Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний

інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

до лабораторної роботи № 3 з дисципліни

«Розробка мобільних застосувань під Android»

Виконав ІП-24 Цюх В.М.

Перевірив Орленко С.П.

Київ 2025

Лабораторна робота №3

Тема: ДОСЛІДЖЕННЯ СПОСОБІВ ЗБЕРЕЖЕННЯ ДАНИХ

ЗАВДАННЯ

Написати програму під платформу Андроїд, яка доповнює програму, що розроблена за лабораторною роботою 2, роботою зі сховищами.

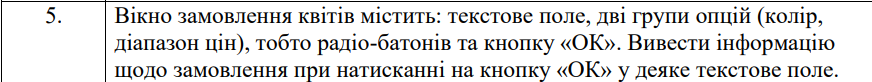
Тобто при натисканні на кнопку «ОК» додатково:

- здійснюється запис результату взаємодії з інтерфейсом до сховища (файл або базу даних);

- користувач інформується відповідним повідомленням щодо успішності запису.

Також інтерфейс необхідно доповнити кнопкою «Відкрити», натискання на яку призводить до переходу на іншу Діяльність, у якій відображається вміст даних, що зберігаються у сховищі. Якщо дані відсутні (сховище пусте) відобразити відповідне повідомлення. За бажанням можна додатково реалізувати оновлення та видалення даних зі сховища.

Варіант - 5



Лістинг програмного коду

Файл App.tsx

import { **NavigationContainer** } from "@react-navigation/native";

import { **createStackNavigator** } from "@react-navigation/stack";

import { **View**, **StyleSheet**, **Button** } from "react-native";

import **OrderFormFragment** from "./components/OrderFormFragment";

import **OrderResultFragment** from "./components/OrderResultFragment";

import **OrderViewScreen** from "./components/OrderViewScreen";

import { **useState**, **useRef** } from "react";

const Stack = createStackNavigator**()**;

*// Головний екран, на якому користувач заповнює форму*

function HomeScreen({ navigation }) {

  const [orderInfo, setOrderInfo] = useState**(**""**)**;

  const clearForm = useRef<() => *void*>**(**() => {}**)**;

  return **(**

    <View *style*={styles.**container**}>

      <OrderFormFragment

*setOrderInfo*={setOrderInfo}

*setClearForm*={(clearFunction) => **(**clearForm.**current** = clearFunction**)**}

      />

      {orderInfo ? **(**

        <OrderResultFragment

*orderInfo*={orderInfo}

*setOrderInfo*={setOrderInfo}

*clearForm*={() => clearForm.current?.**()**}

        />

**)** : null}

      <Button

*title*="Відкрити"

*onPress*={() => navigation.navigate**(**"OrderView"**)**}

      />

    </View>

**)**;

}

export default function App() {

  return **(**

    <NavigationContainer>

      <Stack.Navigator>

        <Stack.Screen

*name*="Home"

*component*={HomeScreen}

*options*={{ **title:** "Замовлення" }}

        />

        <Stack.Screen

*name*="OrderView"

*component*={OrderViewScreen}

*options*={{ **title:** "Перегляд замовлення" }}

        />

      </Stack.Navigator>

    </NavigationContainer>

**)**;

}

const styles = StyleSheet.create**(**{

**container:** {

**flex:** 1,

**justifyContent:** "center",

**padding:** 20,

  },

}**)**;

Файл OrderFormFragment.tsx

import { **useState**, **useEffect** } from "react";

import { **View**, **Text**, **TextInput**, **Button**, **StyleSheet**, **Alert** } from "react-native";

import \* as **FileSystem** from "expo-file-system";

import **RadioGroup** from "./RadioGroup";

interface *OrderFormFragmentProps* {

  setOrderInfo: (info: *string*) => *void*;

  setClearForm: (clearFunction: () => *void*) => *void*;

}

*// Компонент для форми замовлення*

export default function OrderFormFragment({

  setOrderInfo,

  setClearForm,

}: *OrderFormFragmentProps*) {

  const [name, setName] = useState**(**""**)**;

  const [color, setColor] = useState**(**""**)**;

  const [priceRange, setPriceRange] = useState**(**""**)**;

  const filePath = FileSystem.**documentDirectory** + "order.txt";

  const saveToFile = async (content: *string*) => {

    try {

      await FileSystem.writeAsStringAsync**(**filePath, content**)**;

      Alert.alert**(**"Успіх", "Замовлення збережено у файл!"**)**;

    } catch **(**error**)** {

      Alert.alert**(**"Помилка", "Не вдалося зберегти дані."**)**;

    }

  };

  const handleSubmit = async () => {

    if **(**!name || !color || !priceRange**)** {

      Alert.alert**(**"Помилка", "Будь ласка, заповніть всі поля!"**)**;

      return;

