คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

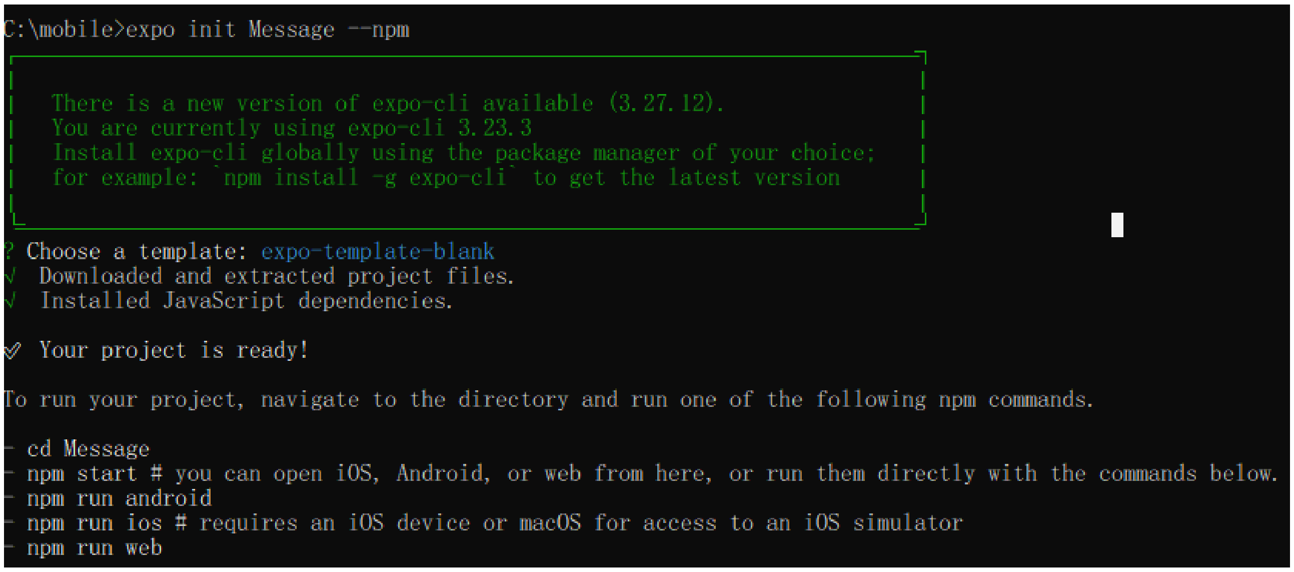
ชื่อ.....................นายศรัณย์ ยันตะบุษย์...................

รหัสนักศึกษา..............61070218...........................

Lab 6 : Core APIs

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

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



```
C:\mobile>expo init Message --npm
┌─────────────────────────────────────────────────────────────────────────┐
│                                                                         │
│   There is a new version of expo-cli available (3.27.12).               │
│   You are currently using expo-cli 3.23.3                               │
│   Install expo-cli globally using the package manager of your choice;   │
│   for example: `npm install -g expo-cli` to get the latest version      │
│                                                                         │
└─────────────────────────────────────────────────────────────────────────┘
? Choose a template: expo-template-blank
√ Downloaded and extracted project files.
√ Installed JavaScript dependencies.

✅ Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.

- cd Message
- npm start # you can open iOS, Android, or web from here, or run them directly with the commands below.
- npm run android
- npm run ios # requires an iOS device or macOS for access to an iOS simulator
- npm run web
```

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

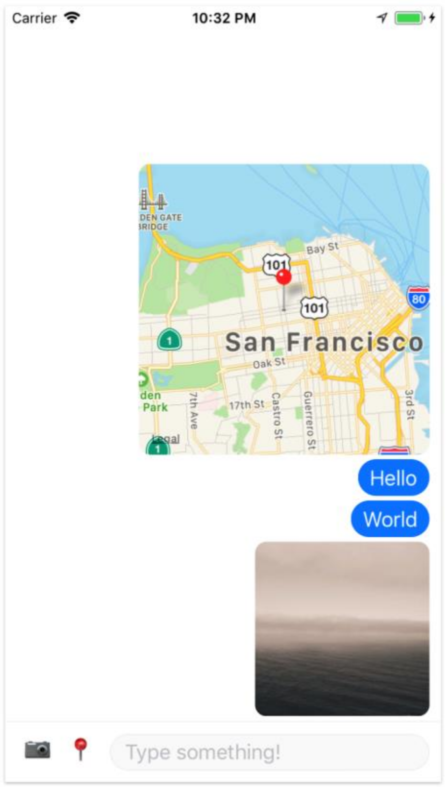
   Hint: (Component) View, Text, Image

   (APIs) NetInfo, MessageList, Statusbar, Alert

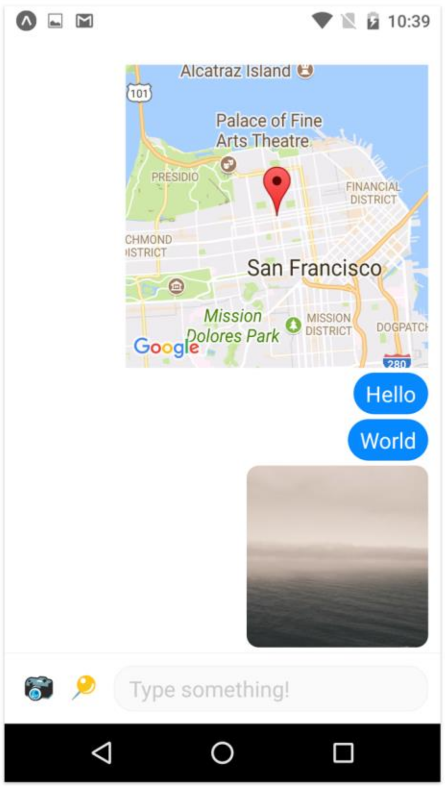
ชื่อ.....................นายศรัณย์ ยันตะบุษย์...................

รหัสนักศึกษา..............61070218...........................

