



UPPSALA
UNIVERSITET

IT 16 053

Examensarbete 15 hp
Augusti 2016

Cross-platform development with React Native

Aymen Beshir



UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Angströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Cross-platform development with React Native

Aymen Beshir

In this project a mobile application for dog owners is built, which allows dog owners to create their own profile. The customer is a dog whisperer with the aspiration to create a platform for dog owners where they can share and access articles and experiences and structure their dog's life.

This mobile application is built for both Android and iOS. Building native mobile applications has never been easier given the many resources and frameworks available for developers. But since the frameworks are often built for one specific mobile operating system the process can be aggravating. In this project the native mobile application is built with the framework react native, which allows us to build an application with the same code for two mobile operating systems.

In this essay the UI and the programming language of react native JSX will be discussed. The focus is essentially on building a good application with a user friendly design, rather than building an application that is completely ready for the market. The resulting product is a functional application with a simple design that is easy to understand, even for the first time users.

Handledare: Emil Paulsson
Ämnesgranskare: Justin Pearson
Examinator: Olle Gällmo
IT 16053

Contents

1	Introduction	4
2	Background	5
2.1	React.js	5
2.2	React Native	6
2.2.1	Native	6
2.2.2	JavaScript/JSX	7
2.2.3	NPM	8
3	Existing application in React native	9
4	Important components	10
4.1	View	10
4.1.1	StyleSheet	10
4.1.2	ListView	10
4.1.3	FlexBox	12
4.2	Navigator	12
4.2.1	NavigationBar	12
4.3	AnimatedView	13
5	Platforms AndroidSDK and Xcode	14
5.1	AndroidSDK	14
5.2	Xcode	15
6	Interface	16
6.1	Kunskapsbank/Övningsbank	18
6.2	Mina listor	20
6.3	Min hund	21
7	Getting started with React Native	23
8	Hello World with React Native	24
9	Crossing platforms iOS and Android	26
9.1	Android vs iOS design	26
10	Working Agile	27

11 Discussion/Result	28
11.1 Books	30
11.2 Webpages	30

1 Introduction

The mobile application market is growing and showing no signs of declining anytime soon. One of the biggest reasons for its growth is the breakthrough and widespread adoption of the smart phone, which has caused the mobile app market to grow substantially [9]. The purpose of this thesis is to create an application that is compatible with both iOS and Android for a customer of the company Interactive Solutions. There are a lot of different stages to take into consideration before an app can be released on the market. One of the most fundamental questions to answer is if an app is going to be available for iOS or Android, or maybe both. When creating an app that is compatible with both iOS and Android the programming has to be done separately and are written in two different development environments. This originally meant higher costs and high resource requirements for companies, but Facebook introduced the React Native framework as a solution to this problem[6]. React Native aims to facilitate the process of writing across different mobile operating systems and make it easier to implement the same code for both iOS and Android [10].

The application that is going to be built with the React Native is to a customer. The customer is a dog whisperer who wants to create a platform for dog owners where users can share articles and experiences to help, inspire and create a dog profile were the dog owner is able to structure their dog's life. The name of this application is Canine Of Sweden and Canine is another definition of a dog[7]. Given that the application is ordered from a customer the app is made mostly of how the customer and the company wants the application to function. In order to make the application available for both iOS and Android the framework React Native will be used. The implemented programming languages for the two different mobile operating systems are based on JavaScript and React. The code will be compiled in different platforms given the operating system, Xcode[3] C for iOS and Android SDK[4] for Android. Will React Native make it easier or more complicated when implementing the code from Android to iOS or contrariwise?

2 Background

To be able to explain what React Native is one has firstly understand what React.js is.

2.1 React.js

React.js is a library for the web programming language JavaScript, and it is used for creating User Interfaces of websites[2]. In May 2013 Facebook introduced React and since then it has grown enormously, both internally and externally. Many developers choose to work with React because it gives more time to concentrate on creating a good product instead of fighting with the framework. Examples of big websites that use React are Netflix, Facebook and many more[12].

