**SCHOOL OF COMPUTING (SOC)**

**DIPLOMA IN INFORMATION TECHNOLOGY**

**ST0504 – Mobile Application Development**

2022/2023 SEMESTER 2

ASSIGNMENT 2

**Assignment 2 Submission:**

Download the dependencies below:

"dependencies": {

"@expo/ngrok": "^4.1.0",

"@expo/vector-icons": "^13.0.0",

"@react-navigation/bottom-tabs": "^6.5.3",

"@react-navigation/native": "^6.1.2",

"@react-navigation/stack": "^6.3.11",

"expo": "~47.0.12",

"expo-checkbox": "^2.2.2",

"expo-constants": "^14.0.2",

"expo-status-bar": "~1.4.2",

"react": "18.1.0",

"react-native": "0.70.5",

"react-native-circular-progress-indicator": "^4.4.2",

"react-native-paper": "^5.1.3",

"react-native-reanimated": "~2.12.0",

"react-native-safe-area-context": "^4.4.1",

"react-native-screens": "^3.18.2",

"react-native-svg": "13.4.0",

"react-native-vector-icons": "^9.2.0"

},

Add the plugin below to babel.config.js

plugins: ['react-native-reanimated/plugin'],

**1. User Interface coding**

|  |
| --- |
| **New List Page - JavaScript Code** |
| import React, { useState, Fragment } from 'react';  import { Text, View, StyleSheet, TextInput, Switch, TouchableOpacity, SafeAreaView, FlatList, StatusBar} from 'react-native';  import AntDesign from 'react-native-vector-icons/AntDesign';  const TASKS = [  { id: 1, name: "Task1" },  { id: 2, name: "Task2" },  { id: 3, name: "Task3" },  { id: 4, name: "Task4" },  { id: 5, name: "Task5" }  ]  export default function App({ navigation }) {  const [starred, setStarred] = useState(false);  const toggleStarred = () => {  setStarred(previousState => !previousState);  }  const renderTask = ({ item }) => {  return (  <View style={[styles.taskBox, { flex: 1 }]}>  <View style={{ flex: 14 }}>  <TouchableOpacity>  <Text style={styles.task}>{item.name}</Text>  </TouchableOpacity></View>  <View style={{ flex: 1 }}>  <TouchableOpacity style={styles.remove}>  <Text style={styles.minus}>X</Text>  </TouchableOpacity>  </View>  </View>  )  }  return (  <Fragment>  <SafeAreaView style = {{flex: 0, backgroundColor: 'black'}} />  <StatusBar barStyle = 'light-content' />  <SafeAreaView style={styles.container}>  <View style={[styles.bar, { flex: 1 }]}>  <View style={[styles.backButton, { flex: 3 }]}>  <TouchableOpacity style={{ flex: 1, flexDirection: "row", alignItems: "center" }} onPress={() => {navigation.navigate('NavigationBar')}}>  <AntDesign name="leftcircleo" style={styles.returnIcon} />  </TouchableOpacity>  <Text style={styles.barText}>  New List  </Text>  </View>  <View style={{ flex: 3 }} />  </View>  <View style={[styles.settings, { flex: 8 }]}>  <View style={[styles.titleBox, { flex: 1 }]}>  <TextInput style={styles.titleInput} placeholderTextColor="#888" placeholder='Untitled List...'