Ministerul Educaţiei, Tineretului şi Sportului al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

Departamentul Informatica si Ingineria sistemelor

RAPORT

Lucrare de laborator nr.1

la Programarea aplicațiilor mobile

Tema: **UI View Model**

A efectuat: st. gr. FI-181

Damean Alxandra

A verificat: prof.

Rusu Cristian

Chişinău 2020

**Obiective :** Dezvoltarea unei aplicații pe una din platformele stabilite la Labora torul Nr.0 , utilizând mediul de dezvoltare corespunzător acesteia.

**Scopul:**

De prezentat o aplicație ce rulează pe un dispozitiv sau emulator, ce va conține pe interfața sa, următoarele elemente:

4 butoane (ce vor executa condițiile de mai jos)

1 TextBox (pentru input)

2 Radio butoane (pentru camera față, spate)



**Android Studio** este un mediu de dezvoltare pentru colaborarea cu platforma Android.

Prin Android Studio se poate vedea automat interfața aplicației pe diferite dimensiuni și rezoluții, chiar și în diferite limbi. O funcție ceva mai avansată este Memory Monitor, introdusă pentru a oferi informații legate de ocuparea resurselor – o treabă de care dezvoltatorii Android ar trebui să țină seama mai des.

**Android Studio** are o interfață grafică pentru instalare din care se stabilesc setările de bază, se adaugă un emulator și ceva exemple de aplicații. O modalitate foarte eficientă pentru un începător este să folosească unul din exemple și pe baza lui să înceapă să extindă funcționalitatea aplicației sale.

1. **De creat un push notification pe ecranul dispozitivului care se va trata peste 10s.**

**Code\*source:**

 useEffect(() => {

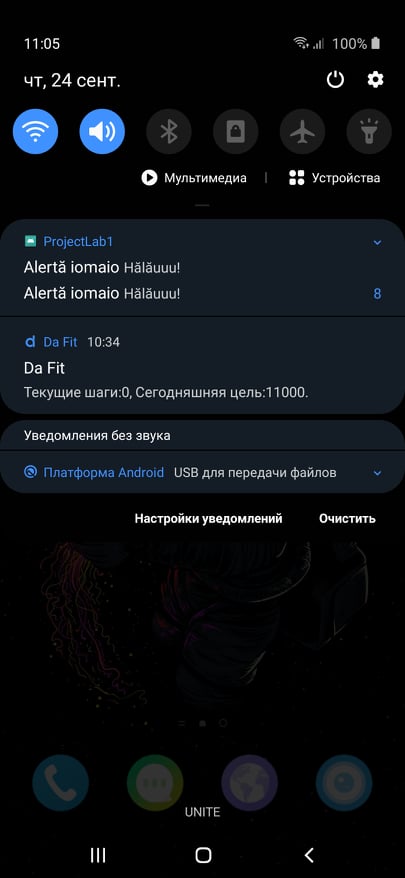
    setTimeout(() => {

      PushNotification.localNotification({ title: 'Alertă iomaio', message: 'Hălăuuu!' })

    }, 10000)

  }, [])

**View mode:**



**2.De utilizat browserul intern al dispozitivului, pentru a inițializa o căutare în Google, conform cuvântului cheie introdus în TextBox.**

