НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

з дисципліни

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

Виконав:

студент групи ІВ-81

ЗК ІВ-8102

Бєлов Микита Сегрійович

Київ 2021

**Варіант № 1**

**Лістинг коду**

**Movie.js**

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

import { SafeAreaView, View, VirtualizedList, StyleSheet, Text, StatusBar, Image, TextInput, TouchableOpacity, Dimensions } from 'react-native';

import { MaterialCommunityIcons } from '@expo/vector-icons';

import Swipeable from 'react-native-gesture-handler/Swipeable';

import { styleConfig } from "../style";

import \* as data from '../MoviesList.json';

export *const* DATA = data.Search

import { choosePoster } from "../assets/global/choosePoster";

import { tt } from "../assets/global/tt";

*const* window = Dimensions.get("window");

*const* screen = Dimensions.get("screen");

*const* getItemCount = (*data*) *=>* *data*.length;

*const* getItem = (*data*, *index*) *=>* {

    return ({

        id: `${*data*[*index*].imdbID}`,

        title: `${*data*[*index*].Title}`,

        year: `${*data*[*index*].Year}`,

        type: `${*data*[*index*].Type}`,

        poster: `${*data*[*index*].Poster}`

    })

};

export default *function* Movies({ *navigation* }) {

*const* [dimensions, setDimensions] = useState({ window, screen });

*const* [selectedData, setSelectedData] = useState([]);

*const* [fullData, setFullData] = useState([]);

*const* [reloade, setReloade] = useState(false);

*const* [search, setSearch] = useState('');

    useEffect(() *=>* {

        setSelectedData(DATA);

        setFullData(DATA);

    }, []);

*function* Item({ *id*, *title*, *year*, *type*, *poster* }) {

        return (

            <*TouchableOpacity*

                activeOpacity={0.5}

                onPress={

                    () *=>* *navigation*.navigate('Details', {

                        id: *id*,

                        title: *title*,

                        year: *year*,

                    })}>

                <*Swipeable*

                    renderRightActions={LeftActions}

                    onSwipeableRightOpen={() *=>* {

*const* onDelete = DATA.findIndex(*elem* *=>* *elem*.imdbID === *id*)

                        DATA.splice(onDelete, 1);

                        searching(search)

                        setReloade(!reloade)

                    }}>

                    <*View* style={portrait.item}>

                        <*View* style={portrait.posterViev}>

                            <*Image*

                                style={orientation().poster}

                                source={choosePoster(*poster*)}

                            />

                        </*View*>

                        <*View* style={orientation().textViev}>

                            <*Text* style={portrait.title}>{*title*}</*Text*>

                            <*Text* style={portrait.details}>{*year*}</*Text*>

                            <*Text* style={portrait.details}>{*type*}</*Text*>

                        </*View*>

                    </*View*>

                </*Swipeable*>

            </*TouchableOpacity*>

        )

    }

*const* onChange = ({ *window*, *screen* }) *=>* {

        setDimensions({ window, screen });

    };

*const* searching = (*text*) *=>* {

        setSelectedData(DATA);

        setFullData(DATA);

        if (*text*) {

*const* newData = fullData.filter(*function* (*item*) {

*const* itemData = *item*.Title

                    ? *item*.Title.toUpperCase()

                    : ''.toUpperCase();

*const* textData = *text*.toUpperCase();

                return itemData.indexOf(textData) > -1;

            });

            setSelectedData(newData);

            setSearch(*text*);

        } else {

            setSelectedData(fullData);

            setSearch(*text*);

        }

    };

*const* LeftActions = () *=>* {

        return (

            <*View* style={portrait.rightAction}>

                <*Text* style={portrait.actionText}>Delete</*Text*>

            </*View*>

        )

    }

*const* orientation = () *=>* {

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

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

            return portrait

        } else {

            return landscape

        }

    }

    useEffect(() *=>* {

        Dimensions.addEventListener("change", onChange);

        return () *=>* {

            Dimensions.removeEventListener("change", onChange);

        };

    });

    return (

        <*SafeAreaView* style={portrait.container}>

            {

                React.useLayoutEffect(() *=>* {

                    navigation.setOptions({

                        headerRight: () *=>* (

                            <*TouchableOpacity*

                                activeOpacity={0.5}

                                onPress={() *=>* { navigation.navigate('Create')}}>

                                <*MaterialCommunityIcons* style={portrait.addIcon} name="plus" color={'#808082'} size={30} />

                            </*TouchableOpacity*>

                        ),

                    });

                }, [navigation])

            }

            <*View* style={portrait.sectionStyle}>

                <*MaterialCommunityIcons* style={portrait.searchIcon} name="magnify" color={'#808082'} size={26} />

                <*TextInput*

                    style={portrait.textInputStyle}

                    placeholder={'Find movie you want'}

                    clearButtonMode={'while-editing'}

                    onChangeText={(*text*) *=>* searching(text)}

                    value={search}

                />

            </*View*>

            <*VirtualizedList*

                data={selectedData}

                ItemSeparatorComponent={() *=>* {return(<*View* style={portrait.separator}/>)}}

                renderItem={({ *item* }) *=>* (

                    <*Item* id={item.id} title={item.title} year={item.year} type={item.type} poster={item.poster} />

                )}

                getItemCount={getItemCount}

                getItem={getItem}

            />

        </*SafeAreaView*>

    );

}

*const* portrait = StyleSheet.create({

    container: {

        flex: 1,

        marginTop: StatusBar.currentHeight,

        backgroundColor: styleConfig.bg,

    },

    separator: {

        marginLeft: 'auto',

        marginRight: 'auto',

        backgroundColor: styleConfig.separator,

        width: '92%',

        height: 0.5,

    },

    item: {

        flexDirection: 'row',

        backgroundColor: styleConfig.bg,

        height: 'auto',

        justifyContent: 'center',

        marginHorizontal: 0,

        padding: 20,

    },

    title: {

        fontSize: 30,

        color: styleConfig.color

    },

    poster: {

        width: 86,

        height: 150,

        borderRadius: 1,

        borderColor: styleConfig.color,

        borderWidth: 3,

        marginTop: 6

    },

    posterViev: {

        flex: 2

    },

    textViev: {

        flex: 10,

        marginLeft: 55,

    },

    details: {

        fontSize: 16,

        marginTop: 10,

        color: styleConfig.color

    },

    textInputStyle: {

        flex: 1,

        height: 40,

        margin: 2,

        borderRadius: 10,

        color: styleConfig.searchColor.color

    },

    sectionStyle: {

        flexDirection: 'row',

        justifyContent: 'center',

        alignItems: 'center',

        backgroundColor: styleConfig.searchColor.bg,

        height: 40,

        borderRadius: 12,

        marginTop: 10,

        marginHorizontal: 6,

        marginBottom: 3,

    },

    searchIcon: {

        margin: 8,

        color: styleConfig.searchColor.color

    },

    rightAction: {

        flex: 1,

        justifyContent: 'center',

        backgroundColor: 'red',

    },

    actionText: {

        color: '#fff',

        padding: 20,

        textAlign: 'right'

    },

    addIcon: {

        textAlign: 'right',

        marginHorizontal: 16,

        marginBottom: 5,

        marginTop: 2,

        color: 'white'

    }

});

*const* landscape = StyleSheet.create({

    textViev: {

        marginRight: 20,

        flex: 10,

        marginLeft: -20

    },

    poster: {

        width: 80,

        height: 135,

        borderRadius: 1,

        marginTop: 6,

        borderColor: styleConfig.color,

        borderWidth: 3,

    },

})

**Create.js**

import React, { useState } from 'react';

import { TextInput, StyleSheet, View, Text, Button, ScrollView } from 'react-native';

import { DATA } from "../screens/Movie";

*let* newPosterCounter = 0

import { styleConfig } from "../style";

*const* inputColor = styleConfig.searchColor

export default *function* Create({ *navigation*, *route* }) {

*const* [title, onChangeTitle] = useState('');

*const* [type, onChangeType] = useState('');

*const* [year, onChangeYear] = useState('');

*const* [show, setShow] = useState(false)

*const* newItem = () *=>* {

        onChangeTitle('')

        onChangeType('')

        onChangeYear('')

        if (title != '') {

*const* obj = {

                "Title": title,

                "Type": type,

                "Year": year,

                "imdbID": newPosterCounter + 1,

            }

            DATA.push(obj)

            newPosterCounter++

*navigation*.navigate('Movie')

        } else {

            setShow(true)

            setTimeout(() *=>* {

                setShow(false)

            }, 4000);

        }

    }

*const* arr = []

    return (

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

            <*ScrollView* contentContainerStyle={{ flexGrow: 1 }}

                keyboardShouldPersistTaps='handled'

            >

                <*Text* style={styles.textStyle}>Title</*Text*>

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

                    <*TextInput*

                        style={styles.textInputStyle}

                        onChangeText={(*text*) *=>* onChangeTitle(*text*)}

                        value={title}

                        clearButtonMode={'while-editing'}

                    />

                </*View*>

                <*View* style={{ padding: 2 }}>

                    {show ? (<*Text* style={styles.textTipStyle}>This field must be filled!</*Text*>) : null}

                </*View*>

                <*Text* style={styles.textStyle}>Type</*Text*>

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

                    <*TextInput*

                        style={styles.textInputStyle}

                        onChangeText={(*text*) *=>* onChangeType(*text*)}

                        value={type}

                        clearButtonMode={'while-editing'}

                    />

                </*View*>

                <*View* style={{ padding: 2 }}></*View*>

                <*Text* style={styles.textStyle}>Year</*Text*>

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

                    <*TextInput*

                        keyboardType={'numeric'}

                        style={styles.textInputStyle}

                        onChangeText={(*text*) *=>* {

                            onChangeYear(*text*)

*let* arr = *text*.split('')

                            arr.forEach(*function*(*item*, *index*, *array*) {

                                if(*item* === ',') {

                                    onChangeYear(year)

                                }

                            })

                        }}

                        value={year}

                        clearButtonMode={'while-editing'}

                        maxLength={4}

                    />

                </*View*>

                <*View* style={{ padding: 2 }}></*View*>

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

                    <*Button*

                        style={styles.buttonStyle}

                        title="Add new movie"

                        onPress={newItem}

                    />

                </*View*>

            </*ScrollView*>

        </*View*>

    );

}

*const* styles = StyleSheet.create({

    screen: {

        flex: 1,

        backgroundColor: styleConfig.bg,

        paddingHorizontal: 20,

        paddingVertical: 0,

    },

    textInputStyle: {

        flex: 1,

        height: 40,

        marginLeft: 10,

        borderRadius: 10,

        color: inputColor.color

    },

    sectionStyle: {

        flexDirection: 'row',

        justifyContent: 'center',

        alignItems: 'center',

        backgroundColor: inputColor.bg,

        height: 40,

        borderRadius: 10,

        marginTop: 10,

        marginHorizontal: 8,

        borderColor: '#808082',

        borderWidth: 1,

    },

    buttonStyleContainer: {

        marginVertical: 16,

    },

    buttonStyle: {

        color: styleConfig.color,

    },

    textStyle: {

        marginLeft: 16,

        color: styleConfig.color,

        marginTop: 20,

        fontSize: 16,

    },

    textTipStyle: {

        paddingTop: 6,

        marginLeft: 16,

        position: 'absolute',

        fontSize: 12,

        color: 'blue',

    },

});

**Details.js**

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

import { View, Dimensions, Text, StyleSheet, SafeAreaView, ScrollView, Image, ActivityIndicator } from 'react-native';

import { styleConfig } from "../style";

import { choosePoster } from "../assets/global/choosePoster";

import { tt } from "../assets/global/tt";

*const* window = Dimensions.get("window");

*const* screen = Dimensions.get("screen");

export default *function* Details({ *route* }) {

*const* { id, title, year } = *route*.params;

*const* data = tt(id)

*const* [dimensions, setDimensions] = useState({ window, screen });

*const* onChange = ({ *window*, *screen* }) *=>* {

        setDimensions({ window, screen });

    };

    useEffect(() *=>* {

        Dimensions.addEventListener("change", onChange);

        return () *=>* {

            Dimensions.removeEventListener("change", onChange);

        };

    });

*const* orientation = () *=>* {

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

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

            return styles

        } else {

            return landscape

        }

    }

    return (

        <>

            <*SafeAreaView*>

                <*ScrollView* style={{ backgroundColor: styleConfig.bg }}>

                    <*View* style={orientation().infoScreen}>

                        <*View* style={orientation().infoImageSection}>

                            <*Image*

                                style={orientation().infoImage}

                                source={choosePoster(data.Poster) }

                            />

                        </*View*>

                        <*View* style={orientation().infoScreenTextView}>

                            <*Text* style={orientation().titleText}>Title</*Text*>

                            <*Text* style={orientation().subText}>{data.Title != '' ? data.Title : title}</*Text*>

                            <Text style={orientation().titleText}>Runtime</Text>

                            <Text style={orientation().subText}>{data.Runtime}</Text>

                            <Text style={orientation().titleText}>Genre</Text>

                            <Text style={orientation().subText}>{data.Genre}</Text>

                            <Text style={orientation().titleText}>Awards</Text>

                            <Text style={orientation().subText}>{data.Awards}</Text>

                            <Text style={orientation().titleText}>Rating</Text>

                            <Text style={orientation().subText}>{data.imdbRating}</Text>

                            <Text style={orientation().titleText}>Actors</Text>

                            <Text style={orientation().subText}>{data.Actors}</Text>

                            <Text style={orientation().titleText}>Language</Text>

                            <Text style={orientation().subText}>{data.Language}</Text>

                            <Text style={orientation().titleText}>Plot</Text>

                            <Text style={orientation().subText}>{data.Plot}</Text>

                            <Text style={orientation().titleText}>Director</Text>

                            <Text style={orientation().subText}>{data.Director}</Text>

                            <Text style={orientation().titleText}>Country</Text>

                            <Text style={orientation().subText}>{data.Country}</Text>

                            <Text style={orientation().titleText}>Production</Text>

                            <Text style={orientation().subText}>{data.Production}</Text>

                            <Text style={orientation().titleText}>Released</Text>

                            <Text style={orientation().subText}>{data.Released}</Text>

                            <Text style={orientation().titleText}>Year</Text>

                            <Text style={orientation().subText}>{data.Year != '' ? data.Year : year}{'\n'}</Text>

                        </View>

                    </View>

                </ScrollView>

            </SafeAreaView>

        </>

    );

}

const styles = StyleSheet.create({

    titleText: {

        color: styleConfig.color,

        fontWeight: '600',

        fontSize: 20,

        marginVertical: 2,

    },

    subText: {

        color: styleConfig.color,

        fontWeight: '400',

        marginTop: 2,

        marginBottom: 8,

        fontSize: 18,

    },

    infoScreen: {

        paddingHorizontal: 13,

        paddingTop: 10,

        paddingBottom: 40,

        backgroundColor: styleConfig.bg

    },

    infoImageSection: {

        alignItems: 'center'

    },

    infoImage: {

        width: 180,

        height: 270,

        borderWidth: 2,

        borderColor: '#EEE'

    },

    infoScreenTextView: {

        marginTop: 30,

    },

    loading: {

        position: 'absolute',

        left: 0,

        right: 0,

        top: 0,

        bottom: 0,

        alignItems: 'center',

        justifyContent: 'center',

        backgroundColor: styleConfig.bg,

        color: styleConfig.color

    }

});

const landscape = StyleSheet.create({

    infoScreen: {

        paddingHorizontal: 16,

        paddingTop: 10,

        paddingBottom: 40,

        backgroundColor: styleConfig.bg,

        flex: 1,

        flexDirection: 'row',

    },

    infoImage: {

        width: 170,

        height: 300,

        borderWidth: 2,

        borderColor: styleConfig.color,

        marginTop: 6,

    },

    infoScreenTextView: {

        paddingLeft: 14,

        flexShrink: 1

    },

    titleText: {

        color: styleConfig.color,

        fontWeight: '600',

        fontSize: 21,

        marginVertical: 1,

    },

    subText: {

        color: styleConfig.color,

        fontWeight: '400',

        fontSize: 19,

    },

    loading: {

        position: 'absolute',

        left: 0,

        right: 0,

        top: 0,

        bottom: 0,

        alignItems: 'center',

        justifyContent: 'center',

        backgroundColor: styleConfig.bg,

        color: styleConfig.color

    }

})

**Висновок**

Під час виконання лабораторної роботи ми покращили нашу програму, а саме додали туди пошук, видалення на свайп уліво, кнопку для додавання нових постерів яка навігує нас на новий скрін з формою. Програма працює як у портреті так і у лендскейпі без переривань та ворнінгів.