Lab 6 : Core APIs

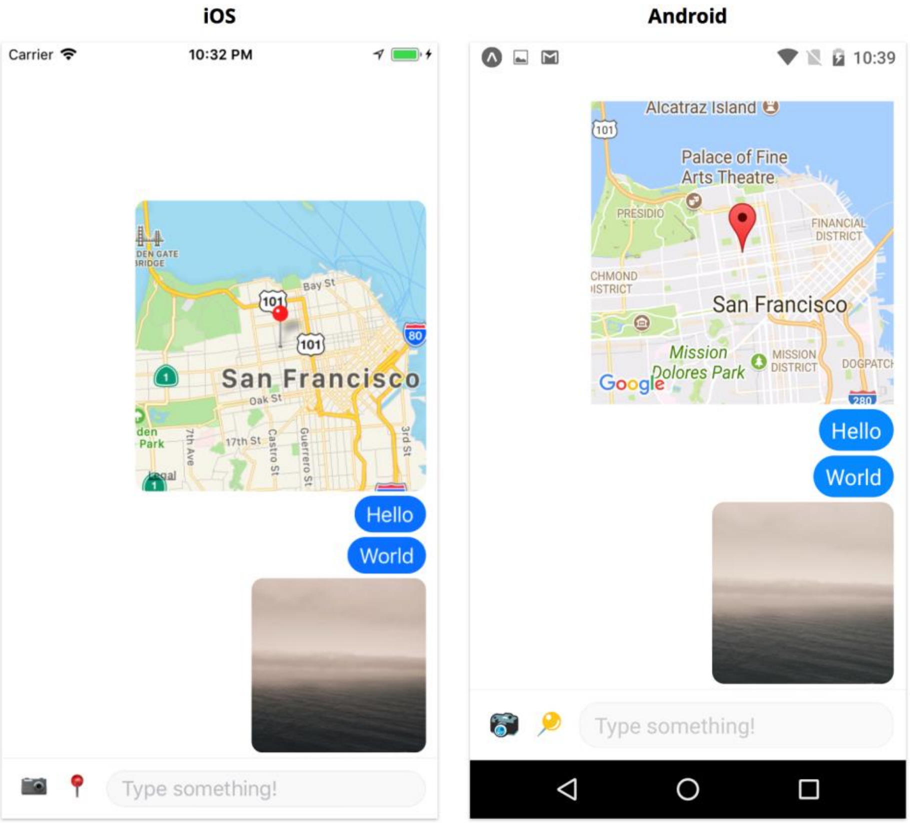
1. ให้นักศึกษาทำการสร้าง New Project โดยให้ Folder มีดังนี้ Mobile\<รหัสนักศึกษา>\Message

   Expo init <path\folder\StudentID\Project name>

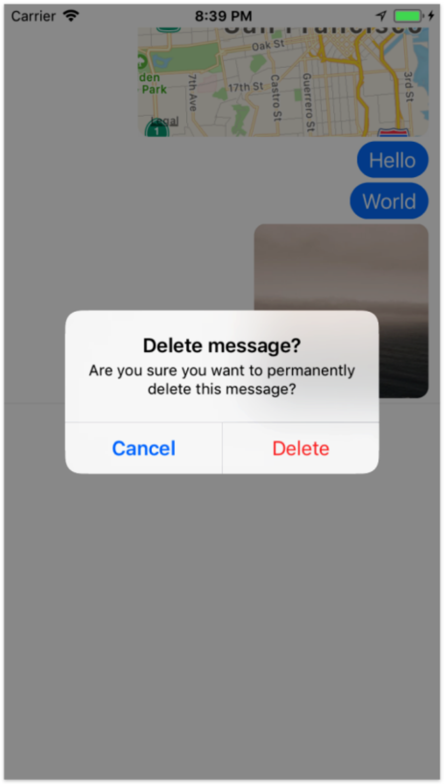
2. ให้นักศึกษาเขียนโปรแกรมเพื่อรับส่งข้อความและรูปภาพ (ดังแสดงตามรูปภาพข้างล่างนี้) โดยการปฏิบัติครั้งนี้ให้นักศึกษาทำการสร้าง NetInfo, StatusBar, MessageList, Alert

   Hint: (Component) View, Text, Image

   (APIs) NetInfo, MessageList, Statusbar, Alert



iOS



Hello

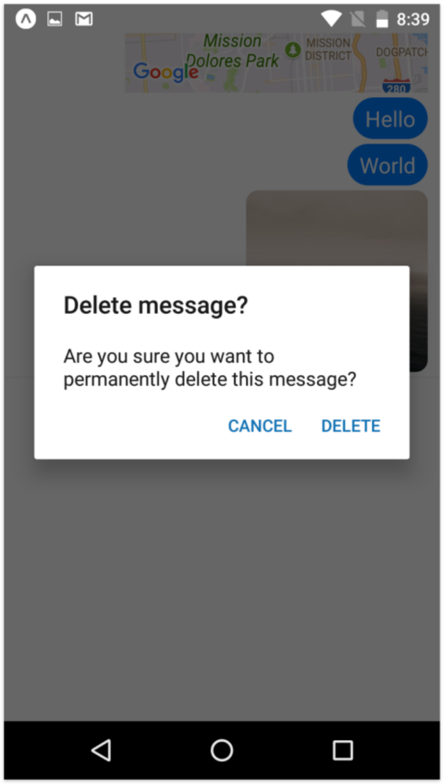
World

Delete message?

Are you sure you want to permanently delete this message?

