**SCHOOL OF COMPUTING (SOC)**

**DIPLOMA IN INFORMATION TECHNOLOGY**

**ST0504 – Mobile Application Development**

2022/2023 SEMESTER 2

ASSIGNMENT 2

**Class:** DIT/FT/1B/05

**Names:** Dawn Oh Le Qian (2222923), Jeanette Ong Jing Xuan (2222808)

**Assignment 2 Submission:**

**1. User Interface coding**

|  |
| --- |
| **User Interface 1 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written.  State the command if you use external library  (example-> npm install react-datepicker –save)  import { useState } from "react";  import {    StyleSheet,    Text,    View,    Image,    TouchableOpacity,    SafeAreaView,    TextInput  } from "react-native";  export default function SignIn() {    const [email, onChangeEmail] = useState(null);    const [password, onChangePassword] = useState(null);    return (      <View style={styles.container}>        <Image source={require("./assets/logo.png")} style={styles.image} />        <Text style={styles.title}>Sign In</Text>        <Text style={styles.text}>Welcome back!</Text>        <SafeAreaView>          <Text style={styles.inputTitle}>Email</Text>          <TextInput            style={styles.input}            onChangeText={onChangeEmail}            value={email}          />          <Text style={styles.inputTitle}>Password</Text>          <TextInput            secureTextEntry={true}            style={styles.input}            onChangeText={onChangePassword}            value={password}          />        </SafeAreaView>        <View style={styles.btnView}>          <TouchableOpacity style={styles.button}>            <Text style={styles.btnText}>Sign in</Text>          </TouchableOpacity>          <View style={styles.bottom}>            <Text style={styles.bottomText}>              <Text>Don't have an Account? </Text>              <Text style={styles.signup}>Sign up</Text>            </Text>          </View>        </View>      </View>    );  }  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: "#fff",      alignItems: "flex-start",      fontFamily: "Roboto",    },    title: {      fontSize: 30,      marginLeft: 40,      fontWeight: "bold",    },    text: {      color: "#989eb1",      marginLeft: 40,      marginTop: 10,      fontWeight: "600",    },    image: {      margin: 50,      width: "30%",      height: "10%",      alignSelf: "center",    },    input: {      borderColor: "#989eb1",      borderBottomWidth: 1,      height: 35,      padding: 10,      margin: 15,      marginLeft: 40,      minWidth: 300,    },    inputTitle: {      color: "#f85f6a",      marginLeft: 40,      marginTop: 20,      fontWeight: "700",    },    btnView: {      marginTop: 20,      width: "100%",      height: 60,      alignItems: "center",    },    button: {      justifyContent: "center",      alignSelf: "center",      backgroundColor: "#f85f6a",      padding: 15,      paddingHorizontal: 120,      borderRadius: 10,    },    btnText: {      color: "white",      fontWeight: "900",    },    bottom: {      flexDirection: "column",      margin: 10,    },    bottomText: {      color: "#989eb1",      fontWeight: "900",    },    signup: {      color: "#f85f6a",    },  }); |
| User Interface 1 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here |

|  |
| --- |
| **User Interface 2 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  import { useState } from "react";  import {    StyleSheet,    Text,    View,    Image,    TouchableOpacity,    SafeAreaView,    TextInput  } from "react-native";  export default function SignIn() {    const [email, onChangeEmail] = useState(null);    const [password, onChangePassword] = useState(null);    const [confirmPassword, onChangeConfirmPassword] = useState(null);    const checkIfPasswordsMatch = () => {      if (email === "" || email === null) {        alert('Please enter email');      }      else if (password === "" || password === null){        alert('Please enter password');      }      else if (password === confirmPassword) {        alert('Account created successfully');      } else {        alert('Passwords do not match');      }    }    return (      <View style={styles.container}>        <Image source={require("./assets/logo.png")} style={styles.image} />        <Text style={styles.title}>Sign Up</Text>        <Text style={styles.text}>Welcome back!</Text>        <SafeAreaView>          <Text style={styles.inputTitle}>Email</Text>          <TextInput            style={styles.input}            onChangeText={onChangeEmail}            value={email}          />          <Text style={styles.inputTitle}>Password</Text>          <TextInput            secureTextEntry={true}            style={styles.input}            onChangeText={onChangePassword}            value={password}          />          <Text style={styles.inputTitle}>Confirm Password</Text>          <TextInput            secureTextEntry={true}            style={styles.input}            onChangeText={onChangeConfirmPassword}            value={confirmPassword}          />        </SafeAreaView>        <View style={styles.btnView}>          <TouchableOpacity style={styles.button} onPress={checkIfPasswordsMatch}>            <Text style={styles.btnText}>Continue</Text>          </TouchableOpacity>          <View style={styles.bottom}>            <Text style={styles.bottomText}>              <Text>Have an Account? </Text>              <Text style={styles.signup}>Sign up</Text>            </Text>          </View>        </View>      </View>    );  }  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: "#fff",      alignItems: "flex-start",      fontFamily: "Roboto",    },    title: {      fontSize: 30,      marginLeft: 40,      fontWeight: "bold",    },    text: {      color: "#989eb1",      marginLeft: 40,      marginTop: 10,      fontWeight: "600",    },    image: {      margin: 50,      width: "30%",      height: "10%",      alignSelf: "center",    },    input: {      borderColor: "#989eb1",      borderBottomWidth: 1,      height: 35,      padding: 10,      margin: 15,      marginLeft: 40,      minWidth: 300,    },    inputTitle: {      color: "#f85f6a",      marginLeft: 40,      marginTop: 20,      fontWeight: "700",    },    btnView: {      marginTop: 20,      width: "100%",      height: 60,      alignItems: "center",    },    button: {      justifyContent: "center",      alignSelf: "center",      backgroundColor: "#f85f6a",      padding: 15,      paddingHorizontal: 120,      borderRadius: 10,    },    btnText: {      color: "white",      fontWeight: "900",    },    bottom: {      flexDirection: "column",      margin: 10,    },    bottomText: {      color: "#989eb1",      fontWeight: "900",    },    signup: {      color: "#f85f6a",    },  }); |
| User Interface 2 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here    Details entered      Alert telling user account is created    If user password inputs don’t match    If email is not entered    If password |