    }

    const orderData = `Замовлення для: ${name}\nКолір: ${color}\nЦіна: ${priceRange}`;

    setOrderInfo**(**orderData**)**;

    await saveToFile**(**orderData**)**;

  };

  const clearForm = () => {

    setName**(**""**)**;

    setColor**(**""**)**;

    setPriceRange**(**""**)**;

  };

  useEffect**(**() => {

    setClearForm**(**clearForm**)**;

  }, **[])**;

  return **(**

    <View *style*={styles.**container**}>

      <Text *style*={styles.**label**}>Ім'я замовника:</Text>

      <TextInput

*style*={styles.**input**}

*placeholder*="Введіть ваше ім'я"

*value*={name}

*onChangeText*={setName}

      />

      <Text *style*={styles.**label**}>Виберіть колір:</Text>

      <RadioGroup

*options*={**[**"Червоний", "Білий", "Жовтий"**]**}

*selected*={color}

*onSelect*={setColor}

      />

      <Text *style*={styles.**label**}>Виберіть діапазон цін:</Text>

      <RadioGroup

*options*={**[**"100-200 грн", "200-300 грн", "300-400 грн"**]**}

*selected*={priceRange}

*onSelect*={setPriceRange}

      />

      <Button *title*="ОК" *onPress*={handleSubmit} />

    </View>

**)**;

}

const styles = StyleSheet.create**(**{

**container:** { **width:** "100%", **padding:** 10 },

**label:** { **fontSize:** 16, **fontWeight:** "bold", **marginVertical:** 5 },

**input:** {

**borderWidth:** 1,

**borderColor:** "#ccc",

**borderRadius:** 5,

**padding:** 10,

**marginBottom:** 10,

  },

}**)**;

Файл OrderResultFragment.tsx

import { **View**, **Text**, **Button**, **StyleSheet** } from "react-native";

interface *OrderResultFragmentProps* {

**orderInfo**: *string*;

  setOrderInfo: (info: *string*) => *void*;

  clearForm: () => *void*;

}

*// Компонент для відображення результатів замовлення*

export default function OrderResultFragment({

  orderInfo,

  setOrderInfo,

  clearForm,

}: *OrderResultFragmentProps*) {

  return **(**

    <View *style*={styles.**container**}>

      <Text *style*={styles.**result**}>{orderInfo}</Text>

      <Button

*title*="Cancel"

*onPress*={() => {

          setOrderInfo**(**""**)**;

          clearForm**()**;

        }}

*color*="red"

      />

    </View>

**)**;

}

const styles = StyleSheet.create**(**{

**container:** { **marginTop:** 20 },

**result:** {

**fontSize:** 16,

**fontWeight:** "bold",

**color:** "green",

**marginBottom:** 10,

  },

}**)**;

Файл RadioGroup.tsx

*// Компонент групи радіокнопок для вибору кольору або цінового діапазону*

import { **TouchableOpacity**, **View**, **Text**, **StyleSheet** } from "react-native";

interface *RadioGroupProps* {

**options**: *string***[]**;

**selected**: *string*;

  onSelect: (value: *string*) => *void*;

}

export default function RadioGroup({

  options,

  selected,

  onSelect,

}: *RadioGroupProps*) {

  return **(**

    <View *style*={styles.**container**}>

      {options.map**(**(option) => **(**

        <TouchableOpacity

*key*={option}

*style*={styles.**radioContainer**}

*onPress*={() => onSelect**(**selected === option ? "" : option**)**}

        >

          {*/\* Візуальне відображення вибраної кнопки \*/*}

          <View

*style*={**[**styles.**radio**, selected === option && styles.**radioSelected]**}

          />

          <Text *style*={styles.**text**}>{option}</Text>

        </TouchableOpacity>

**))**}

    </View>

**)**;

}

const styles = StyleSheet.create**(**{

**container:** {

**marginBottom:** 10,

  },

**radioContainer:** {

**flexDirection:** "row",

**alignItems:** "center",

**marginBottom:** 5,

  },

**radio:** {

**width:** 20,

**height:** 20,

**borderRadius:** 10,

**borderWidth:** 2,

**borderColor:** "#555",

**marginRight:** 10,

  },

**radioSelected:** {

**backgroundColor:** "#555",

  },

**text:** {

**fontSize:** 16,

  },

}**)**;

Скріни виконання програми у віртуальному телефоні

Зображення, що містить текст, знімок екрана, мультимедіа, Пристрій зв’язку

Вміст, створений ШІ, може бути неправильним.Зображення, що містить текст, знімок екрана, мультимедіа, Пристрій зв’язку

Вміст, створений ШІ, може бути неправильним.Зображення, що містить текст, знімок екрана, Пристрій зв’язку, мультимедіа

Вміст, створений ШІ, може бути неправильним.