Cancel | Delete

Android



Hello

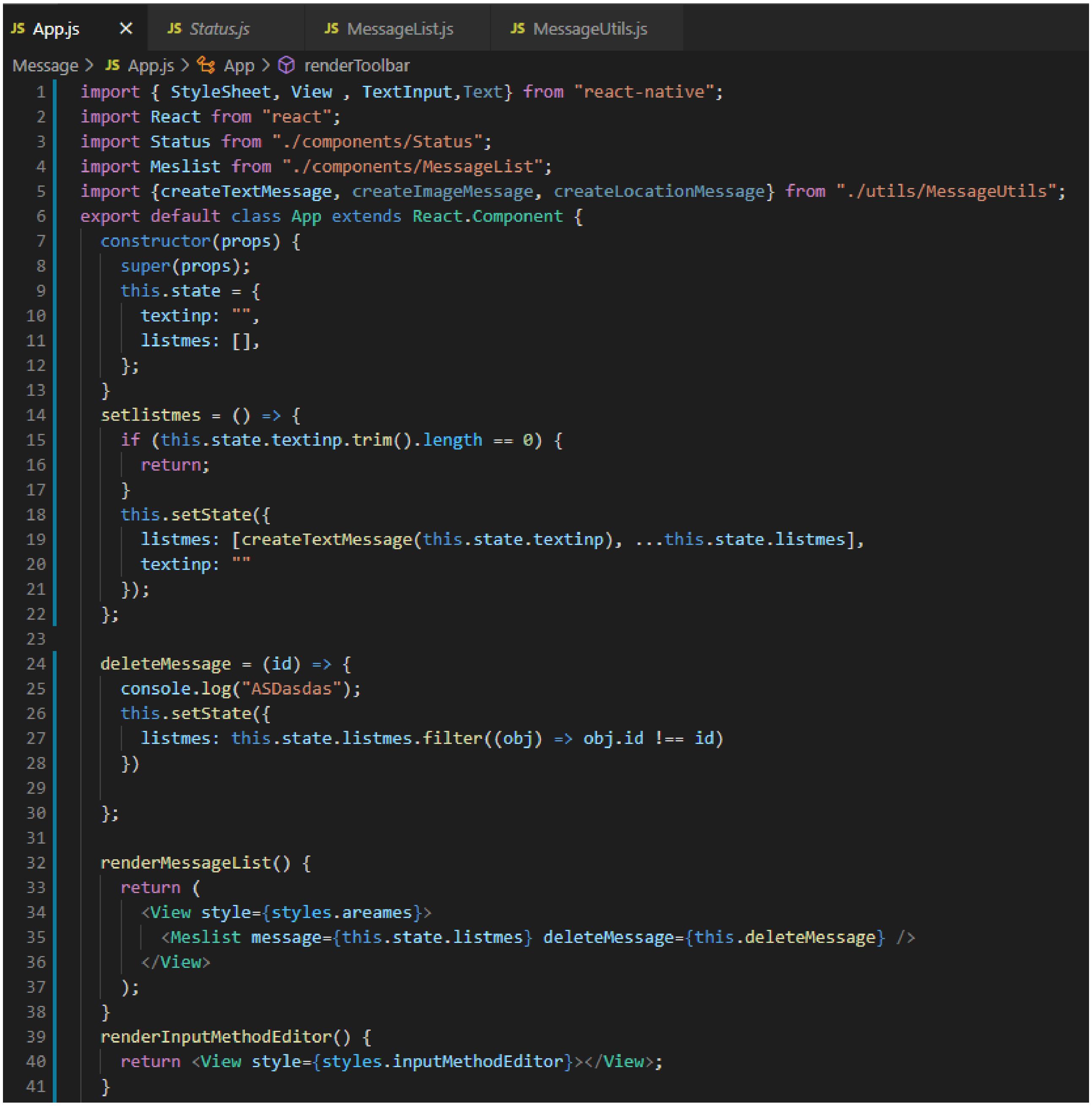
World

Delete message?

Are you sure you want to permanently delete this message?

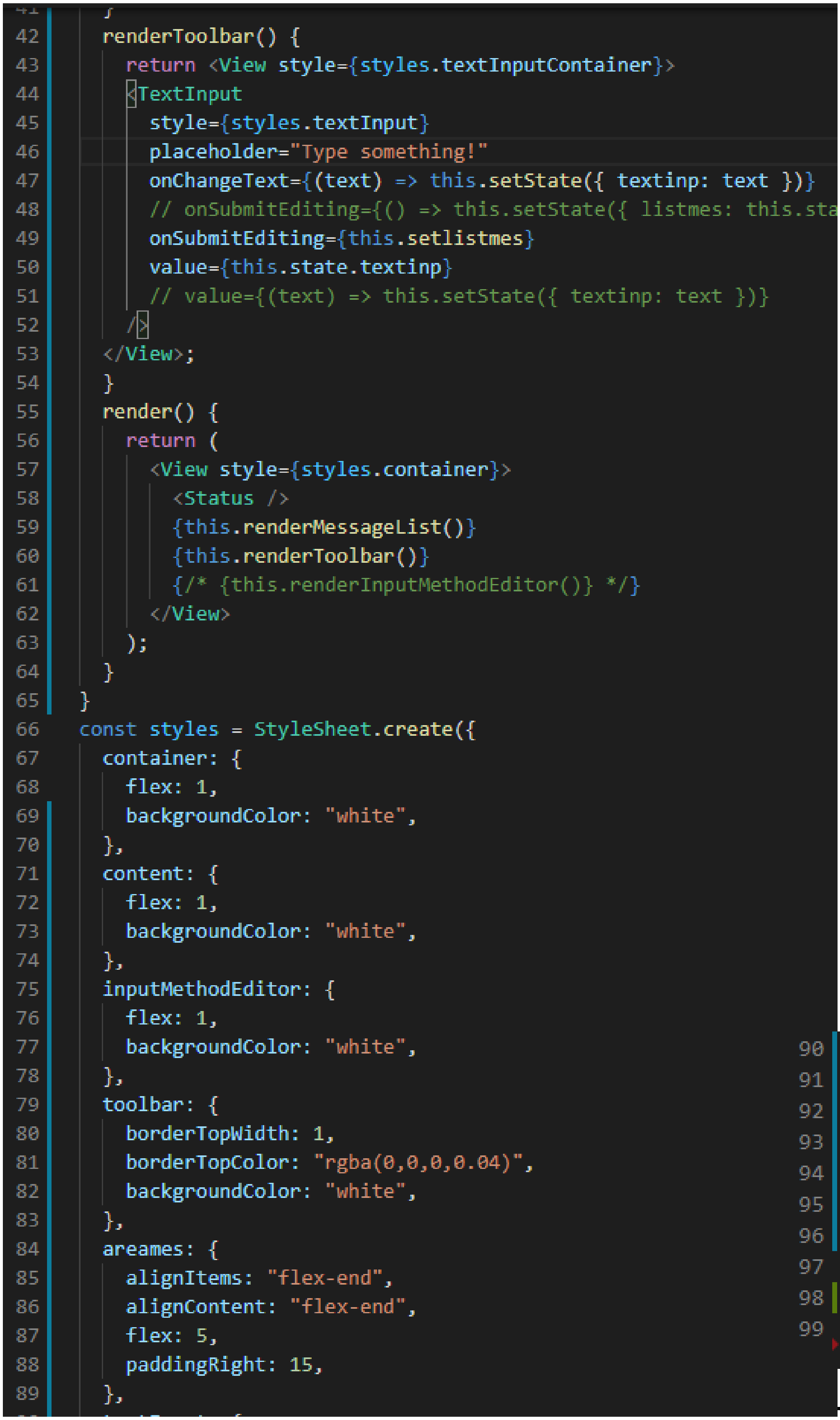
CANCEL DELETE

App.js (1/2)

