ReactJS a Viable Career Option?

Stuart Boaler

Stuart Boaler

25.04.2019

Author(s) Stuart Boaler	
Degree programme **Business Information Technology**	
Report/thesis title ReactJS a Viable Career Option?	**Number of pages and appendix pages** **39**

The objective of the report is to ascertain whether following a career path in ReactJS is a viable option. ReactJS is an open-source library built using JavaScript that was developed by Facebook, React Native is an open-source framework that uses ReactJS to create mobile applications. It sets itself apart by allowing cross-platform development that uses the native rendering APIs of the respective platforms. Obviously, this puts in an excellent position against its competition with regard to development time and performance.

The background section will give a brief history of JavaScript then go on to a general overview of what ReactJS / Native are, why they were created and will list the advantages and disadvantages that come with using the library /framework. The information present was gathered from various reputable sources from all over the web.

The results go on to show, that despite being relatively new, ReactJS / Native is already the favoured library / framework within web / mobile development community. It has shown steady growth since its inception though as to be expected is now to beginning to slow down slightly.

The conclusion is that ReactJS because of its ability for cross-platform development and its ease of use will remain a key player in the industry for the foreseeable future. Salaries will remain fairly stable with slight fluctuations dependent on supply / demand but will remain in-line with other leading frameworks / libraries.

Keywords
ReactJS, React Native, Web-Development, Mobile-Development, Career Opportunities

Table of contents

1   Introduction

With so many different frameworks to choose from it can be sometimes difficult to know where to begin. This report will attempt to shed light on one of the latest to emerge and to show that it is a viable option heading into the future. ReactJS is an open-source JavaScript library created by Facebook that is used to create user interfaces and can be used as a base for creating single-page applications. It is also the base for the React Native framework that is used in the development of cross-platform mobile applications.

As it now stands, the mobile market is dominated by Android and iOS, which are created by Google and Apple respectively. As of March 2019, 75.46% of all mobile phones were powered by Android, while 21.09% were iOS. As pictured in Figure 1. (gs.statcounter 2019.)
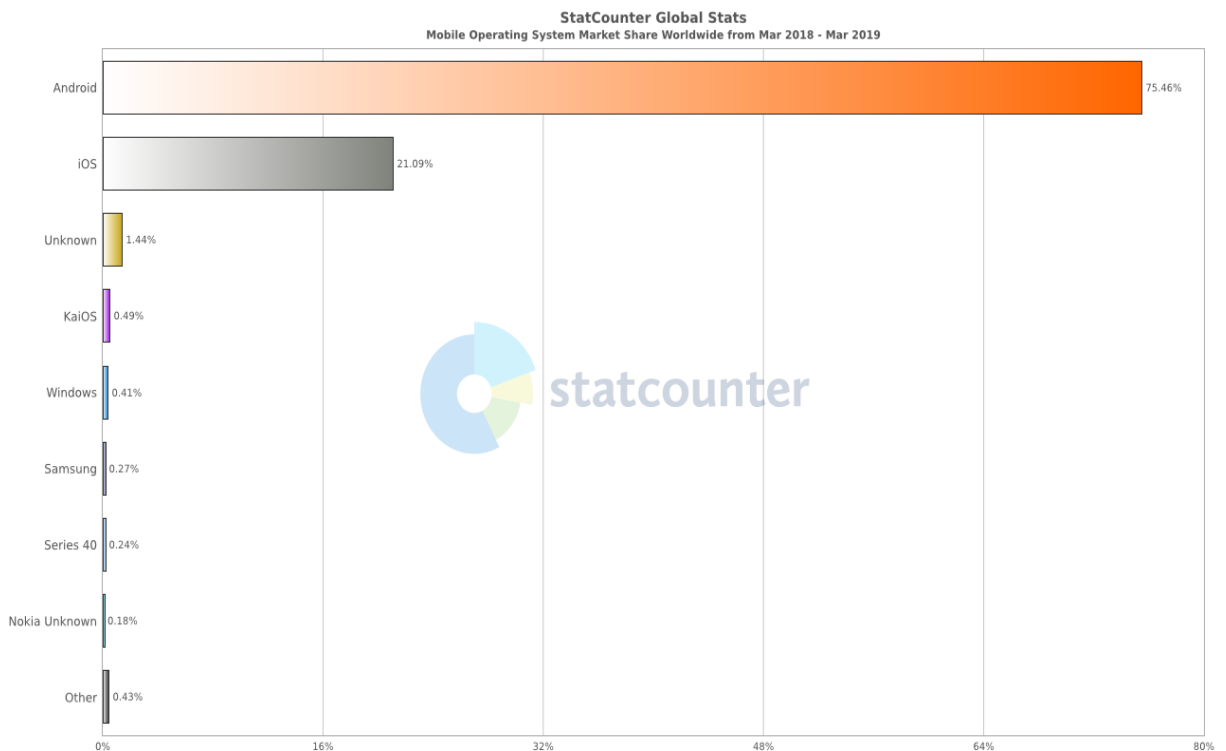
Figure 1. Mobile Operating System Market Share Worldwide.

Usually when developers create applications for mobile, they have a choice to make, do they develop natively for each specific platform on the market or, do they use one of the available cross-platform frameworks available. Obviously, using the native approach has the benefit of using the specified platforms native APIs, however, the drawback is then they have to develop separate applications for each platform. The other route involves using frameworks that use web-views, this can save time but then the application will not be using any of the native built in functions of the chosen platform.

With the introduction of React Native this choice has now been solved, launched in 2015 as an open-source JavaScript framework by Facebook. React Native can be used to create applications for both iOS and Android that use the native UI found within their respective platforms. It is also based on the ReactJS library which was also created by Facebook. (Facebook 2015.)

The author felt that due to how new the library / framework is, much could be found with regard to job availability within the United Kingdom. The author hopes that the following report will help to shed new light on what sort of opportunities will be provided to people who choose to become fluent with React. The main target audience that would benefit from such knowledge is, in the authors opinion students studying software engineering as it can be sometimes difficult to decide which area to focus on.

1.1   Goal of the Thesis

The aim of the thesis is to first take a look at ReactJS and Native, this section will be used to give an overview of the library / framework. It is hoped that readers will be able to gauge what the previously mentioned are capable of, while at the same time listing advantages and disadvantages that come with using them. The second part will look into the IT labour market within the United Kingdom. This section will go on to list various sources that show React is a key player within the industry, with comparable salaries to other frameworks and that there is a demand for React developers.

The author specifically chose this subject as although there is currently an abundance of reports on how to use React, there is very little in regard to career opportunities. With how new the library / framework is, it was felt that this would be an excellent subject for the author to focus on when considering how rapidly it has grown since release. Also, as the author is currently studying IT at Haaga-Helia and has taken courses pertaining to React, it would help the author himself in gauging the viability of choosing to become fluent with ReactJS and would help also help other students who are unsure which path to take.

This report will be focusing specifically on a library / framework within JavaScript. As such, the author assumes that those interested in the report will already posses some familiarity with programming in general.

1.1.1   Research Questions

In the report the following questions will be answered:

1. Is there a demand for ReactJS developers?
2. What are the average salaries?

Sub Questions:

1. What is JavaScript?
2. What is ReactJS / Native?
3. What are the advantages / disadvantages of ReactJS / Native?

1.2   Scope of the Thesis

The report is focused on ReactJS / Native and whether there is a demand for developers in that field within the United Kingdom. As such, all research will be based on verified sources including books, journals and trusted online web-sites.

1.3   Out of Scope

There are many different libraries / frameworks available currently for mobile / web development. However, they will not be included, this report will be focusing on ReactJS / Native. Also, as it is looking at the current job market there will be no in-depth guides in how to create applications using said library / framework.

1.4   Abbreviations and Definitions

CSS (Cascading Style Sheets) is a language that is used to set the style of a HTML document.

DOM (Document Object Model) is the programming interface for HTML and XML documents. The DOM represents the page as nodes and objects, thus allowing various programming languages to connect to the page. (Mozilla Foundation, 2018.)

MVC (Model View Controller) is an architectural pattern that is used to separate applications into three components, the view, model and controller.

2   Theoretical framework

The ReactJS library is built using the JavaScript language and React Native is a cross-platform framework for making mobile applications that uses ReactJS. This section will be devoted to first giving a brief history of JavaScript and why it was created as it is always good to know from whence something came. Then, we will move on to looking into ReactJS and React Native, look at what they are, why they were created and what are the advantages and disadvantages of using them.


2.1   JavaScript a Brief History

The year is 1992, a company called Nombas begins to develop an embedded scripted language by the name of C-minus-minus (Cmm). The idea was to create a scripting language that could replace macros, while remaining similar to C and C++ so that existing developers would find it easy to make the switch. It was originally packaged as a shareware product under the name CEnvi, which would expose developers to the power of such languages. Due to a negative connotation with the name, Nombas decided to change it to ScriptEase. It would go on to become the main driver for Nombas products. (Zakas 2005, 1.)

As surfing the Web became more popular, so too did the demand for client-side scripting languages. At this time most Internet users connected to the web via 28.8 kbps modems, as web pages were growing in size and complexity this would become a problem. Imagine filling out a simple contact form, hitting the send button, then having to wait 30 seconds only to be told you forgot to fill in a field. Due to this, Netscape, who at the time were one of the leaders in cutting edge technological innovation, began to consider the development of a client-side scripting language to handle simple processing. (Zakas 2005, 1.)