></TextInput>  </View>  <View style={[styles.starredSection, { flex: 1 }]}>  <View style={{ justifyContent: "center" }}>  <Text style={styles.settingName}>Starred List</Text>  </View>  <View>  <Switch  trackColor={{ false: "#4C4C4C", true: "#03EF62" }}  onValueChange={toggleStarred}  thumbColor="#FFFFFF"  value={starred}  style={{ transform: [{ scaleX: 1.6 }, { scaleY: 1.6 }] }}  />  </View>  </View>  <View style={{ flex: 1, width: "100%" }}>  <View style={{ backgroundColor: "black", }}>  <Text style={styles.tasks}>Tasks</Text>  </View>  </View>  <View style={{ flex: 4, width: "100%" }}>  <View style={{ flex: 1 }}>  <FlatList style={{ flex: 1 }} data={TASKS} keyExtractor={(item) => item.id} renderItem={({ item }) => { return renderTask({ item }) }} />  </View>  <View></View>  </View>  <View style={{ flex: 1, width: "100%", alignItems: "center" }}>  <TouchableOpacity style={styles.newTask} onPress = {() => {navigation.navigate('NewTaskScreen')}}>  <Text style={styles.newTaskText}>+</Text>  </TouchableOpacity>  </View>  <View style={{ flex: 1 }} />  <View style={[{ flex: 1.2, alignItems: "center", width: "100%" }, styles.settingSection]}>  <TouchableOpacity style={styles.createButton}>  <Text style={styles.buttonText}>Create</Text>  </TouchableOpacity>  </View>  <View style={{ flex: 0.2 }} />  </View>  </SafeAreaView>  </Fragment>  )  }  const styles = StyleSheet.create({  container: {  flex: 1,  flexDirection: "column",  alignItems: "center"  },  bar: {  flex: 1,  flexDirection: "row",  backgroundColor: "black",  },  backButton: {  flexDirection: "row",  alignItems: "center"  },  barText: {  color: "white",  fontSize: 28,  fontWeight: "bold",  paddingTop: 2  },  returnIcon: {  color: "white",  fontSize: 28,  paddingLeft: 20,  paddingRight: 5,  paddingTop: 5  },  settings: {  flexDirection: "column",  alignItems: "center",  width: "100%"  },  titleBox: {  backgroundColor: "#333333",  width: "75%",  marginTop: 40,  marginBottom: 20,  alignItems: "center",  justifyContent: "center",  borderRadius: 10  },  titleInput: {  fontSize: 28,  color: "white",  },  starredSection: {  flexDirection: "row",  alignItems: "center",  justifyContent: "space-between",  width: "75%",  marginBottom: 20  },  settingName: {  fontSize: 20,  fontWeight: "500"  },  tasks: {  fontSize: 20,  fontWeight: "bold",  color: "white",  padding: 8,  paddingLeft: 20  },  taskBox: {  flex: 1,  flexDirection: "row",  paddingBottom: 8,  width: "87%",  alignItems: "center"  },  task: {  fontSize: 24,  color: "black",  fontWeight: "500",  paddingLeft: 20,  backgroundColor: "#E2E2E2",  paddingTop: 14,  paddingBottom: 14,  marginLeft: 16,  marginRight: 10,  borderRadius: 10  },  remove: {  width: 66,  height: 66,  borderRadius: 10,  backgroundColor: "red",  justifyContent: "center"  },  minus: {  color: "white",  textAlign: "center",  fontSize: 20,  padding: 0  },  newTask: {  backgroundColor: "#88AAFF",  width: "90%",  borderRadius: 10,  marginTop: 10  },  newTaskText: {  textAlign: "center",  fontWeight: "bold",  fontSize: 30  },  createButton: {  backgroundColor: "#03EF62",  width: "40%",  borderRadius: 20  },  buttonText: {  fontSize: 26,  fontWeight: "bold",  textAlign: "center",  padding: 6  }  }); |
| New List Page – Screen Capture of the Interface |
|  |

