Metropolia

Tam Long Thai

Developing a cross-platform restaurant mobile application with React Native

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor’s Thesis

6th May 2022

Abstract

Author:	Tam Long Thai
Title:	Developing a cross-platform restaurant mobile application with React Native.
Number of Pages:	31 pages
Date:	6th May 2022
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	Mobile Solutions
Supervisors:	Ilkka Kylmäniemi, Senior Lecturer

The title of the report is “Developing a cross-platform restaurant mobile application with React Native”. The restaurant market analysis and the essentials of React Native in mobile development are the key ideas that will be explored in this thesis. In addition, this thesis is going to cover a step-by-step guidance on how to create a functional restaurant application.

The project for this thesis is going to be implemented with the React Native framework for Kung Food Panda restaurant chains in Finland.

Keywords: React, React Native, Mobile Development, Cross-platform Development, Foodservice, Restaurant Industry.

Contents

List of Abbreviations

UX:	User Experience
UI:	User Interface
SPA:	Single Page Application
DOM:	Document Object Model
API:	Application Programming Interface
MVC:	Model-View-Controller
WHO:	World Health Organisation
GPS:	Global Positioning System
HTML:	HyperText Markup Language
CSS:	Cascading Style Sheets
XML:	Extensible Markup Language
JSX:	JavaScript JSX
UCD:	User-centred Design
IDE:	Integrated Development Environment

1 Introduction

The main purpose of this thesis is to provide an instruction for using React Native to develop a cross-platform restaurant application for mobile devices. Also, the restaurant market will be analysed to provide an insight on why an application is necessary for restaurant business. The final part of this thesis will introduce step-by-step guidelines on how to develop and deploy a restaurant application with React Native.

According to Statcounter, the mobile operating system market share is composed of two main operating systems: Android (72.83%) and iOS (26.35%) by June 2021. [1] There are major differences between the two operating systems and they both have different development platforms: Java or Kotlin for Android development and Swift for iOS development. However, React Native provides an efficient solution for the mentioned problem with its “write once, run anywhere” feature. It means that a developer only needs one code base to render the application on multiple platforms or devices. See figure 1 below.

React Native, initially released by Facebook in March 2015, is a JavaScript framework that allows developers to create cross-platform mobile applications on both iOS and Android natively. [2] React Native is powered by React.js (or React in short), a JavaScript library mainly used for developing UX or UI components on web-platforms. A prototype of React was first introduced as FaxJS by a Facebook software engineer named Jordan Walke. One year later, Walke officially created React, the technology was then deployed on Facebook Ads and later adopted by Instagram, which was bought by Facebook in April 2012. With React becoming popular within the developer community, Facebook decided to make the technology open-sourced at JSConf US in May 2013. [3]

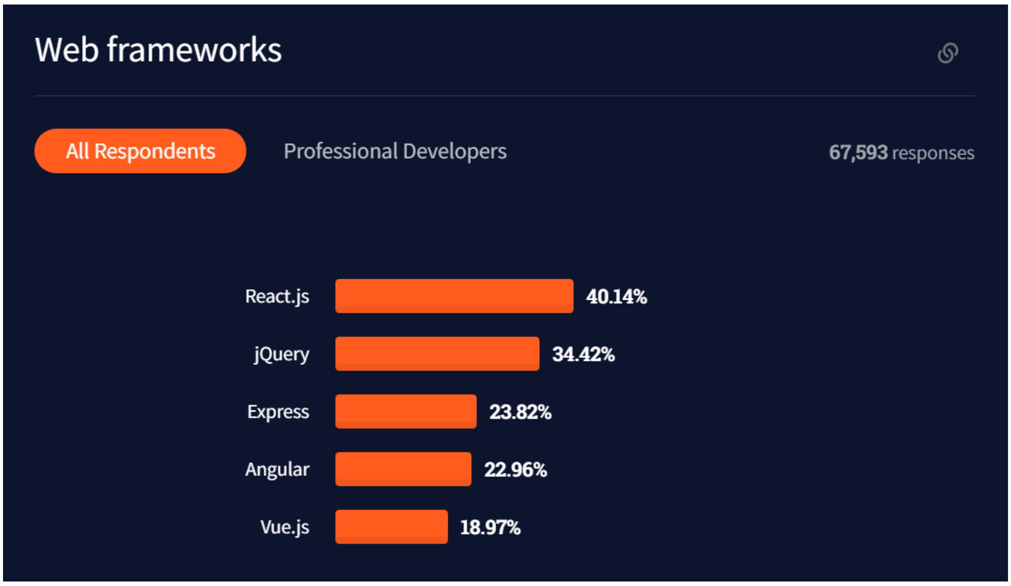


Figure 1. The most common web frameworks in 2021 by Stack Overflow.

While other frameworks are based on traditional multi-pages design, React stands out for its simplicity with the Single Page Application (SPA) model. SPA allows the application to load and unload different view models to a single page instead of loading through multiple pages. This can be achieved with multiple React features such as automatic UI state management, virtual DOM manipulation or APIs. As a result, developers can create a more fluid user experience. [4]

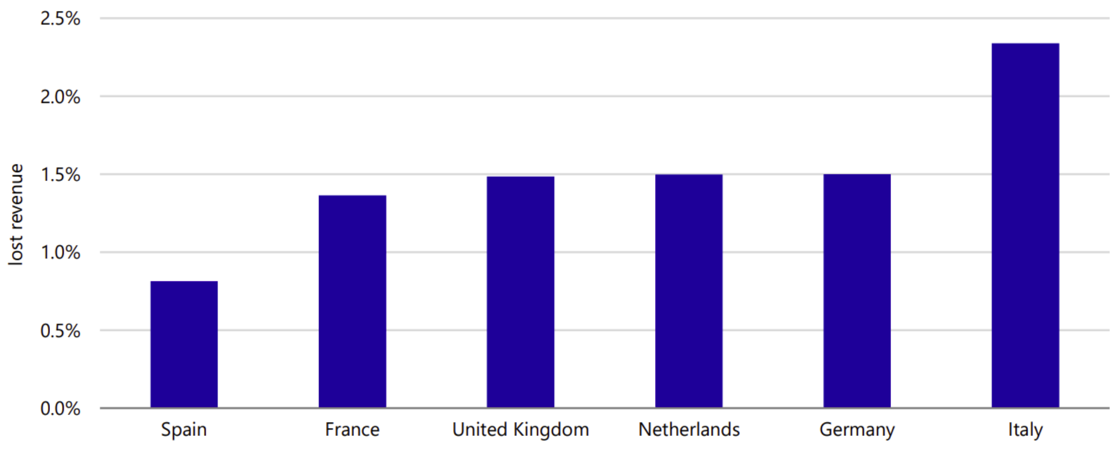
Moreover, React only functions as the View layer in the MVC model, which allows developers to integrate other technologies for the Model and Controller layers. This means maintaining the existing code of the application or using the new technology on the application will be more straightforward. [4]

2 Foodservice market analysis

In late 2019, the first case of coronavirus disease 2019 (COVID-19) was discovered in Wuhan, one of the major cities in China. After a failed attempt to contain the virus in China, COVID-19 quickly spread around the world, causing a massive outbreak. The outbreak caused several stages of dishevelment in many countries such as China, India, United Kingdom, France, Italy, Russia, United States, and Brazil. In March 2020, WHO officially announced COVID-19 as a global pandemic.

The COVID-19 pandemic has definitely had a significant influence on the global economy including the food industry. In several nations, the government has imposed temporary restrictions on foodservice to prevent the spread of COVID-19, making it illegal for consumers to visit restaurants, bars, and cafes. The restriction created a ripple effect to the restaurant industry. According to the data from OpenTable, the restaurant occupancy including reservations and walk-ins in the United States dropped 50-60% from February to March 2020. [5]

Furthermore, more than 100 countries had imposed restrictions on both international and domestic travel to battle with the COVID-19 pandemic. According to the data from FlightStats, air traffic fell by 63% in March 2020 compared to the same time period in 2019. In many major cities, data shows that intercity travel dropped up to 70% during the lockdown period. [6] Accordingly, travel and tourism had been heavily imperilled. Travel and tourism play an important role in providing a large number of potential consumers for the foodservice industry. For this reason, the lockdown had put the food industry at a great risk. The report by RaboResearch on Impact of Coronavirus on European Foodservice states that the total revenues of restaurants, bars and cafes in Europe consists of 20% to 40% from the travel and tourism demands. [7] As a result, the total revenue has been adversely impacted as shown in Figure 2.