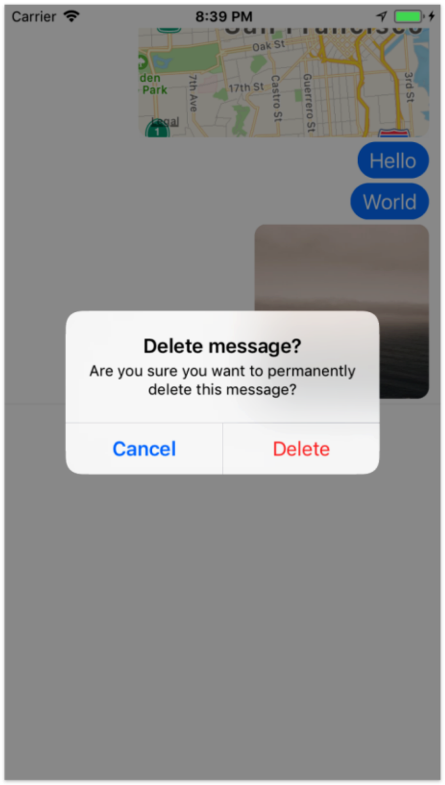
iOS



Android

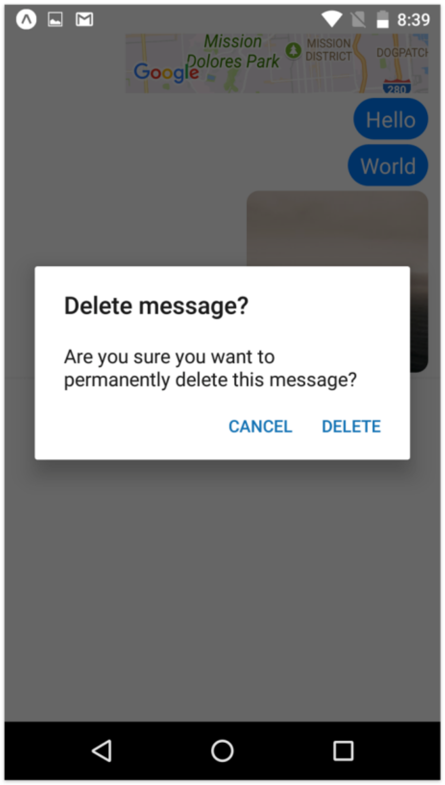


iOS

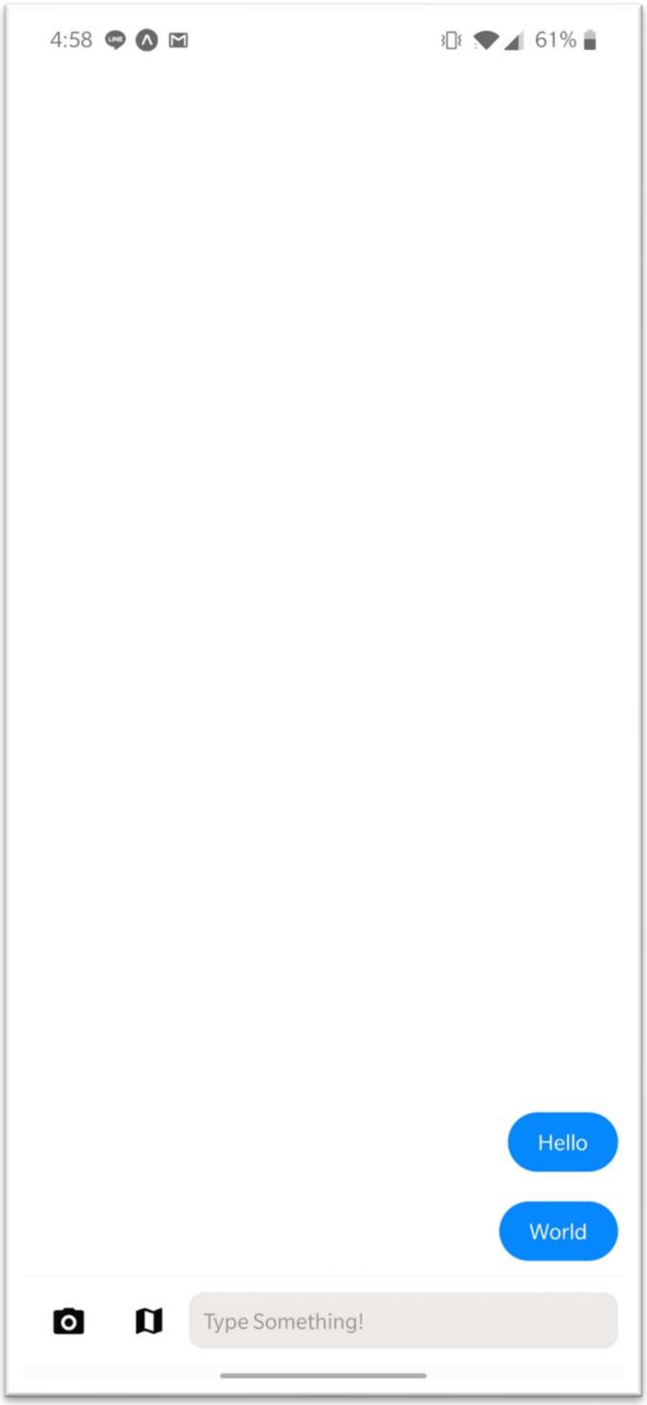


Delete message?
Are you sure you want to permanently delete this message?