|  |
| --- |
| **User Interface 3 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  expo install react-native-svg@13.4.0 –npm  npm install react-native-svg-transformer –save  npm install apexcharts –save  npm install react-native-redash  npm install react-native-reanimated  expo install react-native-chart-kit --npm  import { useRef } from "react";  import {    View,    StyleSheet,    Text,    Dimensions,    TouchableOpacity,    Image,    FlatList,    SafeAreaView,    DrawerLayoutAndroid,  } from "react-native";  import { ProgressChart } from "react-native-chart-kit";  const screenWidth = Dimensions.get("window").width;  const Item = ({ title, amount, index }) => (    <View style={styles.entireItem}>      <View style={styles.SeparatorLine} />      <View style={styles.item}>        <View style={styles.left}>          <Text style={styles.color}>            <Image source={images[index]} style={styles.icons} />            {title}          </Text>        </View>        <View style={styles.right}>          <Text style={styles.color}>{amount}</Text>        </View>      </View>    </View>  );  const NavItem = ({ title, index, selected }) => (    <View  style={[selected === 0 ? styles.navColor : null, styles.listItem]}>      <TouchableOpacity>        <Text style={styles.navItemText}>          <Image source={navImages[index]} style={styles.icons} />          {"     "}<View style={styles.itemText}><Text>{title}</Text></View><Image source={arrows[selected]} style={[styles.icons, styles.arrows]} />        </Text>      </TouchableOpacity>    </View>  );  const images = [    require("./assets/food.png"),    require("./assets/transport.png"),    require("./assets/others.png"),  ];  const navImages = [    require("./assets/home.png"),    require("./assets/calendar.png"),    require("./assets/statistics.png"),    require("./assets/transactions.png"),  ];  const arrows = [    require("./assets/redArrow.png"),    require("./assets/greyArrow.png"),  ];  const renderItem = ({ item }) => (    <View>      <Item title={item.title} amount={item.amount} index={item.index} />    </View>  );  const renderNavItem = ({ item }) => (    <View>      <NavItem title={item.title} index={item.index} selected={item.selected} />    </View>  );  export default function App() {    const data = {      data: [0.4],    };    const list = [      {        id: 1,        title: "Food",        amount: "$6",        index: 0,      },      {        id: 2,        title: "Transport",        amount: "$2",        index: 1,      },      {        id: 3,        title: "Others",        amount: "$3",        index: 2,      },    ];    const navigationItems = [      { index: 0, title: "Home", selected: 0 },      { index: 1, title: "Calendar", selected: 1 },      { index: 2, title: "Statistics", selected: 1 },      { index: 3, title: "Transactions", selected: 1 },    ];    const drawer = useRef(null);    const navigationView = () => (      <View>        <TouchableOpacity          style={styles.close}          activeOpacity={0.5}          onPress={() => drawer.current.closeDrawer()}        >          <Image            source={require("./assets/closeButton.png")}            style={styles.closeImage}          />        </TouchableOpacity>        <FlatList          data={navigationItems}          renderItem={renderNavItem}          keyExtractor={(item) => item.index.toString()}          style = {styles.selectionList}        />      </View>    );    return (      <DrawerLayoutAndroid        ref={drawer}        drawerWidth={200}        renderNavigationView={navigationView}      >        <View style={styles.container}>          <TouchableOpacity            style={styles.hamburger}            activeOpacity={0.5}            onPress={() => drawer.current.openDrawer()}          >            <Image source={require("./assets/hamburger.png")} />          </TouchableOpacity>          <Text style={styles.title}>Home</Text>          <Text style={styles.text}>Welcome back!</Text>          <View style={styles.SeparatorLine} />          <Text style={styles.todaySpendings}>Today's Spendings</Text>          <ProgressChart            data={data}            width={screenWidth - 15}            height={150}            chartConfig={{              backgroundGradientFrom: "#fff",              backgroundGradientTo: "#fff",              color: (opacity = 1) => `rgba(248, 95, 106, ${opacity})`,            }}            hideLegend={true}            radius={55}            style={styles.chart}          />          <View style={styles.money}>            <Text style={styles.value}>$11</Text>          </View>          <View style={styles.box}>            <Text style={styles.percentage}>              You spent {data.data \* 100}% of your recommended budget            </Text>          </View>          <SafeAreaView style={styles.spendingList}>            <FlatList              data={list}              renderItem={renderItem}              keyExtractor={(item) => item.id}            />          </SafeAreaView>        </View>      </DrawerLayoutAndroid>    );  }  const styles = StyleSheet.create({    container: {      backgroundColor: "#fff",      alignItems: "flex-start",      fontFamily: "Roboto",    },    title: {      fontSize: 30,      marginLeft: 40,      fontWeight: "500",      color: "35424a",    },    text: {      marginLeft: 40,      marginTop: 10,      fontWeight: "400",      color: "#35424a",    },    hamburger: {      marginTop: 50,      marginBottom: 25,      marginLeft: 30,      width: "6%",      height: "3%",    },    listItem: {      alignItems: "flex-start"    },    itemText: {      width: 100,    },    arrows: {      alignSelf: "flex-end"    },    close: {      marginTop: 50,      marginLeft: 15,    },    closeImage: {      height: 25,      width: 25,    },    selectionList: {      marginTop: 30,    },    navColor: {      backgroundColor: 'rgba(248, 95, 106, 0.1)'    },    navItemText: {      fontSize: 20,      marginLeft: 20,      marginVertical: 10,    },    SeparatorLine: {      borderBottomColor: "rgba(222, 226, 230, 0.5)",      marginHorizontal: 25,      marginVertical: 10,      borderBottomWidth: 2,      alignSelf: "stretch",    },    todaySpendings: {      color: "#989eb1",      fontWeight: "600",      alignSelf: "center",    },    money: {      position: "absolute",      top: "22%",      left: 0,      right: 0,      bottom: 0,      alignItems: "center",    },    chart: {      marginLeft: 7,      alignItems: "center",    },    value: {      color: "#f85f6a",      fontWeight: "700",      fontSize: 25,    },    box: {      width: "80%",      padding: 15,      marginTop: 25,      alignSelf: "center",      backgroundColor: "rgba(222, 226, 230, 0.25)",    },    percentage: {      color: "#92d36e",      alignSelf: "center",      fontSize: 24,    },    spendingList: {      marginTop: 50,    },    icons: {      height: 17,      width: 17,    },    entireItem: {      marginVertical: 5,    },    item: {      flexDirection: "row",      alignSelf: "center",    },    left: {      alignSelf: "flex-start",      width: "50%",      marginLeft: 95,      marginVertical: 5,    },    right: {      width: "40%",      alignSelf: "flex-end",      marginVertical: 5,    },    color: {      color: "#989eb1",      fontWeight: "600",    },  }); |
| User Interface 3 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here    When hamburger icon is clicked |