A developer by the name of Brendan Eich, who was working for Netscape began to develop a scripting language by the name of LiveScript, this would tie-in with the release of Netscape Navigator 2.0 in 1995. The intention was to use it in both the browser and on the server. To complete LiveScript in time for release Netscape joined forces with Sun Microsystems to form a development alliance. Just before the release of Navigator 2.0, the name LiveScript was changed to JavaScript. This renaming was made in order to capitalize on Java as the new internet buzz-word of the time. The renaming was a success and JavaScript would go on to become a foundation of the Web. (Zakas 2005, 1.)
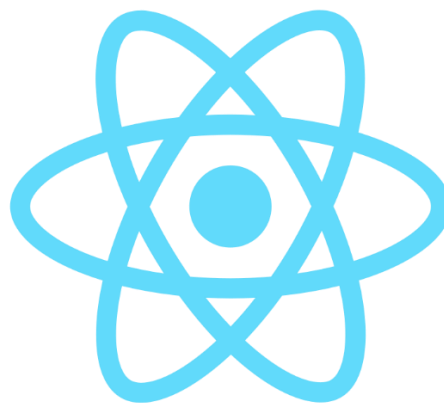
2.2 What is ReactJS?

ReactJS is a relatively new library that was built using JavaScript and created by Facebook and Instagram. The idea behind it was to create a library that could build user interfaces that have the ability to respond to users' actions along with creating and maintaining states. (Sengupta, Singhal, & Corvalan 2016, 1).

Among developers ReactJS is considered to be the V in MVC. As such, it encourages people to build reusable components and rethink their UI and best practices. As the need for performance and portability among user interfaces has increased due to Internet ready devices and rapid development cycles of projects. So too has the need for a library that can enable code to increase in performance and quality. (Sengupta, Singhal, & Corvalan 2016, 2.)

React tries to solve the problem of the view layer by creating abstract representations of the views. It does this by breaking down parts of the view into components. These components then encapsulate both the logic needed to control the display of the view and the view itself. (Vipul & Prathamesh 2016, 1.)

One of the main reasons for creating

According to Sengupta, Singhal, & Corvalan (2016,2), "ReactJS encourages the best practices shown here:"

- Following a pattern
- Separating concerns
- Splitting your UI into components
- Communication between components wit one-way dataflow
- Use of properties and states appropriately

2.2.1 Who Uses React?

As an emerging library that is used to build web UI components ReactJS has taken the development world by storm, it is being used by some of the largest companies on the web. (Sengupta, Singhal, & Corvalan 2016, 1.)

The following is a list of some of the largest companies to use ReactJS:

- Atlassian
- AirBnB
- Cloudflare
- Dropbox
- Facebook
- Instagram
- Khan Academy
- Netflix
- PayPal
- Reddit
- Yahoo

2.2.2   Correlated Technologies

Now that we have seen what React is and who is using it, in this section we can go on to see how React fits into web development and what other technologies React ties into. Nothing works inside a vacuum and this holds true for React, it is part of an ecosystem of technologies that is being used by many developers worldwide, as can be seen in Figure 2. (Stackoverflow 2019a.)
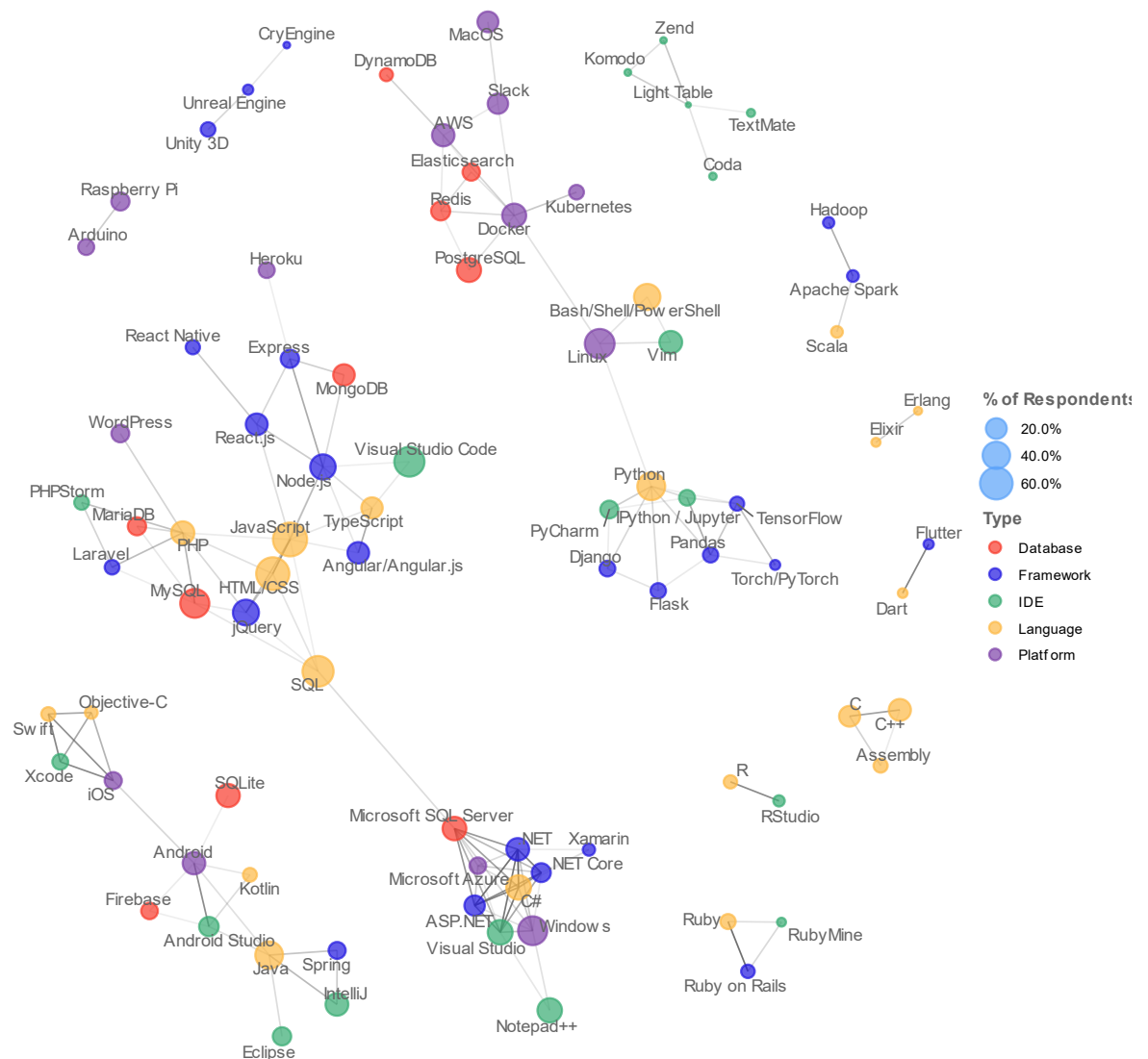

Figure 2 How Technologies Are Connected

2.2.3　Why was React Created?

The creation of React came about in response to problems that can occur when creating applications using JavaScript. These problems begin to become more evident as an application grows in size and complexity. The reason for this of course, is that JavaScript was never created for the purpose of creating large applications. It was originally designed for the purpose of enabling web pages to be more interactive. (Masiello & Friedmann 2017, 8.)

Possibly one of the largest concerns is how unpredictable shared mutable states can become when using JavaScript. This is because objects will be passed to the states and views of the application, when this happens objects can easily become mutated by a rogue view. Once this occurs, bugs in the application can become incredibly difficult to find. And again, this can be multiplied exponentially as an application grows in size or the team size increases. (Masiello & Friedmann 2017, 8.)

Then, there is the problem of the Document Object Model itself and the effect it has on the efficiency of an application. It is known as being slow, obviously with a small application this is not a problem as updates to the DOM are infrequent. Obviously, as the size and complexity of an application increases so to does the need to re-render the DOM. In the case of React this would be unworkable as it is declarative, so any time there is a change in state it has to re-render. Due to this the React developers had to come up with a solution. (Masiello & Friedmann 2017, 8.)

The React developers found the solution to this problem by using something that is known as the virtual DOM. They did this by making it as the name would suggest a virtual representation of the DOM. Any alterations that happen will first be done in the memory, after this React will compare the virtual with the actual and conduct only the minimum number of changes. This of course can cause a large increase in performance as the entire DOM does not require re-rendering. (Masiello & Friedmann 2017, 9.)

The Virtual DOM acts as a sort of layer between how things are supposed to look, and the work being carried out to render all of this on to the page. To minimise the impact of having to re-render the whole DOM React works in the following way pictured below in Figure 3. (Eisenmann 2016, 26.)
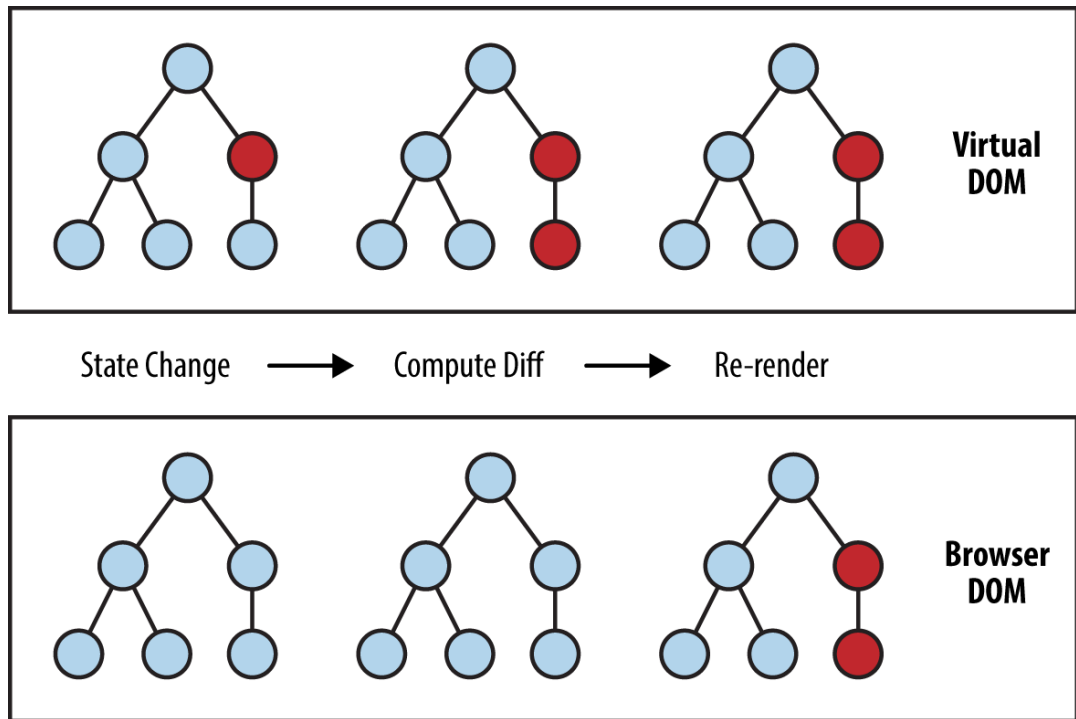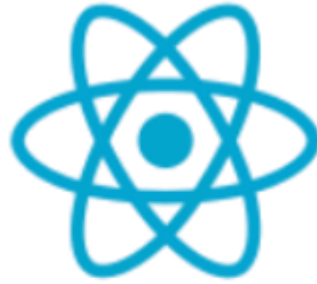
Figure 3. Performing calculations in the Virtual DOM limits re-rendering in the Browser DOM

2.3 What is React Native?

React Native is an open-source JavaScript framework created by Facebook that is used for creating mobile applications. As such it is built using common web technologies, this gives the added bonus of not having to learn any new languages when jumping in to React Native. (Masiello & Friedmann 2017, 18.)