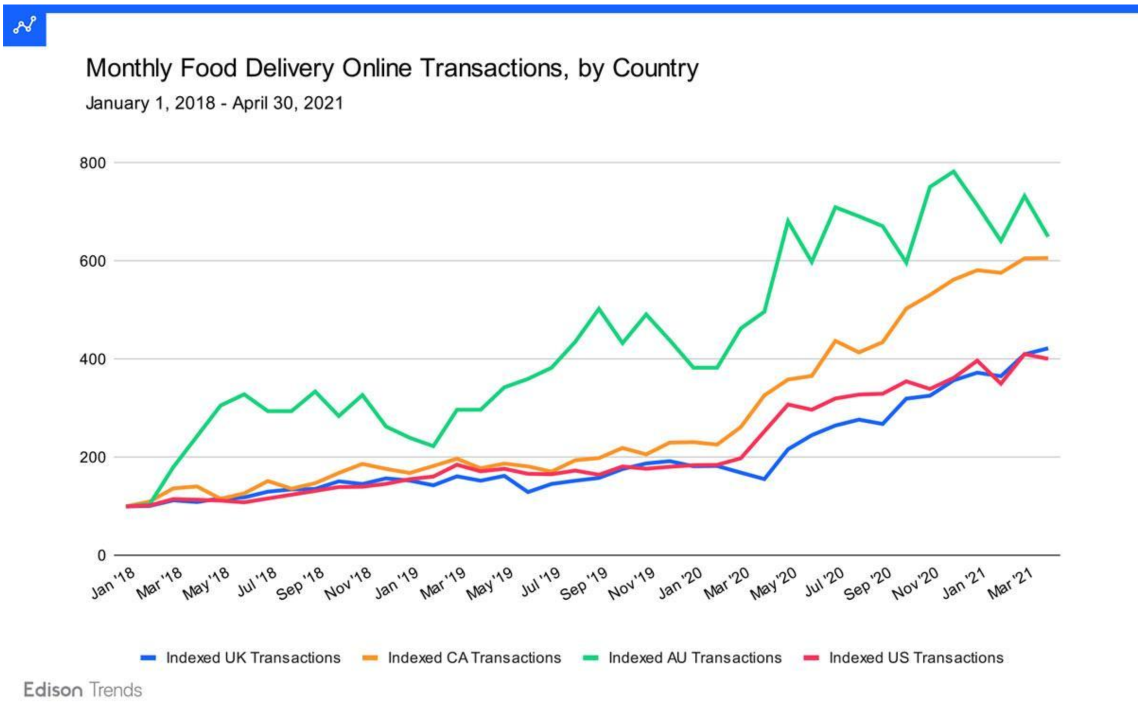
Source: Rabobank 2020

Figure 2. Lost revenue of the foodservice industry in Europe in early 2020.

2.1 The foodservice strategy

The social distancing and lockdown restrictions created a cosmic boost for takeaway and delivery as the habit of consumers in the restaurant industry has been leaning towards ordering in.

According to a report by Edison Trends presented in Figure 3, the food delivery online transaction from the beginning of 2018 to early of 2020 in the United States raised 85%, while it continued to raise up to 96% until the end of 2020. The year-over-year online transaction for mid-2021 (compared to the same time period in 2020) in the United Kingdom, Australia and Canada grew 171%, 30% and 86% respectively. [8]



Edison Trends

Figure 3. Monthly food delivery online transactions in the United States, United Kingdom, Australia, and Canada from 2018 to 2021.

This trend has given the food delivery ecosystem (including consumers, restaurants, and delivery services) a chance to continue its fast-growing expansion. In 2020, the estimated number of active food delivery application users in Europe reaches over 174 million users, while the same statistic in the US is 111 million users. The market revenue of food delivery platforms in Europe has grown 170% between 2015 and 2020, with projected revenue continuing to grow 55% in 2025. Meanwhile, the United States market hit 26.5 billion dollars in 2020 and is expected to nearly double the revenue in 2025. [9]

2.2 Kung Food Panda restaurant strategy

Kung Food Panda is a Finnish fast food restaurant franchise founded in 2015 with a mission to bring tasty Asian fast food for Finnish taste. The franchise serves a range of Chinese-inspired meals such as mandarin chicken, sweet chilli fried chicken, dynamite chicken. After the first opening in Espoo, Kung Food Panda has been expanding to Helsinki Capital Region and major cities including Tampere and Turku. With 9 locations, it is one of the most popular Chinese restaurant franchises in Southern Finland.

With the constant growth of the franchise and the shifting of consumer's habits in foodservice, Kung Food Panda needs a well-executed mobile restaurant application with aims of a more direct approach to the customers and friendlier collaboration with the delivery services. The first and foremost advantage is to reach the customers base in the designated geographical area. In Finland, 96% of the population and 100% of people aged 16 to 24 own a smartphone in 2021. [10] Due to the popularity of smartphones, a location-based application is one way to increase the branding exposure of the restaurant. The application uses modern technology such as iBeacon, GPS, and social media to target the customers locally.

Secondly, Kung Food Panda wants to enhance the customer's online experience with the restaurant. Traditionally, customers have to visit various websites and applications for different information about the restaurant. The process can often be time consuming and inconvenient, therefore decreasing the customer's experience. The mobile application offers the solution by providing necessary features to the customers such as restaurant information (address, menu, opening hours, etc.), online reservation, in-app deals and offers, delivery options.

Last but not least, customer loyalty plays an important role in food industry revenue. Research by Valassis Communication in 2019 found that customers with coupons visit the restaurant more frequently than regular customers. [11] Kung Food Panda rewards customer loyalty with loyalty programs and rewards integrated into the mobile application, thus encouraging returning visits.

3 The fundamental of React Native

3.1 JavaScript and JSX

3.1.1 JavaScript

JavaScript is a high-level programming language that adheres to the ECMAScript standard. It is also classified as a just-in-time compilation and multi-paradigm. The main feature of JavaScript includes curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. [12] According to a survey by Stack Overflow in Figure 4, JavaScript has ranked first as the most widely used programming language for the last 9 years. Mainly acting as a client-side, JavaScript - together with HTML and CSS- is the backbone of over 97% of websites on the world wide webs. [14] Moreover, with the constant development and its popularity, JavaScript can also be used to create web servers or server applications (such as Node.js) and develop front-end frameworks (such as React, React Native, Angular, and Vue).