|  |
| --- |
| **Lists Page- JavaScript Code** |
| import React from 'react';  import { SafeAreaView, View, Text, StyleSheet, Pressable,} from 'react-native';  import Icon from 'react-native-vector-icons/Ionicons'  import CircularProgress from 'react-native-circular-progress-indicator';  function AddList({ navigation }) {  return (  <Pressable  style = {styles.addButtonStyle}  onPress={() => navigation.navigate('NewListsScreen')}  ><Icon name ="add-outline" size={38} color = '#fff'/>  </Pressable>  );  }  function StarredButton() {  return (  <Pressable  style = {styles.starredButtonStyle}  ><Icon name = "star" size = {30} color = "#000" /></Pressable>  )  }  export default function ListsScreen({ navigation }) {  return (  <SafeAreaView>  <View style={styles.headerContainer}>  <Text style={styles.header}>Task Lists</Text>  <AddList navigation={navigation} />  </View>  <View style={styles.allListsContianer}>  <Pressable style={styles.listsContainer} onPress={() => navigation.navigate('TasksScreen')}>  <View style={styles.starAndListName}>  <StarredButton />  <Text style={styles.listStyle}>List 1  </Text>  </View>  <CircularProgress  value={4}  maxValue={5.5}  progressValueFontSize={16}  radius={20}  clockwise={false}  progressValueColor={'#000'}  activeStrokeColor={'#03EF62'}  style={styles.progressCircle}  />  </Pressable>  <View style={styles.listsContainer}>  <Text style={styles.listStyle}>List 2  </Text>  <CircularProgress  value={7}  maxValue={35}  progressValueFontSize={16}  radius={20}  clockwise={false}  progressValueColor={'#000'}  activeStrokeColor={'#03EF62'}  style={styles.progressCircle}  />  </View>  <View style={styles.listsContainer}>  <Text style={styles.listStyle}>List 3  </Text>  <Icon name = "alert-circle-outline" size = {38} style = {styles.alertIcon}></Icon>  </View>  <View style={styles.listsContainer}>  <Text style={styles.listStyle}>List 4  </Text>  <Icon name = "checkmark-circle-outline" size = {38} style = {styles.doneIcon}></Icon>  </View>  </View>  </SafeAreaView>  );  }  const styles = StyleSheet.create({  headerContainer: {  flexDirection: 'row',  paddingHorizontal: 20,  paddingVertical: 20,  paddingHorizontal: 20,  backgroundColor: '#000',  color: '#fff',  alignItems: 'center',  justifyContent: 'space-between',  },  addButtonStyle: {  backgroundColor:'#000'  },  starredButtonStyle: {  backgroundColor: '#fff',  paddingRight:10  },  alertIcon: {  backgroundColor: '#ff0000',  borderRadius: 21,  paddingLeft:3,  overflow: 'hidden',  },  doneIcon: {  backgroundColor: '#03EF62',  borderRadius: 21,  paddingLeft:3,  overflow: 'hidden',  },  header: {  color: '#fff',  alignItems: 'center',  justifyContent: 'center',  fontSize: 24,  fontWeight: 'bold',  },  allListsContianer: {  margin: 10  },  listsContainer: {  borderRadius: 10,  flexDirection: 'row',  justifyContent: 'space-between',  backgroundColor: '#fff',  padding: 15,  alignItems: 'center',  margin: 10,  borderWidth: 2,  borderColor: 'black'  },  starAndListName: {  flexDirection: 'row',  alignItems: 'center'  },  listStyle: {  fontSize: 24  },  }); |
| Lists Page – Screen Capture of the Interface |
|  |

|  |
| --- |
| **Home Page - JavaScript Code** |
| import React, { useState } from 'react';  import { Image, SafeAreaView, View, Text, StyleSheet, ScrollView, SectionList } from 'react-native';  import CheckBox from 'expo-checkbox';  const DATA = [  {  title: { name: "Today's Task", color: '#000'},  data: ['Task 1', 'Task 2', 'Task 3'],  },  {  title: { name: 'Overdue', color: '#ff0000'},  data: ['Task 7', 'Task 8', 'Task 9'],  },  {  title: { name: 'Completed!', color: '#03ef62'},  data: ['Task 4', 'Task 5', 'Task 6'],  },  ];  export default function HomeScreen() {  const Task = ({title }) => {  const [isChecked, setChecked] = useState(false);  return (  <View style={styles.tasks}>  <Text style={styles.tasksTitle}>{title}</Text>  <CheckBox  style={styles.checkBox}  color={isChecked ? 'black' : 'black'}  value = {isChecked}  onValueChange ={setChecked}  />  </View>  );  };  return (  <SafeAreaView style={{ flex: 1, backgroundColor: '#eaeaea' }}>  <View style={styles.imageContainer}>  <Image source={require('../Screens/logo.png')} style={styles.image} />  </View>  <SectionList  sections={DATA}  keyExtractor={(item, index) => item + index}  renderItem={({ item }) => <Task title={item} />}  renderSectionHeader={({ section: { title } }) => (  <Text style={[styles.header, { color: title.color }]}>{title.name}</Text>  )}  stickySectionHeadersEnabled={false}  />  </SafeAreaView>  );  };  const styles = StyleSheet.create({  tasks: {  backgroundColor: '#fff',  padding: 20,  borderRadius: 10,  overflow: 'hidden',  marginHorizontal: 20,  marginVertical: 10,  flexDirection: 'row',  justifyContent: 'space-between'  },  header: {  fontSize: 28,  fontWeight: 'bold',  marginHorizontal: 20,  marginTop: 25,  marginBottom: 10,  },  tasksTitle: {  fontSize: 20,  },  imageContainer: {  padding: 10,  jeustifyContnt: 'center',  alignItems: 'center',  backgroundColor: 'black',  },  image: {  backgroundColor: 'black',  width: 150,  height: 50,  },  checkBox: {  alignItems: 'center'  }  }); |
| Home Page – Screen Capture of the Interface |
|  |