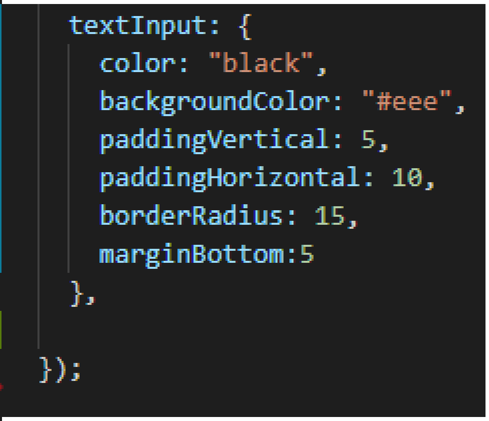


```
import { StyleSheet, View , TextInput,Text} from "react-native";
import React from "react";
import Status from "./components/Status";
import Meslist from "./components/MessageList";
import {createTextMessage, createImageMessage, createLocationMessage} from "./utils/MessageUtils";
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      textinp: "",
      listmes: [],
    };
  }
  setlistmes = () => {
    if (this.state.textinp.trim().length == 0) {
      return;
    }
    this.setState({
      listmes: [createTextMessage(this.state.textinp), ...this.state.listmes],
      textinp: ""
    });
  };

  deleteMessage = (id) => {
    console.log("ASDasdas");
    this.setState({
      listmes: this.state.listmes.filter((obj) => obj.id !== id)
    })

  };

  renderMessageList() {
    return (
      <View style={styles.areames}>
        <Meslist message={this.state.listmes} deleteMessage={this.deleteMessage} />
      </View>
    );
  }
  renderInputMethodEditor() {
    return <View style={styles.inputMethodEditor}></View>;
  }
```

App.js (2/2)

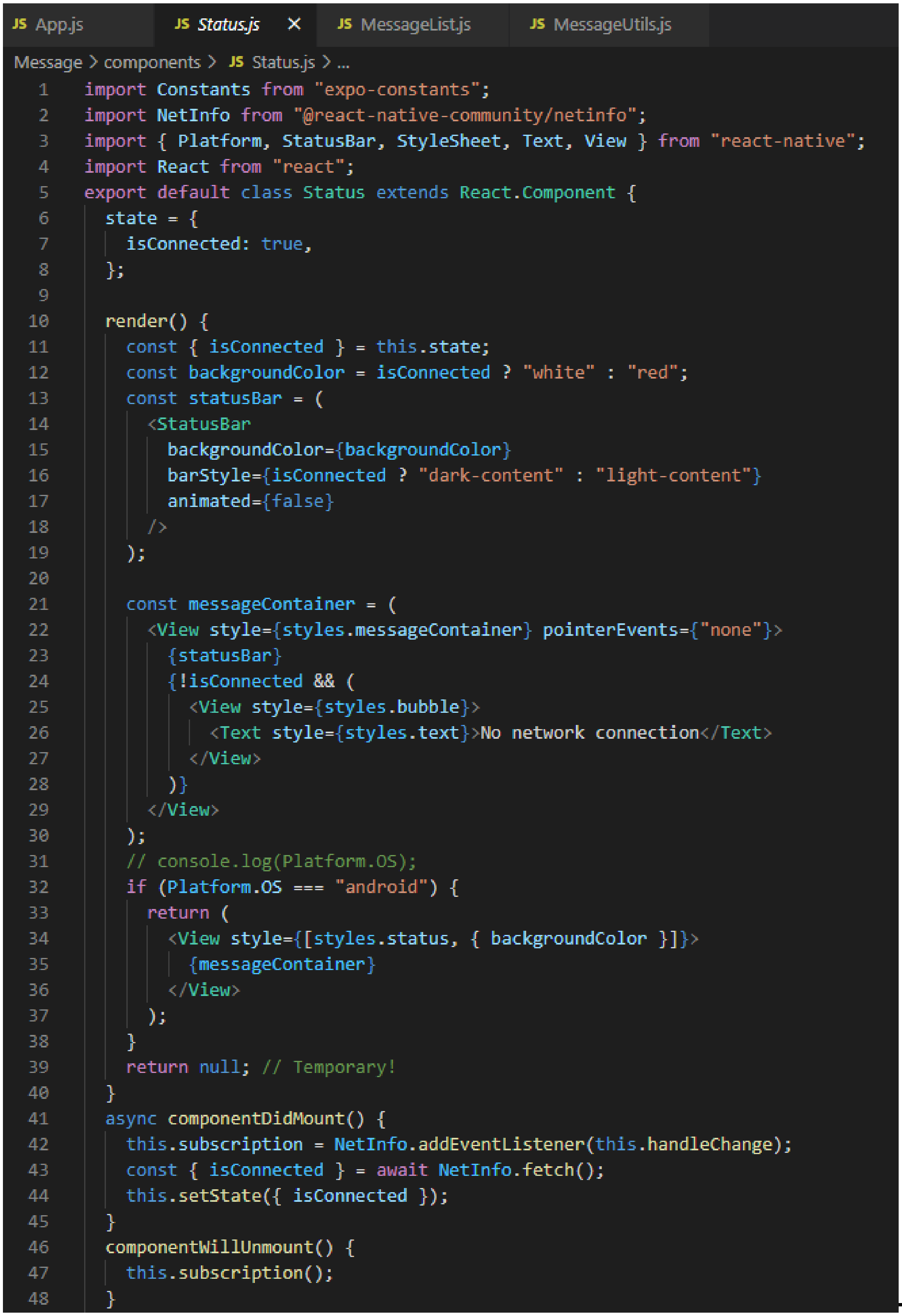


```
  }
  renderToolbar() {
    return <View style={styles.textInputContainer}>
    <TextInput
      style={styles.textInput}
      placeholder="Type something!"
      onChangeText={(text) => this.setState({ textinp: text })}
      // onSubmitEditing={() => this.setState({ listmes: this.sta
      onSubmitEditing={this.setlistmes}
      value={this.state.textinp}
      // value={(text) => this.setState({ textinp: text })}
    />
  </View>;
  }
  render() {
    return (
      <View style={styles.container}>
        <Status />
        {this.renderMessageList()}
        {this.renderToolbar()}
        {/* {this.renderInputMethodEditor()} */}
      </View>
    );
  }
}
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
  },
  content: {
    flex: 1,
    backgroundColor: "white",
  },
  inputMethodEditor: {
    flex: 1,
    backgroundColor: "white",
  },
  toolbar: {
    borderTopWidth: 1,
    borderTopColor: "rgba(0,0,0,0.04)",
    backgroundColor: "white",
  },
  areames: {
    alignItems: "flex-end",
    alignContent: "flex-end",
    flex: 5,
    paddingRight: 15,
  },
```