React is built on components and every component represents a view. Having many different components makes it easy to iterate with the products, because if a change needs to be made the whole system do not have to change, only the specific component. React is for web and creating a native mobile environment is more difficult[2]. For example when building on the web the developer could easily save the files and reload the browser to see the changes, meanwhile working with native application it has to be recompiled and installed on the emulator or the physical device. Facebook knew that they have a native application anyways because it is more consistent and a better feeling experience. Given the different UI components that comes with native apps like navigation, maps and so on.

When it was time to create a platform to build mobile applications Facebook thought of how they could give the user a native experience and at the same time give the developer the same experience as building with React.js. React Native enables you to build world-class application experiences on native platforms using a consistent developer experience based on JavaScript and React[1].

2.2 React Native

React native is a framework that allows the developer to build application on native platforms with libraries of React and JavaScript. Febraury 8 2015 Facebook released the open source for React Native, but only for iOS[11]. The Android open source was released on September the same year[13]. Many application developers were very excited about the framework, especially the React.js developers because it would be the same technique or as Facebook said:

“Learn once, write everywhere”

What they meant is that a developer that knows how to write code in React.js will have no problems writing in React native, but not use the same code. React native is built to give the developer the same experience as for web application, no compilation step and the changes appears instantly[5]. What stands out with React Native is that most of the other mobile application frameworks uses programming languages like Objective-C, Java and so on. Meanwhile React Native gives the web developer a chance to easily work with native mobile applications. React Native uses all of the technology used for React and applies it to native apps. Instead of using div like in React and HTML React Native uses View and Text. The components are written in JavaScript and React Native takes the UI and the minimal amount of data and sends it to the main thread to render it with the native components[5].

2.2.1 Native

The second part of the name React Native is native. Native applications are applications that are adjusted for specific platforms and have full access to the API that communicates directly to the software unit. Native apps are often mentioned in mobile development because mobile applications is often written for a specific platform.

2.2.2 JavaScript/JSX

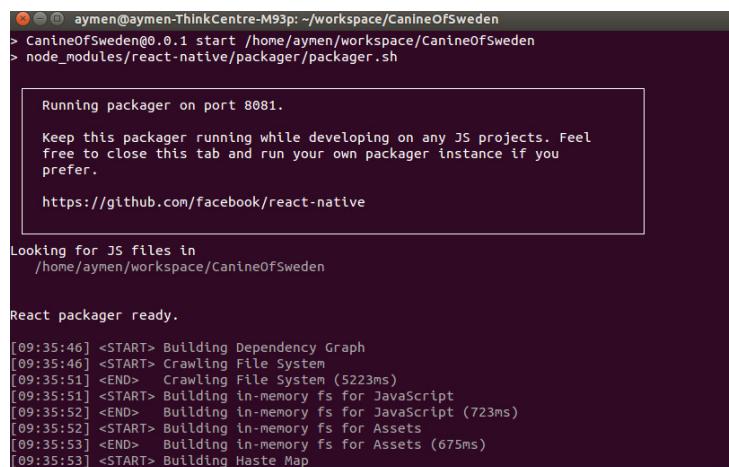
JavaScript is a very common programming language in the web developing world. In React Native the developer can choose to either use JS(JavaScricpt) or JSX library, but React Native strongly advises the developers to use JSX. It is concise and has a familiar syntax for defining tree structures with attributes and is also more common for design developers. For the Canine Of Sweden project JSX was used as the programming language.

JSX is JavaScript with a syntax extension which allows a mixture between JavaScript and XML tags. JSX is a objected oriented programming language that is designed to run on web browsers. Since React Native wanted the developers that are used to develop for web to get the same feeling they chose to use JSX. The JSX code is still compiled in JavaScript. JSX optimizes the code so it runs faster compared to if it was written directly in JavaScript[2].

In contrary to JavaScript, JSX is statically-typed. JavaScript is very famous for having a bad debugging system, most errors will be caught during the compilation process the debugging system gets much more effective. Beside that the actual coding process is very similar to JavaScript, the expressions, creating methods and statements, therefore it is not difficult for a developer that is used with JavaScript to switch to JSX[2].