|  |
| --- |
| **User Interface 4 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  npm install react-native-dropdown-select-list  import { useState } from "react";  import {    StyleSheet,    View,    Text,    Image,    TouchableOpacity,    SafeAreaView,    FlatList,  } from "react-native";  import { SelectList } from "react-native-dropdown-select-list";  const images = [    require("./assets/shopping2.png"),    require("./assets/tips2.png"),    require("./assets/transport2.png"),    require("./assets/entertainment2.png"),  ];  export default function App() {    const [selected, setSelected] = useState("Month");    const selectionOptions = [      { key: 1, value: "Day" },      { key: 2, value: "Month" },      { key: 3, value: "Year" },    ];    const renderItem = ({ item }) => (      <Item        title={item.title}        amount={item.amount}        category={item.category}        index={item.index}        type={item.type}        year = {item.year}        month = {item.month}        day = {item.day}      />    );    const Item = ({ title, amount, category, index, type, year, month, day }) => (      // for Month option, all transactions which occurred in the same month (eg january) will appear      // for year option, all transactions which occurred in the same year (eg 2023) will appear      // for day option, all transactions which occurred in the same day will appear      <View style={selected === "Month" && year === 0 && month < 1 || selected === "Year" && year === 0 || selected === "Day" && day === 0 && month === 0 && year === 0 ? styles.entireItem : styles.hide}>        <View style={styles.item}>          <View style={styles.left}>            <Text style={styles.date}>{title}</Text>          </View>          <View style={styles.right}>            <Text style={type === "gain" ? styles.color1 : styles.color2}>              {amount}            </Text>          </View>        </View>        <View style={styles.item}>          <View style={[styles.left]}>            <Text style={[styles.color, styles.category]}>              <Image source={images[index]} style={styles.icons} /> {category}            </Text>          </View>          <View style={styles.right}>            <Text style={styles.details}>Merchant Details{">"}</Text>          </View>        </View>      </View>    );    const todaysDate = new Date();    const DATA = [      {        id: 1,        title: "12/1/2023",        amount: "$15.30",        category: "Shopping",        index: 0,        type: "loss",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 12,      },      {        id: 2,        title: "8/1/2023",        amount: "$2.42",        category: "tips",        index: 1,        type: "gain",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 8,      },      {        id: 3,        title: "7/1/2023",        amount: "$8.00",        category: "Shopping",        index: 0,        type: "loss",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 7,      },      {        id: 4,        title: "6/1/2023",        amount: "$2.00",        category: "Transport",        index: 2,        type: "loss",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 6,      },      {        id: 5,        title: "4/1/2023",        amount: "$106.99",        category: "Entertainment",        index: 3,        type: "loss",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 4,      },      {        id: 6,        title: "2/1/2023",        amount: "$1.26",        category: "Transport",        index: 2,        type: "loss",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 2,      },      {        id: 7,        title: "1/1/2023",        amount: "$5.00",        category: "tips",        index: 1,        type: "gain",        year: todaysDate.getFullYear() - 2023,        month: todaysDate.getMonth() - 0,        day: todaysDate.getDate() - 1,      },    ];    return (      <View style={styles.container}>        <TouchableOpacity style={styles.hamburger} activeOpacity={0.5}>          <Image source={require("./assets/hamburger.png")} />        </TouchableOpacity>        <Text style={styles.title}>Transactions</Text>        <SelectList          setSelected={(val) => setSelected(val)}          data={selectionOptions}          save="value"          search={false}          placeholder={"Month"}          defaultOption={"Month"}          boxStyles = {styles.selectList}          inputStyles = {{color: "white"}}          dropdownStyles = {{height: 140, marginLeft: 30,}}          arrowicon = {<Image source={require("./assets/arrow.png")} style={styles.arrow}/>}        />        <SafeAreaView style={styles.flatlist}>          <FlatList            data={DATA}            renderItem={renderItem}            keyExtractor={(item) => item.id}          />        </SafeAreaView>      </View>    );  }  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: "#fff",      alignItems: "flex-start",      fontFamily: "Roboto",    },    title: {      fontSize: 25,      marginLeft: 40,      fontWeight: "500",      color: "#35424a",    },    hamburger: {      marginTop: 50,      marginBottom: 30,      marginLeft: 30,      width: "4%",      height: "2%",    },    dropdown: {      width: "30%",    },    selectList: {      marginLeft: 40,      marginVertical: 10,      backgroundColor: "#f85f6a",    },    arrow: {      height: 18,      width: 18,    },    flatlist: {      alignSelf: "center",      flex: 1,      flexGrow: 1,    },    entireItem: {      alignSelf: "center",      marginVertical: 10,      padding: 8,      borderWidth: 1,      borderRadius: 10,      borderColor: "#989eb1",      width: "90%",    },    hide: {      display: "none",    },    item: {      flexDirection: "row",      alignSelf: "center",    },    date: {      color: "#989eb1",      fontWeight: "700",    },    left: {      alignSelf: "flex-start",      width: "50%",      marginLeft: 50,      marginVertical: 5,    },    right: {      width: "40%",      alignSelf: "flex-end",      marginVertical: 5,    },    color1: {      color: "#92d36e",      fontWeight: "700",    },    color2: {      color: "#f85f6a",      fontWeight: "700",    },    category: {      backgroundColor: "#989eb1",      alignSelf: "flex-start",      padding: 5,      borderRadius: 5,      width: "78%",    },    icons: {      width: 17,      height: 17,    },    color: {      color: "white",    },    details: {      paddingVertical: 5,    },  }); |
| User Interface 4 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here  When year is selected    When month is selected    When day is selected |

