

React Native UI Framework

Research Document

Coaches

Name	Email
Patrick IJntema	p.ijntema@fontys.nl
Martijn Ruissen	m.ruissen@fontys.nl

Team Members

Name	Email
Aleksandar Todorov	aleksandar.todorov@student.fontys.nl
Rene Roets	Rene.roets@student.fontys.nl
Domantas Mauruca	d.mauruca@student.fontys.nl

Table of contents

Coaches	1
Team Members	1
Table of contents	2
Changelog	2
Overview:	3
Requirements:	3
Criteria:	4
Responsive:	4
User experience:	4
Customizability:	5
Developer experience:	5
The choices:	6
AntDesign Mobile	6
NativeBase	6
React Native Paper	7
Conclusion:	7
References	8

Changelog

By	Changes	Date
Aleksandar Todorov	Initial document	01/10/2022
Aleksandar Todorov	Adding criteria	10/10/2022

Overview:

This document provides information and research conducted during the analysis stage for the project “Drieam mobile Porfolio application”, regarding the chosen UI framework. It describes the requirements set by the project owner and the limitations that certain technologies have. In the end it is noted which technology would best suit the project needs and why.

Requirements:

The project owner (Drieam) did not have any mandatory requirements regarding the UI framework of the mobile application. However it was noted that they would prefer the app to be inspired by their original desktop web application regarding the styling and user experience. The original Portfolio software was built using a combination of premade components such as buttons and labels by the AntDesign framework. They have also included some custom elements made and styled entirely by them.

Due to the nature of a mobile application, it needs to support a variety of dimensions and dots per inch (DPI). Therefore a big requirement for the UI framework is the support for smartphone responsiveness.

Criteria:

There are a number of criteria that need to be matched by the UI framework so they can fit the given assignment. The different requirements are listed below by importance order.

Criteria	Description	Importance
Responsive	Ability to support a variety of display devices	5
User experience	Satisfy and enhance the user behavior	4
Customizable	Provides a wide range of custom components / styling options	3
Developer experience	Ease of use by the developer and existing community support	4

** Importance is scaled from 1-5 based on stakeholder feedback, developer needs and analysis of user requirements*

Responsive:

The goal of the assignment is to create an application that supports a wide variety of mobile devices. The display resolution is different from device to device therefore the UI library needs to support responsive layouts. To achieve this the best way to ensure compatibility between devices is with the use of display breakpoints, so a library which has this feature would be with priority.

User experience:

The end product will be used by students with different backgrounds so in order to maximize efficiency while using this application, the app needs to have a strong emphasis on user

experience. There can be lots of different factors when taking this into account such as ease of use and simplicity but also a mix of colors and shapes.

Customizability:

This assignment is a by-product of an existing application with an established user base, therefore there is a strong emphasis on UI library customization. The mobile app needs to mimic at least partially the behavior and feel of the original product that it is based on by providing the user with the same iconography and shape language. The selected library needs to support a wide range of customization options in order to fit the acceptance criteria.

Developer experience:

Another important criteria is the developer experience which can be determined by multiple factors. One of them is the need for proper extensive documentation provided by the UI library creators. That would greatly enhance the efficiency and end product delivered in this assignment. Another factor is ease of use by the developer and the time it will take to establish the groundwork for the rest of the assignment. Libraries which are too underdeveloped or have a bad reputation among the community would also be a disadvantage.

The choices:

There are a variety of choices for a UI framework which supports React Native. Some of the criteria for choosing the best suitors would be popularity among the community, support for old and new mobile devices, look and customization and overall mobile responsiveness.

Name	Customizability	Community support	Good mobile support	Documentation	Responsiveness
AntDesign Mobile	✓	✓	X	X	X
NativeBase	✓	✓	✓	✓	✓
React Native Paper	X	✓	✓	✓	✓

AntDesign Mobile

Upon initial research it was concluded that the UI framework that the project owner used for their original “Portfolio” application would not suffice for a mobile application. AntDesign, which is their main choice for their website, does not support the project's main technology stack React Native. However, there is a mobile counterpart AntDesign Mobile [1]. Upon further research it was concluded that the library lacked a lot of the essential features that the full web-library supported. The support for different mobile dimensions is limited and would require a lot of manual labor to make sure the design is fluid across all mobile devices. Another noteworthy point is the lack of solid documentation regarding the different components and usability.

NativeBase

Another very popular option across the React Native community is NativeBase [2]. It is one of the top choices due to its abundance of components and superior documentation provided by the developers. It focuses on responsive design across the spectrum, from mobile devices upto desktop dimensions. The library is utility focused meaning that you can create complex designs with just passing props to each individual component. Another advantage of NativeBase is that it also has extensive support for custom theming, providing the developer with different options to create their designs

React Native Paper

React Native Paper [3] is another highly appraised by the community UI library due to its inspiration from Google's Material Design [4] styling guide. The library has extensive documentation for its use cases and provides an exceptional user experience for the application. React Native Paper was built with responsiveness into account, providing easy to implement components by the developer. A downside however is the lack of customization provided for the library due to the fixed design choices inspired by Google's Material Design styling guide. Because of the lack of custom components, it wouldn't be efficient to use a mixture between the library and an existing custom stylesheet.

Conclusion:

After conducting and analyzing the research results, it was decided that the best tool for the job would be NativeBase. The ideal choice would be AntDesign Mobile, due to the existing use by the stakeholders in the original website application "Portfolio", but due to the limited capabilities provided by the library and lack of proper documentation, it would be inefficient to use in the given context. A third option was provided - React Native

Paper, however due to its strict Google's Material Design styling and lack of customization, it wouldn't be ideal to use in the given assignment.

NativeBase is the ideal choice for the given assignment due to its abundance of components and support for custom stylesheets. The developers have also provided outstanding documentation regarding its use and implementation.

References

- [1] AntDesign Mobile - <https://mobile.ant.design/>
- [2] NativeBase - <https://nativebase.io/>
- [3] React Native Paper - <https://reactnativepaper.com/>
- [4] Material Design - <https://material.io/design>