The truly great part of React Native lies in the fact that it gives developers the opportunity to develop mobile applications for both Android and iOS that are actually natively rendered. Also, the fact that React Native is built using the ReactJS library which is focused on building user interfaces, means it makes it easier for web developers to get involved in the development of mobile applications. Plus, most of the code written can be shared across both platforms, thus cutting down the time it takes to develop an application

As with ReactJS, React Native is written in JavaScript and the XML-esque mark-up JSX. Then the React Native bridge will use the rendering APIs that are native to the respective device, Objective-C for iOS and Java for Android. This causes an application to render using actual mobile UI components as opposed to web-views. (Eisenmann 2016, 17.)

2.3.1 What is JSX

JSX is a syntax that is used to extend JavaScript and make it more in similarity to XML. Though it is similar in many ways to HTML it still remains different, for instance in HTML tags are used, however, when using JSX tags actually become JavaScript objects. As one would think it is used to create user interface components within ReactJS. (Sengupta, Singhal, & Corvalan 2016, 13.)

In React Native all of the views are written using JSX, this allows both the mark-up and the JavaScript to combined into one file. This helps as there is no browser, so combining all styles, mark-up and logic into a single file helps out massively. (Eisenmann 2016, 16.)

In the figures below will be shown the differences between a pure JavaScript component and one written using JSX. The first will show a pure JavaScript component. Figure 4. (Eisenmann 2016, 16.)

```
var HelloMessage = React.createClass({
  displayName: "HelloMessage",

  render: function render() {
    return React.createElement(
      "div",
      null,
      "Hello ",
      this.props.name
    );
  }
});

React.render(React.createElement(HelloMessage, { name: "Bonnie" }), mountNode);
```

Figure 4. Pure JavaScript Component

Then Figure 5. (Eisenmann 2016, 16.) will show what it looks like when using JSX.

```
var HelloMessage = React.createClass({
  render: function() {
    // Instead of calling createElement, we return markup
    return <div>Hello {this.props.name}</div>;
  }
});

// We no longer need a createElement call here
React.render(<HelloMessage name="Bonnie" />, mountNode);
```

Figure 5. JSX Component

2.3.2   Advantages of React Native

One of the largest advantages that comes with using React Native comes in the form of how it renders. The Native in React Native is not there by chance, the name was chosen as it uses the native rendering APIs of the respective platform it is deployed on. This of course gives it an advantage over other cross-platform frameworks on the market as they use web-views for the rendering. Cordova and Ionic for instance use JavaScript, HTML and CSS to render web-views. While this is an option, it is not the most favourable when it comes to performance and on top of that when using web-views you don't have access to the native UI elements that are present on the platform you are working on. This of course can lead to user interfaces that feel slightly off when being used. (Eisenmann 2016, 18.)

Below is an example of how React Native differentiates itself from WebView rendering. Figure 6. (Altexsoft 2018.)
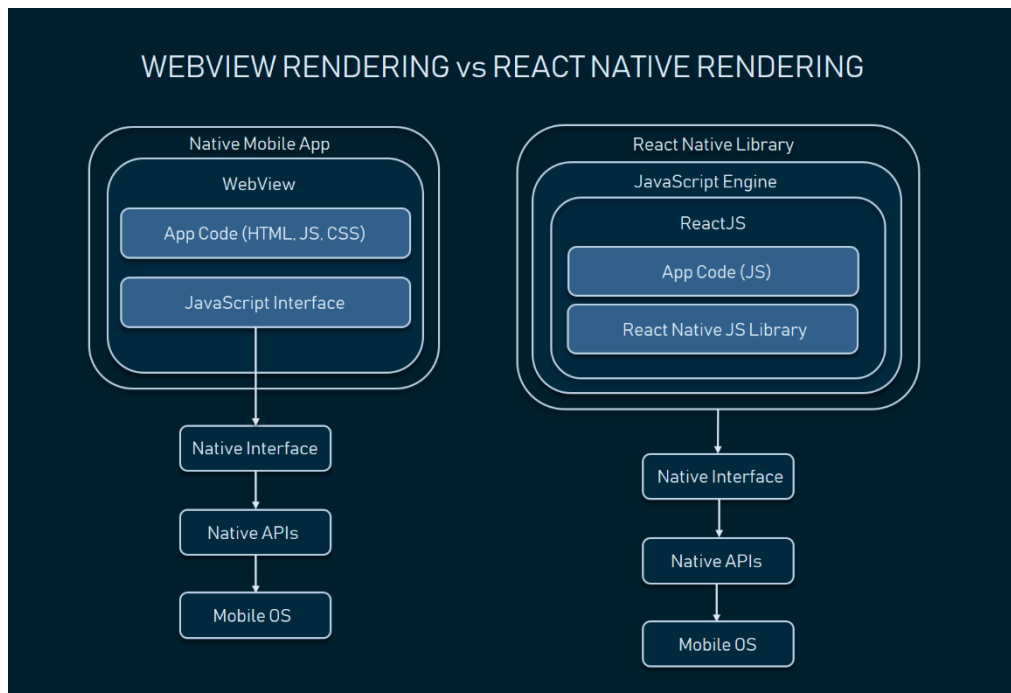
Figure 6. WebView Rendering vs React Native Rendering

As can be seen in the previously listed figure, React Native allows developers to over-come the problem of web-views by actually translating the mark-up into the native UI elements associated with the chosen platform. Another of the advantages comes in the form of how React works separate from the UI thread thus allowing high performance without sacrificing capability. As with React the React Native update cycle follows the same path, whenever there is a change to the props or state React Native will re-render the view.

Though the main difference with React Native is that it will render with native UI libraries instead of HTML and CSS mark-up (Eisenmann 2016, 18.)

2.3.3   Disadvantages of React Native

Of course, the library is not without disadvantages of its own. One of the main problems arise from how new the library is itself. Because of this, documentation for the library can still found to be lacking in some areas. Then there is the problem that not all the features for the various platforms are fully supported.

The reader also has to taken into account that due to how new it is, best practices are still being discovered and that as React Native adds another layer to development it can cause debugging to become more difficult. (Eisenmann 2016, 22.)

3   Empirical

This section of the report is of the upmost importance, it is here that all information regarding the report will be covered. This chapter will cover in detail all of the collected data from the chosen organizations. The first part will use data collected from the Stackoverflow developer survey of 2019, this data will be used to show developer profiles, what are the most popular technologies including frameworks / libraries and average salaries by developer type.

Then the second part will use data taken from IT Jobs Watch UK to paint a picture of the current needs for ReactJS / Native developers in the UK labour market, it will also give expected salaries, number of positions available, job vacancy trends and salary trends.

3.1   Research plan

For any project one of the most critical phases is done before anything else, during the planning phases. This holds true across all fields and industries. The plan for this report started to formulate during a course taken at Haaga-Helia (Multidisciplinary Software Programming). During this course ReactJS / Native were used extensively to create a mobile application in collaboration with Futurice Oy.

The first part of the plan was in deciding what part of ReactJS / Native should the report focus on. As there are already a number of reports pertaining to the use of ReactJS, it was decided to focus on a different area that the author feels is important to students studying software engineering. Once the area was decided the author then went on to planning what should be included in the report. As the target audience are upcoming software developers it was felt that only a brief overview of the library / framework would be needed, where a history would be given and then advantages, and disadvantages would also be listed.

Finally, the author then had to decide what sort of research method would be used. The author decided that to accomplish the report the form of data best employed would be secondary data. The reason for choosing this method of data is simply the sheer size of the development community, the survey done by Stack Overflow is the largest of its kind. The number of respondents covers a far larger sample size than the author could ever hope to accomplish, and it is conducted by the most respected developer website in the world.

After reading "Research Methods" by Nicholas Walliman. it became clear that the use of secondary data holds large advantages, especially for students who do not have the resources available to conduct large scale surveys.

Walliman (2011, 78) states that students who are conducting projects with a shorter time frame cannot possibly hope to retrieve data comparable to that done by a team of experts that is backed up by large amounts of funding. And that all the provided data is available to the public, making any findings easily verifiable. Thus, making the use of secondary data an excellent option for this kind of report.

3.2 Data Gathering

The first source of information was "Research Methods" by Nicholas Walliman. The book gives an excellent introduction into how to conduct research, what sort of methods to use and how to find and review literature. Once read, the research itself began by searching for respected sources via the web, books and reports. The main source of information for the theoretical side where JavaScript, ReactJS and React Native were covered was gained from the various books listed in the references section.

Moving on to the usage statistics and job availability, the main sources of information were gathered from respected sites on the web. These include: Stackoverflow and IT Jobs Watch UK.