|  |
| --- |
| **New Tasks Page - JavaScript Code** |
| import React, { Fragment, useState } from 'react';  import { Text, View, StyleSheet, TextInput, Button, Switch, TouchableOpacity, SafeAreaView, StatusBar } from 'react-native';  import AntDesign from 'react-native-vector-icons/AntDesign';  export default function App({ navigation }) {  const [deadlineEnabled, setDeadlineEnabled] = useState(false);  const [deadline, setDeadline] = useState(null)  const [reminderEnabled, setReminderEnabled] = useState(false);  const [reminder, setReminder] = useState(null)  function toggleDeadline() {  setDeadlineEnabled(previousState => !previousState);  }  function toggleReminder() {  setReminderEnabled(previousState => !previousState);  }  return (  <Fragment>  <SafeAreaView style={{ flex: 0, backgroundColor: 'black' }} />  <StatusBar barStyle='light-content' />  <SafeAreaView style={styles.container}>  <View style={[styles.bar, { flex: 1 }]}>  <View style={[styles.backButton, { flex: 3 }]}>  <TouchableOpacity onPress={() => { navigation.navigate('NewListsScreen') }}>  <AntDesign name="leftcircleo" style={styles.returnIcon} /></TouchableOpacity>  <Text style={styles.barText}>  back  </Text>  </View>  <View style={{ flex: 5 }} />  </View>  <View style={[styles.settings, { flex: 8 }]}>  <View style={[{ flex: 1 }, styles.titleInputContainer]}>  <TextInput style={styles.titleInput} placeholderTextColor="#888" placeholder='My Task...'></TextInput>  </View>  <View style={[styles.settingSection, { flex: 2 }]}>  <View style={styles.settingSectionHeader}>  <View style={{ justifyContent: "center" }}>  <Text style={styles.settingName}>Deadline</Text>  </View>  <View>  <Switch  trackColor={{ false: "#4C4C4C", true: "#03EF62" }}  onValueChange={toggleDeadline}  thumbColor="#FFFFFF"  value={deadlineEnabled}  style={{ transform: [{ scaleX: 1 }, { scaleY: 1 }] }}  />  </View>  </View>  <View style={styles.dateButton}>  <Button disabled={!deadlineEnabled} title="Mon 01 Feb 2023" />  </View>  </View>  <View style={[styles.settingSection, { flex: 1 }]}>  <View style={styles.settingSectionHeader}>  <View style={{ justifyContent: "center" }}>  <Text style={styles.settingName}>Remind Me</Text>  </View>  <View>  <Switch  trackColor={{ false: "#4C4C4C", true: "#03EF62" }}  onValueChange={toggleReminder}  thumbColor="#FFF"  value={reminderEnabled}  style={{ transform: [{ scaleX: 1 }, { scaleY: 1 }] }}  />  </View>  </View>  <View style={styles.dateButton}>  <Button disabled={!reminderEnabled} title="1 Day Before" />  </View>  </View>  <View style={{ flex: 1 }} />  <View style={[{ flex: 4 }, styles.settingSection]}>  <View style={{ flex: 1 }}>  <Text style={styles.settingName}>Notes</Text>  </View>  <View style={[styles.notes, { flex: 4 }]}>  <TextInput multiline={true} style={styles.notesText} placeholderTextColor="#888" placeholder='Enter text here...'></TextInput>  </View>  </View>  <View style={styles.createButtonArea}>  <TouchableOpacity style={styles.createButton}>  <Text style={styles.buttonText}>Create</Text>  </TouchableOpacity>  </View>  </View>  </SafeAreaView>  </Fragment>  );  }  const styles = StyleSheet.create({  container: {  flex: 1,  flexDirection: "column",  alignItems: "center"  },  bar: {  flex: 1,  flexDirection: "row",  backgroundColor: "black",  },  backButton: {  flexDirection: "row",  alignItems: "center"  },  barText: {  color: "white",  fontSize: 28,  fontWeight: "bold",  paddingTop: 2  },  returnIcon: {  color: "white",  fontSize: 28,  paddingLeft: 20,  paddingRight: 5,  paddingTop: 5  },  settings: {  flexDirection: "column",  alignItems: "center",  width: "100%"  },  settingSection: {  flexDirection: "column",  width: "80%"  },  settingSectionHeader: {  flexDirection: "row",  justifyContent: "space-between"  },  settingName: {  fontSize: 18,  fontWeight: "500"  },  titleInputContainer: {  backgroundColor: "#333333",  justifyContent: "center",  alignItems: "center",  width: "65%",  borderRadius: 10,  marginTop: 40,  marginBottom: 20,  padding: 5  },  titleInput: {  fontSize: 28,  color: "white",  paddingLeft: 20,  paddingRight: 20,  borderRadius: 10  },  dateButton: {  backgroundColor: "#CCC",  borderRadius: 10,  overflow: 'hidden'  },  notes: {  backgroundColor: "#333333",  borderRadius: 10,  padding: 8  },  notesText: {  fontSize: 16,  color: "white"  },  createButtonArea: {  flex: 3,  width: "100%",  alignItems: "center"  },  createButton: {  flex: 1,  backgroundColor: "#03EF62",  width: "40%",  justifyContent: "center",  margin: 45,  borderRadius: 10  },  buttonText: {  color: "black",  textAlign: "center",  fontWeight: "bold",  fontSize: 24  }  }); |
| New Tasks Page – Screen Capture of the Interface |
|  |