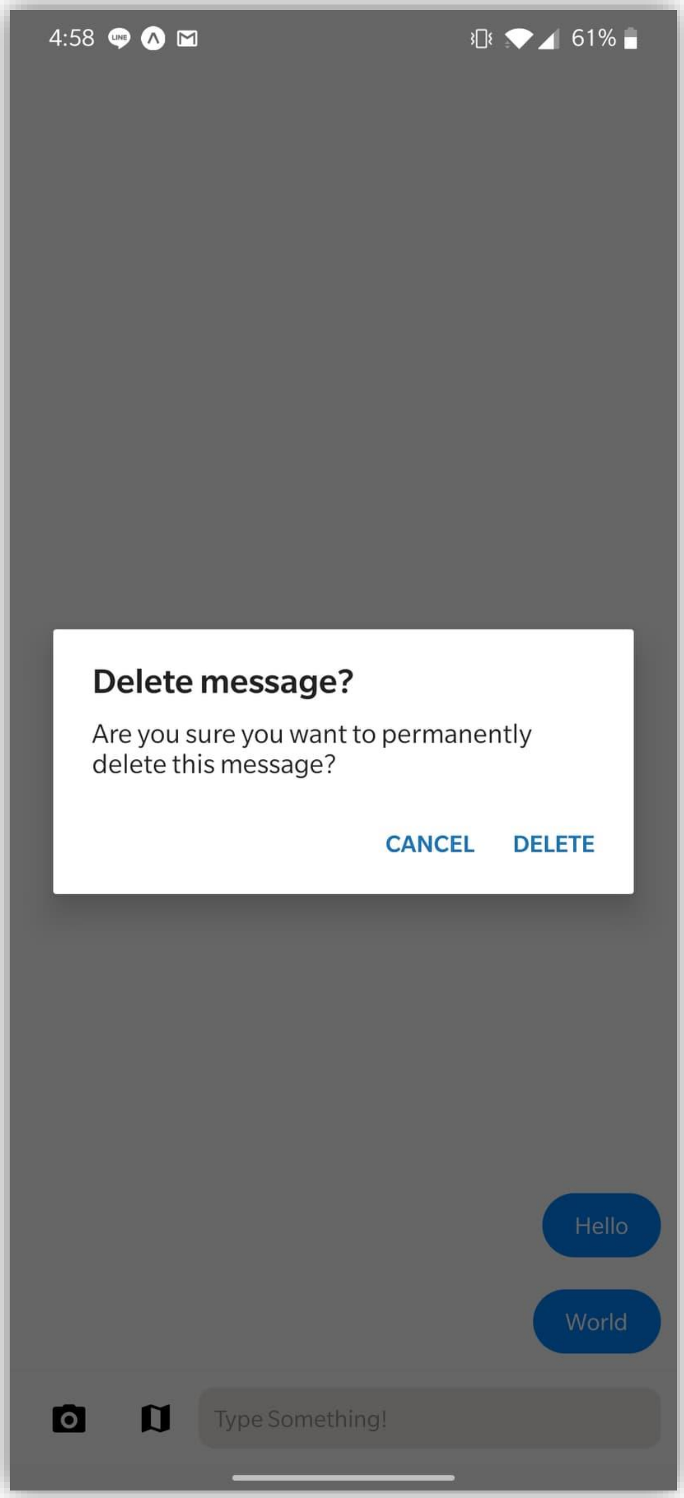
Android



Delete message?
Are you sure you want to permanently delete this message?


4:58
61%
Hello
World
Type Something!

ชื่อ.....................นายศรัณย์ ยันตะบุษย์...................
รหัสนักศึกษา..............61070218...........................



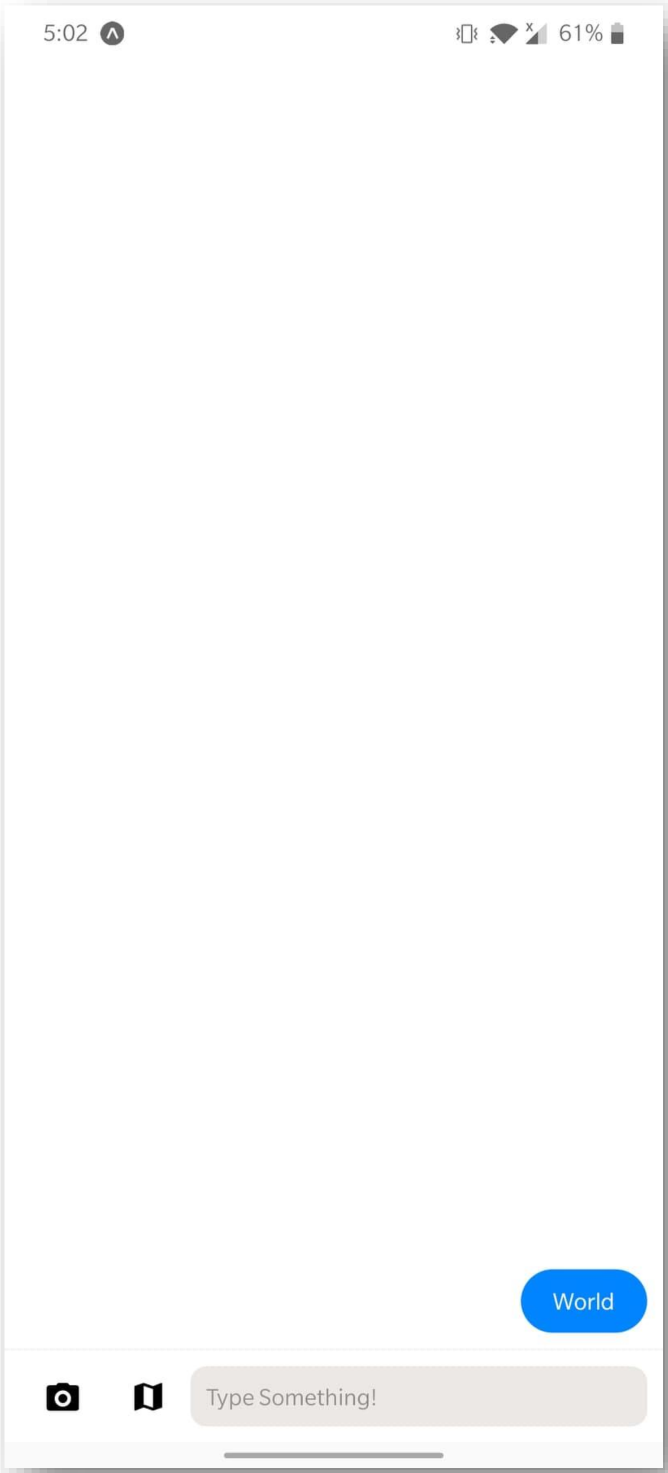
Delete message?