3.3 Usage Statistics

As a large part of the report is given over to showing how ReactJS is viable as a career option this section is especially important. A large portion of the statistics are gathered from the Stackoverflow Developer Survey that is conducted yearly and was completed this year by nearly 90,000 respondents. It gives insights into what technologies are the most popular amongst both professional and upcoming developers.

The second source focuses on the UK job market, IT Jobs Watch provides excellent information on the technology labour market in the United Kingdom, the service provides in-depth statistics on national, regional and local labour markets.

3.4 Theoretical Data

All information in the theoretical section of this report was gained from various printed books that are written by respected authors. The main sources for this information were "Learning React Native" by Bonnie Eisenman and "Mastering React Native" by Masiello, E & Friedmann, j. This was further supplemented with additional information from "Professional JavaScript for Web Developers" and "Getting Started with React" by Nicholas Zakas and Sengupta, Singhal, & Corvalan respectively. These books gave excellent insights firstly into JavaScript and then what React is capable of, why it was created and also what are the limitations of the library / framework.

3.5 Stack Overflow Developer Survey 2019

Stack Overflow, first founded in 2008, is the largest and most trusted online resource in the world for both professional developers and enthusiasts alike. It is used by people to learn, share and advance their career paths. It garners more than 50 million unique visitors each month who come to find help in solving coding problems, developing new skills and finding job opportunities.

The annual developer survey is the largest survey of its type for developers around the globe. This year close to 90,000 developers took part in the survey to provide the data that will be given below. the largest responses came from America with 23.64%, India with 10.23%, Germany with 6.62% and the United Kingdom with 6.47%, while Finland provided 0.62% of the total of respondents. The countries where respondents were from can be seen below in Figure 7. (Stackoverflow 2019b.)

Figure 7. Global Survey Respondents

3.6   Survey Data

In this section can be seen all the demographics of those who have taken part in the de-
veloper survey this year. It gives insight into the locations of developers around the world,
race and ethnicity and also gender.

3.6.1   Location of respondents

The target of the survey was software developers around the globe, with a total of 88,883
respondents from 179 countries. To qualify respondents had to spend a set amount of
time on the survey and answer a set amount of questions. The questions that respondents
received were based on previous answers. The location of the respondents can be seen
in the table below. (Stackoverflow 2019m.)

Table 1. Qualified Responses Worldwide

Location	Amount
Europe	36,073
North America	25,526
Asia	18,273
South America	3,459
Africa	2,850
Australia/Oceania	2,434
Other	268

3.6.2   Race and Ethnicity

Of the 63,991 responses that were received amongst professional developers the following figures listed in the table below were found. (Stackoverflow 2019n.)

Table 2. Race and Ethnicity

Race & Ethnicity	Percentage
White / European	72.5%
South Asian	11.1%
Hispanic/Latino/Latina	7.3%
East Asian	5.3%
Middle Eastern	4.5%
Black / African Descent	3.1%
Multiracial	2.3%
Biracial	1.3%
Native American, Pacific Islander, Indigenous Australian	0.8%

3.6.3   Gender

During the survey people were also asked about their gender identity, from the 70,717 responses of professional developers the figure shown in the table below were attained. (Stackoverflow 2019n.)

Table 3. Gender

Gender	Percentage
Male	92.1%
Female	7.5%
Non-binary, genderqueer, gender non-conforming	1.1%

In this section the report will focus on developer roles, experience, education and demographics. The first part is developer type, It is important to know what positions developers fill in the marketplace, during the survey it was found that 51.9% of the respondents identified as full-stack developers while 18.1% identify as mobile developers and the most combinations are back / front-end and full-stack developers. Figure 8. (Stackoverflow 2019c.)
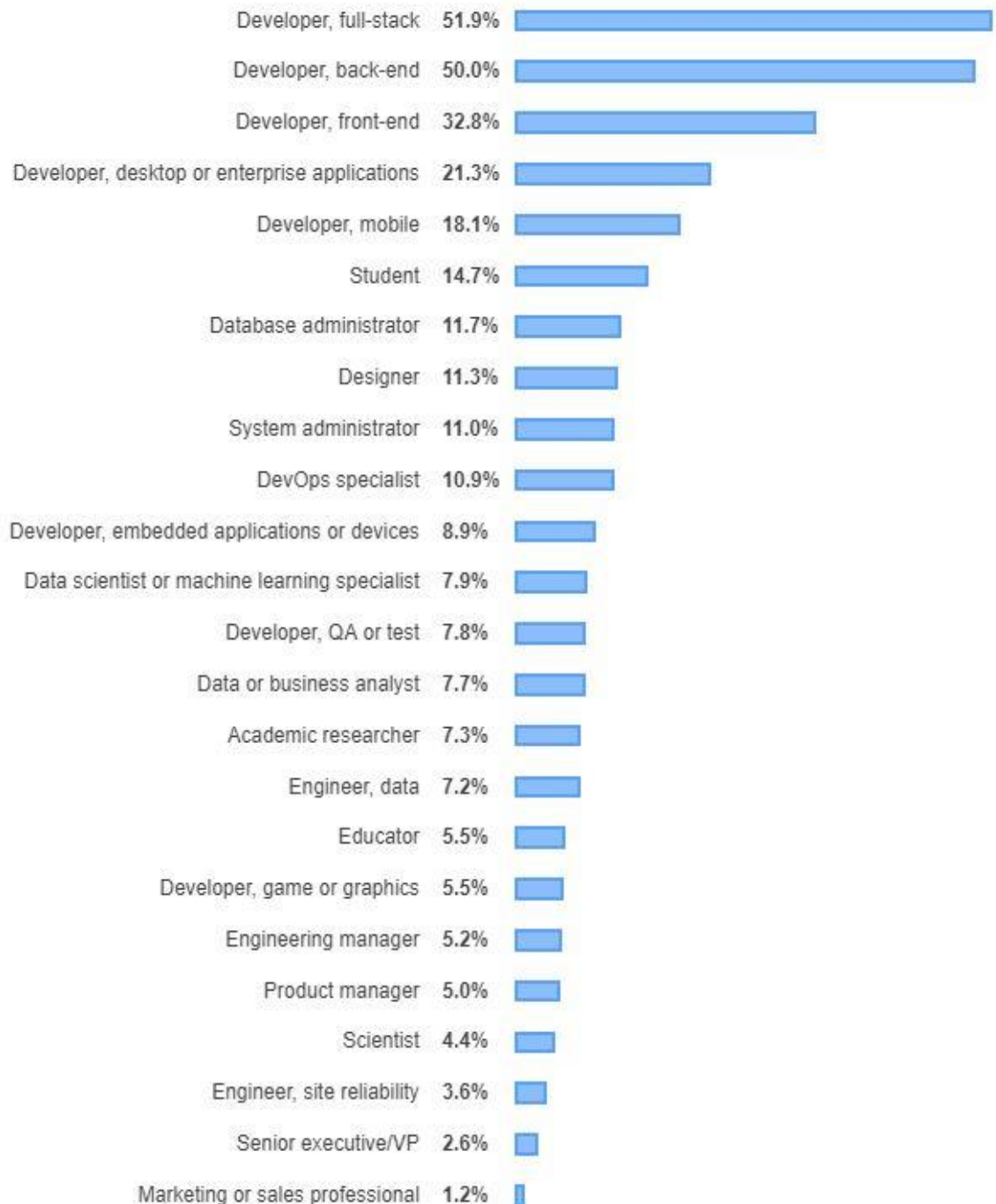
Developer Type	Percentage
Developer, full-stack	51.9%
Developer, back-end	50.0%
Developer, front-end	32.8%
Developer, desktop or enterprise applications	21.3%
Developer, mobile	18.1%
Student	14.7%
Database administrator	11.7%
Designer	11.3%
System administrator	11.0%
DevOps specialist	10.9%
Developer, embedded applications or devices	8.9%
Data scientist or machine learning specialist	7.9%
Developer, QA or test	7.8%
Data or business analyst	7.7%
Academic researcher	7.3%
Engineer, data	7.2%
Educator	5.5%
Developer, game or graphics	5.5%
Engineering manager	5.2%
Product manager	5.0%
Scientist	4.4%
Engineer, site reliability	3.6%
Senior executive/VP	2.6%
Marketing or sales professional	1.2%

Figure 8. Developer Type

3.6.4   Experience

One of the major obstacles for people looking for new avenues of employment is experience, during the survey people were questioned on how many years ago they learnt to code and how many years have been spent coding professionally.

As can be seen in Figure 9. (Stackoverflow 2019d.) There is currently a broad spectrum of experience amongst developers with the largest figure 41% of developers who have less than 5 years' experience spent in a professional capacity.
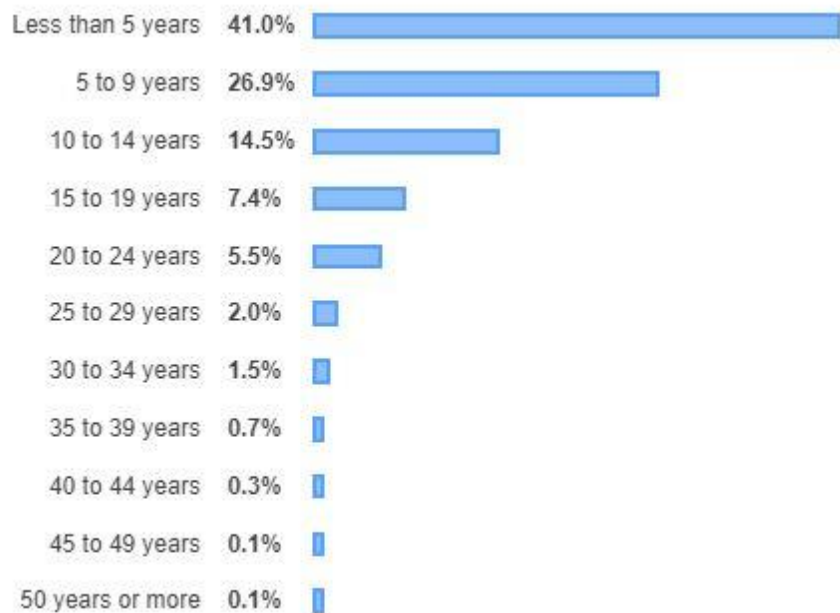
Less than 5 years	41.0%
5 to 9 years	26.9%
10 to 14 years	14.5%
15 to 19 years	7.4%
20 to 24 years	5.5%
25 to 29 years	2.0%
30 to 34 years	1.5%
35 to 39 years	0.7%
40 to 44 years	0.3%
45 to 49 years	0.1%
50 years or more	0.1%

Figure 9. Years Coding Professionally

3.6.5   Education

For many, education is an important steppingstone on the path to a successful career. While some people manage to become professional developers without attaining any form of degree, for the most part the developers who answered the questionnaire have attained one. The largest number of respondents (45.3%) have attained the equivalent of a bachelor's degree. However, as can be seen in Figure 10. (stackoverflow 2019e.) it is not that rare for people without to also attain positions as developers with a total of 27.8% of the respondents whose education is below the level of bachelor.
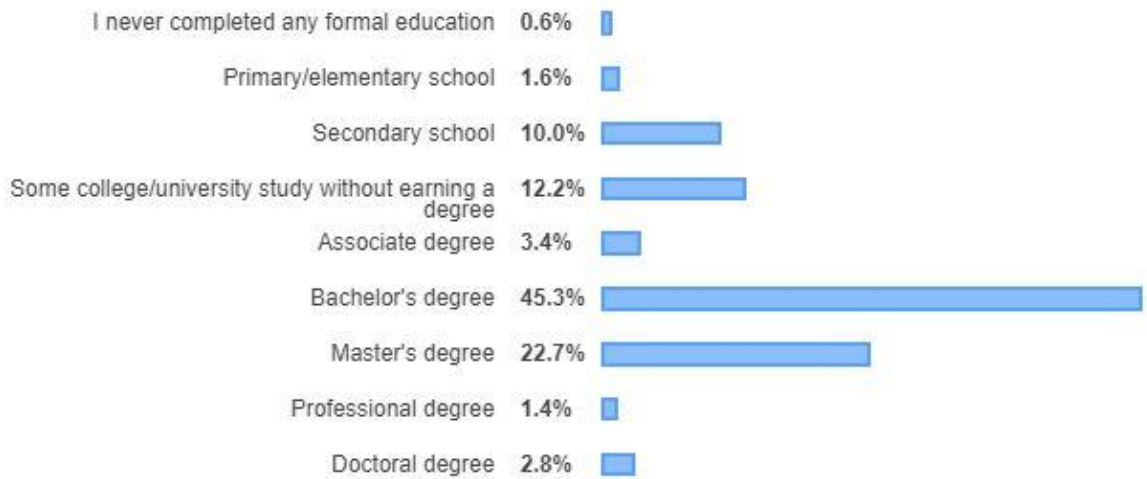
Figure 10. Educational Attainment

Continuing on from the previous, the report will now look at what people who studied at a university chose as their major. Out of those questioned, the largest number with 62.4% majored in either computer science, computer engineering or software engineering with those in second place having majored in another type of engineering discipline with 8.2% as can be seen in Figure 11. (stackoverflow 2019f.)
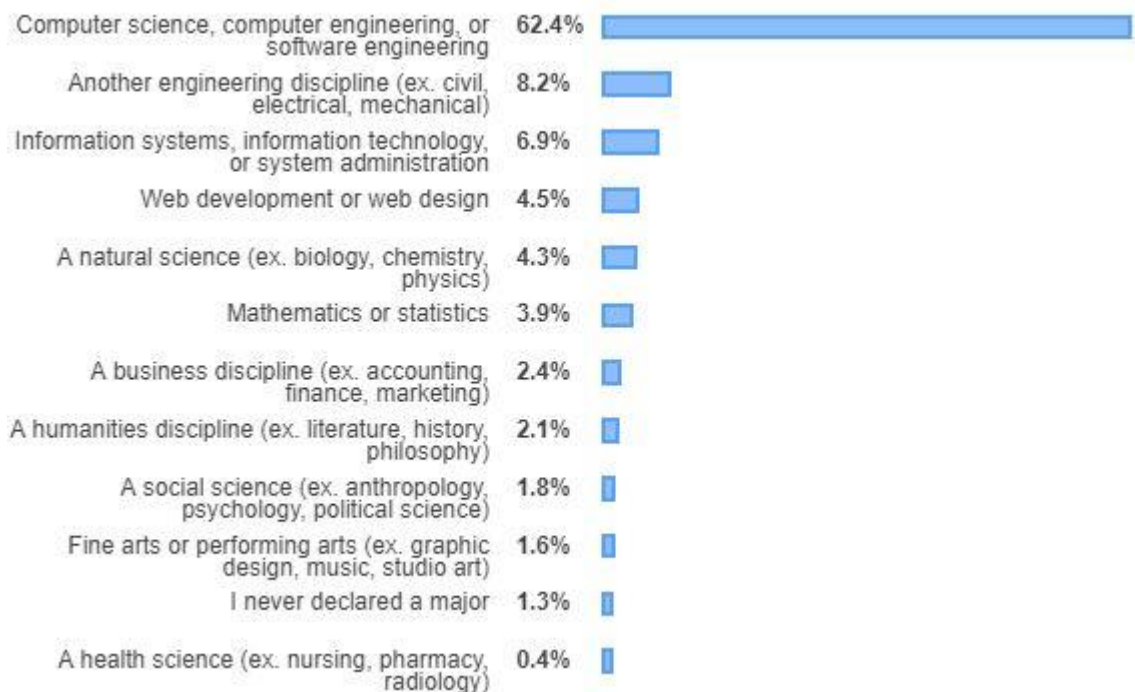

Figure 11. Undergraduate Major

Due to the nature of software development and how it is constantly evolving, the ability to teach oneself is paramount. The survey has found from 70,092 responses that developers are lifelong learners with a total of 86.8% stating that they have taught themselves a new language or framework and a further 60.1% stating they have taken online courses in software development as can be seen below in Figure 12. (stackoverflow 2019g.)

Figure 12. Other Types of Education

3.6.6 Technology

The first thing to be looked at in this section is the most popular programming languages in use today. Amongst professional developers the survey found after 72,525 responses that JavaScript still holds the crown for most popular language, with 69.7% of those questioned indicating they use it. Though as can be seen in the results below python is rising in the ranks and is set to be the fastest growing programming language today. Figure 13. (stackoverflow 2019h.)
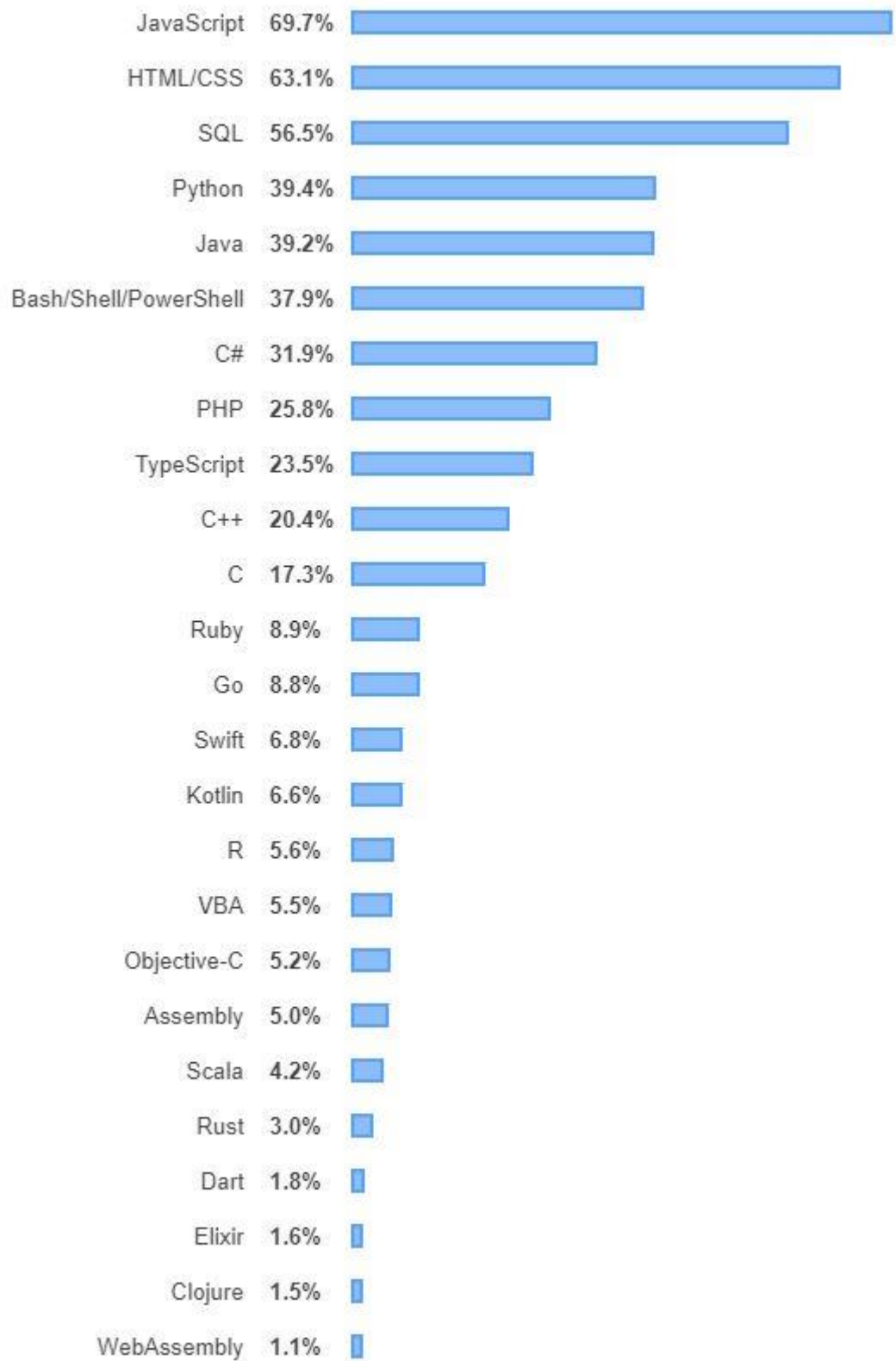
Figure 13. Programming, Scripting and Markup languages

Now that we have seen that JavaScript is still the most popular language used today, it is time to investigate the ReactJS library itself. In the following results developers were questioned on what frameworks they currently hold to be the most loved, dreaded and wanted frameworks, through this we will see what developers think of both ReactJS and React Native.

After the survey the results found that ReactJS is the most loved framework of 2019 by those who are currently working with it and who wish to continue to do so just beating vue.js with 74.5% of the vote. Figure 14 (stackoverflow 2019i.)
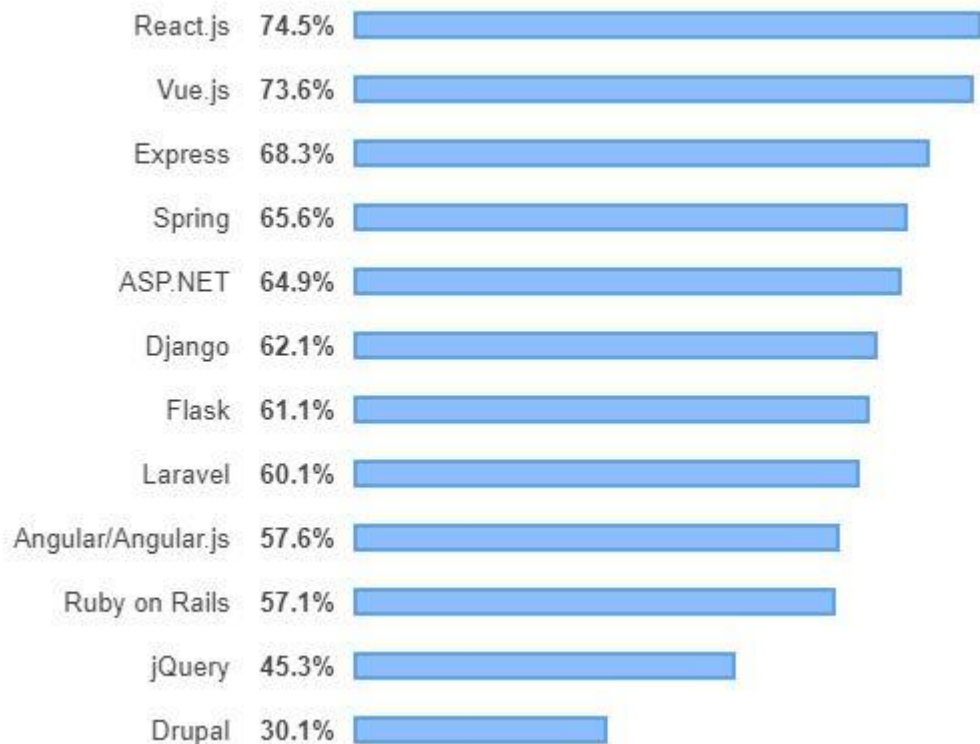

Figure 14. Most Loved, Dreaded and Wanted Web Frameworks

ReactJS was also found to be the most wanted amongst developers who are not currently working with the library but wish to do so in the future, once again beating vue.js with 21.5%. Figure 15 (stackoverflow 2019i.)
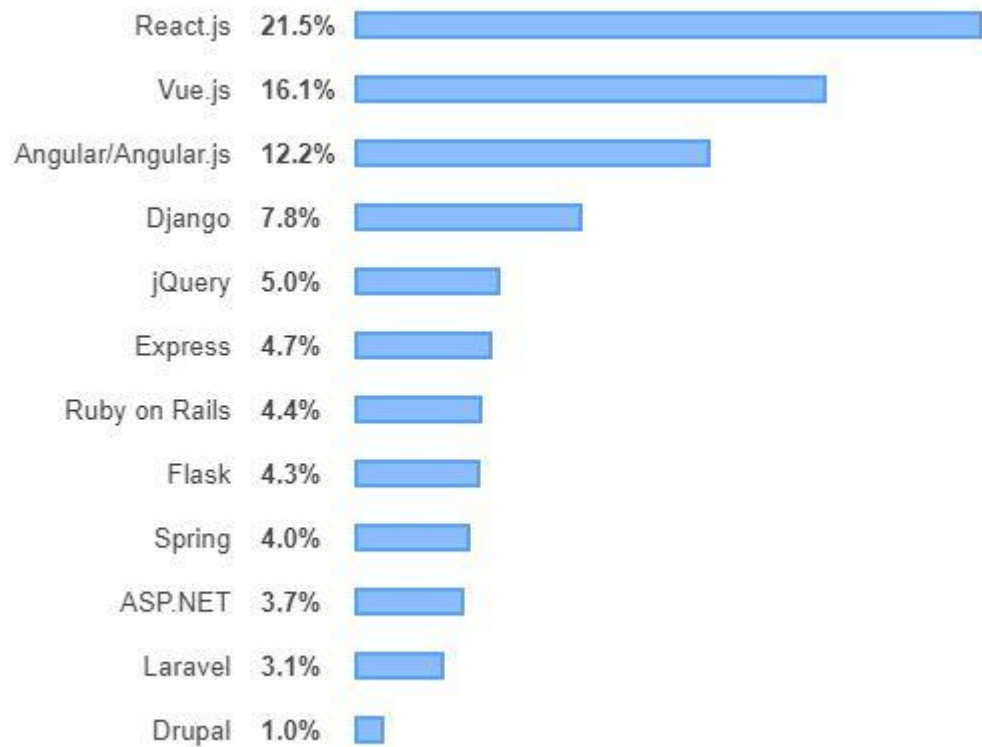

Figure 15. Most Loved, Dreaded and Wanted Web Frameworks

3.6.7 Salaries

The final part in this section will be used to show the results of respondents on salaries globally. The first part will how salary is affected by experience as is shown below in Figure 16 (stackoverflow 2019j.)
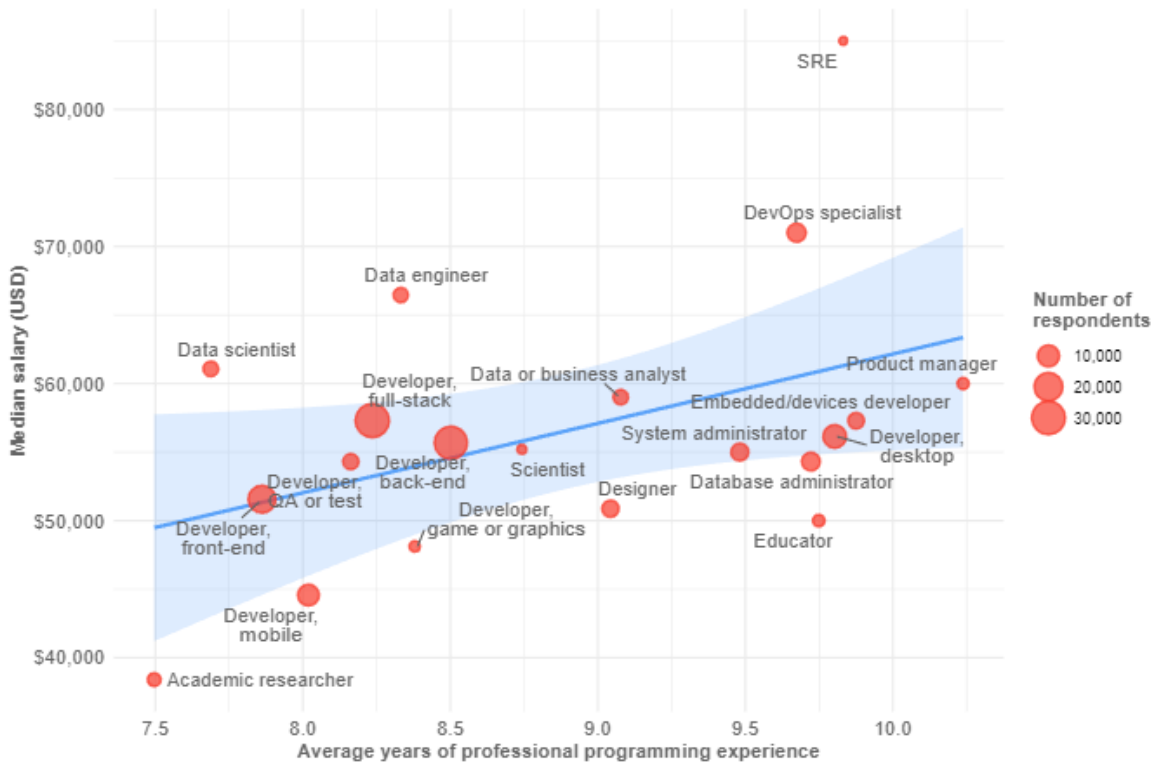
Figure 16. Salary and Experience by Developer Type

Next shall be shown salary and experience by language, as can be seen in Figure 17 (stackoverflow 2019k.)
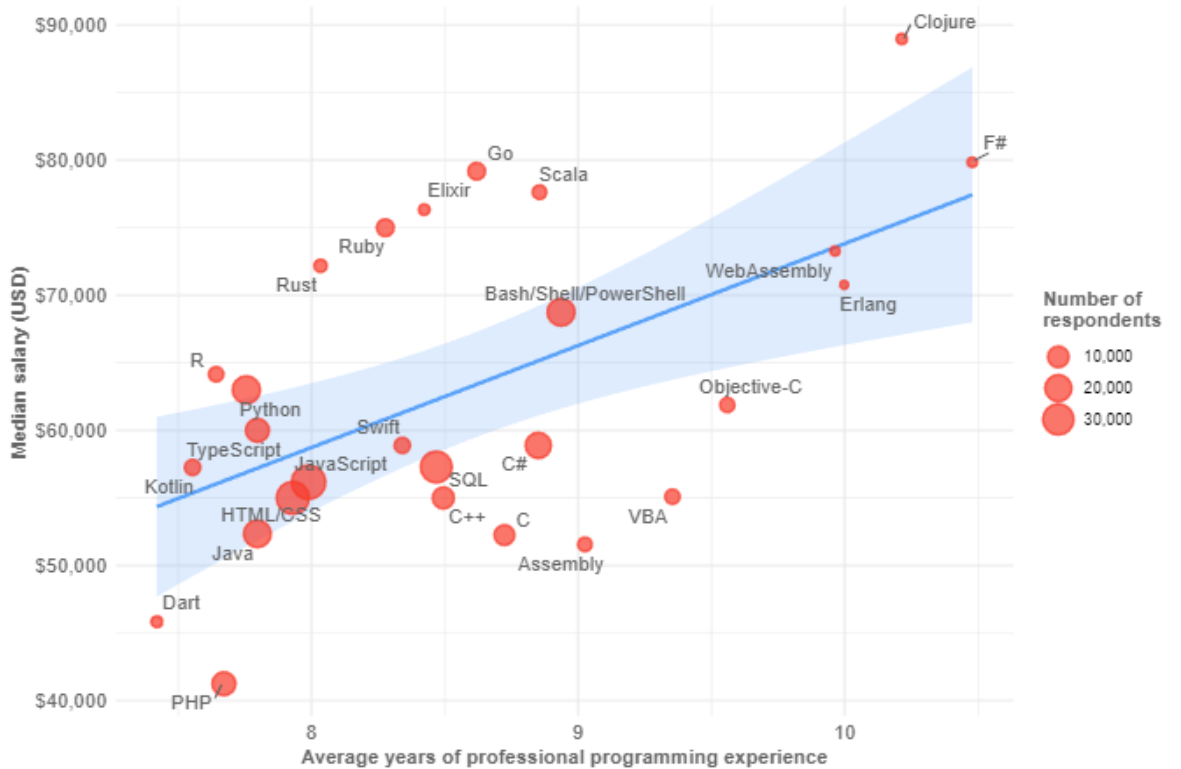

Figure 17. Salary and Experience by Language

The final piece of data from the survey on this section goes on to show working hours per week by country. Figure 18 (stackoverflow 2019l.) gives the following results.
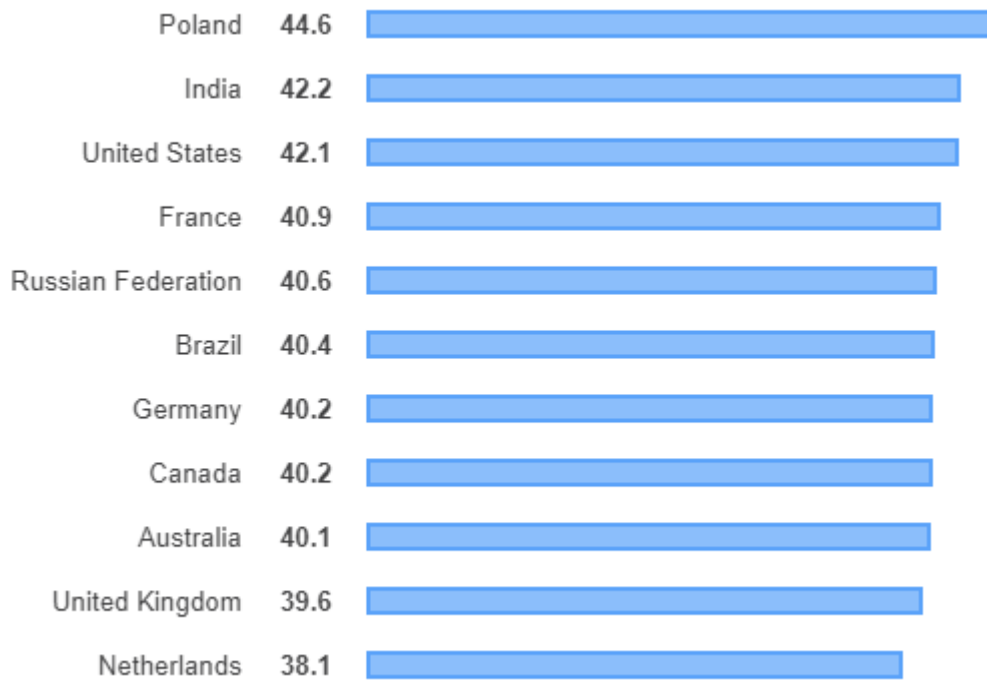

Figure 18. Hours Worked Per Week

3.7 IT Jobs Watch UK

IT Jobs Watch is a service that offers insight into the current information technology labour market in the United Kingdom. The aim of the service is to provide an accurate map of conditions in the UK IT job market. (ITJobsWatch, 2019a.)

The service offers national, regional and local statistics for the following areas:

- Salary benchmarking
- Discovering IT skill sets
- Job role & IT skill trends
- Job role skills requirements

In this section the report will be focusing on data gathered from the UK labour market to help the readers understand, job availability and expected salaries.

3.7.1   ReactJS

The data below provides statistics on permanent job vacancies where skills in React are a requirement. It also includes a benchmarking guide on salaries offered that have required React over the past 6 months with comparisons of the same period from the last 2 years. Figure 19. (ITJobsWatch 2019b.)

React.js UK			
Location UK	6 months to 29 Apr 2019	Same period 2018	Same period 2017
Rank	51	68	109
Rank change year-on-year	▲ +17	▲ +41	▲ +191
Permanent jobs citing React	7,251	7,546	4,769
As % of all permanent IT jobs advertised in the UK	4.95%	4.27%	2.81%
As % of the Libraries, Frameworks & Software Standards category	12.72%	10.50%	6.96%
Number of salaries quoted	6,038	6,612	4,211
UK median annual salary	£55,000	£52,500	£53,000
Median salary % change year-on-year	+4.76%	-0.94%	+0.95%
10th Percentile	£32,500	£32,500	£32,500
90th Percentile	£90,000	£85,000	£85,000
UK excluding London median annual salary	£45,000	£45,000	£42,500
% change year-on-year	-	+5.88%	-5.56%

Figure 19. React Jobs

Following on from that is data that provides job vacancy trends in the UK for the past five years. As can be seen in the graph permanent job availability has continued to grow since its initial release though has started to slow down slightly in the last six months, with contract positions actually taking a quite profound dip in the last two months. Figure 20. (ITJobsWatch 2019b.)

Figure 20. React Job Vacancy Trend

Next is the available data that provides both median and percentile figure for salary trends quoted for IT jobs that require React for the last five years, as can be seen while median and 10th percentiles have remained similar there has been a slow increase in the 90th percentile figures indicating that as people become more experienced with React so to do the salaries increase. Figure 21. (ITJobsWatch 2019b.)
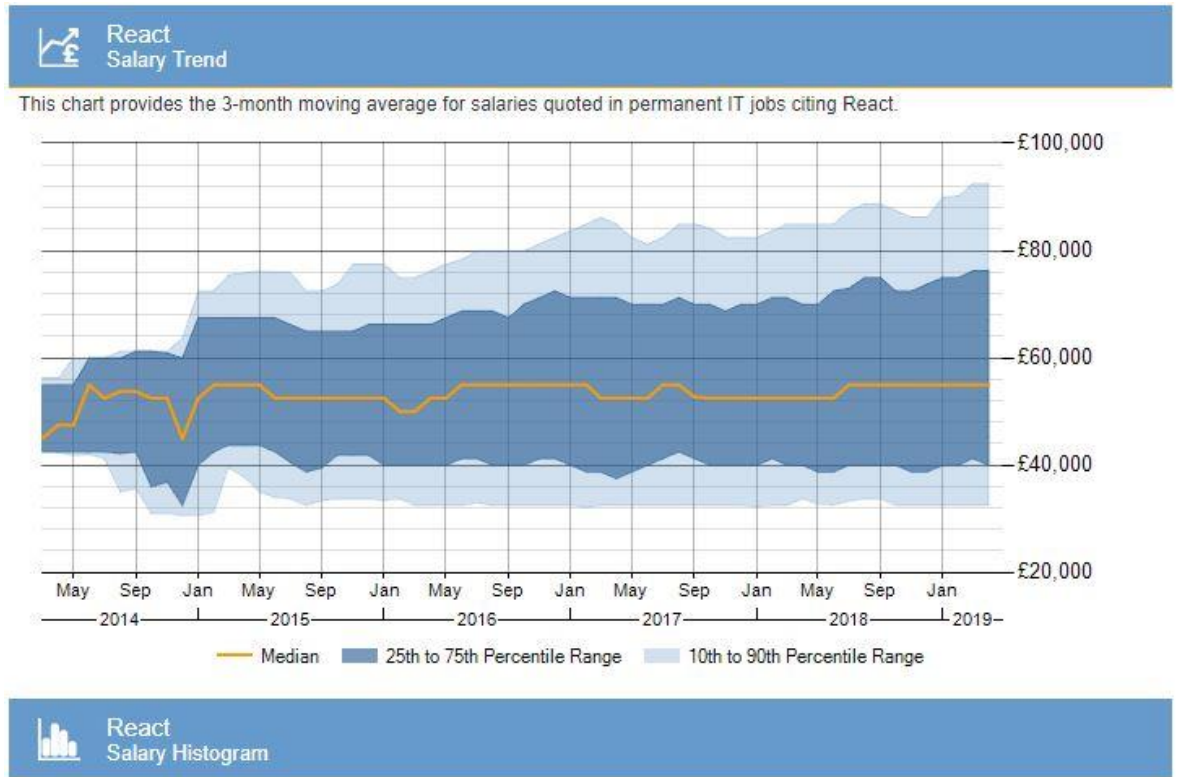

Figure 21. React Salary Trend

Finally, the data below shows what areas of the United Kingdom have the highest number of vacancies. It gives the number of matching jobs, median salary and median salary changes for the past six months and live job vacancies. It should come as no surprise that London has the largest number of jobs by far, this is followed by the South East of England which includes the counties Berkshire, Buckinghamshire, Hampshire, Kent, Oxfordshire, Surrey and Sussex. Following in third place is North England which includes the cities of Liverpool and Manchester. Figure 22 (ITJobsWatch 2019b)

Location	Rank Change on Same Period Last Year	Matching Permanent IT Job Ads	Median Salary Past 6 Months	Median Salary % Change on Same Period Last Year	Live Job Vacancies
England	▲ +13	6,736	£57,500	+9.52%	823
UK excluding London	▲ +15	3,668	£45,000	-	440
London	▲ +15	3,439	£70,000	+7.69%	433
South East	▲ +10	1,161	£50,000	+11.11%	83
North of England	▲ +12	904	£42,500	-5.56%	148
North West	▼ -3	491	£42,500	-	92
South West	▲ +17	476	£42,500	-5.56%	54
Yorkshire	▲ +39	386	£42,500	-5.56%	40
Midlands	▲ +20	360	£42,500	+6.25%	66
East of England	▲ +28	347	£55,000	+10.00%	38
Scotland	▲ +16	314	£50,000	+11.11%	32
West Midlands	▼ -6	192	£44,000	+17.33%	38
East Midlands	▲ +46	168	£42,500	-5.56%	28
Wales	▲ +38	64	£40,000	+5.26%	12
Northern Ireland	▼ -1	48	£50,000	+36.99%	7
North East	▼ -24	27	£40,000	-11.11%	16

Figure 22. React Top 16 Job Locations

3.8 Results

This chapter will be used to give details on the empirical section of the report where the data comes from and how it has helped solve the initial questions of the report. The results are based on data provided by the Stack Overflow Developer survey, which as previously stated is the largest of its kind in the world and IT Jobs Watch.

3.8.1 Demographics

As can be seen in the supporting data (subchapter 3.6), Europe is currently the largest area for developers in the world. This gives students studying in Europe excellent opportunities regarding careers. Furthermore, the United Kingdom ranks in second place for the amount of developers in the survey located in Europe. It also shows that the vast majority of those questioned have spent less than five years programming in a professional capacity and that the largest share of respondents has a bachelor's degree.

3.8.2 Technology

Through all data gathered it can be seen that JavaScript is still the most widely used language today by far. In addition to this, ReactJS is without a doubt the most favoured framework currently available to developers, with the survey showing it is both the most loved and wanted.

3.8.3 Salaries

The survey shows that respondents working with JavaScript earn on average $55k, this increase or decreases dependent on what position they currently fill with mobile developers earning the lowest and dev-ops amongst the highest. The data gathered from IT Jobs Watch also indicates that React developer's median salary is currently £45k, with the obvious exception being London where salaries are much higher.

4 Discussion

4.1 Consideration of Results

To round up, after reviewing all the data available in this report, I believe that ReactJS will continue to be a major driving force within the web / mobile-development community. One of the major reasons is how easy it is to use for those just beginning their professional careers, as it only requires knowledge in JavaScript, HTML and CSS to start.
However, ReactJS also provides those people who already have experience the tools necessary to create high performance applications.

ReactJS has seen huge growth in the job market since its initial inception, though now it is starting to slow down somewhat, there are still an abundance of job vacancies available within the United Kingdom as can be seen in the listed sources within this report. It is also one of the top libraries amongst developers that already use it and the most wanted by those who don't currently.

The median salaries currently hold up well to jobs of a similar nature in the field. Obviously as can be seen from the data, London currently has both the highest number of jobs and also the highest median salaries. This is to be expected, as the cost of living in London is significantly higher than any other part of the UK. Also, the number of vacancies comes as no surprise as London is also the largest population centre in the United Kingdom.

Also, as can be seen in the data from the survey the United Kingdom is currently one of the best countries when it comes to total hours worked with only the Netherlands being lower.

4.2 Trustworthiness

The author feels that all of the data gathered in this report is provided by highly respected sources. The theoretical data from books are all authored by respected individuals within the community. The main statistical data was gathered from Stack Overflow and IT Jobs Watch. Stack Overflow is one of, if not the most respected sources of information on the web with regard to development, not only by developers but also international companies. The survey itself as previously stated is largest of its kind and covers a massive area with concern to developers.

IT Jobs Watch is also a highly respected source of information concerning job vacancies within the United Kingdom, the site garners more than 100,000 visitors each month and provides excellent data on the United Kingdom labour market within the IT sector. Giving excellent insights into job availability, salaries and which locations have the highest number of positions.

4.3 Bias

The author feels there is a definite bias within the development community at large and by association this report. As can be seen in the demographics section of this report, the community is dominated by white male respondents from both Europe and America. This is a problem that has been known for a long time within the profession. However, as this report is on how viable a career in React is, the author feels that this is unavoidable as the entire report is focused on a community that currently suffers from heavy bias.

4.4   Conclusion

The author feels that with its continued growth and the preferential treatment it currently enjoys within the community, ReactJS is here to stay. The salaries are average but that is always to be expected when choosing something of a popular nature, as it comes down to supply and demand. However as can be seen in the results for those who are well versed in ReactJS the salaries have continued to rise year on year.

The salaries are in line with the rest of the industry and the job availability within the United Kingdom is in a good position. Also, the fact the UK has some of the lowest working hours of all those who partook in the survey, indicates that developers will not have to worry too much on issues such being overworked which can lead to burnout. Then there is the data showing that most professionals questioned have been coding for less than five years, this shows that a large portion of our industry is made up of those with less experience. This gives students just entering the field an excellent opportunity as they don't have to be overly daunted by the prospect of not being good enough or not having enough experience. On the other hand, the data also shows that for those who are willing to become well-versed in ReactJS the opportunity for larger salaries are there for the taking.

All in all, the author feels once all the data contained within the report is taken into account, that choosing to follow a career path in ReactJS and becoming a React developer is a safe and viable option and will continue to be so going into the future.

References

AltexSoft, 2018. The Good and the Bad of ReactJS and React Native. URL:
https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-na-
tive/.
Accessed 10 April 2019.

Bonnie Eisenman. 2016. Learning React Native. Building Native Mobile Apps with JavaS-
cript. O'Reilly Media. Sebastopol.

Facebook 2015. React Native. Bringing modern web techniques to mobile. URL:
https://code.fb.com/android/react-native-bringing-modern-web-techniques-to-mobile/.
Accessed: 10 April 2019.

ITJobsWatch 2019a. About IT Jobs Watch. URL:
https://www.itjobswatch.co.uk/about.aspx.
Accessed 15 April 2019.

ITJobsWatch 2019b. React Jobs. URL:https://www.itjobswatch.co.uk/jobs/uk/react.do.
Accessed 26 April 2019.

Masiello, E & Friedmann, j. 2017. Mastering React Native. Packt Publishing. Birmingham.

Mozilla Foundation, 2018. Introduction to the DOM. URL:https://developer.mozilla.org/en-
US/docs/Web/API/Document_Object_Model/Introduction.
Accessed 21 April 2019.

Nicholas Zakas. 2005. Professional JavaScript for Web Developers. Wiley Publishing Inc.
Indianapolis.

Sengupta, Singhal, & Corvalan. 2016. Getting Started with React. Packt Publishing. Bir-
mingham.

StackOverflow 2019a. Developer Survey Results 2019. URL:https://insights.stackover-
flow.com/survey/2019#correlated-technologies.
Accessed 27 April 2019.

StackOverflow 2019b. Developer Survey Results 2019. URL:https://insights.stackover-
flow.com/survey/2019#geography.
Accessed 27 April 2019.

StackOverflow 2019c. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#developer-roles.
Accessed 27 April 2019.

StackOverflow 2019d. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#developer-profile-_-years-coding-professionally.
Accessed 27 April 2019.

StackOverflow 2019e. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#developer-profile-_-educational-attainment.
Accessed 27 April 2019.

StackOverflow 2019f. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#developer-profile-_-undergraduate-major.
Accessed 27 April 2019.

StackOverflow 2019g. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#developer-profile-_-other-types-of-education.
Accessed 27 April 2019.

StackOverflow 2019h. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#technology-_-programming-scripting-and-markup-languages.
Accessed 27 April 2019.

StackOverflow 2019i. Developer Survey Results 2019. URL:https://insights.stackover-flow.com/survey/2019#technology-_-most-loved-dreaded-and-wanted-web-frameworks.
Accessed 27 April 2019.

Stackoverflow 2019j. Developer Survey Results 2019. URL: https://insights.stackover-flow.com/survey/2019#work-_-salary-and-experience-by-developer-type.
Accessed 4 May 2019.

Stackoverflow 2019k. Developer Survey Results 2019. URL: https://insights.stackover-flow.com/survey/2019#work-_-salary-and-experience-by-language.
Accessed 4 May 2019.

Stackoverflow 2019l. Developer Survey Results 2019. URL: https://insights.stackover-flow.com/survey/2019#work-_-hours-worked-per-week.
Accessed 4 May 2019.

Stackoverflow 2019m. Developer Survey Results 2019. URL: https://insights.stackover-flow.com/survey/2019#methodology.
Accessed 4 May 2019.

Stackoverflow 2019n. Developer Survey Results 2019. URL: https://insights.stackover-flow.com/survey/2019#demographics.
Accessed 4 May 2019.

Vipul, AM & Prathamesh, S. 2016. ReactJS by Example. Building Modern Web Applications with React. Packt Publishing. Birmingham

React 2018. A JavaScript library for building user interfaces. URL: https://reactjs.org/.
Accessed: 14 April 2019.

React Native | A framework for building native apps using React. URL: https://facebook.github.io/react-native/.
 Accessed: 21 April 2019.

Walliman, N. 2011. Research Methods: the basics. Routledge. New York.