**Технически университет - София**

**Факултет по Компютърни Системи и Технологии**

**Курсов проект**

**Програмиране за мобилни устройства**

**Темa: „Тестова Система“**

**Изготвил: Марио Веселинов Стоилов**

**Факултетен Номер: 121211122**

Github репозитори - <https://github.com/energy-uktc/QuizApp>

Съдържание:

1. Изисквания към мобилното приложението.
2. Архитектура на приложението. Използвани технологии.
3. Описание на мобилното приложение. Ръководство за употреба.
4. Програмен код на приложението.

# **Изисквания към мобилното приложение**

Да се проектира и реализира програма позволяваща провеждането на тестове за проверяване на знания в дадена област. Да се предостави възможност за добавянето на

нови въпроси. Да се включат и въпроси с отворен отговор (например посочване на отговор конкретно число). Като опции на програмата да се включат режими на тестване с време и без. Възможност за печат на тест, печат или запис на справка с

резултати от положени тестове.

# **Архитектура на приложението. Използвани технологии**

Мобилното приложението е базирано на Expo софтуерна рамка и React Native компоненти за изграждането на кросплатформен потребителски интерфейс и Firebase Realtime Database централизирана NoSQL база данни.

**React Native** е софтуерна рамка с отворен код базирана на JavaScript. Използва се за изграждане на кросплатформени мобилни приложения за iOS и Android. Съществуват свободно поддържани от потребителите разклонения на React Native, които позволяват и разработване на приложения за Windows, MacOS, Apple TV, Android TV. React Native компонентите представляват абстракция върху базовите за платформата компоненти, като по този начин позволява изграждането на потребителски интерфейс, който изглежда естетически консистентно с останалите елементи на мобилната платформа. Например View компонента в React Native се транслира до ViewGroup в Android и UIView в iOS. В таблицата по-долу, взета от официалния сайт на React Native, може да се види връзката между основните компоненти на фреймуърка и съответстващите им в Android и iOS.

React Native използва JSX ( JavaScript XML ) за описване на елементите и вграждането им в JavaScript код. Всеки React Native елемент притежава набор от свойства (props) , които определят как се изобразява един компонент, и състояние (state) ,което съдържа данните с които работи компонента – всяка промяна в състоянието води до повторно изобразяване на елемента.

<https://reactnative.dev/docs/intro-react-native-components>

| **REACT NATIVE UI COMPONENT** | **ANDROID VIEW** | **IOS VIEW** | **WEB ANALOG** | **DESCRIPTION** |
| --- | --- | --- | --- | --- |
| <View> | <ViewGroup> | <UIView> | A non-scrollling <div> | A container that supports layout with flexbox, style, some touch handling, and accessibility controls |
| <Text> | <TextView> | <UITextView> | <p> | Displays, styles, and nests strings of text and even handles touch events |
| <Image> | <ImageView> | <UIImageView> | <img> | Displays different types of images |
| <ScrollView> | <ScrollView> | <UIScrollView> | <div> | A generic scrolling container that can contain multiple components and views |
| <TextInput> | <EditText> | <UITextField> | <input type="text"> | Allows the user to enter text |

**Expo**

**Firebase Realtime Database**