Are you sure you want to permanently delete this message?

CANCEL DELETE

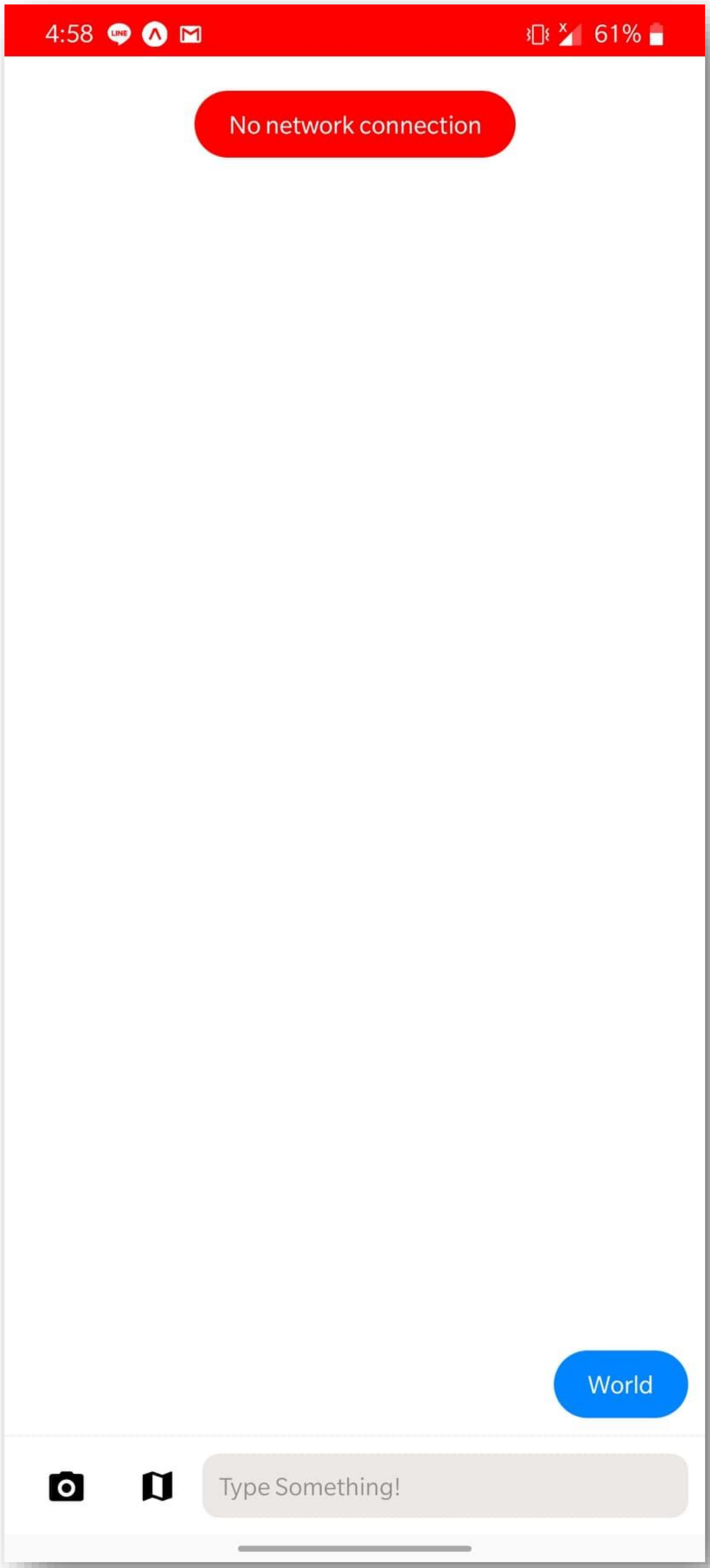
ชื่อ......................นายศรัณย์ ยันตะบุษย์....................

รหัสนักศึกษา...............61070218.............................



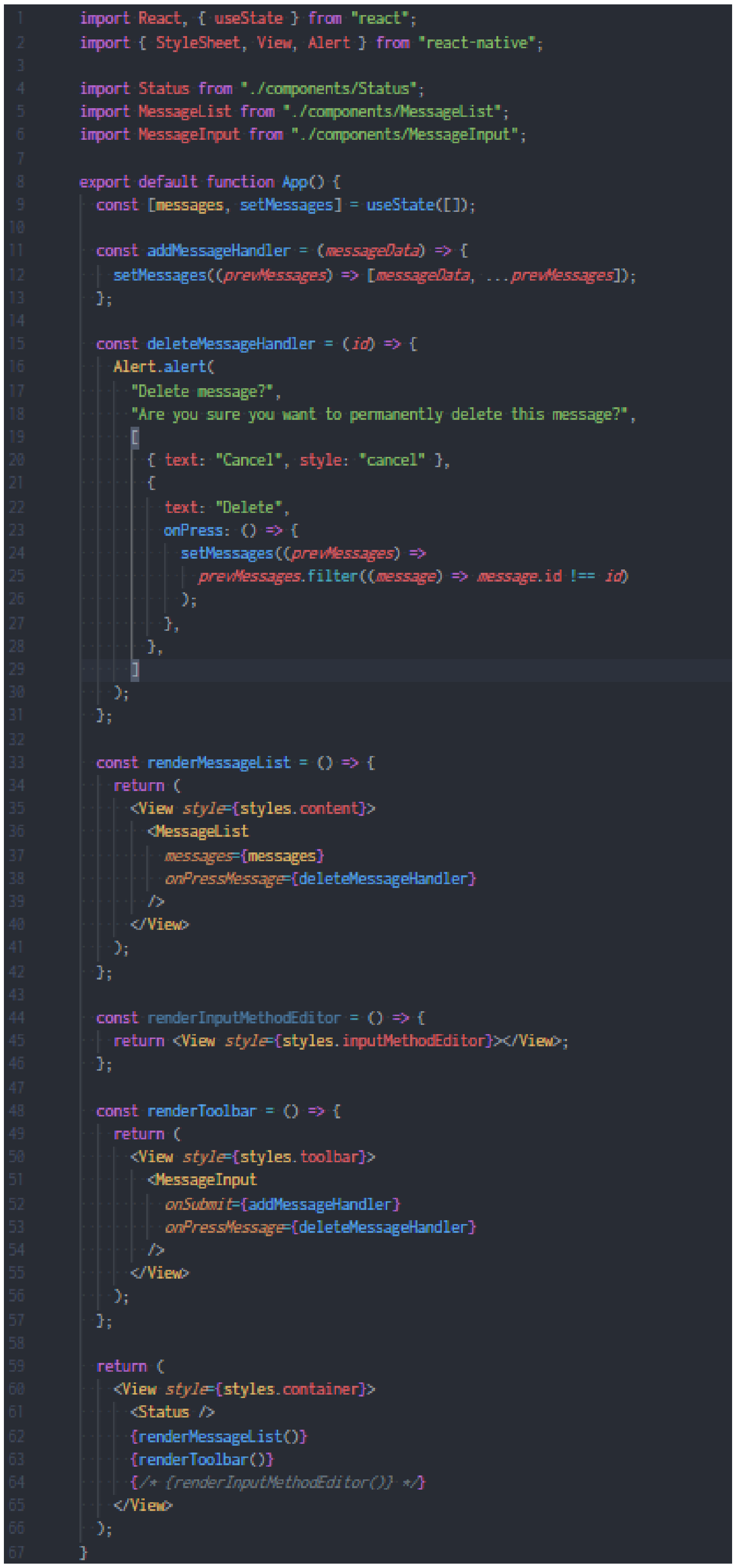
ชื่อ.....................นายศรัณย์ ยันตะบุษย์...................