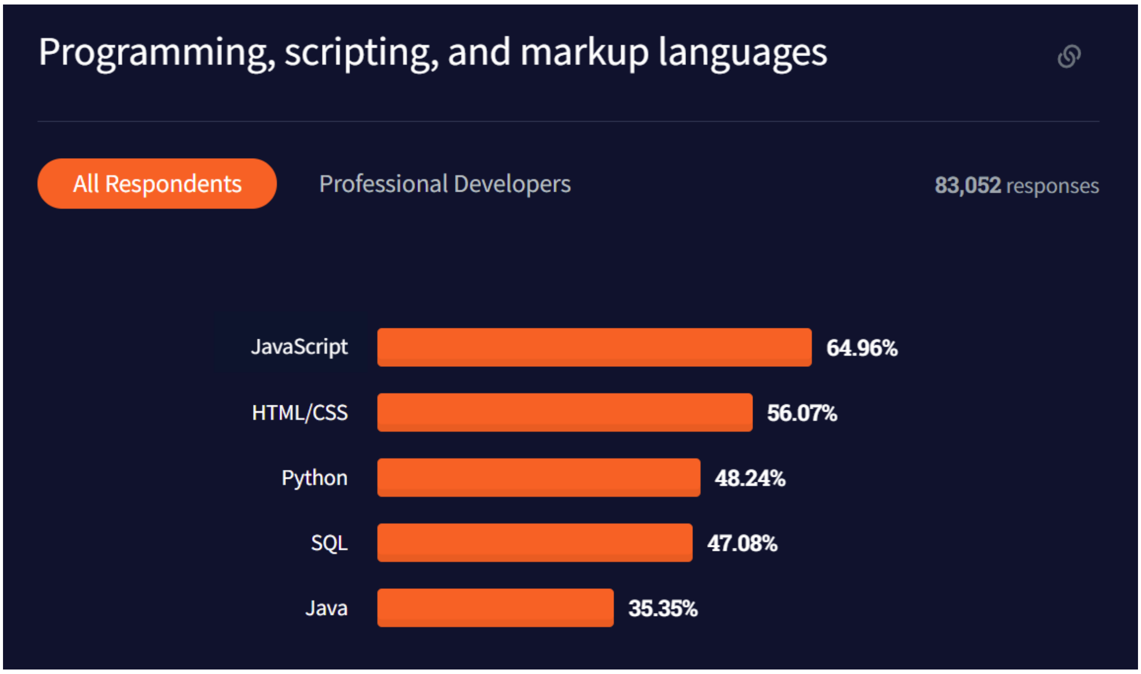
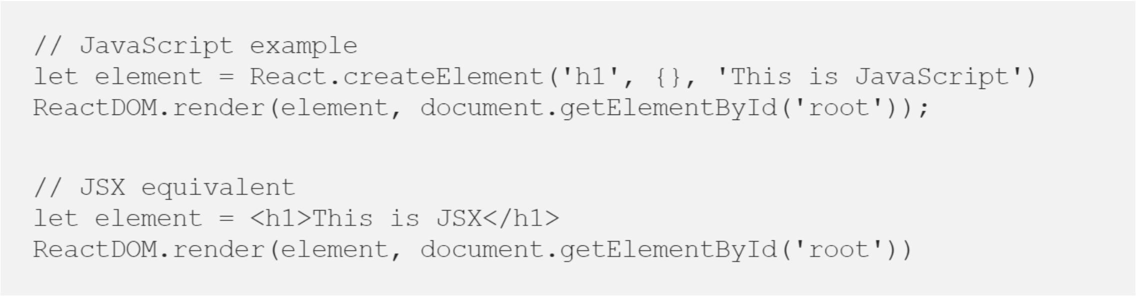


Figure 4. The most common programming languages in 2021 by Stack Overflow.

3.1.2 JSX

JSX, also known as JavaScript XML, is an extension of the JavaScript syntax with the main purpose of defining the UI in a React application. [14] Compared to other frameworks such as Angular or Vue which require a series of steps to create, manipulate and render the element to the DOM, JSX transforms HTML elements into React elements directly. That means developers are able to write HTML elements inside JavaScript without calling for extra methods such as createElement() or appendChild() like in an example in Figure 5.



```
// JavaScript example
let element = React.createElement('h1', {}, 'This is JavaScript')
ReactDOM.render(element, document.getElementById('root'));

// JSX equivalent
let element = <h1>This is JSX</h1>
ReactDOM.render(element, document.getElementById('root'))
```

Figure 5. Difference between JavaScript and JSX.

JSX is not mandatory in React, but it offers a faster, safer, and easier solution in comparison to JavaScript. JSX can simultaneously optimise and compile the source code to JavaScript, thus increasing the running speed significantly. Moreover, one of the main features of JSX is source-code debugger which allows the compiler to detect potential errors during the compilation process. [15]

3.2 Virtual DOM

Real DOM defines the UI of an application and DOM manipulation (see Figure 6) is the process of updating and illustrating the UI when there are changes in the application. DOM is a representation of tree data, which makes the DOM manipulation quick. When the application has multiple UI components that need to be rendered, every changed element and its child nodes are also re-rendered. As a result, DOM manipulation becomes overloaded and slows down the performance of the application. [16]

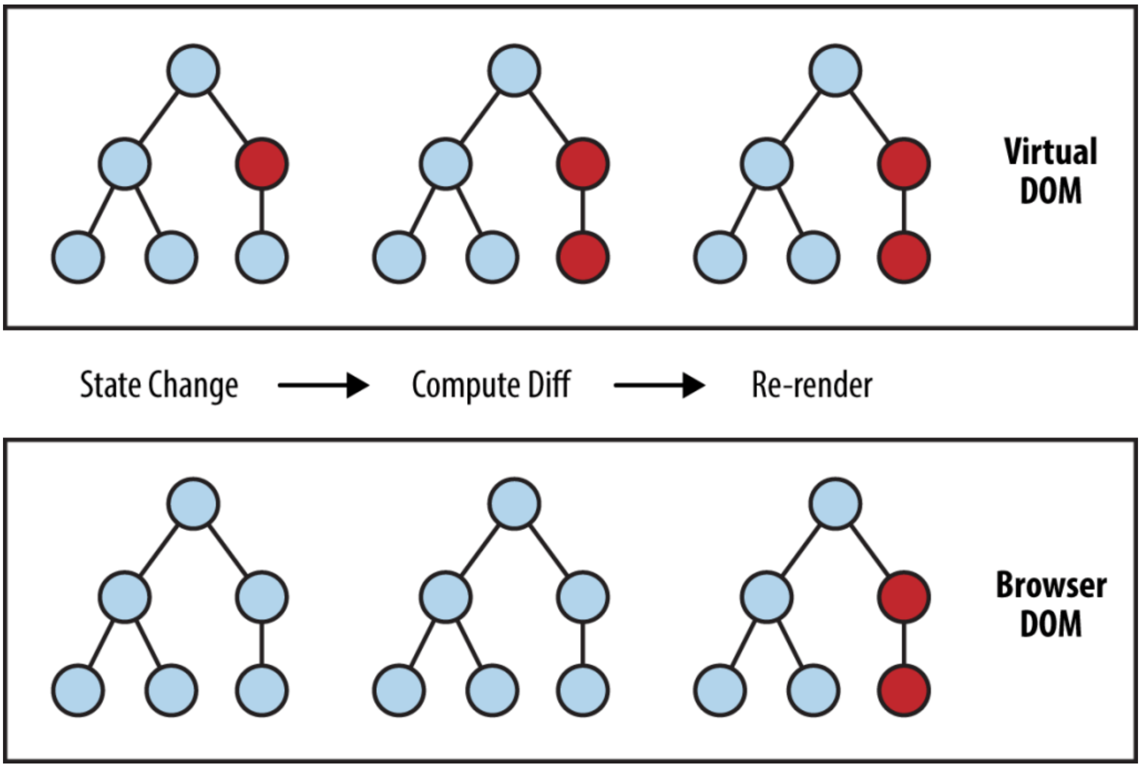
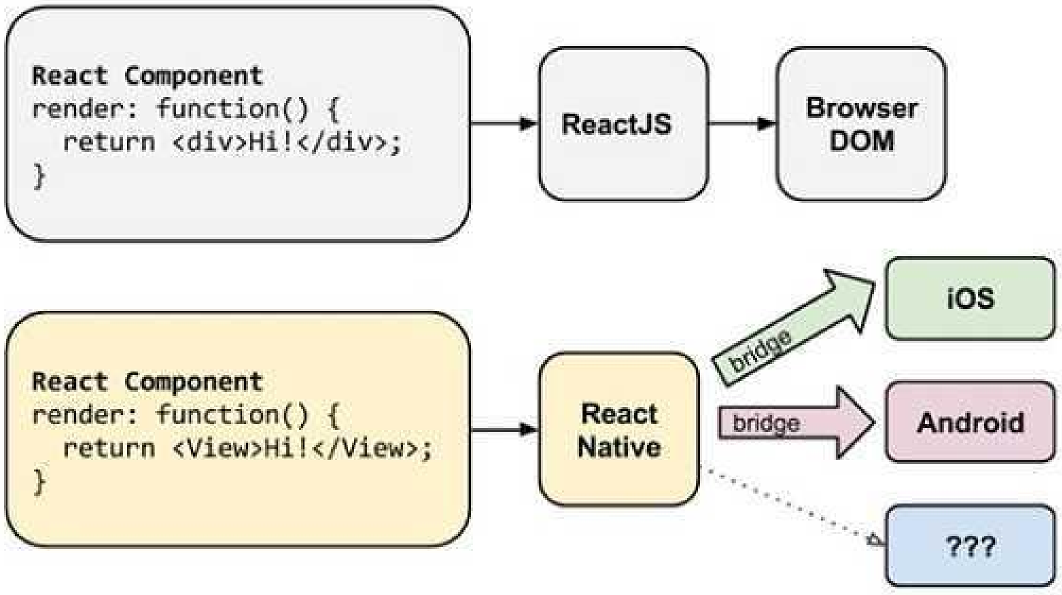


