НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2

з дисципліни

“Програмування мобільних систем / Розроблення клієнтських додатків для мобільних платформ”

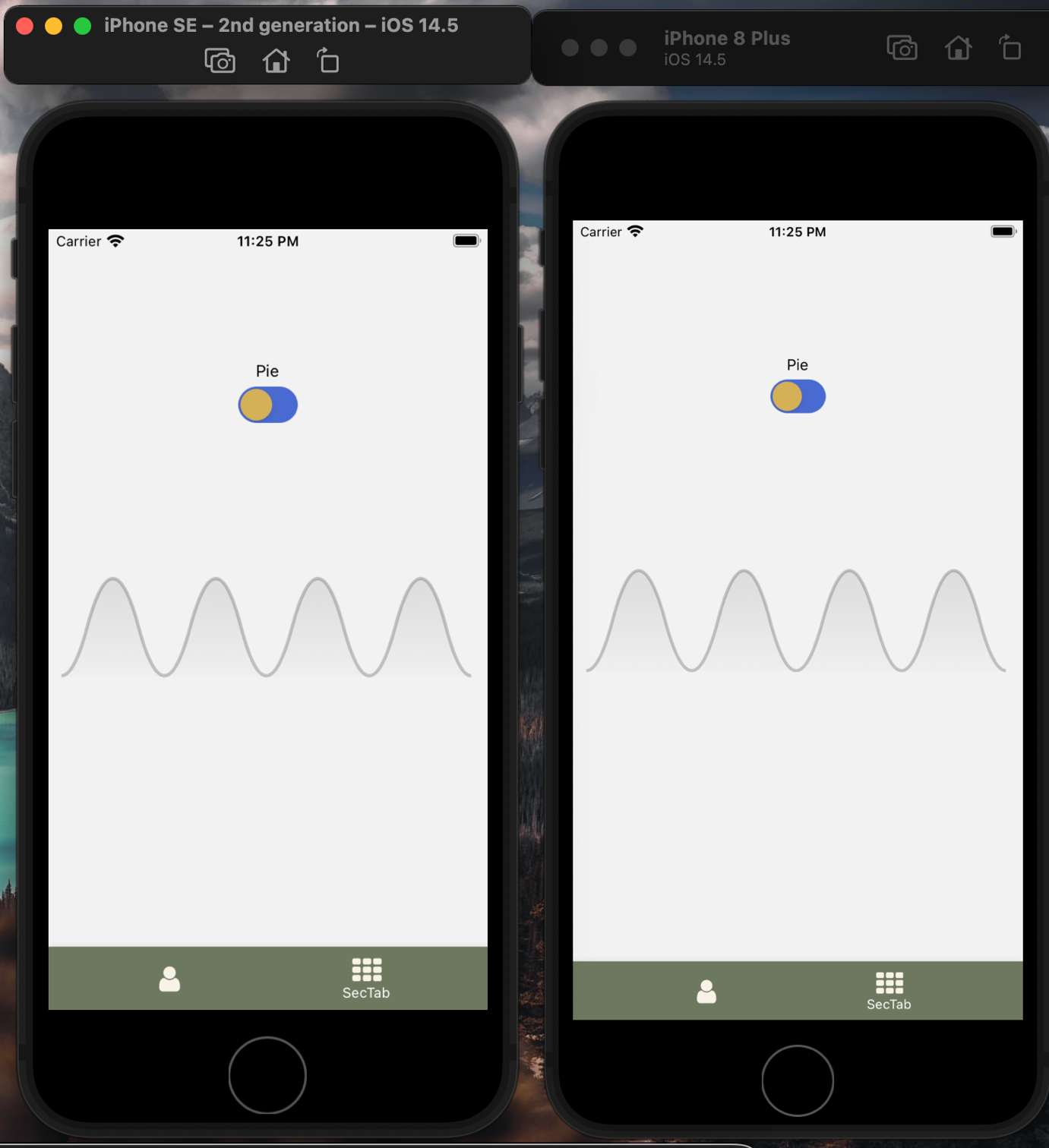
Виконав:

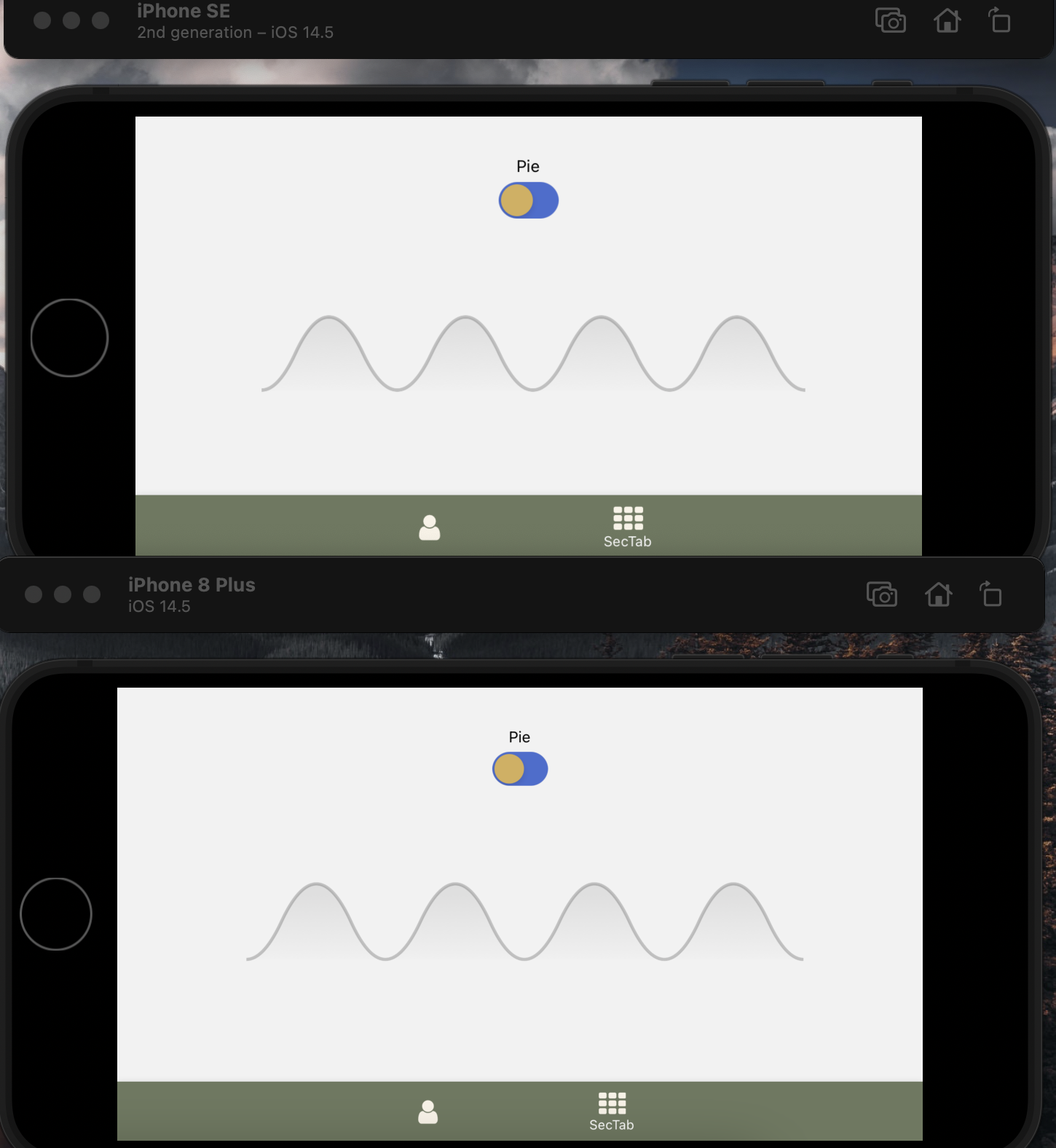
студент групи IO-83

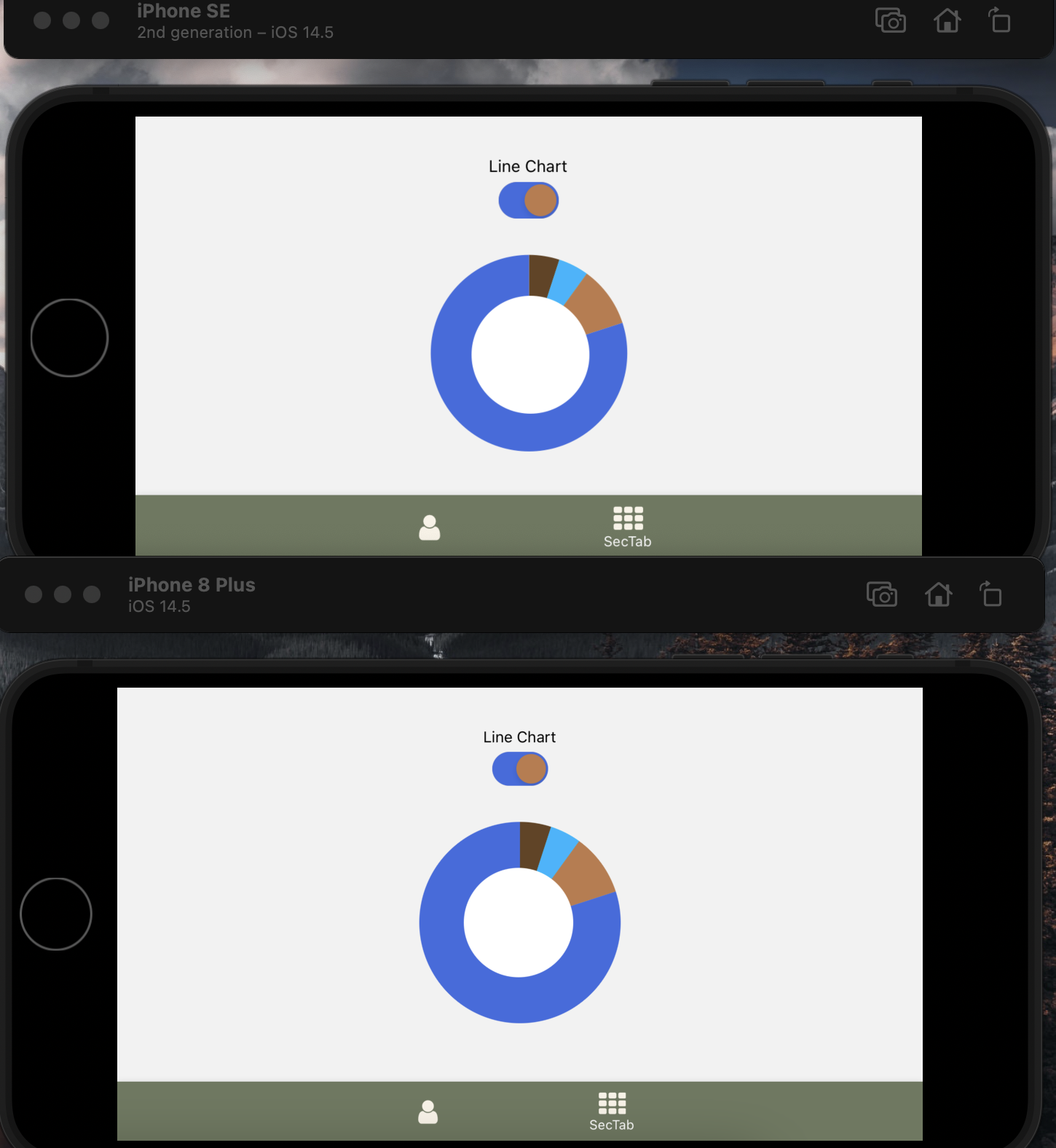
ЗК 8321

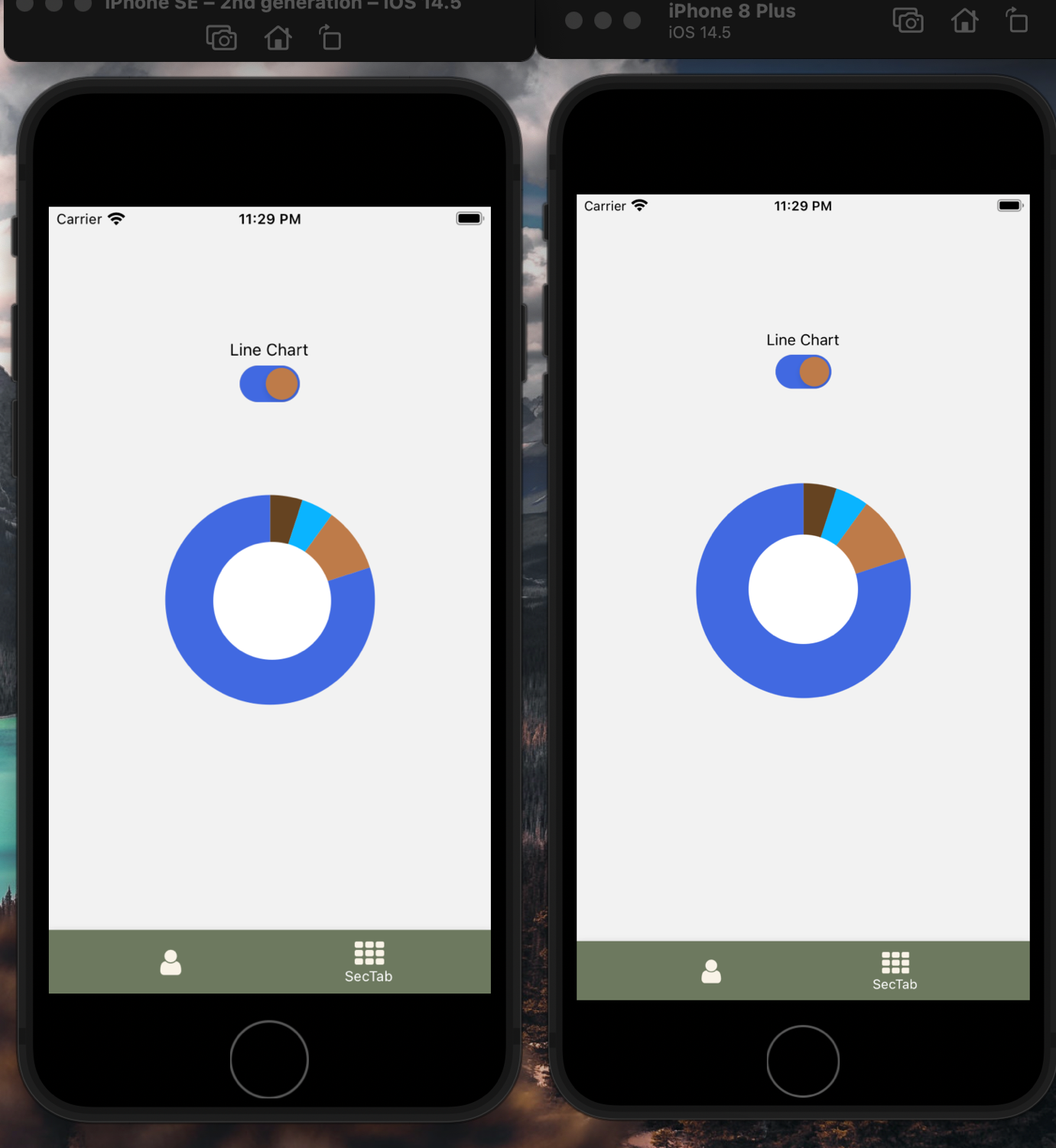
Морозов Андрій

Київ 2021

****

****

****



**Лістинг коду**

App.js

import React from 'react';

import RootNavigator from './screens/RootNavigator'

const App = () => {

return (

<RootNavigator />

);

}

export default App

MainTab.js

import React from "react";

import { View, Text } from 'react-native'

const MainTab = () => {

return (

<View style={{

flex: 1,

justifyContent: 'center',

flexDirection:'column',

alignItems:'center',

}}>

<Text style={{ fontSize: 20 }}>Морозов Андрій</Text>

<Text style={{ fontSize: 20 }}>Група ІО-83</Text>