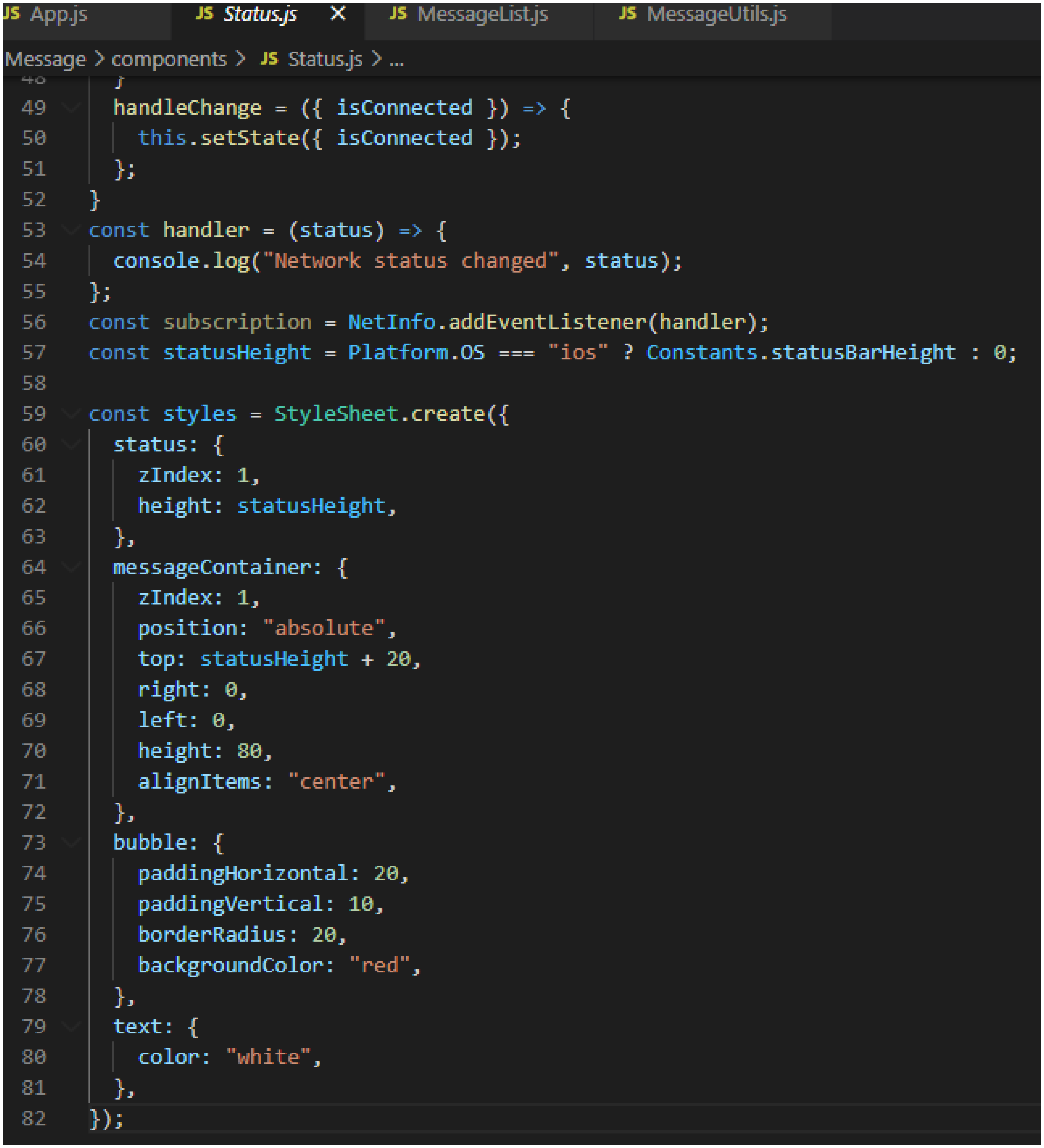
```
  textInput: {
    color: "black",
    backgroundColor: "#eee",
    paddingVertical: 5,
    paddingHorizontal: 10,
    borderRadius: 15,
    marginBottom:5
  },

});
```

Status.js (1/2)