Figure 6. Virtual DOM versus real DOM manipulation. [17]

The virtual DOM represents the virtual version of the real DOM with the similar tree data structure. Every element of the UI is defined as a node of the tree data and when the state of these elements alters, a new virtual DOM tree is created. Both the newly created tree and the previous one will then be compared through a diffing process to find the differences and re-render only those to an updated tree. In the final step, the real DOM gets a batch update from the updated tree. The whole process guarantees the minimal operation on the real DOM. [16]



```
React Component
render: function() {
  return <div>Hi!</div>;
}
```

```
React Component
render: function() {
  return <View>Hi!</View>;
}
```

Figure 7. Communication between React Native and native components through its Bridge. [18]

As depicted in Figure 7, React Native uses a similar method to render the components in iOS or Android by calling for Objective-C API or Java API respectively instead of browser DOM. In this case, the virtual DOM functions as an abstraction layer (also known as a Bridge) for JavaScript threads and native threads to communicate with each other.

3.3 The component lifecycle

In React Native, the UI breaks down into multiple native components. Each component is an individualistic, reusable piece of code that gets arbitrary input (also known as Props) and gives back the equivalent React element to represent a segment of the UI. There are two types of components in React Native: class components (using *extends React.Component* statement and render() method) and function components (using JavaScript function). [19]

A component lifecycle describes the events in a component life. Each component follows three phases of the cycle including mounting, updating, and unmounting. The first phase called mounting is when a new component is generated and mounted on the DOM. After each interaction between the user and UI, the state of the component is modified. This requires the component to undergo the updating phases where the component is being re-rendered. See Figure 8 below. In the last phase of the component lifecycle, the component is unmounted on the DOM. [20]

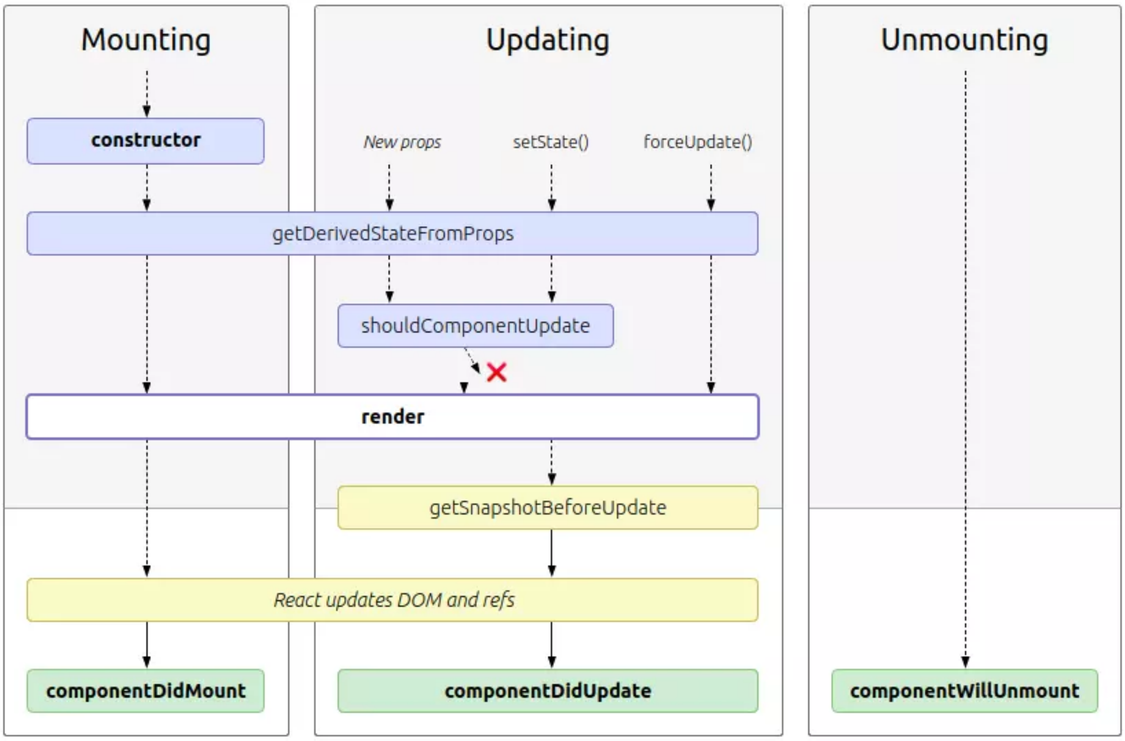


Figure 8. A diagram of the component lifecycle. [21]

The most common lifecycle method in React Native classes is the render() method. It is called in the mounting and updating phases with the main purpose of rendering the component to present the equivalent JSX in the UI. The render() method can only receive and return the same data, which means it is not allowed to interfere with the component state. After the rendering process finishes, the componentDidMount() method is then executed only once. This method handles the update of the component in the DOM. In the updating phase, the componentDidUpdate() method instantly updates the DOM when there are changes of the state and props. In both componentDidMount() and componentDidUpdate() methods, developers can set the state of the component, however it may affect the performance or cause infinity loops. Finally, the componentWillUnmount() method is invoked before the removal of the component from the DOM. [20]

3.4 Props and State

Props (shorts for properties) are an object that contains the data of the component. They can be initiated by the component as default props or passed into the component from the parent. In React Native, props are immutable properties which means they are not allowed to change during a component lifecycle event. [22]

Similar to props, state also contains the data of the component. However, the component state can be modified by calling the setState() function, which means the state is observable. In large scale projects, data flow is controlled by using state management containers such as Redux or Mobx. Since state can only be used within the class components, it has to be passed as props from parent to the child components. [22]

```
// Stateless Component
const DisplayDetails = props => {
 return (
 <div>
   <h1> Props Usage </h1>
   <h3>
     My name is {props.firstName} {props.lastName}, I live in {props.city}.
     Nice to meet you!!{" "}
   </h3>
 </div>
 );
};

// Stateful Component
class Greet extends React.Component {
state = {
  name: ""
};

updateName = e => {
 if (e.key === "Enter" && e.target.value.trim().length != 0) {
 this.setState({
   name: e.target.value
 });
 }
};

render() {
 return (
   <div>
     <h1> State Usage </h1>
     {!this.state.name && (
       <div>
         <h3> What's your good name ?</h3>
         <input
           type="text"
           onKeyPress={this.updateName}
           placeholder="Once done press Enter"
         />
       </div>
     )}
     { this.state.name.trim().length != 0 && (
        <h3> Moi !!! My name is {this.state.name}</h3>
     )}
   </div>
  );
 }
}

const userDetails = {
 firstName: "Long",
 lastName: "Thai",
 city: "Espoo"
};

ReactDOM.render(
 <div>
   <DisplayDetails {...userDetails} /> <hr />
   <Greet />
 </div>,
 document.getElementById("root")
);
```

Listing 1. Examples of stateless components and stateful components. [23]

In the example as shown in Listing 1, the DisplayDetails is a function component (or stateless component). The probs firstName, lastName and city which hold the value of Long, Thai and Espoo respectively are passed into the component DisplayDetails. On the contrary, Greet is a class component (or stateful component). When the onKeyPress event is triggered, the component uses state to store the “name” which has a value input by the user (for example, the input is “Long”). Then it uses *this.state.name* to access the current state data and print out the following sentence “Moi !!! My name is Long”.