<Text style={{ fontSize: 20 }}>ЗК ІО-8321</Text>

</View>

)

}

export default MainTab

SecTab.js

import React, { useState, useEffect } from 'react';

import {StyleSheet, Text, View, Switch, Dimensions} from 'react-native';

import {LineChart, PieChart} from "react-native-chart-kit";

import { data , labels, useScreenDimensions } from '../consts/consts'

import Svg, { Circle } from 'react-native-svg';

const Graphs = ({ navigation }) => {

const dim = Dimensions.get("screen")

const [isEnabled, setIsEnabled] = useState(false);

const toggleSwitch = () => setIsEnabled(previousState => !previousState);

const checkOrientation = () => {

const dim = Dimensions.get('screen');

if (dim.height >= dim.width) {

return portrait

} else {

return landscape

}

}

const screenData = useScreenDimensions();

if (isEnabled) {

return (

<View style={checkOrientation().container}>

<Text>Line Chart</Text>

<Switch

trackColor={{ false: "#BE7B4A", true: "#4169e1" }}

thumbColor={isEnabled ? "#BE7B4A" : "#4169e1"}

ios\_backgroundColor="#BE7B4A"

onValueChange={toggleSwitch}

value={isEnabled}

style={checkOrientation().toggle}

/>

<PieChart

data={[

{

percent: 5,

color: '#654321',

},

{

percent: 5,

color: '#0cb4ff',

},

{

percent: 10,

color: '#BE7B4A',

},

{

percent: 80,

color: '#4169e1',

},

]}

hasLegend={false}

width={

screenData.isLandscape ?