รหัสนักศึกษา..............61070218...........................

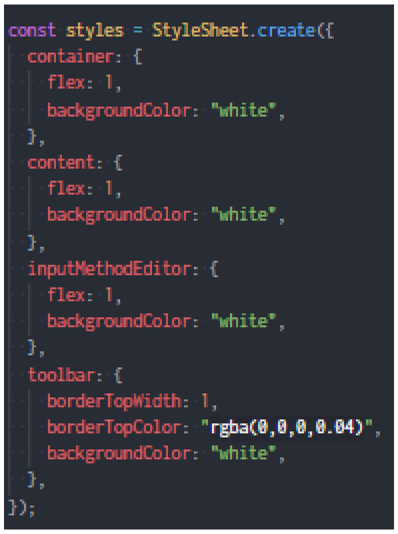


ชื่อ.....................นายศรัณย์ ยันตะบุษย์...................
รหัสนักศึกษา..............61070218...........................

App.js



```
import React, { useState } from "react";
import { StyleSheet, View, Alert } from "react-native";

import Status from "./components/Status";
import MessageList from "./components/MessageList";
import MessageInput from "./components/MessageInput";

export default function App() {
  const [messages, setMessages] = useState([]);

  const addMessageHandler = (messageData) => {
    setMessages((prevMessages) => [messageData, ...prevMessages]);
  };

  const deleteMessageHandler = (id) => {
    Alert.alert(
      "Delete message?",
      "Are you sure you want to permanently delete this message?",
      [
        { text: "Cancel", style: "cancel" },
        {
          text: "Delete",
          onPress: () => {
            setMessages((prevMessages) =>
              prevMessages.filter((message) => message.id !== id)
            );
          },
        },
      ]
    );
  };

  const renderMessageList = () => {
    return (
      <View style={styles.content}>
        <MessageList
          messages={messages}
          onPressMessage={deleteMessageHandler}
        />
      </View>
    );
  };

  const renderInputMethodEditor = () => {
    return <View style={styles.inputMethodEditor}></View>;
  };

  const renderToolbar = () => {
    return (
      <View style={styles.toolbar}>
        <MessageInput
          onSubmit={addMessageHandler}
          onPressMessage={deleteMessageHandler}
        />
      </View>
    );
  };

  return (
    <View style={styles.container}>
      <Status />
      {renderMessageList()}
      {renderToolbar()}
      {/* {renderInputMethodEditor()} */}
    </View>
  );
}
```