2.2.3 NPM

Working with a big project, which in this case means creating an application from scratch, it is necessary to have a tool that takes care of the different packages, making sure that they are synced. Luckily for JavaScript users there is something called NPM, which is a tool that enables JavaScript developers to easily share and reuse code and update the codes that are used or shared. When working with mobile apps there are several packages with different modules for different problems. NPM have something called “npm start” that runs the package start script so that the different packages directory gets synced. This simplifies the process of working with several modules. In the picture below the npm starts and search for the JS files and the different files interact.



```
aymen@aymen-ThinkCentre-M93p: ~/workspace/CanineOfSweden
> CanineOfSweden@0.0.1 start /home/aymen/workspace/CanineOfSweden
> node_modules/react-native/packager/packager.sh

    Running packager on port 8081.

    Keep this packager running while developing on any JS projects. Feel
    free to close this tab and run your own packager instance if you
    prefer.

    https://github.com/facebook/react-native

Looking for JS files in
/home/aymen/workspace/CanineOfSweden

React packager ready.

[09:35:46] <START> Building Dependency Graph
[09:35:46] <START> Crawling File System
[09:35:51] <END>   Crawling File System (5223ms)
[09:35:51] <START> Building in-memory fs for Javascript
[09:35:52] <END>   Building in-memory fs for Javascript (723ms)
[09:35:52] <START> Building in-memory fs for Assets
[09:35:53] <END>   Building in-memory fs for Assets (675ms)
[09:35:53] <START> Building Haste Map
```

3 Existing application in React native

Apps that are created in React Native that works for both Android and iOS are not the easiest to find. React Native was launched in March 2015 and for Android even later when Facebook made an app called Ads manager that was compatible with both iOS and Android. Ads manager creates a platform for Facebook ads, whether it is a small company or a big. The team consisted of eight engineers who started to work on iOS first, since React Native was released for iOS first. These engineers were not necessarily familiar with either JavaScript or React, but they were eager to make a great app for their advertisers.

One of the big challenges the team faced was how to make the navigation flow to work for their desired breadcrumb navigator. A breadcrumb is a trail user interface that allows the user keep track of their location within the program. Their solution was using the component Navigator from React Native to keep track of the different React components by putting them on a stack, making it easy to display one the components and animate between them by pressing a button or using a touch-function. When it was time to make it available for Android the team wanted to reuse as much code as possible, since most of it would be the same, with some exceptions regarding customizing the design of the app for the different operating systems. With the help of React Native's packager features and Reacts abstraction mechanism the team was able to maximize code reusing between the two platforms by excluding all the files that ended with android.js for iOS and excluding all the files that ended with ios.js for Android. This made it possible to nearly reuse 85 % of app code when writing across to Android. After five hard working months and an additional three months they were able to create the first fully functional app on two platforms with the help of React Native.[14]

4 Important components

React Native is built on components and has its standard platform components for the UI. The Components are what makes React Native from the other frameworks because it is easy to use and efficient. In what follow I will go through some of the components that help creating an application with React Native. All the components in React Native start with a capital letter and is inside two HTML tags and closes with two HTML tags and a backslash inside.

4.1 View

View is the most fundamental component in React Native and is supported by the flexbox and style layout. The View component determines how the application is viewed on the screen. Just like div when working with HTML., View is made so that it can be nested inside other views and can have unlimited children types[15]. To get a better insight of how View works there is a good example to show just that:

```
1 <View style = {{height:100, backgroundColor:'blue'}}></View>
```

This example shows a View component that has a style inside of it, a height of 100px and because a width is not assigned React Native is going to assign the width of the device. The backgroundColor element tells React Native what color the background should have.

4.1.1 StyleSheet

StyleSheet is the way the styling is made in React Native and is abstraction similar to CSS. In the section above in the View example, the style was made inside the View component which is not very efficient and makes it more difficult to understand the code. Writing the style outside the render method makes it easier, especially if the same style is going to be used multiple times. By assigning the style into an object it can easily be used[16].

4.1.2 ListView

ListView is a component that is designed to vertically change the list of the changing data. This was crucial when creating an application like Canine Of Sweden since it is an application were there are many different lists

with articles and the view depends on what data is assigned to the different articles[17].