4 Case Study: Restaurant application development for Kung Food Panda restaurants.

4.1 Project analysis

In this chapter, the development process of a React Native mobile application will be demonstrated. The prototype of the application was created for Kung Food Panda restaurants chain, aiming at the mobile users demographic who wants to connect better with the restaurant. The application's main goal is to create a better interaction between Kung Food Panda and its customers by providing the up-to-date information of the restaurant chain, offers and discounts.

The mobile platform application is a further development from an existing web-based application with a number of additional functions. The challenge of this project is to maintain the same level of consistency of the user experience when they switch from web-based application to mobile application. Therefore, this chapter is going to concentrate on how to utilise React Native for a smooth user experience design.

4.2 Project requirement

4.2.1 Application design

The most important function for the application is the user authentication. It is a method of confirming the identity of the customers to give them customised services and other resources. The customer can create a new account or log into an existing account. Both processes will require a working phone number or email, personal information and a secured password. The customer also has additional abilities to log out of the application or log in on multiple devices.

Secondly, the main goal of the application is to encourage the customer to visit Kung Food Panda restaurant. This goal can be achieved with the showcase function. It contains two sections: the first section shows the top 3 nearest restaurants based on the location of the customer and the second section shows all the available Kung Food Panda restaurants in Finland. Furthermore, by clicking on a specific restaurant, the customer will be navigated to the restaurant landing page where they can have access to its information or bookmarking it.

The third requirement is the functionality search. This function allows the customer to explore information about all of the Kung Food Panda restaurants. For instance, the customer can search for the new restaurant in Espoo or find which restaurant serves their favourite food. This function requires an additional database on top of React Native framework.

4.2.2 Application Architecture

In addition to the above mentioned functions, the application should also include some requirements that do not behave accordingly with the user interaction. The purpose of those requirements is to assess the overall quality of the application. In system engineering terms, they can also be known as non-functional requirements. [24]

The most important non-functional requirement is the usability of the application. This is a measurement of how much time and effort for the user to be able to operate the application. Developers and designers must work closely to ensure the consistency level of the usability from the initial planning process to the prototyping phase and the final product delivery. Two characteristics of the application usability are the UI and UX.

The user interface is a graphical and textual representation of the application, it is the only way for the user to interact with the application. Therefore an intuitive design will help the customer to operate the application without any errors. There are certain standards that the user interface must follow. One of the most common designs is the Material Design created by Google in 2014. Since then, Material Design has become the standard to develop a quality and intuitive application for various platforms such as websites, Android and iOS. It breaks the whole application interface into smaller blocks called components; the basic components are display, navigation, actions, input and communication. Google also provides colour system, typography and shapes style to ensure the unity of every design. [25]

The user experience can be judged based on how well the user interacts with the application through the user interface. A widely known framework to create an efficient user experience is user-centered design (UCD). It was firstly presented by Rob Kling in 1977 and later developed into a modern framework by Don Norman. UCD ensures the design and functionality of the application are focused on the user through a series of user testings through every stage of the application development. Three main tools for such testing are persona, scenario and use case. These are different tools to create a specific user profile, fictional event and documentation of possible interaction between the user and the application.

Furthermore, other non-functional requirements should also be taken into consideration such as security - a requirement to protect the application and users data in cases of cyber attack or data breach, reliability - a requirement to measure the frequency between two consecutive application failures, modifiability - a requirement to calculate the necessary time and personnel for the application maintenance and development. Without these requirements, the application will fail to meet the customer’s needs and satisfaction.

4.3 Project workflow

The next step of the project development process is creating a system design for the application based on the presented functions and requirements in the previous section. The workflow is a diagram that represents the interaction process between user and the application. It provides a step by step how the application can be used from start to finish. The main purpose of the workflow diagram is to ensure a certain function will not be missed, eliminate the unnecessary interaction and improve the testing process.

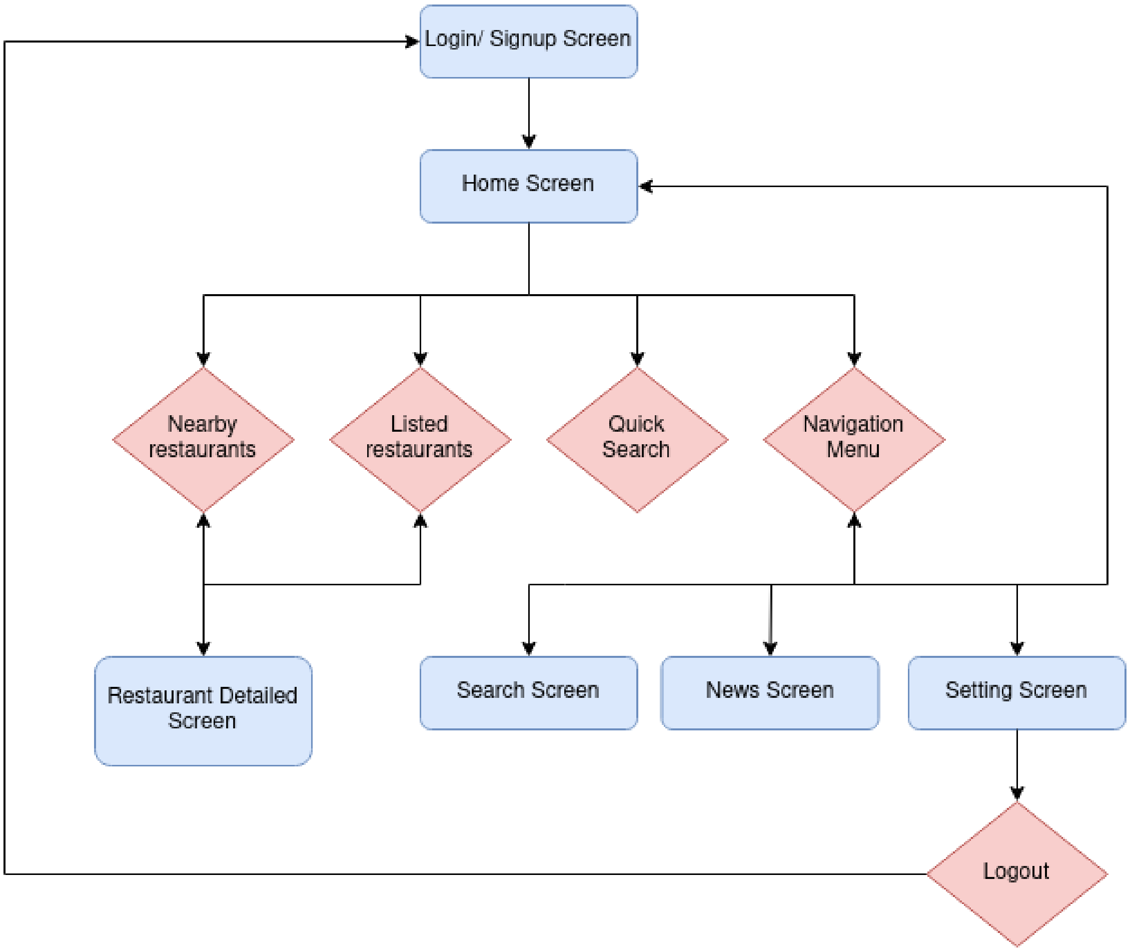
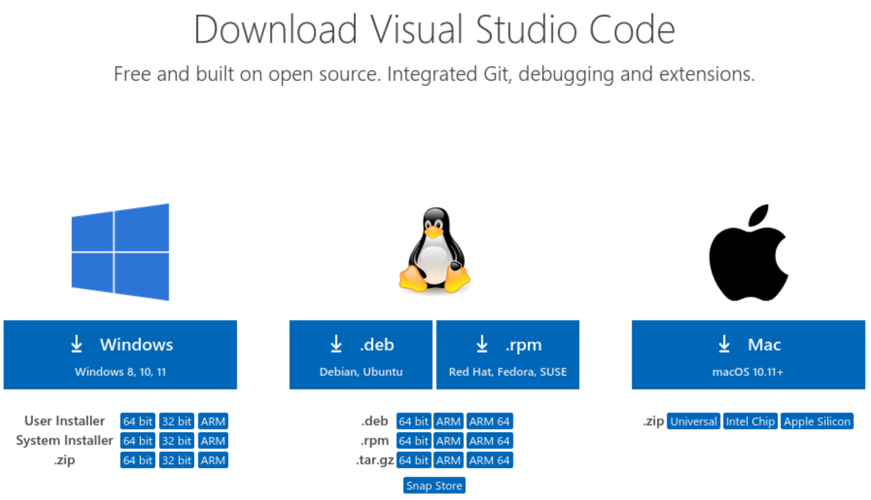


Figure 9. The workflow of the Kung Food Panda application.