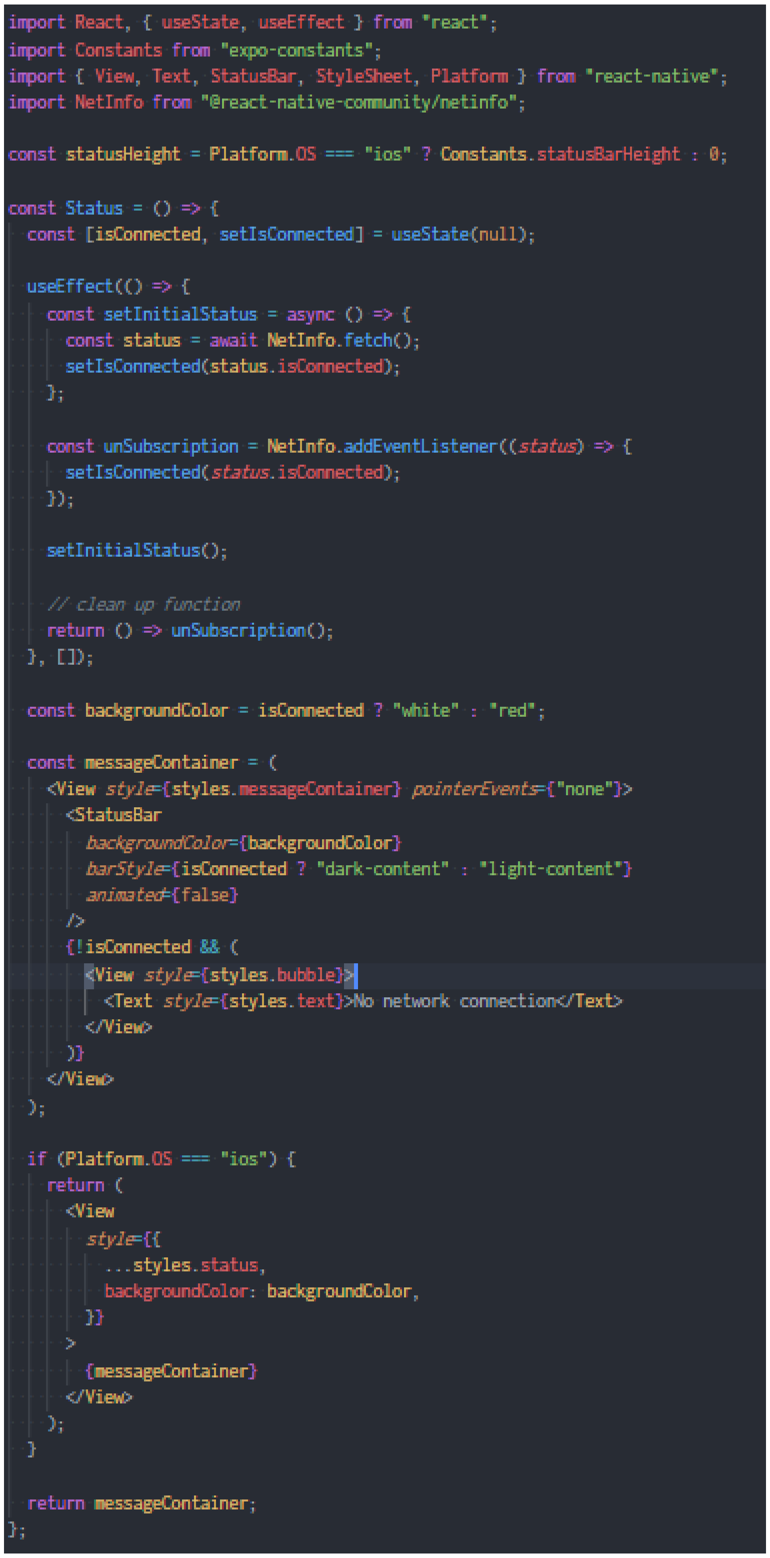


```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: "white",
  },
  content: {
    flex: 1,
    backgroundColor: "white",
  },
  inputMethodEditor: {
    flex: 1,
    backgroundColor: "white",
  },
  toolbar: {
    borderTopWidth: 1,
    borderTopColor: "rgba(0,0,0,0.04)",
    backgroundColor: "white",
  },
});
```

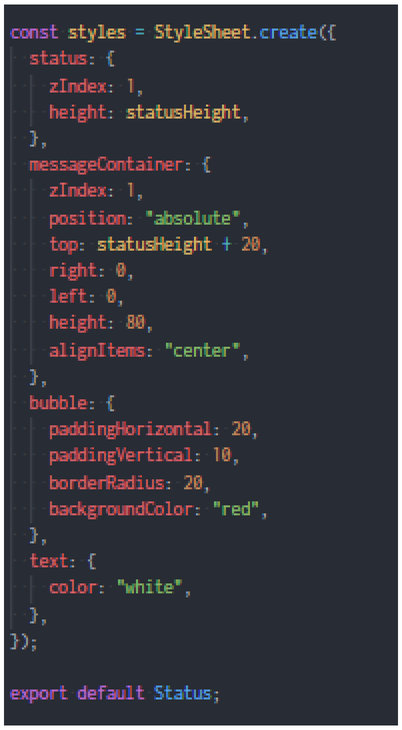
ชื่อ......................นายศรัณย์ ยันตะบุษย์....................

รหัสนักศึกษา...............61070218............................

Status.js

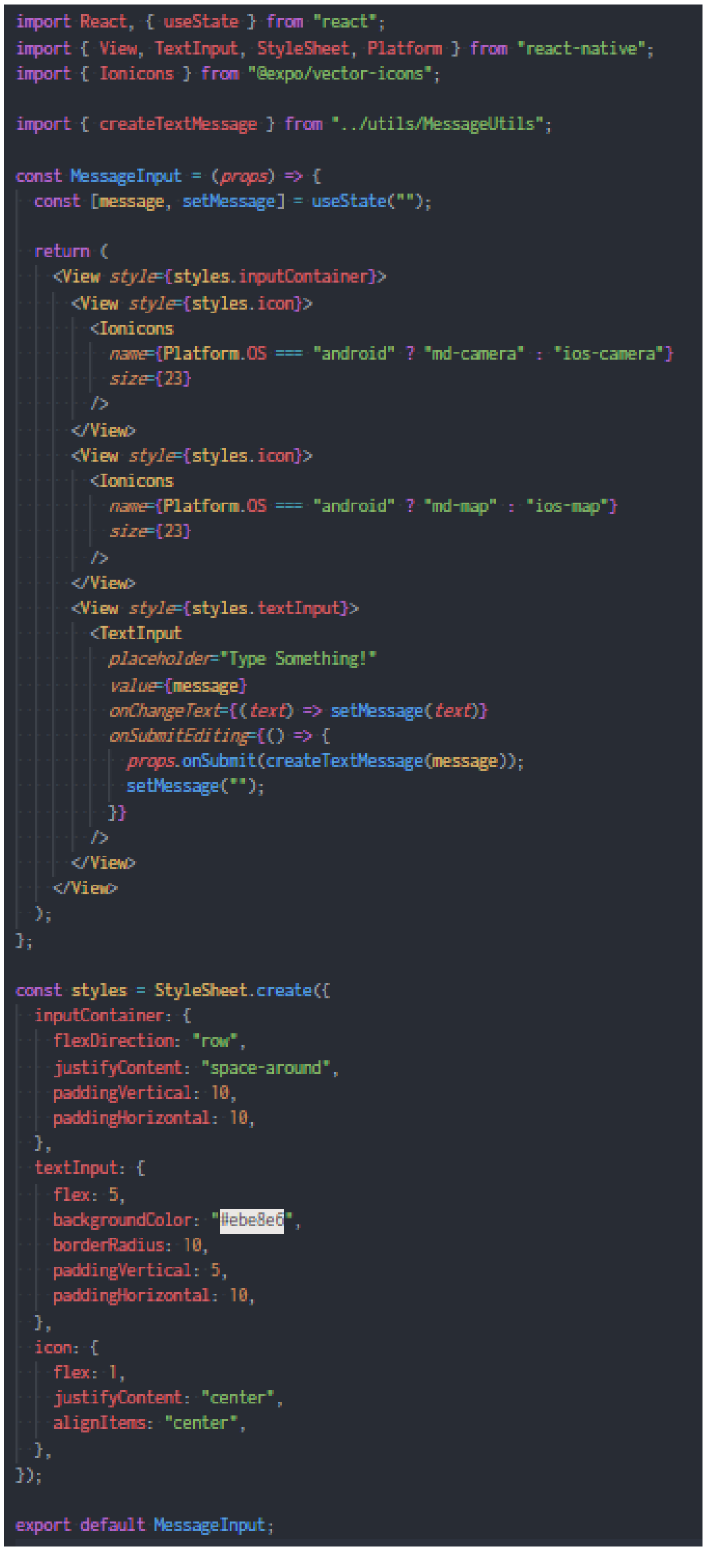


```
import React, { useState, useEffect } from "react";
import Constants from "expo-constants";
import { View, Text, StatusBar, StyleSheet, Platform } from "react-native";
import NetInfo from "@react-native-community/netinfo";

const statusHeight = Platform.OS === "ios" ? Constants.statusBarHeight : 0;

const Status = () => {
  const [isConnected, setIsConnected] = useState(null);

  useEffect(() => {
    const setInitialStatus = async () => {
      const status = await NetInfo.fetch();
      setIsConnected(status.isConnected);
    };

    const unSubscription = NetInfo.addEventListener((status) => {
      setIsConnected(status.isConnected);
    });

    setInitialStatus();

    // clean up function
    return () => unSubscription();
  }, []);

  const backgroundColor = isConnected ? "white" : "red";

  const messageContainer = (
    <View style={styles.messageContainer} pointerEvents={"none"}>
      <StatusBar
        backgroundColor={backgroundColor}
        barStyle={isConnected ? "dark-content" : "light-content"}
        animated={false}
      />
      {!isConnected && (
        <View style={styles.bubble}>
          <Text style={styles.text}>No network connection</Text>
        </View>
      )}
    </View>
  );

  if (Platform.OS === "ios") {
    return (
      <View
        style={{
          ...styles.status,
          backgroundColor: backgroundColor,
        }}
      >
        {messageContainer}
      </View>
    );
  }

  return messageContainer;
};
```