4.1.3 FlexBox

Flexbox makes sure that the element within a container fits to the mobile device it is viewed on. This component makes it very flexible to move around the different elements within the container, which helps when working on the design with React Native. A flex container fills the available space or shrinks them to prevent overflow[18].

4.2 Navigator

The Navigator component is determinant for this application, and for most apps in general. The navigator component makes it possible to keep track of where in app the user is so that it becomes easy to transition between the different scenes in the app[19]. The navigator component is available for both platforms, but there is also a navigator that is only available for iOS[20]. It is very important to carefully decide if the application is going to be available for both of the platforms or only iOS when working with React Native, since there are many components that are only available for iOS and not Android. Canine Of Sweden is an application that is going to be available for both platforms, therefore the Navigator was used and not the iOSNavigator.

4.2.1 NavigationBar

The NavigationBar component was made through the Navigator component. Thanks to the Navigator it was easy to keep track of where the user was and could easily move the user to the next or previous page. To be able to make it easy for the user so that the user can feel in control it was important that a navigation bar that was always available for the user. If the NavigatorIos was used instead of Navigator this would not be necessary because it is already built in. The NavigationBar allows the developer to design a bar and add two buttons, one right button, one left button and a title that informs the user what view it is. In Canine Of Sweden the NavigationBar is used throughout the whole application for users that are logged in, if not the user do not have any access to the NavigationBar.

4.3 AnimatedView

AnimatedView helps the developer when they want to hide a view. The view stays hidden until a specific button is pressed to enable this hidden feature. The developer can easily give different x and y coordinates that decide how the animatedView should be shown and where it should be hidden. For example in the Lägg till hund view it is shown how the animation for choosing year/month/date view is shown only when födelsedatum is pressed and hidden otherwise. The same goes for the hamburgermenu that is used through the whole application. The user can only sees the hamburger icon first, but when the hamburger icon is pressed the menu view with the different features is shown.

5 Platforms AndroidSDK and Xcode

React Native uses JavaScript as its programming language, but when creating an application for two different platforms the code is compiled in two different software development kits, Android SDK and Xcode.

5.1 AndroidSDK

Android is one of the leading, if not the leading operative system for phones with millions applications and over 25 billion downloads [21]. Android SDK stands for Android software development kit. SDK makes it possible for app developers to create apps for Android. Android SDK provides the developers with loads of tools like debugger, libraries and much more[4]. Compare to the iOS development kit Xcode the emulator is downloaded separately. The emulator is called genymotion that is easy to use and fast. Genymotion is a plugin for the Android Studio and is compatible with all android SDK tools. In the picture below, genymotion is used as the emulator and the device (which in this case is the nexus 5) is provided. This emulator offers the ability to easily use and test the application that are created for an Android device.



5.2 Xcode

To be able to compile iOS in React Native a Mac device is required, unlike Android which can be compiled with any operative system. The development environment platform that is used for compiling iOS in React Native is Xcode. Xcode is a common tool for developing apps in the Apple world. This tool supports almost every programming language including C, Objective C, java and of course JavaScript[3]. Similar to Android SDK, Xcode has features like debugger, an interface builder and an emulator.

6 Interface

When working with applications it is important to have a clear design beforehand, otherwise it can become difficult to start the implementation. Therefore, the customer and the designer should agree on a clear and well-structured plan of what the application should look like. like Steve Jobs once said

“Design is not just what it looks like and feel like. Design is how it works.”

Applications like Canine of Sweden in particular with dog owners as target group it has to have a user-friendly design that is easy to understand. The app should not have more clicks than necessary and instead the user should be able to get to the desired function using as few clicks as possible. Accordingly, the design was set to have clear pictures and descriptions to make it easy for the user to orientate. The customer made it very clear that the user must be logged in to the app to be able to use its different functions. As seen in the picture below, the start page that shows when the user clicks on the app offers only two options to choose from; either to log in or to create an account.