Dimensions.get('screen').width :

Dimensions.get('screen').width

}

height={

screenData.isLandscape ?

Dimensions.get("screen").height / 1.8:

Dimensions.get("screen").height / 3

}

chartConfig={{

color: (opacity = 1) => `rgba(0, 0, 0, ${opacity})`,

}}

style={{

alignItems: "center",

marginLeft: '50%',

}}

accessor="percent"

absolute

/>

<View style={

screenData.isLandscape ?

{ zIndex: 1, position: 'absolute', paddingLeft: '39.5%', top: '43.7%' } :

{ zIndex: 1, position: 'absolute', paddingLeft: '35.5%', top: '51.7%' }

}>

<Svg height="180" width="200">

<Circle cx="50" cy="50" r="50" fill="white" />

</Svg>

</View>

</View>

)

} else {

return (

<View style={checkOrientation().container}>

<Text>Pie</Text>

<Switch

trackColor={{ false: "#4169e1", true: "#D4B057" }}

thumbColor={isEnabled ? "#BE7B4A" : "#D4B057"}

ios\_backgroundColor="#4169e1"

onValueChange={toggleSwitch}

value={isEnabled}

style={checkOrientation().toggle}

/>

<LineChart

data={{ labels: labels, datasets: [{ data: data }] }}

width={ screenData.isLandscape ? dim.width : dim.width \* 1.3 }

height={ screenData.isLandscape ? dim.height / 4.5 : dim.height / 6 }

chartConfig={{

backgroundColor: "rgb(242, 242, 242)",

backgroundGradientFrom: "rgb(242, 242, 242)",

backgroundGradientTo: "rgb(242, 242, 242)",

color: (opacity = 1) => `rgba(0, 0, 0, ${opacity})`,

propsForDots: {

r: "0",

strokeWidth: "0",

stroke: "#000",

barPercentage: '1'

}

}}

style={

screenData.isLandscape ?

{

paddingRight: dim.width / 4.5,

marginLeft: dim.width / 10,

marginTop: dim.height / 6.5,

} :

{

paddingRight: dim.width / 4,

marginLeft: dim.width / 9,

marginTop: dim.height / 11,

}

}

withInnerLines={false}

withOuterLines={false}

withHorizontalLabels={false}

withVerticalLabels={false}

bezier

/>

</View>

)

}

}

export default Graphs

const portrait = StyleSheet.create({

container: {

flex: 0,

alignItems: "center",

justifyContent: "center",

marginTop: '30%',

},

toggle: {

flex: 0,

marginTop: 5,

justifyContent: 'center',

alignItems: 'center',

marginBottom: '15%'

},

});

const landscape = StyleSheet.create({

container: {

flex: 0,

alignItems: "center",

justifyContent: "center",

marginTop: '5%',

},

toggle: {

flex: 0,

marginTop: 5,

justifyContent: 'center',

alignItems: 'center',

marginBottom: 10,

},

});

RootNavigator.js

import React from 'react';

import { View } from 'react-native';

import Icon from 'react-native-vector-icons/FontAwesome'

import { createMaterialBottomTabNavigator } from '@react-navigation/material-bottom-tabs';

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

import { MainBtmBarTheme } from '../consts/consts'

import MainTab from "./MainTab";

import SecTab from "./SecTab";

const Tab = createMaterialBottomTabNavigator();

const RootNavigator = () => {

return (

<NavigationContainer theme={ MainBtmBarTheme }>

<Tab.Navigator

shifting={true}

sceneAnimationEnabled={true}

initialRouteName="Creator"

>

<Tab.Screen

name="MainTab"

component={MainTab}

options={{

tabBarLabel: 'MainTab',

tabBarIcon: () => (

<View>

<Icon

style={[{color: '#F9F3E7'}]}

size={25}

name={'user'}

/>

</View>

),

}}

/>

<Tab.Screen

name="SecTab"

component={SecTab}

options={{

tabBarLabel: 'SecTab',

tabBarIcon: () => (

<View>

<Icon

style={[{color: '#F9F3E7'}]}

size={25}

name={'th'}

/>

</View>

),

}}

/>

</Tab.Navigator>

</NavigationContainer>

);

}

export default RootNavigator

Висновок

При виконанні роботи я створив новий компонент який реагує на переключення свіча та показує два різних графіка.