In Figure 9, the workflow of the application is presented. The application’s first step is the authentication of the customer identity with a login screen. If the customer does not have an existing account, they can create a new one. After finishing with the first step, the customer will enter the home screen. This home screen consists of four main components: a list of nearby restaurants, a list of all the available restaurants, a search bar and a bottom menu. When the customer chooses a restaurant from the list, they will be navigated to the restaurant's detailed screen that shows all the information about the restaurant. The customer also has an option to perform a quick search on the home screen. The bottom menu acts as a navigation system for the customer to move in between the main screens such as Main screen, Search screen, News screen and Setting screen. The final action of the workflow is when the customer logs out of their account, which brings them back to the initial login screen.

4.4 Project setup

The first step for the project setup is choosing an integrated development environment (IDE). IDE is a graphical user interface software that provides developers with the needed tools for application development such as source code editor, local build automation and debugger. [26] The chosen IDE for this project was Visual Studio Code developed by Microsoft because of its availability on multiple platforms, lightweight, low resources usage and a wide choice of extra libraries and plug-ins. See Figure 10 below.

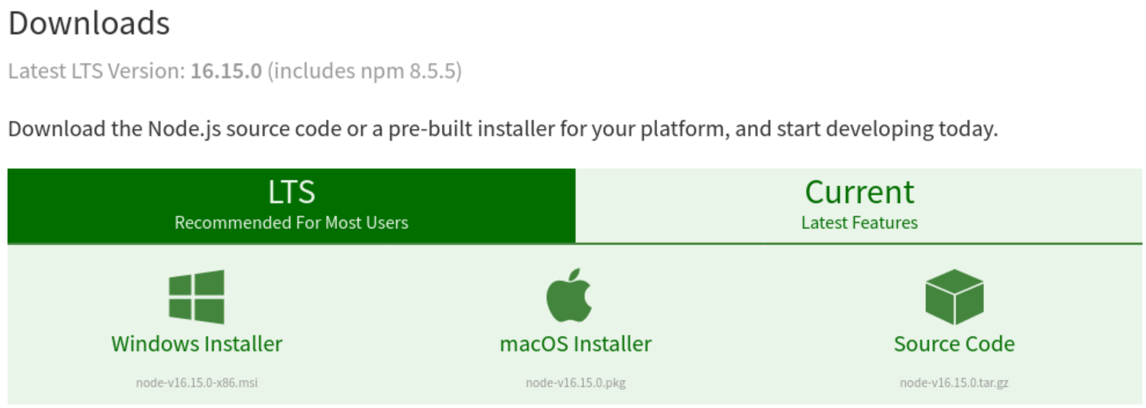


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

Figure 10. Visual Studio Code download page on Microsoft website.

The next step is to install npm and Node.js. Node.js is a runtime environment where the application written in Javascript is executed and. Npm is the node package manager, a database of packages that are needed for the development process. Both of them can be installed via command line or direct download from the official website of Nodejs as shown in Figure 11.



Downloads

Latest LTS Version: **16.15.0** (includes npm 8.5.5)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Figure 11. Node.js download page on Nodejs website.

Finally, an emulator or a physical device is needed to run the application. There are two available emulators at the moment: iOS emulator which is only available exclusively on MacOS operating system and Android emulator. The iOS emulator can be set up via Xcode IDE on MacOS while the Android emulator can be set up via Android Studio IDE.

4.5 Project implementation

4.5.1 Setting up the environment

Once the project setup is completed, the developer needs to set up the development environment for the project. The first step is the installation of react native app globally by typing the below command in Listing 2 into a command line interface.

```
npm install -g create-react-native-app
```

Listing 2. Using npm to install react native.

In the next step, the developer can navigate to the intended location for the project and create a new react native project with the command below in Listing 3.

```
create-react-native-app <Application Name>
```

Listing 3. Creating a new react native project.

After the project is created, the developer can start the project with the command line `npm start`. Once the server is running, the following command line (see Listing 4 below) will be used to render the application on an emulator or a physical device.

```
// For Android devices
react-native run-android
// For iOS devices
react-native run-ios
```

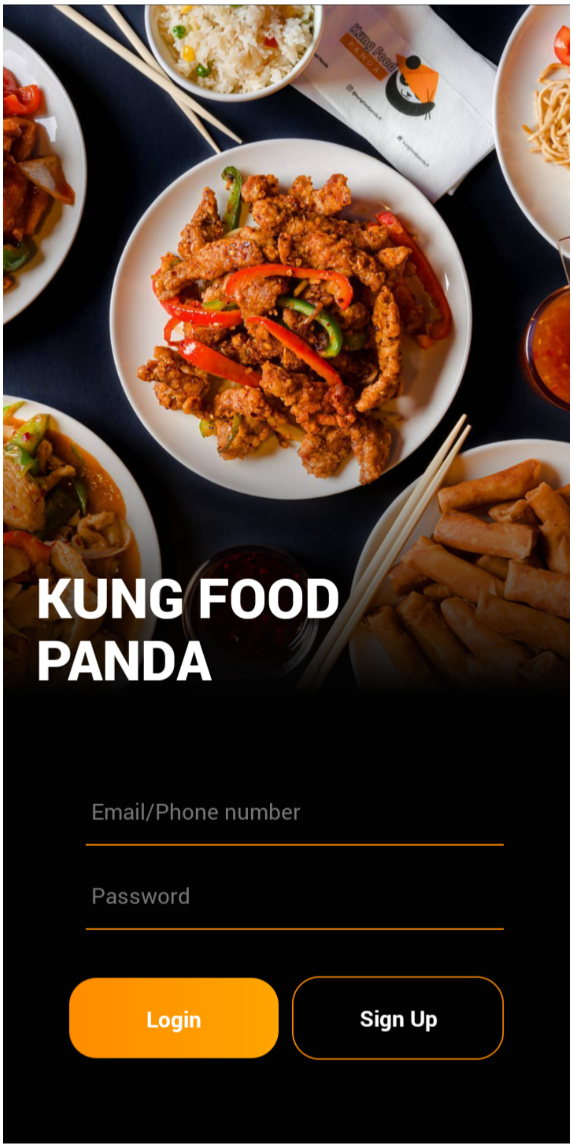
Listing 4. Running react native application on Android/iOS device.

4.5.2 Project development

The first screen that needed to be implemented is the login screen. The purpose of this screen is to authenticate the customer identity with a login and sign up process. The login screen as shown in Figure 12 consists of a React Native Form which requires user inputs for a phone number and password. Two customised buttons for the login and sign up process are also included.

The sign up process is mandatory for a new user in order to use the application. Two inputs (phone number and a secured password) are required in order to complete the registration. The application only requires the user to complete the action once and the input datas will be stored into the application private database.

After the sign up process, the user can start logging in the application with the registered phone number and password. The user information has to match the one in the database so that the login process can run successfully.