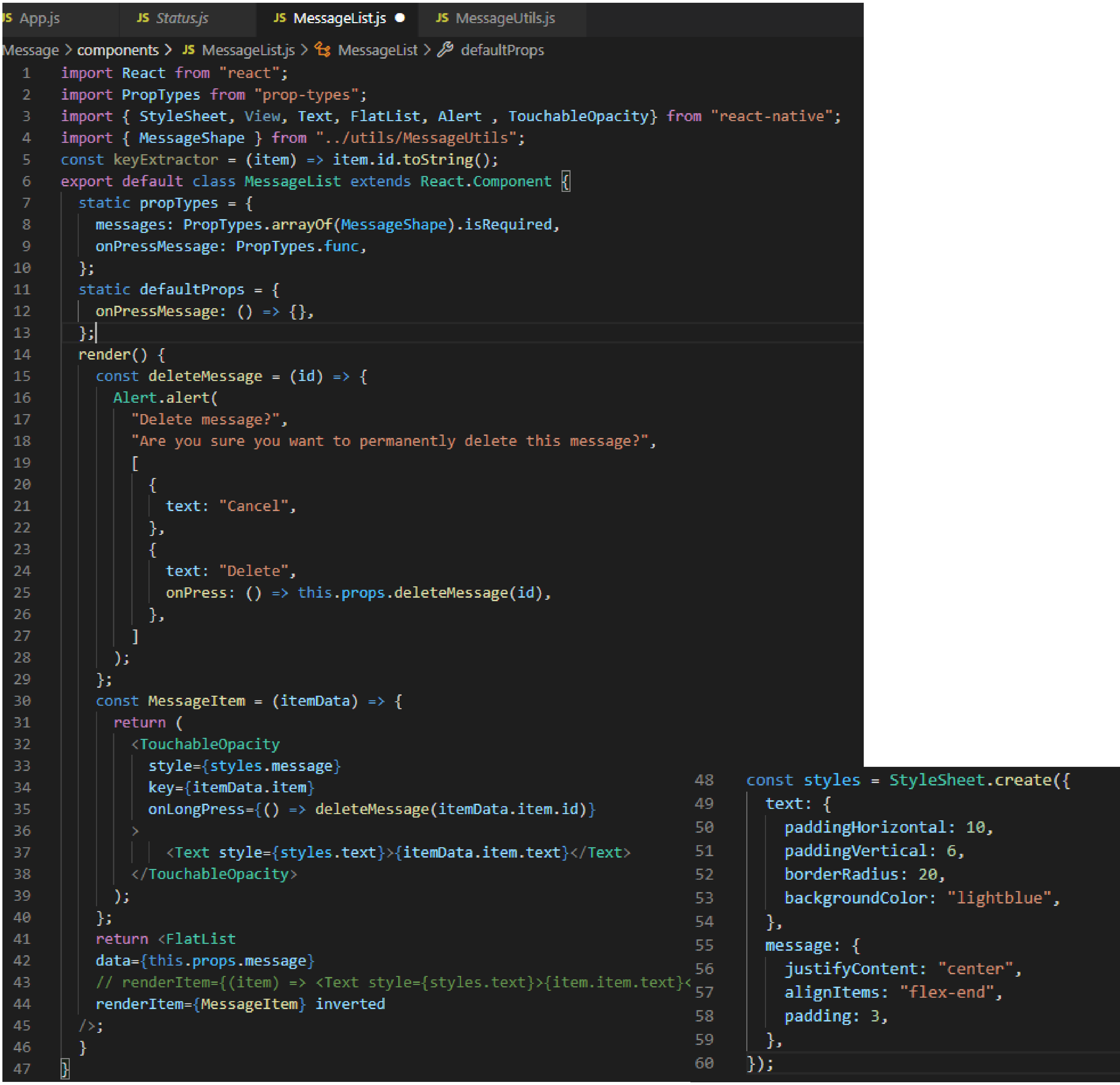
```
import Constants from "expo-constants";
import NetInfo from "@react-native-community/netinfo";
import { Platform, StatusBar, StyleSheet, Text, View } from "react-native";
import React from "react";
export default class Status extends React.Component {
  state = {
    isConnected: true,
  };

  render() {
    const { isConnected } = this.state;
    const backgroundColor = isConnected ? "white" : "red";
    const statusBar = (
      <StatusBar
        backgroundColor={backgroundColor}
        barStyle={isConnected ? "dark-content" : "light-content"}
        animated={false}
      />
    );

    const messageContainer = (
      <View style={styles.messageContainer} pointerEvents={"none"}>
        {statusBar}
        {!isConnected && (
          <View style={styles.bubble}>
            <Text style={styles.text}>No network connection</Text>
          </View>
        )}
      </View>
    );
    // console.log(Platform.OS);
    if (Platform.OS === "android") {
      return (
        <View style={[styles.status, { backgroundColor }]}>
          {messageContainer}
        </View>
      );
    }
    return null; // Temporary!
  }
  async componentDidMount() {
    this.subscription = NetInfo.addEventListener(this.handleChange);
    const { isConnected } = await NetInfo.fetch();
    this.setState({ isConnected });
  }
  componentWillUnmount() {
    this.subscription();
  }
```

Status.js (2/2)



```
  }
  handleChange = ({ isConnected }) => {
    this.setState({ isConnected });
  };
}
const handler = (status) => {
  console.log("Network status changed", status);
};
const subscription = NetInfo.addEventListener(handler);
const statusHeight = Platform.OS === "ios" ? Constants.statusBarHeight : 0;

const styles = StyleSheet.create({
  status: {
    zIndex: 1,
    height: statusHeight,
  },
  messageContainer: {
    zIndex: 1,
    position: "absolute",
    top: statusHeight + 20,
    right: 0,
    left: 0,
    height: 80,
    alignItems: "center",
  },
  bubble: {
    paddingHorizontal: 20,
    paddingVertical: 10,
    borderRadius: 20,
    backgroundColor: "red",
  },
  text: {
    color: "white",
  },
});
```

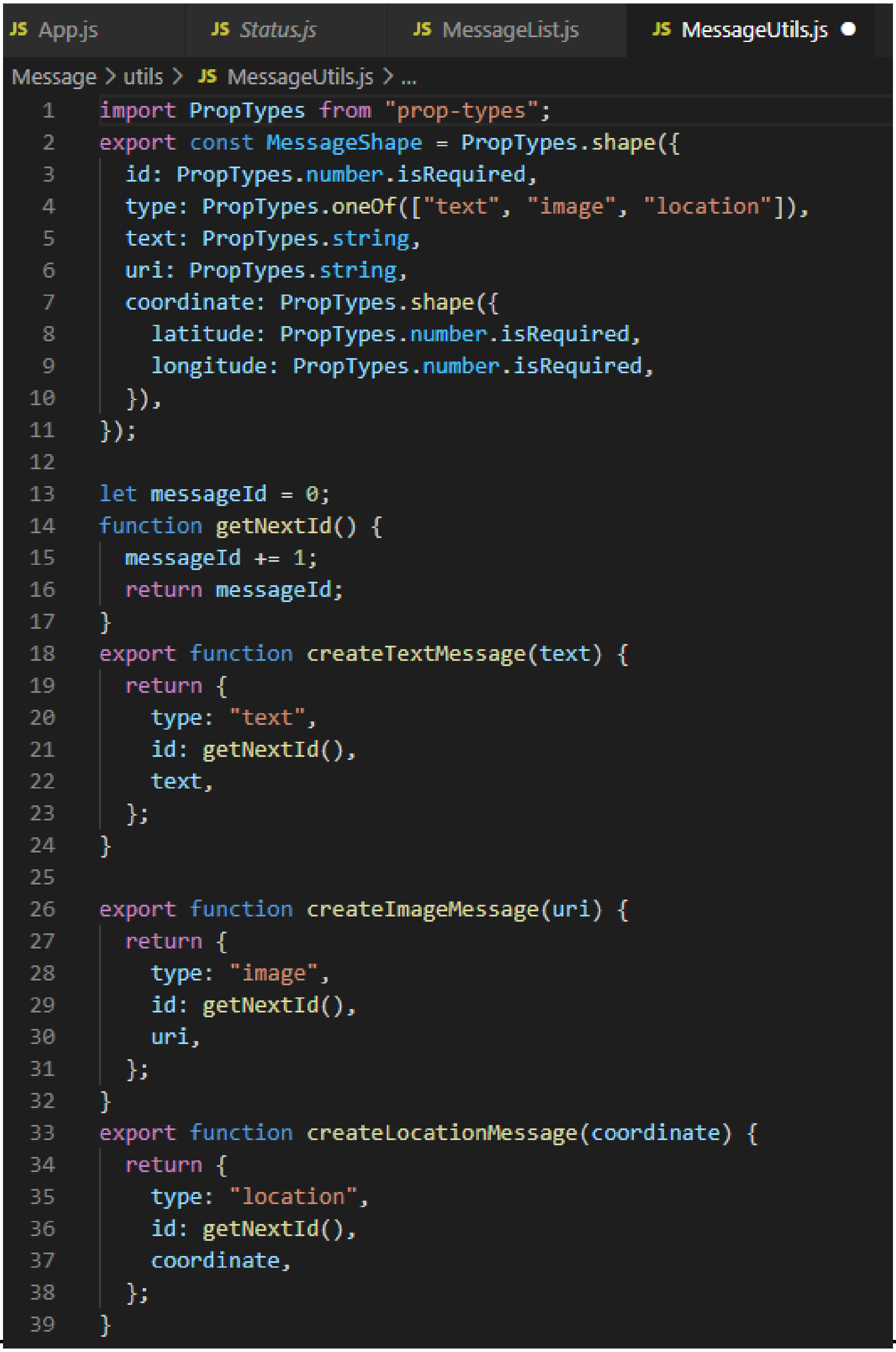
MessageList.js (2/2)