|  |
| --- |
| **User Interface 5 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  npm install react-native-calendars --save  npm install react-native-bottom-drawer-view --save  import { useState, useEffect } from 'react';  import { StyleSheet, Text, View, Image, TouchableOpacity, FlatList } from 'react-native';  import { Calendar } from 'react-native-calendars';  import BottomDrawer from 'react-native-bottom-drawer-view';  const App = () => {    const [selectedDay, setSelectedDay] = useState();    const [markedDates, setMarkedDates] = useState();    const DATA = [      { id: '1', title: 'Food', image: require('./assets/food.png'), selectedImage: require('./assets/foodRed.png'), cost: '$2' },      { id: '2', title: 'Transport', image: require('./assets/transport.png'), selectedImage: require('./assets/transportRed.png'), cost: '$1' },      { id: '3', title: 'Shopping', image: require('./assets/shopping.png'), selectedImage: require('./assets/shoppingRed.png'), cost: '$2' },      { id: '4', title: 'Apparel', image: require('./assets/apparel.png'), selectedImage: require('./assets/apparelRed.png'), cost: '$1' },      { id: '5', title: 'Education', image: require('./assets/education.png'), selectedImage: require('./assets/educationRed.png'), cost: '$1' },      { id: '6', title: 'Entertainment', image: require('./assets/gaming.png'), selectedImage: require('./assets/gamingRed.png'), cost: '$2' },      { id: '7', title: 'Others', image: require('./assets/others.png'), selectedImage: require('./assets/othersRed.png'), cost: '$1' },    ]    const renderItem = ({ item }) => (      <Item        title={item.title}        image={item.image}        style={styles.item}        cost={item.cost}      />    );    function Item({ title, image, style, cost }) {      return (        <View style={{ flexDirection: 'row', justifyContent: 'space-between' }}>          <View style={{ flexDirection: 'row' }}>            <Image source={image} style={styles.icons} resizeMode='contain' />            <Text style={style}>{title}</Text>          </View>          <Text style={styles.cost}>{cost}</Text>        </View>      );    }    useEffect(() => {      const today = new Date();      const year = today.getFullYear();      const month = String(today.getMonth() + 1).padStart(2, '0');      const day = String(today.getDate()).padStart(2, '0');      const currentDate = `${year}-${month}-${day}`;      setSelectedDay(currentDate);      setMarkedDates({        [currentDate]: { selected: true, selectedColor: '#f85f6a' },      });    }, []);    const handleDayPress = (day) => {      setSelectedDay(day.dateString);      setMarkedDates({        [day.dateString]: { selected: true, selectedColor: '#f85f6a' },      });    }    const formatDate = (dateString) => {      const monthArr = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']      const date = new Date(dateString);      const day = String(date.getDate());      const month = monthArr[date.getMonth()];      const year = date.getFullYear();      return `${day} ${month} ${year}`;    }    return (      <View>        <TouchableOpacity>          <View>            <Image source={require('./assets/hamburger.png')} style={styles.hamburger} resizeMode='contain' />          </View>        </TouchableOpacity>        <Text style={styles.calendarTitle}>Calendar</Text>        <>          <Calendar            onDayPress={handleDayPress}            markedDates={markedDates}            monthFormat={'MMM yyyy'}            hideExtraDays={true}            onPressArrowLeft={subtractMonth => subtractMonth()}            onPressArrowRight={addMonth => addMonth()}            enableSwipeMonths={true}            style={styles.calendar}            theme={{              todayTextColor: null,              arrowColor: '#f85f6a',            }}          />          <FlatList            data={DATA}            renderItem={renderItem}            keyExtractor={item => item.id}            style={styles.flatList}          />          <BottomDrawer            containerHeight={190}            shadow={true}   //shadow only appears on ios            startUp={false}            roundedEdges={true}          >            <View>              <View style={styles.header}>                <Text style={styles.blank}>empty</Text>                <Text style={styles.date}>{formatDate(selectedDay)}</Text>                <TouchableOpacity>                  <Image source={require('./assets/cross.png')} style={styles.cross} resizeMode='contain' />                </TouchableOpacity>              </View>              <View style={styles.expensesRow}>                <Text style={styles.expenses}>Expenses</Text>                <Text style={styles.moneyLost}>$10</Text>              </View>              <View style={styles.incomeRow}>                <Text style={styles.income}>Income</Text>                <Text style={styles.moneyEarned}>$0</Text>              </View>              <TouchableOpacity style={styles.plusButton}>                <Text style={styles.plus}>+</Text>              </TouchableOpacity>            </View>          </BottomDrawer>        </>      </View>    );  };  export default App;  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: '#fff',      alignItems: 'center',      justifyContent: 'center',    },    hamburger: {      height: 30,      width: 30,      marginTop: 50,      marginLeft: 40,    },    calendarTitle: {      fontSize: 25,      color: '#35424a',      fontFamily: 'Roboto',      fontWeight: 'bold',      marginLeft: 40,    },    item: {      color: '#606060',      paddingBottom: 20,    },    icons: {      width: 20,      height: 20,      marginRight: 15    },    flatList: {      paddingLeft: 35,      paddingTop: 20,      height: 180,    },    cost: {      paddingRight: 100,      color: '#f85f6a'    },    calendar: {      borderWidth: 1,      borderColor: 'gray',      marginTop: 10,      marginLeft: 15,      marginRight: 15    },    blank: {      color: 'white'    },    date: {      paddingTop: 10,      fontSize: 25,      color: '#35424a'    },    cross: {      height: 30,      width: 30,      marginTop: 13,      marginRight: 10    },    expensesRow: {      flexDirection: 'row',      justifyContent: 'space-evenly',      paddingTop: 20    },    expenses: {      color: '#989eb1',      fontSize: 15    },    moneyLost: {      color: '#f85f6a',      fontSize: 15    },    incomeRow: {      flexDirection: 'row',      justifyContent: 'space-evenly',      paddingTop: 20    },    income: {      color: '#989eb1',      fontSize: 15    },    moneyEarned: {      color: '#92d36e',      fontSize: 15    },    plusButton: {      backgroundColor: '#f85f6a',      borderRadius: 100,      position: 'absolute',      bottom: -50,      right: 20,      zIndex: 1    },    plus: {      fontSize: 30,      color: 'white',      paddingHorizontal: 20,      paddingVertical: 10    },    header: {      flexDirection: 'row',      justifyContent: 'space-around'    }  }); |
| User Interface 5 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here |