|  |
| --- |
| **Settings Page - JavaScript Code** |
| import React, { Fragment } from 'react';  import { SafeAreaView, View, Text, StyleSheet, SectionList, Pressable, ScrollView } from 'react-native';  import Icon from 'react-native-vector-icons/Ionicons';  const SETTINGS = [  {  title: 'PREFERENCES',  data: [  { id: 'language', icon: 'globe-outline', label: 'Language', status: 'English', icon2: 'chevron-forward-outline' },  { id: 'mode', icon: 'contrast', label: 'Dark Mode', status: 'OFF', icon2: 'chevron-forward-outline' },  ]  },  {  title: 'HELP',  data: [  { id: 'resetpw', icon: 'finger-print-outline', label: 'Reset Password', icon2: 'chevron-forward-outline' },  { id: 'bug', icon: 'flag-outline', label: 'Report Bug', icon2: 'chevron-forward-outline' },  { id: 'contact', icon: 'mail-outline', label: 'Contact Us', icon2: 'chevron-forward-outline' },  ]  },  ]  export default function SettingsScreen({ navigation }) {  return (  <SafeAreaView style={{ flex: 1, backgroundColor: '#eaeaea' }}>  <Text style={styles.header}>Settings</Text>  <SectionList  sections={SETTINGS}  keyExtractor={(item, index) => item + index}  renderItem={({ item }) =>  <View style={styles.settingOptions}>  <Icon name={item.icon} size={26} style={styles.iconStyle} />  <Text style={styles.labelStyle}>{item.label}</Text>  <Text style={styles.statusStyle}>{item.status}</Text>  <Icon name={item.icon2} size={26} style={styles.icon2Style} />  </View>  }  renderSectionHeader={({ section: { title } }) => (  <Text style={styles.settingsSectionHeader}>{title}</Text>  )}  />  </SafeAreaView>  )  }  const styles = StyleSheet.create({  header: {  paddingVertical: 20,  paddingHorizontal: 20,  alignItems: 'center',  justifyContent: 'center',  backgroundColor: '#000',  fontSize: 30,  fontWeight: 'bold',  color: '#fff'  },  settingsSectionHeader: {  marginHorizontal: 20,  marginTop: 30,  fontSize: 20,  fontWeight: 'bold',  color: '#a7a7a7'  },  settingOptions: {  flex: 1,  flexDirection: 'row',  alignItems: 'center',  backgroundColor: '#fff',  padding: 15,  marginTop: 10,  marginHorizontal: 20,  borderRadius: 10,  overflow: 'hidden'  },  iconStyle: {  marginRight: 15  },  icon2Style: {  marginLeft: 3,  color: '#616161'  },  labelStyle: {  flex: 1,  fontSize: 22,  fontWeight: '400'  },  statusStyle: {  fontSize: 20,  color: '#616161',  }  }) |
| Settings Page – Screen Capture of the Interface |
|  |