JS App.js
JS Status.js
JS MessageList.js ●
JS MessageUtils.js
Message > components > JS MessageList.js > MessageList > defaultProps

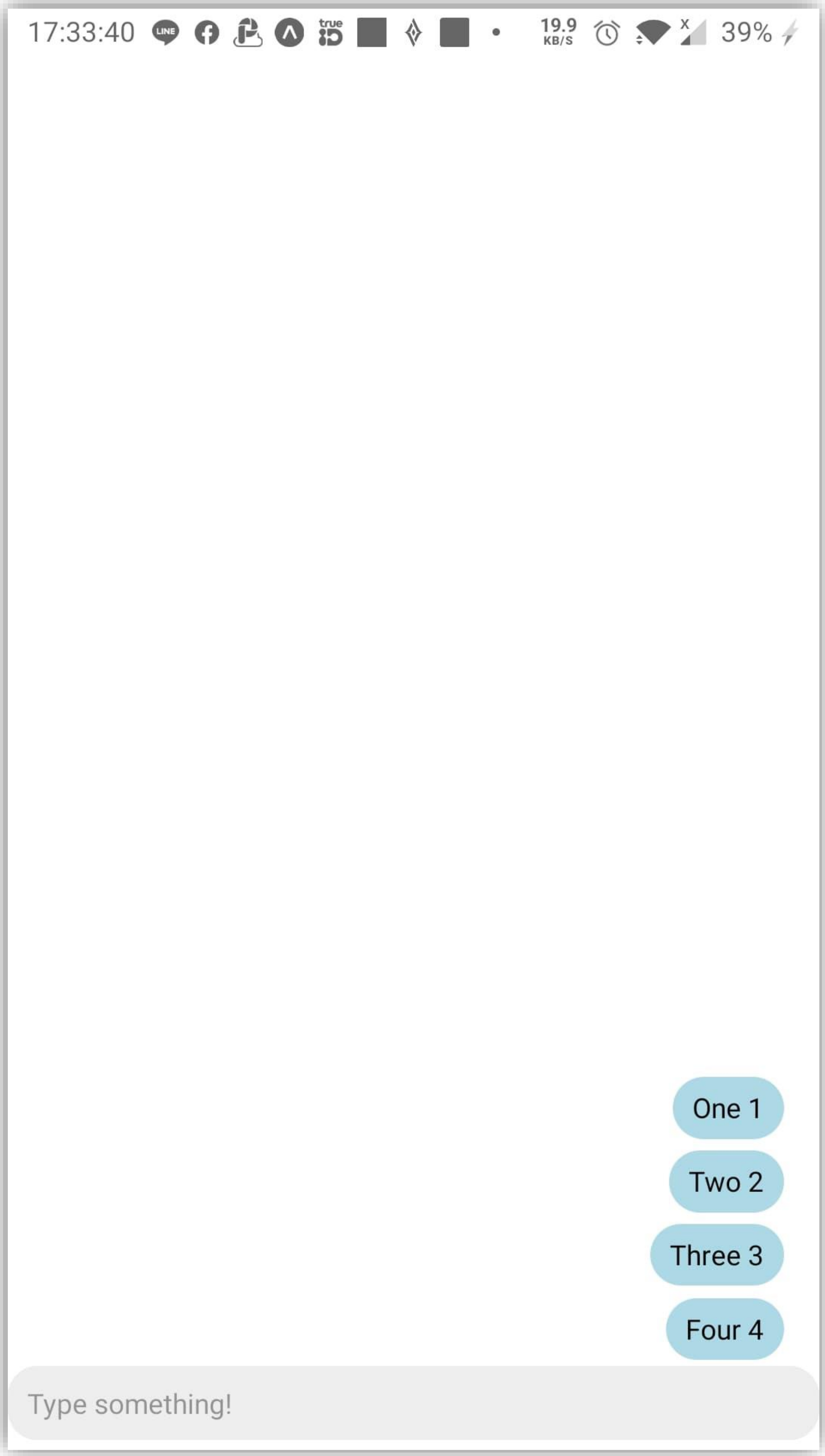
```
import React from "react";
import PropTypes from "prop-types";
import { StyleSheet, View, Text, FlatList, Alert , TouchableOpacity} from "react-native";
import { MessageShape } from "../utils/MessageUtils";
const keyExtractor = (item) => item.id.toString();
export default class MessageList extends React.Component {
  static propTypes = {
    messages: PropTypes.arrayOf(MessageShape).isRequired,
    onPressMessage: PropTypes.func,
  };
  static defaultProps = {
    onPressMessage: () => {},
  };
  render() {
    const deleteMessage = (id) => {
      Alert.alert(
        "Delete message?",
        "Are you sure you want to permanently delete this message?",
        [
          {
            text: "Cancel",
          },
          {
            text: "Delete",
            onPress: () => this.props.deleteMessage(id),
          },
        ]
      );
    };
    const MessageItem = (itemData) => {
      return (
        <TouchableOpacity
          style={styles.message}
          key={itemData.item}
          onLongPress={() => deleteMessage(itemData.item.id)}
        >
          <Text style={styles.text}>{itemData.item.text}</Text>
        </TouchableOpacity>
      );
    };
    return <FlatList
    data={this.props.message}
    // renderItem={(item) => <Text style={styles.text}>{item.item.text}<
    renderItem={MessageItem} inverted
  />;
  }
}
```

```
const styles = StyleSheet.create({
  text: {
    paddingHorizontal: 10,
    paddingVertical: 6,
    borderRadius: 20,
    backgroundColor: "lightblue",
  },
  message: {
    justifyContent: "center",
    alignItems: "flex-end",
    padding: 3,
  },
});
```