**Code\*source:**

 const find = text => {

    const urlPath = `https://www.google.com/search?q=${encodeURIComponent(text)}`

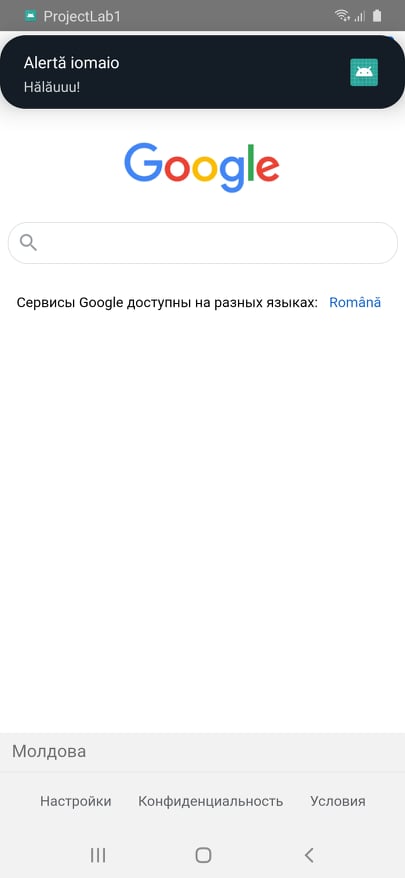
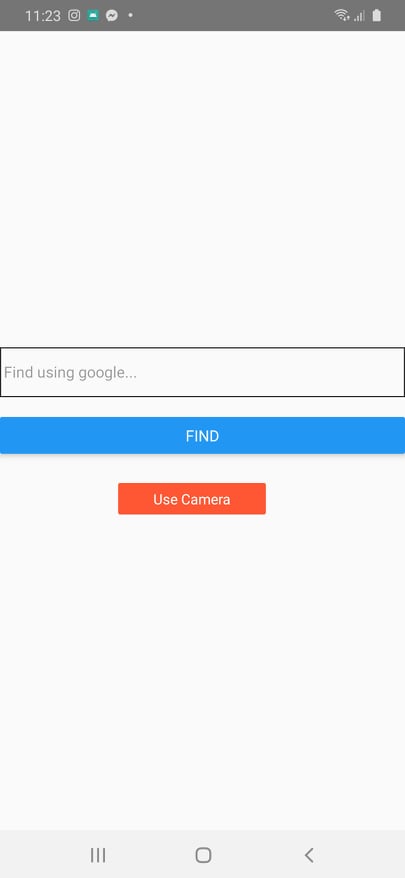
    setIsBrowser({

      show: true,

      val: urlPath

    })

  }



**3.De rulat evenimentul de lucru a camerei frontale și/sau spate la selectarea unuia din cele 2 radio butoane propuse.**

**Code\*source:**

**Camera action:**

import React, { Component, ReactElement, Fragment } from 'react'

import { RNCamera } from 'react-native-camera'

import { StyleSheet, View, Text, TouchableOpacity, Alert } from 'react-native'

import CheckBox from '@react-native-community/checkbox'

// import { savePhotoFile } from './savePhotoFileApi'

export  class CameraAction extends Component{

  constructor(props) {

    super(props)

    this.state = {

      isFrontCamera: true,

    }

    this.takePicture = this.takePicture.bind(this)

  }

  takePicture = async () => {

    if (this.camera) {

      const options = {

        quality: 0.999,

        base64: true,

      }

      try {

        const data = await this.camera.takePictureAsync(options)

        this.props.saveImage(data.base64)

        // console.log(data)

      } catch (error) {

        console.warn(error)

      }

    }

  }

  render() {

    return (

      <Fragment>

        <RNCamera

          ref={(ref) => {

            this.camera = ref

          }}

          style={styles.preview}

          type={

            this.state.isFrontCamera ? RNCamera.Constants.Type.front : RNCamera.Constants.Type.back

          }

          flashMode={RNCamera.Constants.FlashMode.off}

        />

        <View style={styles.container}>

          <View style={{ flexDirection: 'row', backgroundColor: 'white', alignItems: 'center' }}>

            <Text>Front:</Text>

            <CheckBox

              value={this.state.isFrontCamera}

              disabled={this.state.isFrontCamera}

              onValueChange={() => {

                this.setState({ isFrontCamera: true })

              }}

            />

          </View>

          <TouchableOpacity onPress={this.takePicture} style={styles.capture}>

            <Text>Screen </Text>

          </TouchableOpacity>

          <View style={{ flexDirection: 'row', backgroundColor: 'white', alignItems: 'center' }}>

            <Text>Back:</Text>

            <CheckBox

              value={!this.state.isFrontCamera}

              disabled={!this.state.isFrontCamera}

              onValueChange={() => {

                this.setState({ isFrontCamera: false })

              }}

            />

          </View>

        </View>

      </Fragment>

    )

  }

}

const styles = StyleSheet.create({

  container: {

    flex: 0,

    flexDirection: 'row',

    justifyContent: 'center',

  },

  capture: {

    flex: 0,

    backgroundColor: '#a7a7a7',

    borderRadius: 5,

    padding: 6,

    alignSelf: 'center',

  },

  preview: {

    flex: 1,

    justifyContent: 'flex-end',

    alignItems: 'center',

  },

})