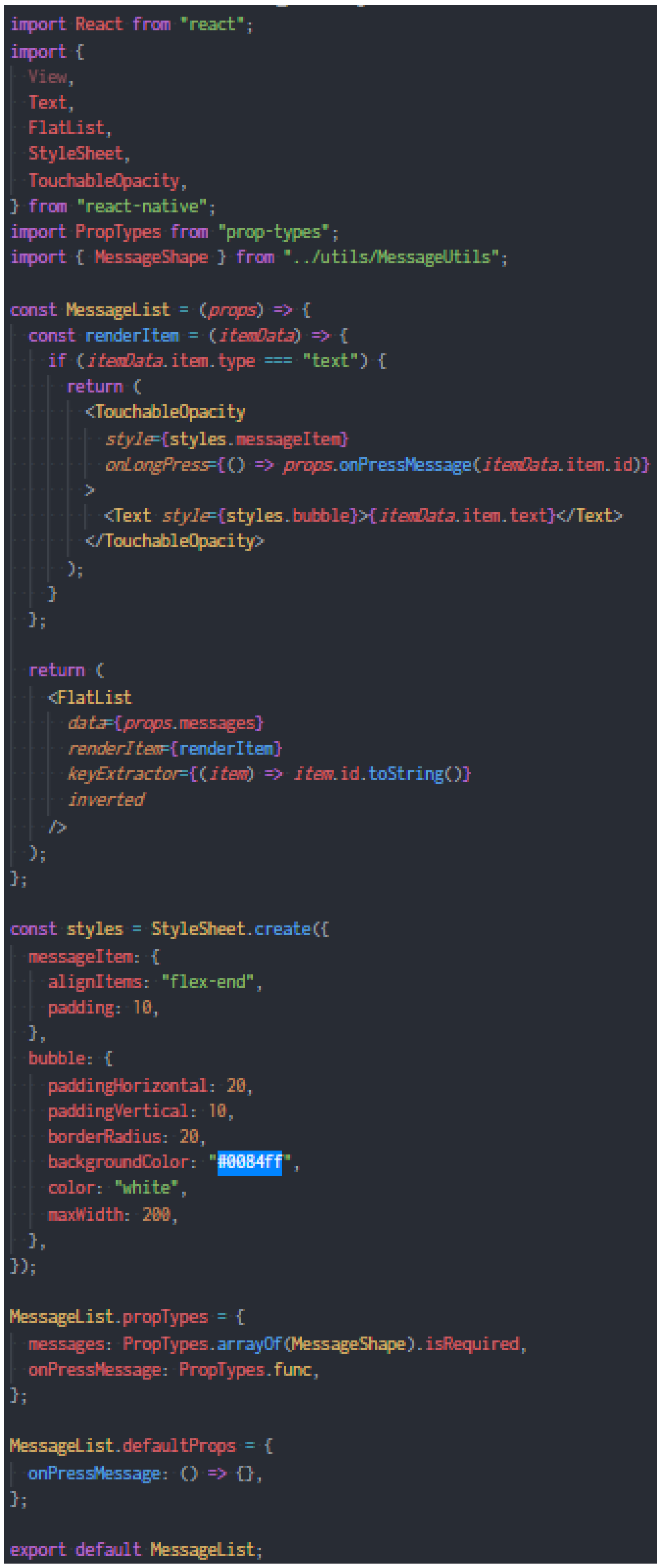
```
const styles = StyleSheet.create({
  status: {
    zIndex: 1,
    height: statusHeight,
  },
  messageContainer: {
    zIndex: 1,
    position: "absolute",
    top: statusHeight + 20,
    right: 0,
    left: 0,
    height: 80,
    alignItems: "center",
  },
  bubble: {
    paddingHorizontal: 20,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

export default Status;
```