The image on the start page is purposely shadowed so that these two

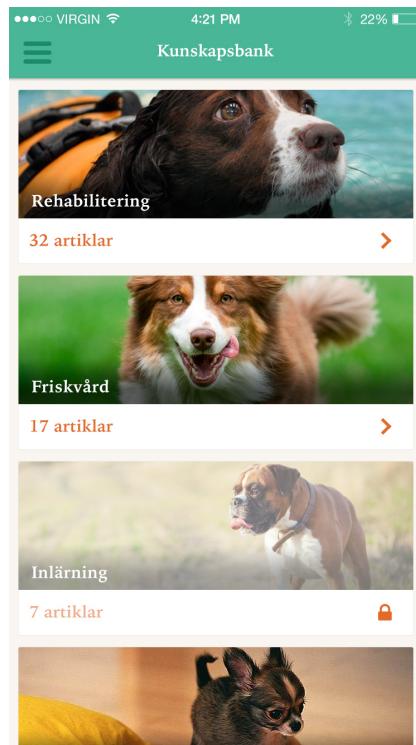


buttons gets full attention and help the user understand what to do next. Once the user has logged in, the home page appears. The home page is designed to easily access the different features and offers a short description for each of them. To distinguish the two, the name is bold and has white color meanwhile the description has a lighter color, which makes it easy for the user to separate the two. On the top-left in there is a so called hamburger menu. The reason of having the hamburger menu in almost every page is to be consistent and facilitate the process of switching between the different features. For the hamburger menu to even occur the user must be logged in. From the hamburger menu it is possible to access all the features which are:

- Start Page
- Kunskapsbank
- Övningsbank
- Min hund
- Mina listor

6.1 Kunskapsbank/Övningsbank

Kunskapsbank is where the articles are to be found. This is the most important feature, therefore the interface is clear so that the user easily can find the desired article. Every category view is divided into two parts, a renderHeader and a renderRow provided by React Native's documentation called listView. The renderHeader is the big picture as shown in picture below. The picture is shadowed which enhances the title of the category and makes it easy for the user to understand. The renderRow is a container that is shown under the picture that shows how many articles it consists of.



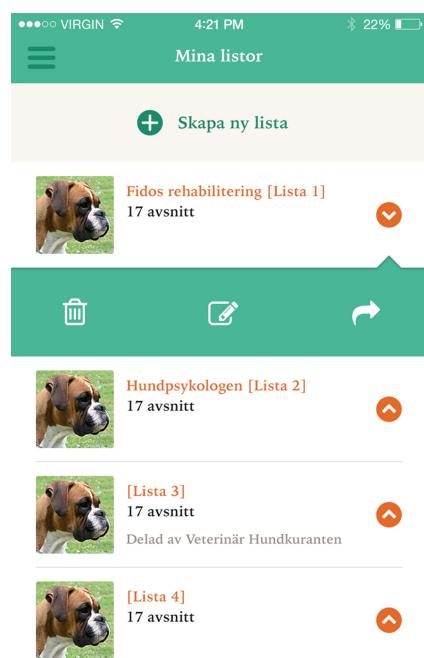
When choosing one of the categories a new view is shown with more specific categories for the different articles. Given that the app is primarily made for dog owners there is no specific age group for that demographic, which makes it important to create an application that is understandable for everyone. If a user wants an article about how to rehab their dog the user would enter “Rehabilitering”. This category consists of 32 articles, but these are not shown directly since that would mean that the user would have to go through all of them, which would not be efficient nor user friendly. The interface is constructed to make it possible to add as many categories inside a category

as required. This type of interface does not only make it easier for the user, but also gives the owners and admins of the application the possibility to divide the articles into as many categories as they like.

If the user goes deep in to a category there is a breadcrumb so that the user can locate and by clicking on the breadcrumb the user can go back to earlier locations within the app.