**Butoanele pentru camera:**

import React, { useEffect } from 'react';

import {

  ScrollView,

  Text,

  TextInput,

  TouchableOpacity,

  View,

  StyleSheet

} from 'react-native';

import {WebView} from 'react-native-webview'

export class CameraButton extends React.Component {

    constructor(props){

        super(props)

    }

    render() {

        return <View>

            <TouchableOpacity

                onPress={() => {this.props.onPress()}}

                activeOpacity={0.7}

                style={styles.headerButton}

                >

                <View>

                    <Text style={styles.customBtnText}>Use Camera</Text>

                </View>

            </TouchableOpacity>

        </View>

    }

}

const styles = StyleSheet.create({

    headerButton: {

        flex: 0,

        backgroundColor: '#FF5733',

        paddingHorizontal: 9,

        paddingVertical: 6,

        marginTop: 30,

        marginHorizontal: 5,

        borderRadius: 2,

        maxWidth: 150,

        minWidth: 60,

        justifyContent: 'center',

        alignItems: 'center',

        marginLeft:120

      },

      customBtnText: {

        fontSize: 13,

        color: '#fff',

      },

})

**Camera:**

import React, { useEffect } from 'react';

import {

    ScrollView,

    Text,

    TextInput,

    TouchableOpacity,

    View,

    StyleSheet

} from 'react-native';

import { CameraAction } from './CameraAction'

import {CameraResult} from './CameraResult'

export class Camera extends React.Component {

    constructor(props) {

        super(props)

        this.state = {

            b64Img: '',

            imageIsShowed: false

        }

        this.saveImage = this.saveImage.bind(this)

    }

    saveImage(b64Img){

        this.setState({b64Img: b64Img, imageIsShowed: true})

    }

    render() {

        return this.state.imageIsShowed ? <CameraResult img={this.state.b64Img} newImageCallback={() => this.setState({imageIsShowed: false})} /> : <CameraAction saveImage={this.saveImage} />

    }

}

**Camera result:**

import React, { useEffect } from 'react';

import {

    Image,

    Button,

    View,

    StyleSheet

} from 'react-native';

export class CameraResult extends React.Component {

    constructor(props) {

        super(props)

    }

    render() {

        return     <View style={styles.container}>

        <Image

          style={styles.photoResult}

          source={{ uri: `data:image/gif;base64,${this.props.img}` }}

        />

        <Button title="New photo" onPress={() => this.props.newImageCallback()} />

      </View>

    }

}

const styles = StyleSheet.create({

    container: {

      flex: 1,

      justifyContent: 'center',

      alignItems: 'center',

    },

    photoResult: {

      flex: 1,

      width: '100%',

      height: '100%',

    },

  })

**Camera:**

import React, { useEffect } from 'react';

import {

    Image,

    Button,

    View,

    StyleSheet

} from 'react-native';

export class CameraResult extends React.Component {

    constructor(props) {

        super(props)

    }

    render() {

        return     <View style={styles.container}>

        <Image

          style={styles.photoResult}

          source={{ uri: `data:image/gif;base64,${this.props.img}` }}

        />

        <Button title="New photo" onPress={() => this.props.newImageCallback()} />

      </View>

    }

}

const styles = StyleSheet.create({

    container: {

      flex: 1,

      justifyContent: 'center',

      alignItems: 'center',

    },

    photoResult: {

      flex: 1,

      width: '100%',

      height: '100%',

    },

  })



**4. De tratat evenimentul de captare a pozei executate cu una din cele doua camere, și de afișat într-o altă Activitate**



**Concluzie:** În urma efectuării acestei lucrări de laborator am realizat sarcinile de a crea 4 funcții. Toate aceste funcții le-am scris aparte avînd fiecare structura aparte, prin Native React. Prima sarcina a fost elaborarea unei notificări unde am creat o notificare. Apoi am realizat căutarea în google. Am învățat cum creez o camera, și în urma acestui laborator am înțeles cum este mai ușor să structurăm funcțiile de executare. Apoi am creat coordonarea dintre una sau mai multe activități, cu transmitere de informație sau cu accesarea lor.