MessageInput.js



```
import React, { useState } from "react";
import { View, TextInput, StyleSheet, Platform } from "react-native";
import { Ionicons } from "@expo/vector-icons";

import { createTextMessage } from "../utils/MessageUtils";

const MessageInput = (props) => {
  const [message, setMessage] = useState("");

  return (
    <View style={styles.inputContainer}>
      <View style={styles.icon}>
        <Ionicons
          name={Platform.OS === "android" ? "md-camera" : "ios-camera"}
          size={23}
        />
      </View>
      <View style={styles.icon}>
        <Ionicons
          name={Platform.OS === "android" ? "md-map" : "ios-map"}
          size={23}
        />
      </View>
      <View style={styles.textInput}>
        <TextInput
          placeholder="Type Something!"
          value={message}
          onChangeText={(text) => setMessage(text)}
          onSubmitEditing={() => {
            props.onSubmit(createTextMessage(message));
            setMessage("");
          }}
        />
      </View>
    </View>
  );
};

const styles = StyleSheet.create({
  inputContainer: {
    flexDirection: "row",
    justifyContent: "space-around",
    paddingVertical: 10,
    paddingHorizontal: 10,
  },
  textInput: {
    flex: 5,
    backgroundColor: "#ebe8e6",
    borderRadius: 10,
    paddingVertical: 5,
    paddingHorizontal: 10,
  },
  icon: {
    flex: 1,
    justifyContent: "center",
    alignItems: "center",
  },
});

export default MessageInput;
```

MessageList.js

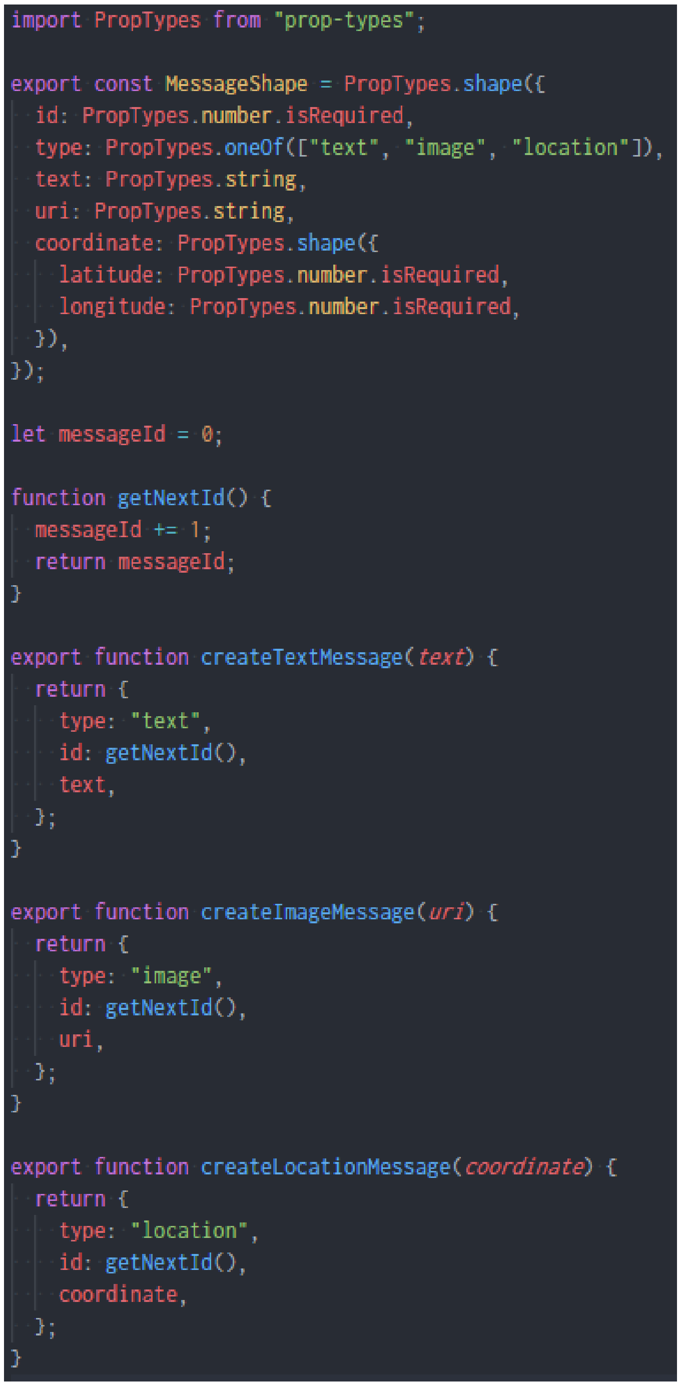


```
import React from "react";
import {
  View,
  Text,
  FlatList,
  StyleSheet,
  TouchableOpacity,
} from "react-native";
import PropTypes from "prop-types";
import { MessageShape } from "../utils/MessageUtils";

const MessageList = (props) => {
  const renderItem = (itemData) => {
    if (itemData.item.type === "text") {
      return (
        <TouchableOpacity
          style={styles.messageItem}
          onLongPress={() => props.onPressMessage(itemData.item.id)}
        >
          <Text style={styles.bubble}>{itemData.item.text}</Text>
        </TouchableOpacity>
      );
    }
  };

  return (
    <FlatList
      data={props.messages}
      renderItem={renderItem}
      keyExtractor={(item) => item.id.toString()}
      inverted
    />
  );
};

const styles = StyleSheet.create({
  messageItem: {
    alignItems: "flex-end",
    padding: 10,
  },
  bubble: {
    paddingHorizontal: 20,
    paddingVertical: 10,
    borderRadius: 20,
    backgroundColor: "#0084ff",
    color: "white",
    maxWidth: 200,
  },
});

MessageList.propTypes = {
  messages: PropTypes.arrayOf(MessageShape).isRequired,
  onPressMessage: PropTypes.func,
};

MessageList.defaultProps = {
  onPressMessage: () => {},
};

export default MessageList;
```

ชื่อ.....................นายศรัณย์ ยันตะบุษย์...................

รหัสนักศึกษา...............61070218............................

MessageUtils.js



```
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,
  };
}
```