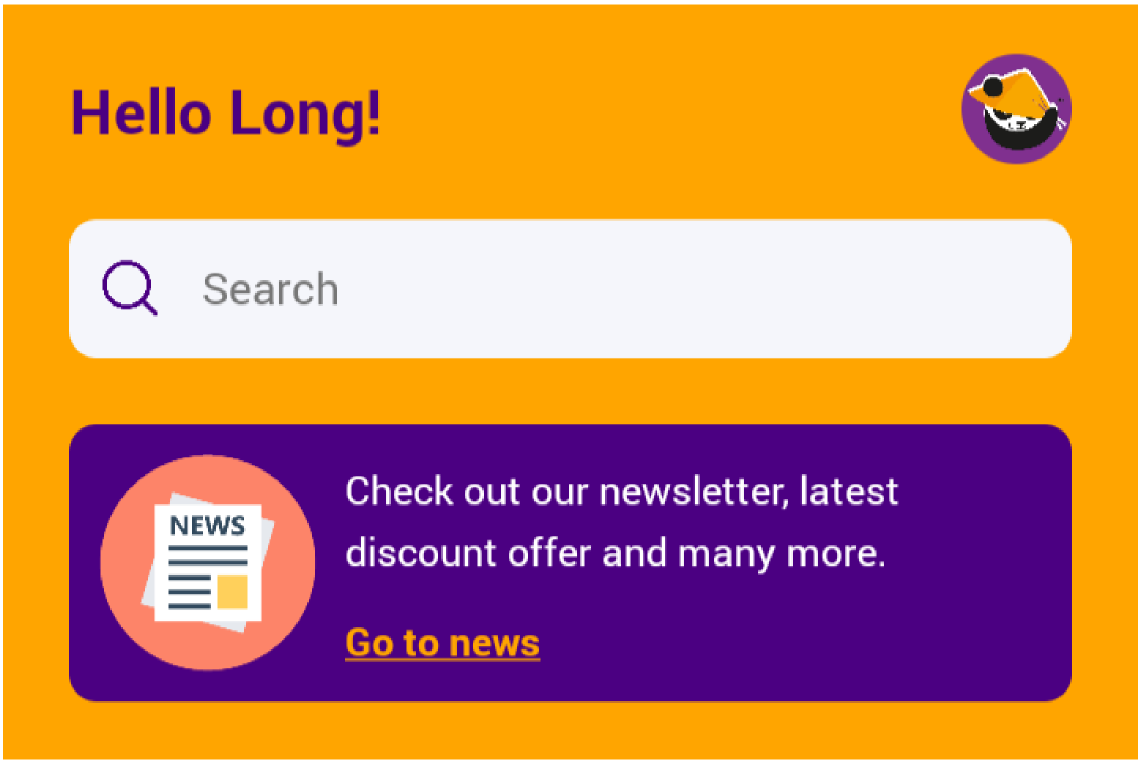
KUNG FOOD PANDA

Email/Phone number

Password

Figure 12 Kung Food Panda application login/sign up screen.

Once the user logged in the application, the application will direct the user to the home screen. It contains three sections: the header section, the main section and the bottom navigation sections.



Hello Long!

Search

Check out our newsletter, latest discount offer and many more.

Go to news

Figure 13 The header section in the main screen.

The header section has a search bar where the user can perform a quick search about all the Kung Food Panda restaurants information (see Figure 13). This search bar requires text input from the device keyboard. When the user types in a certain keyword, a dropdown list will appear with the results or suggestion for the search.

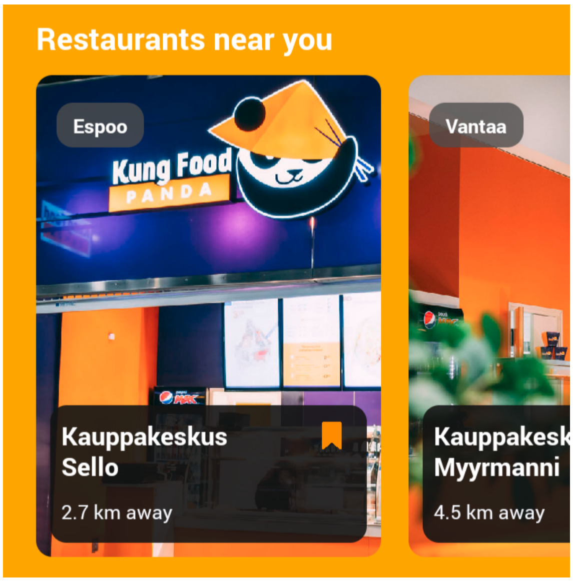
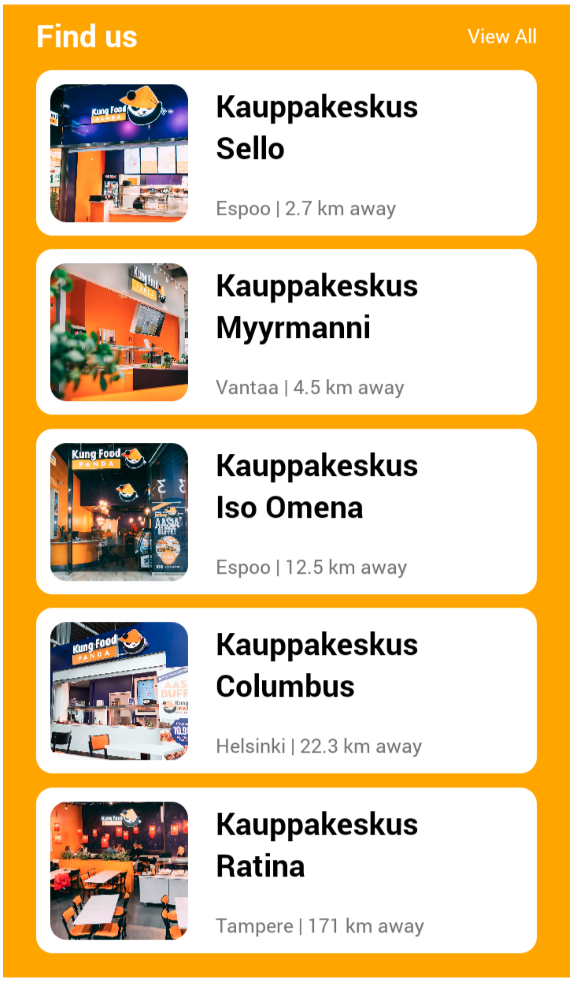

Restaurants near you

Figure 14 The nearby restaurants list.

In Figure 14, the first part of the main section is a carousel list of nearby restaurants based on the customer location. It is presented by a React Native Flatlist component and each item inside the list has an image, restaurant name, bookmark icon, distance and city data. In order to sort the list from the nearest restaurant to the farthest, the below code in Listing 5 was implemented by comparing the distance data of each restaurant and using the `sort()` method. In addition, the list only shows the top three nearest restaurants by using the `slice()` method to pick the first three items from the array data.

```
<FlatList
data={dummyData.restaurantList.sort((a,b) => (parseFloat(a.distance )>
parseFloat(b.distance)) ? 1 : -1).slice(0, 3)}
return (
…
)
/>
```

Listing 5. Using react native method to sort and pick certain data.



Find us View All

Kauppakeskus Sello
Espoo | 2.7 km away

Kauppakeskus Myyrmanni
Vantaa | 4.5 km away

Kauppakeskus Iso Omena
Espoo | 12.5 km away

Kauppakeskus Columbus
Helsinki | 22.3 km away

Kauppakeskus Ratina
Tampere | 171 km away

Figure 15 The available restaurants list.

The second part of the main section is another list that shows all the Kung Food Panda restaurants in Finland (see Figure 15). The restaurant's data are fetched from a mock-up data file for this project but it can also be fetched from an API or a database. The number of items in the list is shown based on the device screen sizes and the user can access the full lists by going to the View All button on the top right corner of this section.