6.2 Mina listor

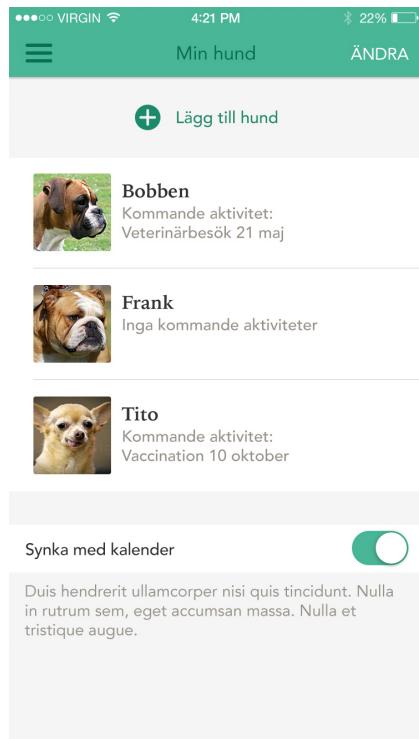
In the feature Mina listor the user can save their articles and easily access them later on. The view is a picture with a description text explaining how to add a list. The text “Lägg till text” is wrapped around a component from React Native called TouchableOpacity that makes it possible to call on a function that shows a new view for actually creating the list. TouchableOpacity is a component that is wrapped around a view and shows the user by increasing the opacity when the view is pressed on. To create a list the user has to give it a name, add a picture and a description text.



6.3 Min hund

Min hund is a feature where the user can make a profile for their dog. When adding a dog profile the owner will be able to add what year it is born, name, race and a picture of the dog.

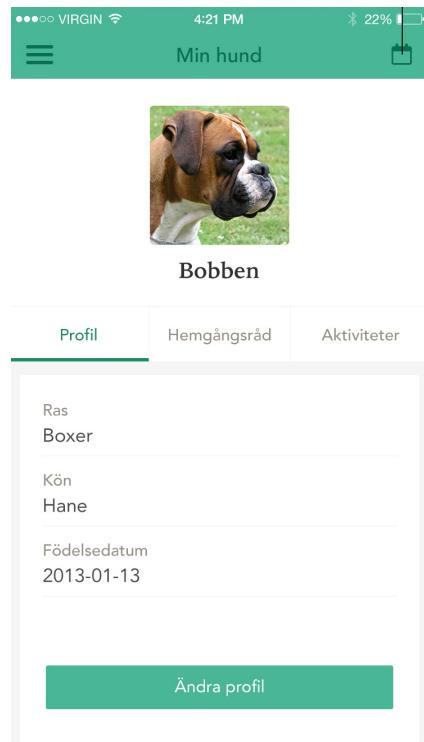
After actually creating a dog profile, the owner is able to save different



appointments for example if the owner has a veterinary appointment it can easily add a date and time to easily remember. The dog profile also allows the owner to save an exercise with the dog it can easily decide what excursive and take one from Övningsbanken and save it on their dog's profile.

With these feature it will be very easy to structure a dog's life and not forgetting an appointment with the veterinarian for example. As shown in the picture below the dog profile has three different menus. The feature where the dog profile is shown, Hemgångsråd, is where the user is able to add an exercise through övningsbanken or an activity were the user can choose an exercise of their choice. Hemgångsråd is a list with exercises that the owner has saved from övningsbanken and can be accessed from the list.

When adding an exercise to the dogs profile there are some input fields that needs to be filled in so that the owner knows when to do the exercise



and how many reps/sets the dog should do. In the pictures it is shown how easily and activity can be added through the Min hund feature. The text and the different inputs are shown very clearly to make it user friendly and easy to navigate the different views. In the second picture the activities are saved. By clicking on one of the activities the information of the activity is shown. When creating a new activity object the user fills in the type of activity and between what dates the dog is going to do the activity, as seen in the third picture.

7 Getting started with React Native

The react native framework is available via GitHub. The framework can be grabbed either by cloning it using git or download as a zip file. React Native uses Node.js, which is a JavaScript run time to build JavaScript code[5]. First install Homebrew using the instructions shown on Homebrew website and when that is done it is time to install Node.js by performing this on the terminal:

```
brew install node
```

Afterwards Homebrew is used to install watchman by typing this on the terminal:

```
brew install Watchman
```

React native uses Watchman to figure out when the code is changed so that it rebuilds accordingly. It makes it possible to build every time something is changed.

Next it is time to use NPM to install React Native CLI tool by typing this on the terminal:

```
npm install -g react-native-cli
```

This installation uses Node Package Manager to fetch the CLI so that it is installed globally. NPM is a package manager for almost every programming language and in this case JavaScript.

8 Hello World with React Native

Depending on what platform the application should be on different files are edited. In this case it is the iOS platform so the file index.ios.js file is used. On top of the file write:

```
'use strict'
```

This makes it possible to use Strict mode, which is an improved error handler and disables some Java-script language features that is not needed which make JavaScript better. After that add:

```
React = { 'react native'}
```

By writing this the react-native modules are loaded and assigns it to React. Afterwards it is time to style by writing:

```
1 var styles = React.StyleSheet.create{
2   text:{ 
3     color: 'black',
4     backgroundColor: 'white',
5     fontSize: 30,
6     margin: 80
7   }
8 );
9 }
```

By doing this a style is defined and will later on to be applied to “hello world” text. Someone who has used CSS before can see the similarity, since React Native uses CSS to style the application user interface.

Now for the application itself and defining a JSX class by typing:

```
1 class HelloWorld extends React.Component{
2   render(){
3     return(
4       <Text style = {styles.text}>Hello world</Text>
5     )
6   }
7 }
```

In the end of the file add:

```
1 React.AppRegistry.registerComponent('HelloWorld', function() {  
2   return PropertyFinderApp  
3 })
```

AppRegistry provides the root component and entry point of the application.

9 Crossing platforms iOS and Android

When it comes to the actual code they are identical, since both of them are written in JavaScript. However, there are some components that are made only for IOs because of its earlier release. This is important to take into consideration when creating an application for both platforms. There are two different files depending on the platform. React Native has a simple and good way of doing exactly that by adding the name of the platform on the file. For example, if it is a component that should only run on iOS and not Android the name of the file should be index.ios.js, and for the contrary index.android.js.

9.1 Android vs iOS design

The biggest difference is how the application is displayed depending on its platform. One has to be prepared that the view will look different depending on the platform. In contrary to desktop, a mobile a device screen is relatively small and therefore there are some common design rules for both platforms. Obviously both platforms use finger clicking and not a mouse pointer. Let us go through some differences between iOS and Android design. The NavigationBar title is left assigned on Android meanwhile in the iOS it is centered. An Android phone has a back button, meanwhile the iOS does not which is often resolved by implementing a back button in top left of the screen. The differences are many because of the device differences. When it was time to design Canine Of Sweden it was important to find a common ground for the design. For example the NavigationBar is an important component for this application and to make it look similar, the left button was only a hamburger icon in the start page of the application and after clicking on one of the features it turns into a back arrow so that the iOS users can move backwards to previous views. The Android users might find it unnecessary but these were the compromises that had to be made giving the circumstances.

10 Working Agile

When working for a customer, things can change quickly and as a distributor it is important to be flexible. The customer might want the app to look one way one day, and then another way the next. As a result of that the company has chosen an agile working method with the customer and internally within the team. Working agile requires frequent follow-ups to evaluate the process, both internally and externally, which in this case meant weekly meetings with the team and meetings every other week with the customer.

The first meeting was a lot like a workshop where a lot of ideas were exchanged back and forth, but the main focus was to get a clear view of what the design should look like. The programmer's task is to get as much of the design implemented as possible until the next meeting. The team working for the customer consisted of one programmer, one designer and one project leader. Internally, the focus of the project leader was checking how far the different members of the team have come during the weekly meetings. For example the project leader could ask the programmer to add the application on a physical device, in this case an iPhone or an Android. By doing this the project leader made sure that the app was on the right track given the customer's demands, but also based on the project leader's own experience. This method allows the project leader to test the application to see how it would feel for the user and later on give the programmer relevant feedback to make some adjustments needed. As a programmer it is important that someone from the outside can give you feedback. An important aspect, given frequent meetings with the customer, is to keep the original structure even though the customer might want to make a lot of changes along the way. Changing the design and some functional details is okay, but as a company it is important to limit the customer and work within the frames that were agreed on originally.