MessageUtils.js (2/2)



```
import PropTypes from "prop-types";
export const MessageShape = PropTypes.shape({
  id: PropTypes.number.isRequired,
  type: PropTypes.oneOf(["text", "image", "location"]),
  text: PropTypes.string,
  uri: PropTypes.string,
  coordinate: PropTypes.shape({
    latitude: PropTypes.number.isRequired,
    longitude: PropTypes.number.isRequired,
  }),
});

let messageId = 0;
function getNextId() {
  messageId += 1;
  return messageId;
}
export function createTextMessage(text) {
  return {
    type: "text",
    id: getNextId(),
    text,
  };
}

export function createImageMessage(uri) {
  return {
    type: "image",
    id: getNextId(),
    uri,
  };
}
export function createLocationMessage(coordinate) {
  return {
    type: "location",
    id: getNextId(),
    coordinate,
  };
}
```

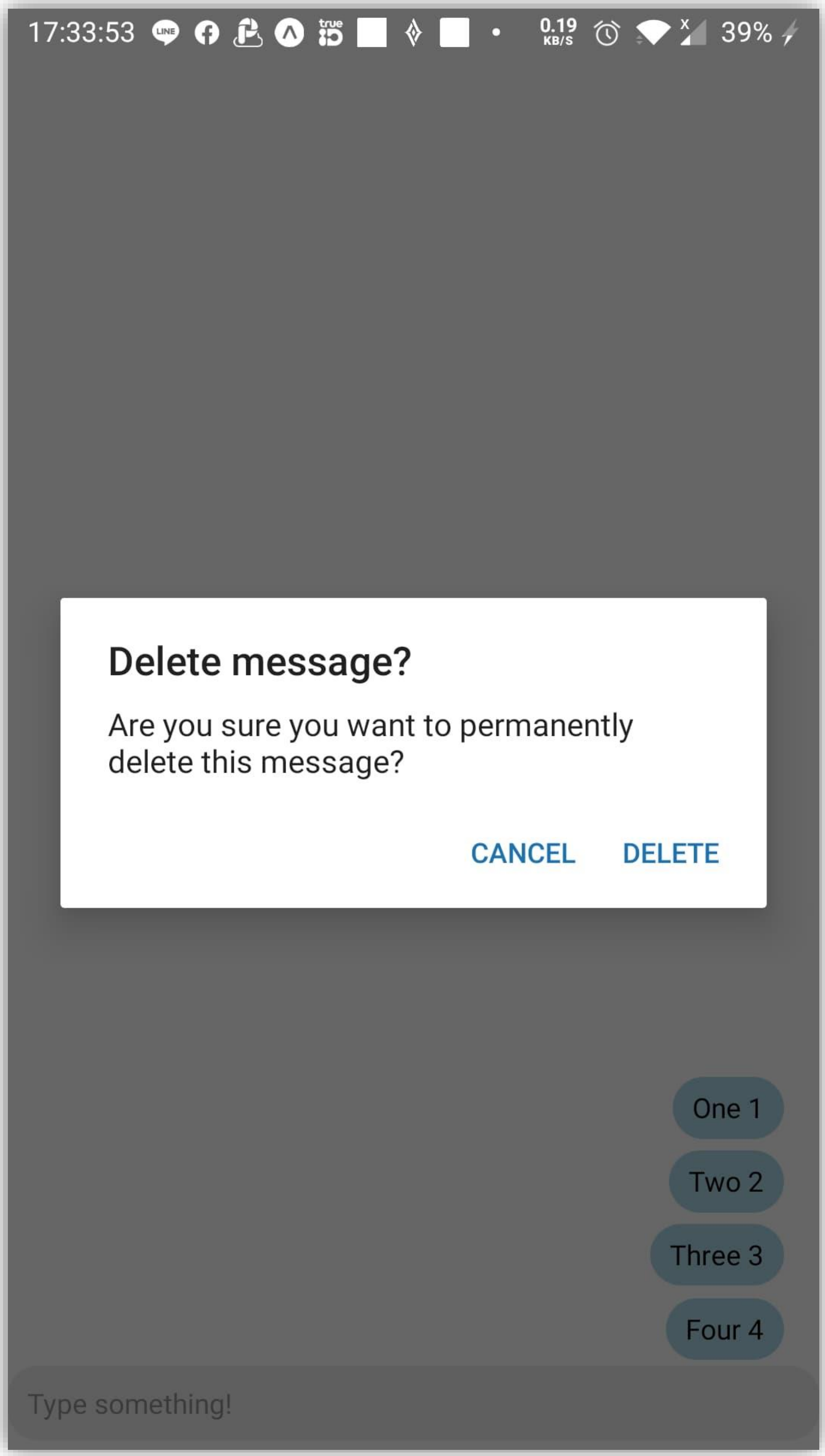


One 1

Two 2

Three 3

Four 4



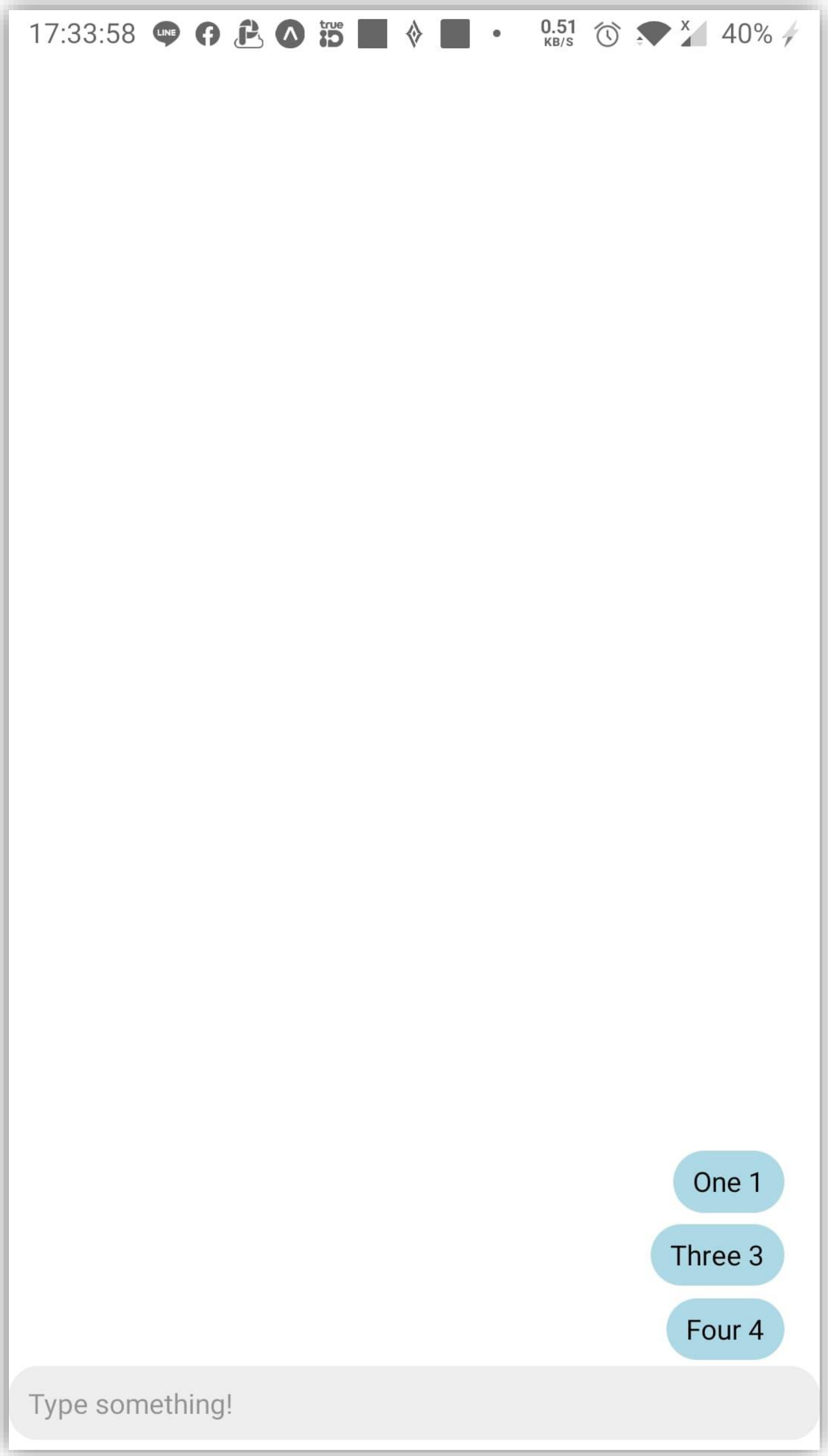
17:33:53

Delete message?

Are you sure you want to permanently delete this message?

ชื่อ.........นาย...สหัสวรรษ......หิรัญเพชร......................

รหัสนักศึกษา..............61070239..............................

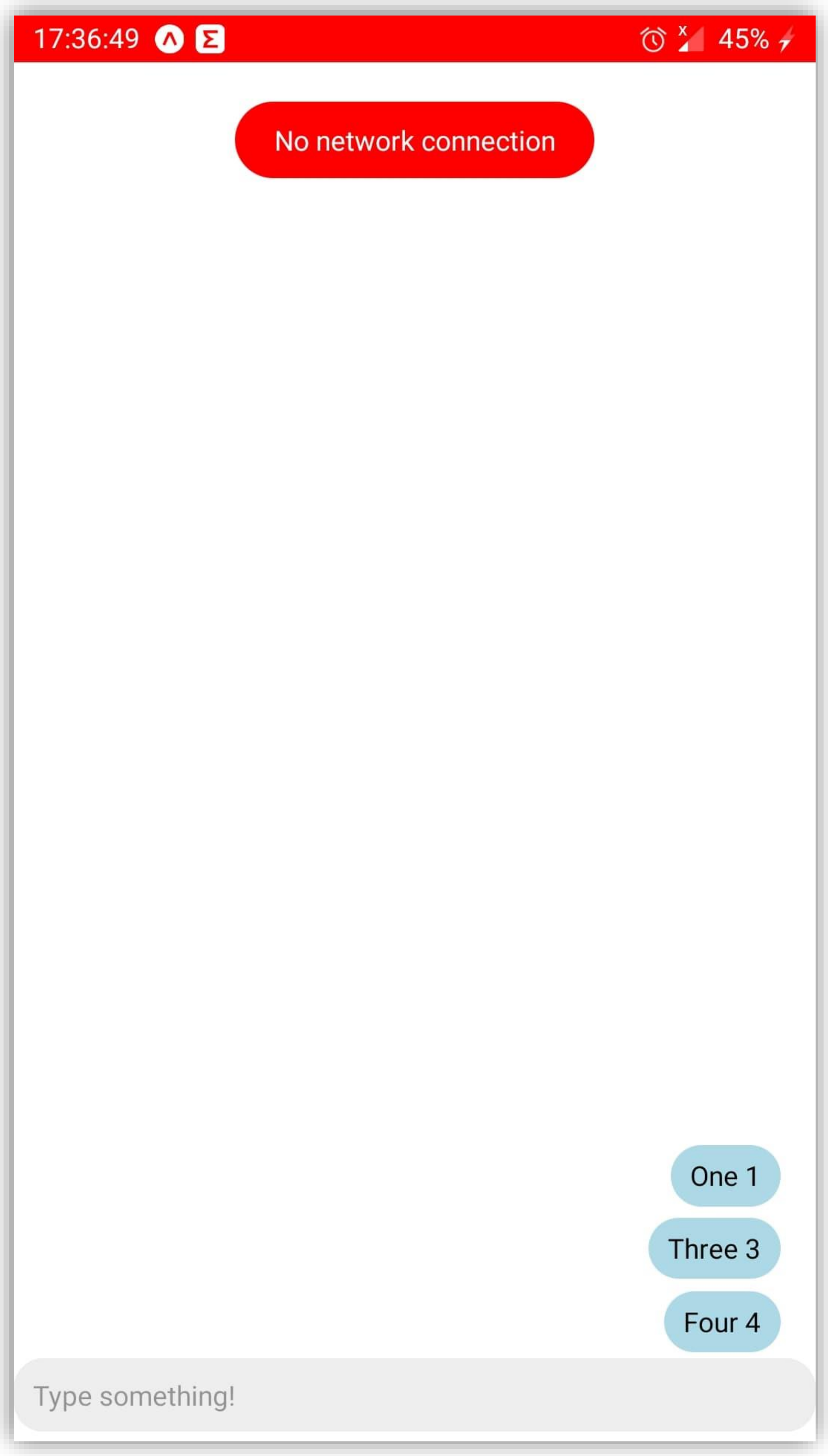


One 1

Three 3

Four 4

Type something!



No network connection

One 1

Three 3

Four 4

Type something!