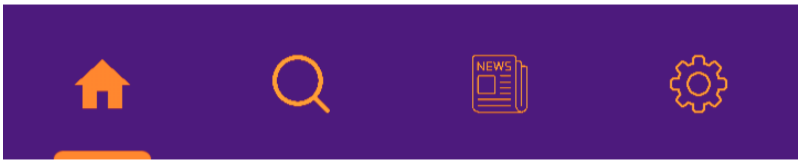
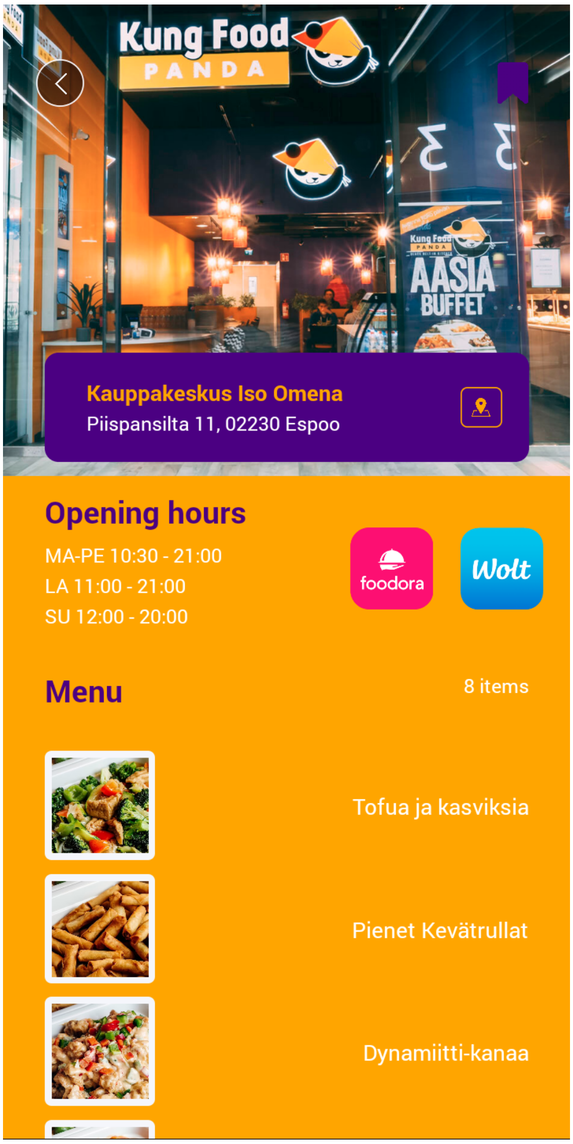


Figure 16 The navigation bottom.

The final section in the home screen is the bottom navigation section as illustrated in Figure 16, which is a navigation bar. It aids the user in transitioning between different screens of the application. Each screen is represented with an icon and the icon is highlighted to notify the user which screen they are currently visiting.



Kauppakeskus Iso Omena
Piispansilta 11, 02230 Espoo

Opening hours
MA-PE 10:30 - 21:00
LA 11:00 - 21:00
SU 12:00 - 20:00

Menu
8 items

Tofua ja kasviksia
Pienet Kevätrullat
Dynamiitti-kanaa

Figure 17 The detailed restaurant screen.

When the user clicks on a specific restaurant on the home screen, they will be navigated to the corresponding detail screen (see Figure 17) for that restaurant. This screen provides all the information about that restaurant such as address, opening hours, delivery option and menu. It creates a unique user experience by utilising other tools such as the open-in-app function. When the user wants to navigate from their location to the restaurant, they can tap on the map button to open the Google Maps application for directions. The Wolt and Foodora icon also directs the customers to the restaurant page on the third party food delivery application.

4.6 Project evaluation

The prototype of the application has received a great number of positive feedback from the customer during the final user testing. Testers displayed satisfaction with the finding nearest restaurant function. As the result of utilising React Native tools, the application has created a smooth design on multiple platforms which assist the testers during the interaction with the application.

This project showcases the most important strength of React Native framework: improving the user experience design. React Native allows developers to build the application UI by creating isolated components and provides useful components libraries and UI toolkits. This can help shorten the development process for any project that uses React Native as the main framework.

The next step for the Kung Food Panda project is to implement new functionalities and possible improvement to the UI. Base on the user testing result, several suggestions has been made for future development such as adding credit card to the application, coupon system and the rating or feedback system

5 Conclusion

In this thesis, a great number of related topics have been analysed. First and foremost, the foodservice market scenario during the last three years has been presented. It provides the challenges of the market during the global pandemic but also gives a new opportunity for different sections of the foodservice industry to rise. Secondly, this study provides a solution for a specific case study of the Kung Food Panda restaurant chain

With the rise of the global smartphone market, the mission of the company is to connect further with their customers through a certain tool. The plan is to create a restaurant application that works on multiple mobile platforms. Several modern programming languages such as Kotlin, Swift or frameworks such as Flutter, Angular JS, Vue JS were taken into consideration. However, React Native arose as a strong candidate for cross-platform mobile application development.

React Native is a Javascript framework that has been created and developed by Facebook since the first quarter of 2015. After several years of development, it has built a strong developer community with its cross-platform development, code reusability features, hot reloading and stable release every few months. React Native provides a competition for other native applications with its equality in performance. For this reason, React Native is the chosen framework for the Kung Food Panda application.

However, the disadvantages of React Native should also be discussed. As a fairly young technology, it received several significant changes in the last few releases. For instance, a lot of old components became incompatible with the new version of React Native and it creates a harder work for the code maintenance. Another problem of the framework is debugging, it requires additional tools such as Flipper to manage the debugging process.

Despite the above issues, Facebook has provided continuous development for the technology and support for the developer community. This means that React Native is still a promising framework for the future of mobile development.

References

1. Mobile Operating System Market Share Worldwide [Online]. URL:https://gs.statcounter.com/os-market-share/mobile/worldwide
2. Bonnie Eisenman. Learning React Native: Building Native Mobile Apps with JavaScript; Chapter 1; 2015.
3. The History of React.js on a Timeline [online]. URL:https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/
4. Introducing React [online]. URL:https://www.kirupa.com/react/introducing_react.htm
5. Chart: How coronavirus is devastating the restaurant business [online]. URL:https://www.vox.com/recode/2020/3/16/21181556/coronavirus-chart-restaurant-business-local
6. Coronavirus: The world in lockdown in maps and charts [online]. URL:https://www.bbc.com/news/world-52103747
7. Maria Castroviejo Citing references [online]. Impact of Coronavirus on European Foodservice.
8. Understanding food delivery growth across the US, UK, Canada and Australia [online]. URL:https://trends.edison.tech/research/global-food-delivery-2021.html
9. Food Delivery App Revenue and Usage Statistics [online]. URL:https://www.businessofapps.com/data/food-delivery-app-market/
10. J. Clausnitzer Citing references [online]. Smartphone penetration rate in Finland 2017-2021
11. Valassis Research: Consumers More Likely to Dine out if Restaurants Sweeten the Deal [online]. URL:https://www.businesswire.com/news/home/20190430005616/en/Valassis-Research-Consumers-More-Likely-to-Dine-out-if-Restaurants-Sweeten-the-Deal
12. DIVAGAR N. EVERYONE CAN HACK -1; Page 33; 2020.

13. Usage statistics of JavaScript as client-side programming language on websites [online].
    URL:https://w3techs.com/technologies/details/cp-javascript/
14. React documentation: Introducing JSX [online].
    URL: https://reactjs.org/docs/introducing-jsx.html
15. What is JSX? [online].
    URL:https://jsx.github.io/
16. React Virtual DOM Explained in Simple English [online].
    URL:https://programmingwithmosh.com/react/react-virtual-dom-explained/
17. Bonnie Eisenman. Learning React Native: Building Native Mobile Apps with JavaScript; Chapter 2; 2015.
18. How to Create Apps for Android and iOS Using React Native? [online].
    URL:https://www.neoito.com/blog/how-to-make-an-app-for-android-and-ios-using-react-native/
19. React documentation: Components and Props [online].
    URL:https://reactjs.org/docs/components-and-props.html
20. React documentation: React.Component [online].
    URL:https://reactjs.org/docs/react-component.html
21. React lifecycle methods diagram [online].
    URL:https://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/
22. ReactJS: Props vs. State [online].
    URL:https://lucybain.com/blog/2016/react-state-vs-pros/
23. React Props vs State [online].
    URL:https://programmingwithmosh.com/javascript/react-props-vs-state/
24. Non-functional Requirements [online].
    URL:http://users.csc.calpoly.edu/~jdalbey/SWE/QA/nonfunctional.html
25. Google’s Material Design documentation [online].
    URL:https://material.io/design/introduction
26. What is an IDE? [online].
    URL:https://www.redhat.com/en/topics/middleware/what-is-ide