11 Discussion/Result

React Native is web based which gives the developer an advantage if the developer has previous experience with HTML, CSS and especially JavaScript. Not having previous web developing experience and barely some JavaScript experience it can be very frightening to use this framework, which is exactly how it felt in the beginning. After some research, practicing and some good documentation provided by React Native it did not take long before it felt natural.

Canine of Sweden is an app with many different views, which could get a little tiring and almost monotonous. Having very little experience with layout interface, most of the time was spent learning different layout techniques. Flexbox was a very important tool when it came to that topic, making sure that the elements behave predictably. In the beginning the layout was adjusted to the display when the app was tested on an Iphone 6. Later on the app was tested on an iPhone 5 display and the layout looked totally different. Flexbox insured that the elements behaved the same for different displays, which is an essential part of the development process.

For this specific app the focus lied in the view, that it looked right and it was important to let colleagues take a look and give their input to make it as user friendly as possible. Given the lack of time, the company wanted the frontend to be the main priority, which resulted in that the backend part was not part of the task. Most of the view is done, except for some adjustments needed for the both platforms.

The choice of which platform to develop the app for first was based on what platform had the least documentation and components. The reason for this is that the app should have the same functionality for both platforms. Since iOS was available on React Native before Android iOS had more components and documentation than Android and it became clear that Android was the platform to start with.

Since Android had not been available in React Native for long it was hard to find previous work task when a problem occurred, but thankfully there were colleagues to ask for advice and some documentation available, which made it all work out in the end.

Having no actual experience with interface logic and never worked as a front-end developer it was a lot of patiently learning different techniques. One of

them being working with flex box and understanding it. Because as a front end developer it is important that the interface looks good, making it easy to understand for anyone using the application and that can get very difficult.

Not so much working with JSX which was surprising given that it was lacking experience in that department to. But JSX was very similar to javaScript what differ was the HTML tags which was only positive.

When it was time to use the same code for iOS, the question remained was it possible to reuse the code from Android? The answer is yes but with a few compiling errors. What can be problematic when developing with iOS generally is that a Mac laptop is a must, after solving that the rest is easy. The only tool that is required is Xcode. Having never worked with Xcode was not a problem because the only thing that was handled in Xcode was the compiling part and the code was written on the editor Webstorm.

References

11.1 Books

- [1] Eisenman, B. 2015. *Learning React Native*. O'Reilly Media
- [2] Frankie, B. 2015. *Developing a React Edge*. Bleeding Edge Press
- [3] Fritz, A. 2015. *Xcode 6 Start to Finish*. Addison-Wesley Professional
- [4] Barry, B. 2011. *Android Application Development All-in-One For Dummies*. For Dummies
- [5] Holmes, E. 2015. *Getting Started with React Native*. Packt Publishing Ltd

11.2 Webpages

- [6] <https://code.facebook.com/posts/1014532261909640/react-native-bringing-modern-web-techniques-to-mobile/>
- [7] <http://www.thefreedictionary.com/canine>
- [8] <https://facebook.github.io/react-native/>
- [9] <http://digiday.com/platforms/mobile-apps-driving-digital-growth-5-charts>
- [10] <http://techcrunch.com/2015/09/14/react-native-android/>
- [11] <https://code.facebook.com/posts/1014532261909640/react-native-bringing-modern-web-techniques-to-mobile/>
- [12] <http://libscore.com/React>
- [13] <http://www.reactnativeandroid.com>
- [14] <https://code.facebook.com/posts/1189117404435352/react-native-for-android-how-we-built-the-first-cross-platform-react-native-app/>
- [15] <https://facebook.github.io/react-native/docs/view.html>
- [16] <https://facebook.github.io/react-native/docs/stylesheet.html>
- [17] <https://facebook.github.io/react-native/docs/listview.html>
- [18] <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

- [19] <https://facebook.github.io/react-native/docs/navigator.html>
- [20] <https://facebook.github.io/react-native/docs/navigatorios.html>
- [21] <http://officialandroid.blogspot.se/2012/09/google-play-hits-25-billion-downloads.html>