генерировать доступные цветовые сочетания, гармоничные и скруглённые углы, техника оптических размеров в UI-типографике, контент в центре внимания, вложенные UI-элементы, стандартный дизайн переключателей, сокращение визуального шума и вывод контента на первый план.

В качестве специфичного компонента под Android был выбран ToastAndroid. Для IOS выбран ActionSheetIOS.

Код программы:

```
import React from "react";
import Constants from "expo-constants";
import {
  StyleSheet,
  Text,
  View,
  ScrollView,
  ImageBackground,
  TouchableOpacity,
  ToastAndroid,
  Dimensions,
  Platform,
  ActionSheetIOS,
  Alert, FlatList,
} from "react-native";

const ITEMS = [
  {item: "Item #1", uri: "https://picsum.photos/100/100?random=1"},
  {item: "Item #2", uri: "https://picsum.photos/100/100?random=2"},
  {item: "Item #3", uri: "https://picsum.photos/100/100?random=3"},
  {item: "Item #4", uri: "https://picsum.photos/100/100?random=4"},
  {item: "Item #5", uri: "https://picsum.photos/100/100?random=5"},
  {item: "Item #6", uri: "https://picsum.photos/100/100?random=6"},
  {item: "Item #7", uri: "https://picsum.photos/100/100?random=7"},
  {item: "Item #8", uri: "https://picsum.photos/100/100?random=8"},
  {item: "Item #9", uri: "https://picsum.photos/100/100?random=9"},
  {item: "Item #10", uri: "https://picsum.photos/100/100?random=10"}
]

export default class App extends React.Component {
  handlePlatform(text) {
    if (Platform.OS === "android") {
      this.handlePressAndroidToast(text);
    }
  }
}
```

```

    } else if (Platform.OS === "ios") {
      this.handlePressIOS(text);
    }
  }

  handlePressAndroidToast(text) {
    ToastAndroid.show(text, ToastAndroid.SHORT);
  }

  handlePressIOS(text) {
    ActionSheetIOS.showActionSheetWithOptions(
      {
        options: ["Cancel", "Generate number", "Magic"],
        destructiveButtonIndex: 2,
        cancelButtonIndex: 0,
        userInterfaceStyle: "dark",
      },
      (buttonIndex) => {
        if (buttonIndex === 0) {
        1);
        } else if (buttonIndex === 1) {
          Alert.alert(text, "Result: " + Math.floor(Math.random() * 100) +
        } else if (buttonIndex === 2) {
          Alert.alert(text, "🔮");
        }
      }
    );
  }

  renderItem = ({item, index}) => {
    return (
      <TouchableOpacity
        style={styles.item}
        key={index}
        onPress={() => this.handlePlatform(item.item)}
      >
        <ImageBackground
          style={styles.image}
          source={{ uri: item.uri }}
        />
        <Text
          style={{
            fontSize: 20,
            marginStart: Dimensions.get("window").width / 5,
          }}
        >
          {item.item}
        </Text>
      </TouchableOpacity>
    );
  }

  render() {
    return (
      <View style={styles.container}>
        <View style={styles.header}>
          <Text style={{ color: "#ffb42f", textAlign: "center", fontSize:
24 }}>
            My List Image
          </Text>
        </View>
        <FlatList data={ITEMS} renderItem={this.renderItem}/>
      </View>
    );
  }

```

```
}  
}  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    paddingTop: Constants.statusBarHeight,  
    backgroundColor: "#fff",  
  },  
  header: {  
    backgroundColor: "black",  
    width: Dimensions.get("window").width,  
  },  
  item: {  
    flexDirection: "row",  
    alignItems: "center",  
    margin: 5,  
    backgroundColor: "#ffb42f",  
    borderRadius: 10,  
  },  
  image: {  
    width: 100,  
    height: 100,  
  },  
});
```

Пример работы:



