Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа №10**

**по дисциплине**

**«Технологии разработки мобильных приложений»**

**Знакомство с React Native**

**Выполнил**:

ст. гр. ПРИ-120

Д. А. Грачев

**Принял**:

Петрова А. И.

Владимир, 2023

ЦЕЛЬ РАБОТЫ

Реализовать приложение с использованием фреймворка React Native

ВЫПОЛНЕНИЕ РАБОТЫ

1. Реализуем базовый пример с надписью «Hello World» по примеру из методических указаний

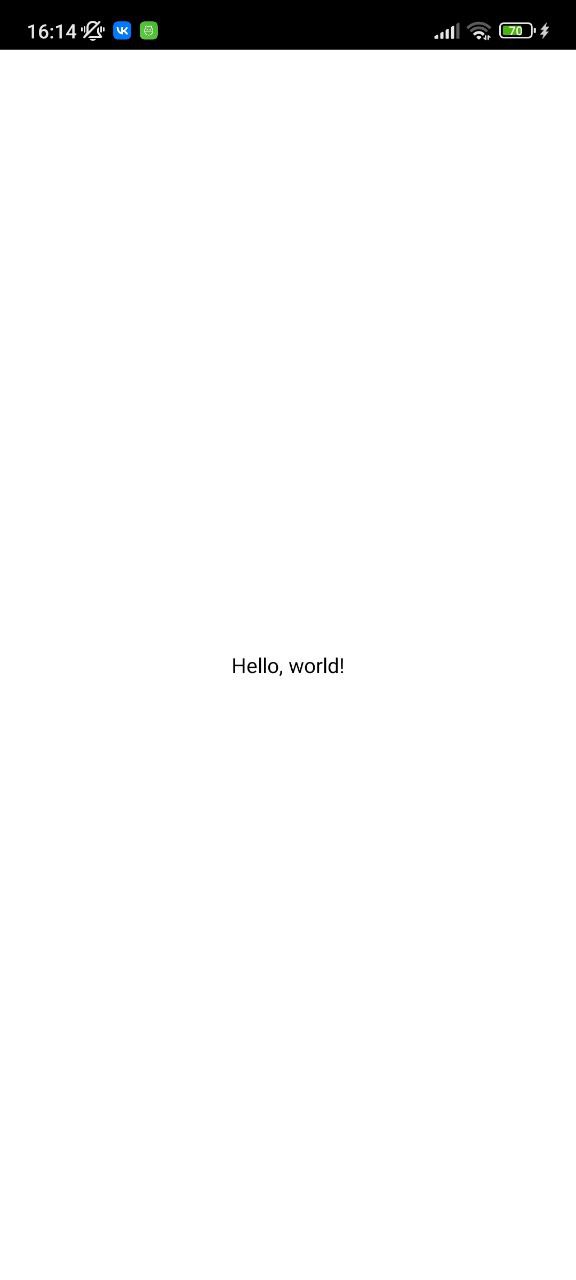


Рисунок 1. Hello World

1. Реализуем пример с кнопкой, которая меняет текст при ее удержании



Рисунок 2. Пример с кнопкой

1. Реализуем пример со список из методических указаний

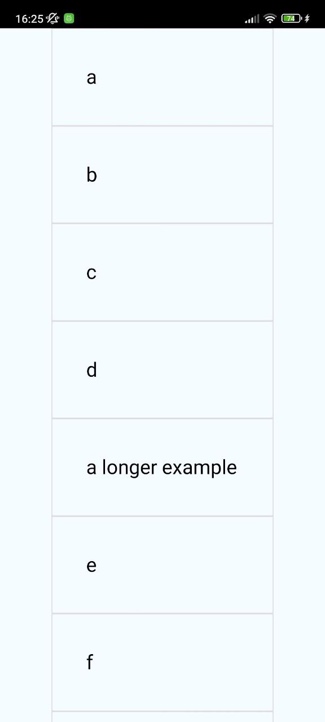


Рисунок 3. Пример со списком

1. С помощью пакета @react-navigation/native-stack создадим навигацию по страницам в приложении

*const* Navigator = () => {  
 *const* Stack = createNativeStackNavigator();  
  
 *return* (  
 <Stack.Navigator initialRouteName={'Main'}>  
 <Stack.Screen name={'Main'} component={Main} />  
 <Stack.Screen name={'Button'} component={ButtonExample} />  
 <Stack.Screen name={'HelloWorld'} component={HelloWorld} />  
 <Stack.Screen name={'ListExample'} component={ListExample} />  
 </Stack.Navigator>  
 );  
};

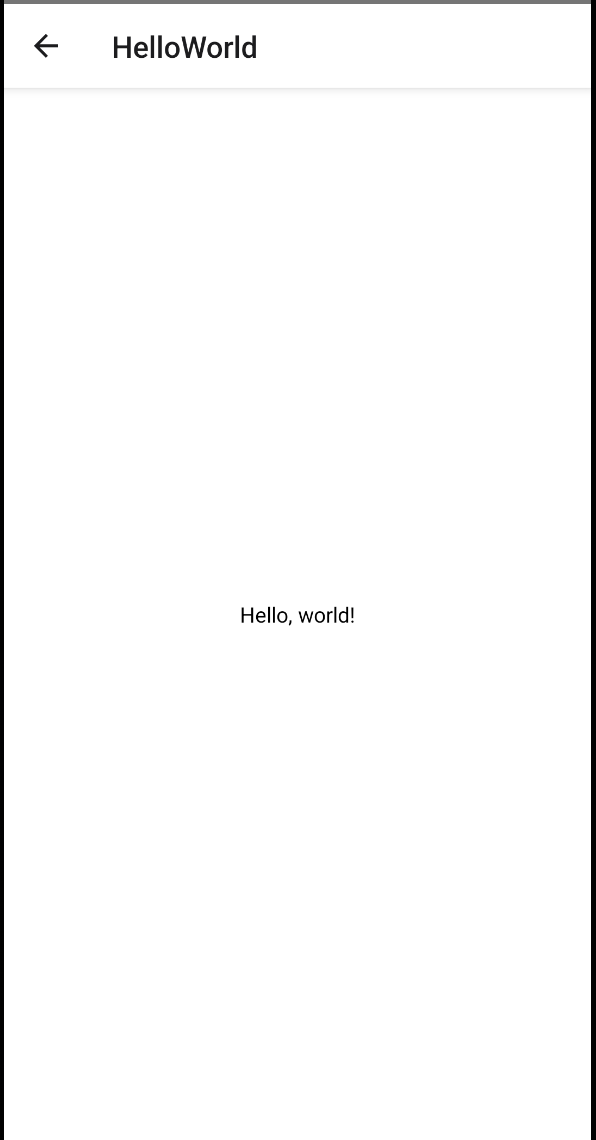
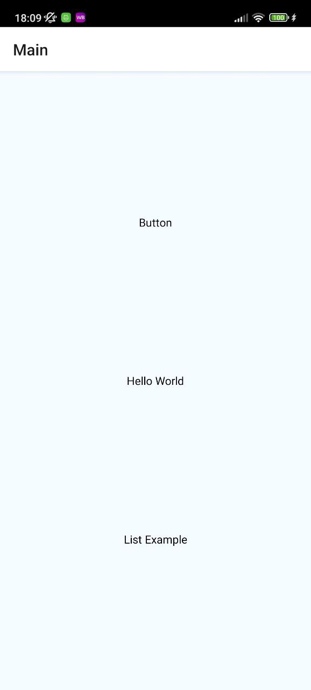


Рисунок 4. Навигация по страницам

1. Изменим цвет и размер кнопки

button: {  
 backgroundColor: '#3caa3c',  
 borderRadius: 50,  
 height: 300,  
 width: 200,  
 justifyContent: 'center',  
},



Рисунок 5. Измененная кнопка

1. Изменим список добавив туда фотографии книг, их названия и упоминания авторов



Рисунок 6. Измененный список

1. Добавим изменения в код так, чтобы при нажатии на элемент списка, его фон менялся

*const* [key, setKey] = useState(-1);  
*const* getRandomColor = () => {  
 *const* letters = '0123456789ABCDEF';  
 *let* hex = '#';  
 *for* (*let* i = 0; i < 6; i++) {  
 hex += letters[Math.floor(Math.random() \* 15)];  
 }  
 *return* hex;  
};  
  
*const* handleTouchRow = key => {  
 setKey(key);  
};

<Pressable  
 onPress={() => handleTouchRow(item.key)}>  
 <View  
 style={{  
 ...styles.row,  
 backgroundColor:  
 item.key === key ? getRandomColor() : 'transparent',  
 }}>  
 <Image source={item.photo} style={styles.image} />  
 <View>  
 <Text style={styles.text}>{item.name}</Text>  
 <Text style={styles.text}>{item.description}</Text>  
 </View>  
 </View>  
</Pressable>

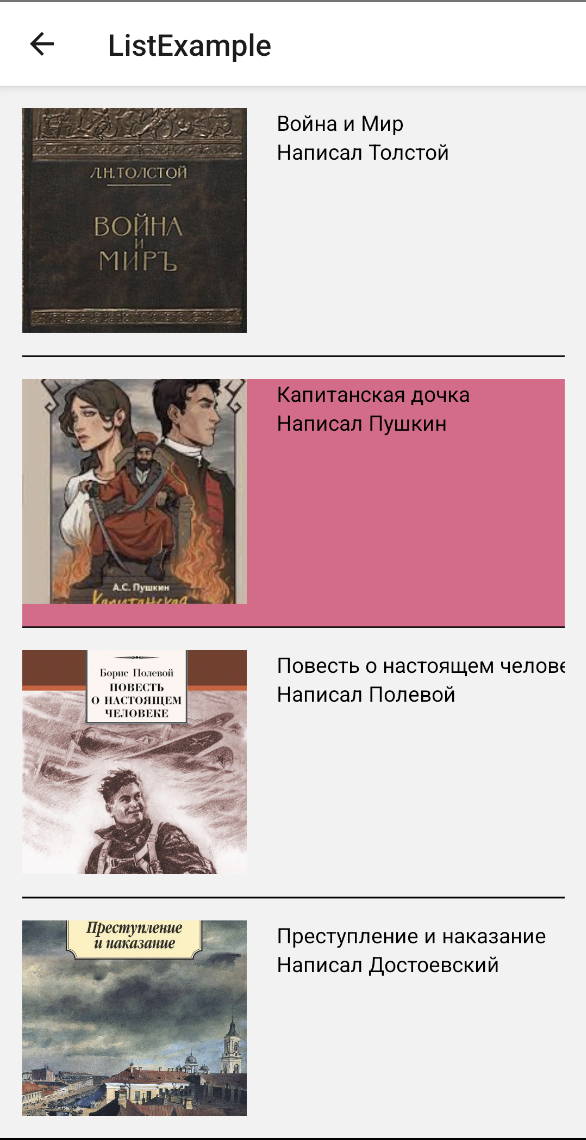


Рисунок 7. Список с измененным фоном элемента

1. Добавим смену кнопки на картинку при ее удержании

*const* ButtonExample = () => {  
 *const* [pressing, setPressing] = useState(*false*);  
  
 *const* togglePressing = () => {  
 setPressing(prevState => !prevState);  
 };  
  
 *return* (  
 <View style={styles.container}>  
 <TouchableHighlight  
 onPressIn={togglePressing}  
 onPressOut={togglePressing}  
 style={{borderRadius: 50}}>  
 <View style={styles.button}>  
 {pressing ? (  
 <Image  
 source={require('../assets/povest.jpg')}  
 style={styles.button}  
 />  
 ) : (  
 <Text style={styles.welcome}>  
 {pressing ? 'Лягушка в обмороке!' : 'Влюблённая жаба!'}  
 </Text>  
 )}  
 </View>  
 </TouchableHighlight>  
 </View>  
 );  
};

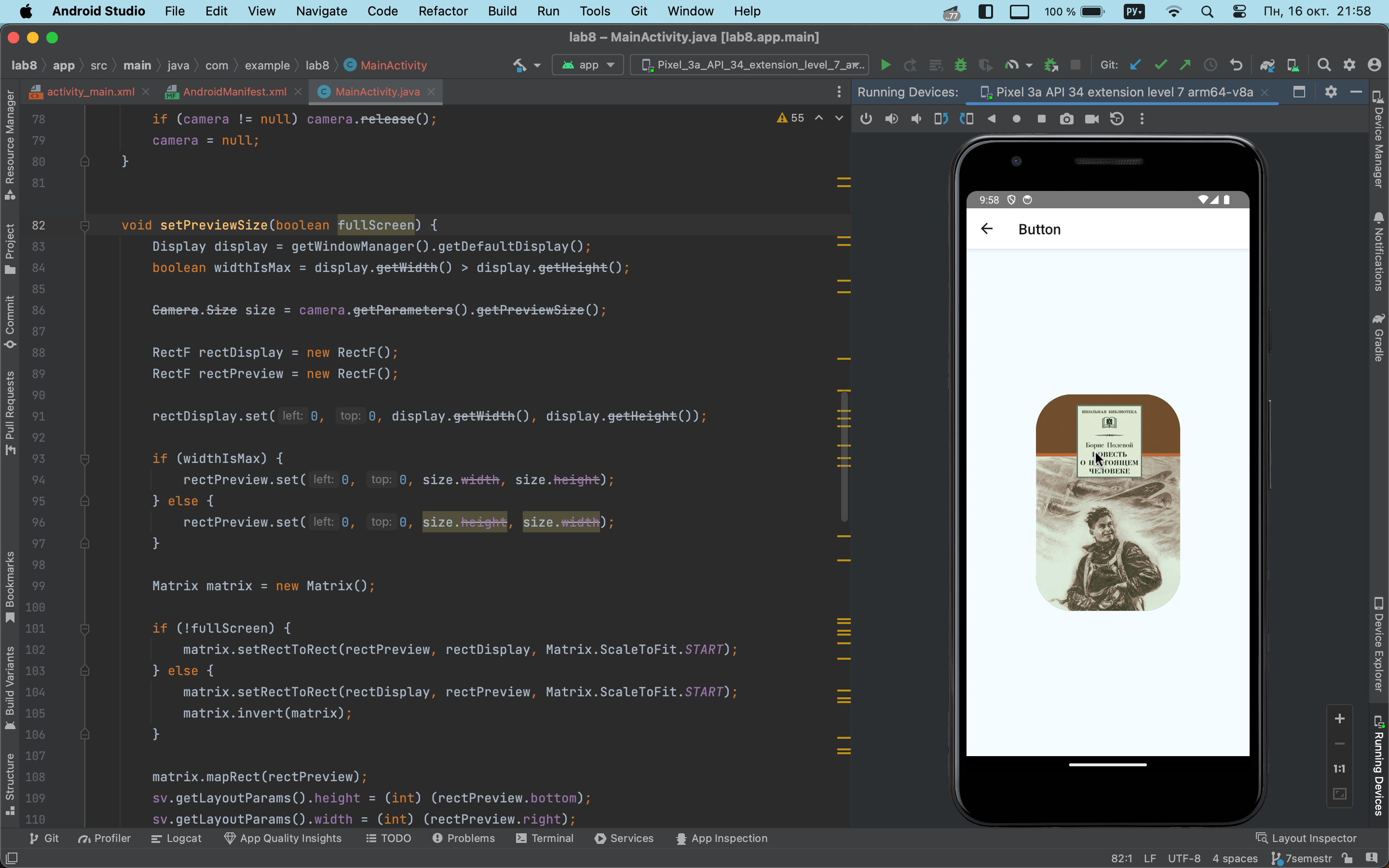


Рисунок 8. Кнопка с картинкой

1. Добавим возможность удаления элемента при помощи длительного удержания

<Pressable  
 onPress={() => handleTouchRow(item.key)}  
 onLongPress={() => handleLongPress(item.key)}>

*const* handleLongPress = key => {  
 setData(prevState => prevState.filter(item => item.key !== key));  
};



Рисунок 9. Удаление элементов

ВЫВОД

В ходе выполнения лабораторной работы были получены практические навыки по реализации приложения с использованием фреймворка React Native