|  |
| --- |
| **User Interface 6 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  import { useState } from 'react';  import { StyleSheet, Text, View, Image, TextInput, FlatList, TouchableOpacity } from 'react-native';  const App = () => {    const [value, setValue] = useState('');    const [selected, setSelected] = useState();    const DATA = [      { id: '1', title: 'Food', image: require('./assets/food.png'), selectedImage: require('./assets/foodRed.png') },      { id: '2', title: 'Transport', image: require('./assets/transport.png'), selectedImage: require('./assets/transportRed.png') },      { id: '3', title: 'Shopping', image: require('./assets/shopping.png'), selectedImage: require('./assets/shoppingRed.png') },      { id: '4', title: 'Apparel', image: require('./assets/apparel.png'), selectedImage: require('./assets/apparelRed.png') },      { id: '5', title: 'Education', image: require('./assets/education.png'), selectedImage: require('./assets/educationRed.png') },      { id: '6', title: 'Entertainment', image: require('./assets/gaming.png'), selectedImage: require('./assets/gamingRed.png') },      { id: '7', title: 'Others', image: require('./assets/others.png'), selectedImage: require('./assets/othersRed.png') },    ];    function Item({ id, title, image, selectedImage, onPress, style }) {      return (        <TouchableOpacity onPress={onPress} style={{ style }}>          <View style={{ flexDirection: 'row' }}>            <Image source={selected === id ? selectedImage : image} style={styles.icons} resizeMode='contain' />            <Text style={style}>{title}</Text>          </View>        </TouchableOpacity>      );    }    const handlePress = (item) => {      setSelected(item.id);    };    return (      <View>        <View style={{ flexDirection: 'row', paddingTop: 30, paddingLeft: 10 }}>          <Image            source={require('./assets/greyArrow.png')}            style={styles.backArrow}          />          <Text style={styles.back}>Back</Text>        </View>        <Text style={styles.expenses}>Expenses</Text>        <Text style={styles.amountSpent}>Amount Spent</Text>        <TextInput          keyboardType='number-pad'          style={styles.inputLine}          onChangeText={text => setValue(text)}          value={value}   //value of input        />        <Text style={styles.typeOfExpense}>Type of Expense</Text>        {selected ? (          <View style={styles.tag}>            <Image source={DATA.find(item => item.id === selected).image} style={styles.tagIcon} resizeMode='contain' />            <Text style={styles.tagText}>{DATA.find(item => item.id === selected).title}</Text>          </View>        ) : (          null        )}        <FlatList          data={DATA}          renderItem={({ item }) => (            <Item              id={item.id}              title={item.title}              image={item.image}              selectedImage={item.selectedImage}              onPress={() => handlePress(item)}              style={selected === item.id ? styles.itemPress : styles.item}            />          )}          keyExtractor={item => item.id}          style={styles.flatList}        />        <TouchableOpacity>          <Text style={styles.button}>Done</Text>        </TouchableOpacity>      </View>    );  };  export default App;  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: '#fff',      alignItems: 'center',      justifyContent: 'center',    },    backArrow: {      width: 10,      height: 20,      marginTop: 23,      paddingRight: 30    },    back: {      marginTop: 20,      fontSize: 15,      color: '#989eb1',      fontFamily: 'Roboto'    },    expenses: {      fontSize: 25,      color: '#35424a',      fontFamily: 'Roboto',      fontWeight: 'bold',      paddingTop: 10,      paddingLeft: 45    },    amountSpent: {      fontSize: 17,      color: '#f85f6a',      fontFamily: 'Roboto',      paddingTop: 20,      paddingLeft: 45    },    inputLine: {      width: 200,      borderBottomColor: '#ebebeb',      borderBottomWidth: 2,      marginLeft: 45    },    typeOfExpense: {      fontSize: 17,      color: '#f85f6a',      fontFamily: 'Roboto',      paddingTop: 35,      paddingLeft: 45,      paddingBottom: 10    },    item: {      color: '#606060',      paddingBottom: 20,    },    itemPress: {      color: '#f85f6a',      paddingBottom: 20,    },    icons: {      width: 20,      height: 20,      marginRight: 15    },    tag: {      flexDirection: 'row',      marginLeft: 45,      marginRight: 200,      borderRadius: 5,      borderWidth: 1,      borderColor: '#d6d6d6'    },    tagIcon: {      width: 20,      height: 20,      margin: 10    },    tagText: {      paddingTop: 10,      color: '#606060'    },    flatList: {      paddingLeft: 50,      paddingTop: 20    },    button: {      backgroundColor: '#f85f6a',      color: 'white',      textAlign: 'center',      fontWeight: 'bold',      fontSize: 20,      marginTop: 20,      marginLeft: 40,      marginRight: 40,      paddingTop: 10,      paddingBottom: 10,      borderRadius: 5    }  }); |
| User Interface 6 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here    When transport (or any other type of expense) is clicked |