|  |
| --- |
| **Tasks Page - JavaScript Code** |
| import React, { Fragment } from 'react';  import { useState } from 'react';  import { Text, View, SafeAreaView, StyleSheet, SectionList, Alert, Button, TouchableOpacity,StatusBar } from 'react-native';  import { Ionicons, FontAwesome, MaterialIcons, AntDesign } from '@expo/vector-icons';  const TASKS = [  { title: 'Mon 9 Jan', data: ['Bake Cake', 'Eat Cake'] },  { title: 'Wed 11 Jan', data: ['Do Homework'] },  { title: 'Thu 12 Jan', data: ['Get a Haircut'] },  ];    export default function App({ navigation }) {  const Bar = () => {  return (  <View style={[styles.bar, { flex: 1, flexDirection: "row" }]}>  <View style={[styles.backButton, { flex: 3 }]}>  <TouchableOpacity onPress={() => { navigation.navigate('NavigationBar') }}><AntDesign name="leftcircleo" style={styles.returnIcon} /></TouchableOpacity>  <Text style={styles.barText}>  List 1's Tasks  </Text>  </View>  <View style={{ flex: 1 }} />  <View style={{ flex: 1, justifyContent: "center" }}>  <Ionicons name="settings-outline" style={styles.returnIcon} />  </View>  </View>  )  }  return (  <Fragment>  <SafeAreaView style={{ flex: 0, backgroundColor: 'black' }} />  <StatusBar barStyle='light-content' />  <SafeAreaView style={styles.container}>  <Bar />  <Tasks />  </SafeAreaView>  </Fragment>  );  }    const Tasks = () => {  return (  <View style={{ flex: 9 }}>  <SectionList  sections={TASKS}  renderItem={renderTask}  renderSectionHeader={renderDay}>  </SectionList>  </View>  )  }  //Functions  const renderDay = ({ section }) => {  return (  <Text style={[styles.day, { color: section.title == "Wed 11 Jan" ? "#03EF62" : "#888888" }, styles.day]}>{section.title}</Text>  )  }  const renderTask = ({ item }) => {  return (  <View style={styles.task}>  <Text style={styles.taskText}>  {item}  </Text>  <View style={[styles.button, { color: "black" }]}>  <Button title="" onPress={() => alert(item + " Task Done!")} />  </View>  </View>  )  }  const styles = StyleSheet.create({  container: {  flex: 1,  flexDirection: "column"  },  //Top Bar Styles  bar: {  justifyContent: "center",  backgroundColor: "black",  marginBottom: 10  },  backButton: {  flexDirection: "row",  alignItems: "center"  },  barText: {  color: "white",  fontSize: 28,  fontWeight: "bold",  paddingTop: 2,  paddingLeft: 10  },  returnIcon: {  color: "white",  fontSize: 28,  paddingLeft: 20,  paddingRight: 5,  paddingTop: 5  },  day: {  textAlign: "center",  fontSize: 18,  backgroundColor: "black",  fontWeight: "bold"  },  //Styles of all the tasks  task: {  padding: 12,  flexDirection: "row",  justifyContent: "space-between",  },  button: {  width: 24,  border: "solid 1px black",  borderRadius: 4,  height: 24,  backgroundColor: "black",  },  taskText: {  fontSize: 20  },  }); |
| Tasks Page – Screen Capture of the Interface |
| **Text  Description automatically generated** |

**2. Individual Contribution**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Assignment 2 Contribution | % Contribute to the Assignment 2 |
|  | Example |  |  |
| 1 | Chia Jun Hao, Javier | Home Page  Settings Page  Lists Page | 50% |
| 2 | Dillan Tan | Tasks Page  Add Tasks Page  Add Lists Page | 50% |
|  |  | **Total** | 100 % |