|  |
| --- |
| **User Interface 7 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  import { useState } from 'react';  import { StyleSheet, Text, View, Image, TextInput, FlatList, TouchableOpacity } from 'react-native';  const App = () => {    const [value, setValue] = useState('');    const [selected, setSelected] = useState();    const DATA = [      { id: '1', title: 'Allowance', image: require('./assets/allowance.png'), selectedImage: require('./assets/allowanceRed.png') },      { id: '2', title: 'Commissions', image: require('./assets/commissions.png'), selectedImage: require('./assets/commissionsRed.png') },      { id: '3', title: 'Salary', image: require('./assets/salary.png'), selectedImage: require('./assets/salaryRed.png') },      { id: '4', title: 'Tips', image: require('./assets/tips.png'), selectedImage: require('./assets/tipsRed.png') },      { id: '5', title: 'Others', image: require('./assets/others.png'), selectedImage: require('./assets/othersRed.png') },    ];    function Item({ id, title, image, selectedImage, onPress, style }) {      return (        <TouchableOpacity onPress={onPress} style={{ style }}>          <View style={{ flexDirection: 'row' }}>            <Image source={selected === id ? selectedImage : image} style={styles.icons} resizeMode='contain' />            <Text style={style}>{title}</Text>          </View>        </TouchableOpacity>      );    }    const handlePress = (item) => {      setSelected(item.id);    };    return (      <View>        <View style={{ flexDirection: 'row', paddingTop: 30, paddingLeft: 10 }}>          <Image            source={require('./assets/greyArrow.png')}            style={styles.backArrow}          />          <Text style={styles.back}>Back</Text>        </View>        <Text style={styles.income}>Income</Text>        <Text style={styles.amountEarned}>Amount Earned</Text>        <TextInput          keyboardType='number-pad'          style={styles.inputLine}          onChangeText={text => setValue(text)}          value={value}   //value of input        />        <Text style={styles.typeOfIncome}>Type of Income</Text>        {selected ? (          <View style={styles.tag}>            <Image source={DATA.find(item => item.id === selected).image} style={styles.tagIcon} resizeMode='contain' />            <Text style={styles.tagText}>{DATA.find(item => item.id === selected).title}</Text>          </View>        ) : (          null        )}        <FlatList          data={DATA}          renderItem={({ item }) => (            <Item              id={item.id}              title={item.title}              image={item.image}              selectedImage={item.selectedImage}              onPress={() => handlePress(item)}              style={selected === item.id ? styles.itemPress : styles.item}            />          )}          keyExtractor={item => item.id}          style={styles.flatList}        />        <TouchableOpacity>          <Text style={styles.button}>Done</Text>        </TouchableOpacity>      </View>    );  };  export default App;  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: '#fff',      alignItems: 'center',      justifyContent: 'center',    },    backArrow: {      width: 10,      height: 20,      marginTop: 23,      paddingRight: 30    },    back: {      fontSize: 15,      color: '#989eb1',      fontFamily: 'Roboto',      marginTop: 20,    },    income: {      fontSize: 25,      color: '#35424a',      fontFamily: 'Roboto',      fontWeight: 'bold',      paddingTop: 10,      paddingLeft: 45    },    amountEarned: {      fontSize: 17,      color: '#f85f6a',      fontFamily: 'Roboto',      paddingTop: 20,      paddingLeft: 45    },    inputLine: {      width: 200,      borderBottomColor: '#ebebeb',      borderBottomWidth: 2,      marginLeft: 45    },    typeOfIncome: {      fontSize: 17,      color: '#f85f6a',      fontFamily: 'Roboto',      paddingTop: 35,      paddingLeft: 45,      paddingBottom: 10    },    item: {      color: '#606060',      paddingBottom: 20,    },    itemPress: {      color: '#f85f6a',      paddingBottom: 20,    },    icons: {      width: 20,      height: 20,      marginRight: 15    },    tag: {      flexDirection: 'row',      marginLeft: 45,      marginRight: 200,      borderRadius: 5,      borderWidth: 1,      borderColor: '#d6d6d6'    },    tagIcon: {      width: 20,      height: 20,      margin: 10    },    tagText: {      paddingTop: 10,      color: '#606060'    },    flatList: {      paddingLeft: 50,      paddingTop: 20,      marginBottom:100    },    button: {      backgroundColor: '#f85f6a',      color: 'white',      textAlign: 'center',      fontWeight: 'bold',      fontSize: 20,      marginLeft: 40,      marginRight: 40,      paddingTop: 10,      paddingBottom: 10,      borderRadius: 5    }  }); |
| User Interface 7 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here    When commissions (or any type of income is clicked) |
| **User Interface 8 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save) |
| import { StyleSheet, Text, View, Image, TouchableOpacity, Dimensions, FlatList } from 'react-native';  import { PieChart } from 'react-native-chart-kit';  import BottomDrawer from 'react-native-bottom-drawer-view';  /\*Reasons for changes in pie chart  1. Percentage in the legend --> no need to look at two places at once and users can see both the category and percentage at the same place  2. Legend at the right side of the pie chart --> same level of vision and users can have a side by side comparison between the pie chart and legend  \*/  const App = () => {    const data = [      {        name: "Food",        expenses: 5,        color: "#ff9f9f",        legendFontColor: "black",        legendFontSize: 15      },      {        name: "Entertainment",        expenses: 2,        color: "#f8c9c3",        legendFontColor: "black",        legendFontSize: 15      },      {        name: "Transport",        expenses: 3,        color: "#e5b5af",        legendFontColor: "black",        legendFontSize: 15      },      {        name: "Others",        expenses: 1,        color: "#ffe7e6",        legendFontColor: "black",        legendFontSize: 15      },    ];    const expenses = [      { id: '1', title: 'Food', cost: '$300' },      { id: '2', title: 'Transport', cost: '$50' },      { id: '3', title: 'Entertainment', cost: '$30' },    ]    function Item({ id, title, cost }) {      if (parseInt(id) == expenses.length) {        return (          <View style={styles.expenses2}>            <Text style={styles.sectionTitles}>{title}</Text>            <Text style={styles.money}>{cost}</Text>          </View>        );      } else if (id=='1') {        return (          <>            <View style={styles.expenses}>              <Text style={styles.sectionTitles}>{title}</Text>              <Text style={styles.money}>{cost}</Text>            </View>            <Image source={require('./assets/line.png')} style={styles.line} />          </>        );      } else {        return (          <>            <View style={styles.expenses2}>              <Text style={styles.sectionTitles}>{title}</Text>              <Text style={styles.money}>{cost}</Text>            </View>            <Image source={require('./assets/line.png')} style={styles.line} />          </>        );      }    }    const renderItem = ({ item }) => (      <Item        id={item.id}        title={item.title}        cost={item.cost}      />    );    const screenWidth = Dimensions.get("window").width;    const chartConfig = {      backgroundGradientFrom: "white",      backgroundGradientTo: "white",      color: (opacity = 1) => `rgba(0,0,0, ${opacity})`,      barPercentage: 0.5,    };    return (      <View>        <View>          <TouchableOpacity>            <View>              <Image source={require('./assets/hamburger.png')} style={styles.hamburger} resizeMode='contain' />            </View>          </TouchableOpacity>          <Text style={styles.statistics}>Statistics</Text>        </View>        <View style={styles.subHeader}>          <Text style={styles.pieGraphText}>Pie Chart</Text>          <View style={styles.roundedArrows}>            <TouchableOpacity>              <Image source={require('./assets/arrowLeft.png')} style={styles.leftRoundedArrow} resizeMode='contain' />            </TouchableOpacity>            <TouchableOpacity>              <Image source={require('./assets/arrowRight.png')} style={styles.rightRoundedArrow} resizeMode='contain' />            </TouchableOpacity>          </View>        </View>        <PieChart          data={data}          width={screenWidth}          height={220}          chartConfig={chartConfig}          accessor="expenses"          backgroundColor="transparent"        />        <BottomDrawer          containerHeight={253}          roundedEdges={true}        >          <View style={styles.drawerHeader}>            <TouchableOpacity>              <Image source={require('./assets/leftArrow.png')} style={styles.arrows} resizeMode='contain' />            </TouchableOpacity>            <Text style={styles.drawerTitle}>March</Text>            <TouchableOpacity>              <Image source={require('./assets/rightArrow.png')} style={styles.arrows} resizeMode='contain' />            </TouchableOpacity>          </View>          <FlatList            data={expenses}            renderItem={renderItem}            keyExtractor={item => item.id}          />          <View style={styles.goal}>            <Image source={require('./assets/tick.png')} style={styles.tick} resizeMode='contain' />            <Text style={styles.monthlyGoal}>Monthly goal reached</Text>          </View>        </BottomDrawer>      </View>    );  };  export default App;  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: '#fff',      alignItems: 'center',      justifyContent: 'center',      fontFamily: 'roboto',    },    hamburger: {      height: 30,      width: 30,      marginTop: 50,      marginLeft: 40    },    statistics: {      fontSize: 25,      marginLeft: 40,      fontWeight: "bold",    },    subHeader: {      flexDirection: 'row',      justifyContent: 'space-between',      marginTop: 15,      marginBottom: 20    },    pieGraphText: {      color: '#35424a',      fontSize: 17,      paddingLeft: 50    },    roundedArrows: {      flexDirection: 'row',      marginRight: 20    },    leftRoundedArrow: {      width: 25,      height: 25,      marginRight: 5    },    rightRoundedArrow: {      width: 25,      height: 25,      marginLeft: 5    },    drawerHeader: {      flexDirection: 'row',      justifyContent: 'space-around'    },    arrows: {      width: 25,      height: 25,      marginTop: 25    },    drawerTitle: {      fontSize: 25,      marginTop: 20,      color: '#35424a'    },    expenses: {      flexDirection: 'row',      justifyContent: 'space-evenly',      marginTop: 25    },    sectionTitles: {      fontSize: 15,      color: '#acacac',      fontWeight: 'bold'    },    money: {      fontSize: 15,      color: '#989eb1'    },    line: {      height: 3,      width: 320,      marginLeft: 50,      marginTop: 10    },    expenses2: {      flexDirection: 'row',      justifyContent: 'space-evenly',      marginTop: 10    },    tick: {      width: 25,      height: 25,      marginRight: 10    },    monthlyGoal: {      fontSize: 20,      color: '#92d36e'    },    goal: {      flexDirection: 'row',      justifyContent: 'center',      marginBottom:15    }  }); |
| User Interface 8 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here |
| **User Interface 9 - JavaScript Code** |
| Copy and paste the JavaScript code that you have written  State the command if you use external library  (example-> npm install react-datepicker –save)  import { StyleSheet, Text, View, Image, TouchableOpacity, Dimensions, FlatList } from 'react-native';  import { LineChart } from 'react-native-chart-kit';  import BottomDrawer from 'react-native-bottom-drawer-view';  const App = () => {    const data = {      labels: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12'],      datasets: [        {          data: [3, 1, 5, 4, 4, 3, 2, 3, 2, 4, 4, 5],          color: (opacity = 1) => `rgba(248,95,106, ${opacity})`        }      ],    };    const DATA = [      { id: '1' },      { id: '2' },      { id: '3' },      { id: '4' },      { id: '5' },      { id: '6' },      { id: '7' },      { id: '8' },      { id: '9' },      { id: '10' },      { id: '11' },      { id: '12' }    ]    function Item({ id }) {      return (        <Text style={id === '12' ? styles.dec : id === '11' ? styles.nov : styles.others}>{id}</Text>      );    }    const renderItem = ({ item }) => (      <Item        id={item.id}      />    );    const info = [      { id: '1', title: 'Budget', cost: '$2000' },      { id: '2', title: 'Costs', cost: '$1500' },      { id: '3', title: 'Savings', cost: '$500' },    ]    function Item2({ id, title, cost }) {      if (parseInt(id) == info.length) {        return (          <View style={styles.info2}>            <Text style={styles.sectionTitles}>{title}</Text>            <Text style={styles.money}>{cost}</Text>          </View>        );      } else if (id == '1') {        return (          <>            <View style={styles.info}>              <Text style={styles.sectionTitles}>{title}</Text>              <Text style={styles.money}>{cost}</Text>            </View>            <Image source={require('./assets/line.png')} style={styles.line} />          </>        );      } else {        return (          <>            <View style={styles.info2}>              <Text style={styles.sectionTitles}>{title}</Text>              <Text style={styles.money}>{cost}</Text>            </View>            <Image source={require('./assets/line.png')} style={styles.line} />          </>        );      }    }    const renderItem2 = ({ item }) => (      <Item2        id={item.id}        title={item.title}        cost={item.cost}      />    )    const screenWidth = Dimensions.get("window").width;    const chartConfig = {      backgroundGradientFrom: "white",      backgroundGradientTo: "white",      color: (opacity = 1) => `rgba(0,0,0, ${opacity})`,      barPercentage: 0.5,    };    return (      <View>        <View>          <TouchableOpacity>            <View>              <Image source={require('./assets/hamburger.png')} style={styles.hamburger} resizeMode='contain' />            </View>          </TouchableOpacity>          <Text style={styles.statistics}>Statistics</Text>        </View>        <View style={styles.subHeader}>          <Text style={styles.lineGraphText}>Line Graph</Text>          <View style={styles.roundedArrows}>            <TouchableOpacity>              <Image source={require('./assets/arrowLeft.png')} style={styles.leftRoundedArrow} resizeMode='contain' />            </TouchableOpacity>            <TouchableOpacity>              <Image source={require('./assets/arrowRight.png')} style={styles.rightRoundedArrow} resizeMode='contain' />            </TouchableOpacity>          </View>        </View>        <LineChart          data={data}          width={screenWidth}          height={220}          chartConfig={chartConfig}          withShadow={false}          withInnerLines={false}          withOuterLines={false}          withDots={true}          withHorizontalLabels={false}          withVerticalLabels={false}          hidePointsAtIndex={[0, 1, 3, 4, 5, 6, 7, 8, 9, 10, 11]}        />        <FlatList          data={DATA}          renderItem={renderItem}          keyExtractor={item => item.id}          style={styles.flatList}          horizontal={true}        />        <BottomDrawer          containerHeight={253}          shadow={true}          roundedEdges={true}        >          <View style={styles.drawerHeader}>            <TouchableOpacity>              <Image source={require('./assets/leftArrow.png')} style={styles.arrows} resizeMode='contain' />            </TouchableOpacity>            <Text style={styles.drawerTitle}>March</Text>            <TouchableOpacity>              <Image source={require('./assets/rightArrow.png')} style={styles.arrows} resizeMode='contain' />            </TouchableOpacity>          </View>          <FlatList            data={info}            renderItem={renderItem2}            keyExtractor={item => item.id}          />          <View style={styles.goal}>            <Image source={require('./assets/tick.png')} style={styles.tick} resizeMode='contain' />            <Text style={styles.monthlyGoal}>Monthly goal reached</Text>          </View>        </BottomDrawer>      </View>    );  };  export default App;  const styles = StyleSheet.create({    container: {      flex: 1,      backgroundColor: '#fff',      alignItems: 'center',      justifyContent: 'center',    },    hamburger: {      height: 30,      width: 30,      marginTop: 50,      marginLeft: 40    },    statistics: {      fontSize: 25,      marginLeft: 40,      fontWeight: "bold",    },    dec: {      fontSize: 10,      paddingLeft: 20    },    nov: {      fontSize: 10,      paddingLeft: 20    },    others: {      fontSize: 10,      paddingLeft: 20    },    subHeader: {      flexDirection: 'row',      justifyContent: 'space-between',      marginTop: 15,      marginBottom: 20    },    lineGraphText: {      color: '#35424a',      fontSize: 17,      paddingLeft: 50    },    roundedArrows: {      flexDirection: 'row',      marginRight: 20    },    leftRoundedArrow: {      width: 25,      height: 25,      marginRight: 5    },    rightRoundedArrow: {      width: 25,      height: 25,      marginLeft: 5    },    flatList: {      marginLeft: 40    },    drawerHeader: {      flexDirection: 'row',      justifyContent: 'space-around'    },    arrows: {      width: 25,      height: 25,      marginTop: 25    },    drawerTitle: {      fontSize: 25,      marginTop: 20,      color: '#35424a'    },    info: {      flexDirection: 'row',      justifyContent: 'space-evenly',      marginTop: 25    },    sectionTitles: {      fontSize: 15,      color: '#acacac',      fontWeight: 'bold'    },    money: {      fontSize: 15,      color: '#989eb1'    },    line: {      height: 3,      width: 320,      marginLeft: 50,      marginTop: 10    },    info2: {      flexDirection: 'row',      justifyContent: 'space-evenly',      marginTop: 10    },    tick: {      width: 25,      height: 25,      marginRight: 10    },    monthlyGoal: {      fontSize: 20,      color: '#92d36e'    },    goal: {      flexDirection: 'row',      justifyContent: 'center',      marginBottom: 15    }  }); |
| User Interface 9 – Screen Capture of the Interface |
| Paste a screenshot of the output of your User Interface here |

**2. Individual Contribution**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Name | Assignment 2 Contribution | % Contribute to the Assignment 2 |
|  | Example |  |  |
| 1 | Dawn Oh Le Qian | Code the Sign in page, Sign up page, Home page, Navigation page, Transactions page | 50% |
| 2 | Jeanette Ong Jing Xuan | Code the Expenses page, Income page, Calendar page, Statistics page | 50% |
|  |  | **Total** | 100 % |