

# Cross-Platform Analysis and Development of Online Catering Platform ( Kunyahku )

Indra Swarna, James Purnama, Randy Anthony

Department of Information Technology, Swiss German University, Tangerang 15143, Indonesia

## Article Information

Received: 29 January 2021  
Accepted: 17 March 2021  
Published: 18 March 2021  
DOI: 10.33555/jaict.v7i2.106

## Corresponding Author:

Name: Indra Swarna  
Email: indra.swarna@student.sgu.ac.id

ISSN 2355-1771  
eISSN 2723-4827

## ABSTRACT

*Kunyahku is an online catering platform that provides catalogs of menus from local caterer near the user and become the distributor that will take the order and deliver them. In developing an application, the most important aspect for small start-up companies is to have a development stack that can deliver the app into multiple platforms by using a single code base to reduce the cost of developers, improve efficiency and reduce the time to develop the application itself. To provide the necessity of building an application for Kunyahku, this paper is aimed to find the most suitable cross platform framework for developing the Kunyahku application. This research focuses on comparison between existing popular cross-platform and proving the suitable one to be used in development cycle in the case of Kunyahku requirements. Sets of test and comparison were conducted in order to compare the most suitable cross platform between React Native and Flutter such as render test and comparing each advantage coming out from using each of the cross-platform. After all the requirements are gathered and then the most suitable framework will be chosen which is Flutter. Performance analysis and the benefits of using Flutter framework are elaborated and thus proving it is the most suitable development stack to be used in Kunyahku case and the development time can be reduced since it is a cross-platform technology that output the application into three different platform rather than managing separate platform for each different one. Testing shows that flutter also met all the requirements thus proving flutter address the problems of scalability and faster development compared to developing each native platform one by one.*

**Keywords:** *Cross-Platform Development, Catering Application, Mobile Development, Flutter, React Native*

## 1. Introduction

Catering is a necessary activity in the market. Almost all events (small and big) require catering to supply their consumption. The rise of technology nowadays has also help to improve this industry and there are many caterings that is online based nowadays. The growing start-up company that involved in the catering business have also appeared and exist in the market which have provided those who need to find catering to supply the food for their events.

Firstly, the current online catering platform usually uses simple interface such as website and there are still room for improvement that can be used as a material for future works and the mobile application used by most company are using native apps which uses different resources to compile their code into separate platforms such as android and iOS which each takes a lot of time for developing it. Also, the design of native apps usually is limited that makes it less attractive to the user. The amount of time invested on developing each platform is long and also costly thus disrupt the momentum of the start-up company that need follow trends otherwise will be left out and some start-up may not able to fund it since IT start-up initially does not have a lot of budget to support development and require a fast launching time thus in this case require such a framework that support all of these problem which is Cross-Platform Framework that are able to solve this problem such as Flutter.

Building IT Start-up requires a good launching strategy and it need a fast idea generation, validation, and market research. Usually, most of small IT start-up that have less than 10 people in the team tends to face problems such as limited time and budget to develop their application which is very crucial. The momentum might disappear if a lot of time are wasted while developing an application thus require a fast development cycle and good result. Having to meet the current standards of developing an application, there is a need to have multiple platform (Android, iOS, and Web) for Kunyahku able to compete in the market whilst fulfilling the user requirement for Kunyahku.

Furthermore, the time needed to create each platform is consuming a lot of time and if each platform are developed using their native language such as Android using java and iOS using swift, there is a need to hire different programmers or a need for the developer in the team to learn multiple language and managing multiple platform for the application which will be complicated in terms of updates and further improvement.

Each platform may differ in terms of applying the use case into to the platform. The problems of some implementation of library also become issues since each platform have different company that manage it such as Android managed by Google while iOS by Apple, and there will be no uniformity in the code and the styling in each platform may be differ from one to another.

The objective of the research is to prove that by using cross-platform framework, it can help to improve efficiency in managing and developing codes for three different platform. The amount of development time can also be reduced since there will be lesser code to be written as there will only be a single code base that can output three different platform at the same time. Also minimizing the amount of effort and cost towards creating multiple application that have the same functionality. Analyzing Kunyahku requirements and choosing the most suitable cross-platform framework for the case of Kunyahku. In addition, to prove that the final product developed using the cross-platform framework is acceptable and really solve the problems mentioned in the previous section. Showing that all the requirements can be fulfilled, and the quality of the product are comparable towards the native performance.

## 2. Related works

React JS is a popular web development language and it is widely used by many developers in the world. Then after several years after ReactJS was launched, react native was launched as an internal hackathon project of Facebook to unify the development process for iOS and Android. It is currently become the strongest communities in open-source language for cross-platform. With React Native we build apps for both Android and iOS but we do not have to write any Java or Swift code but using only one code which is JavaScript which improve time efficiency of development (Hansson & Vidhall, 2016). The base code of React Native is JavaScript and view rendered using their technology called JSX. Developed and maintained by Facebook from 2016, React native is being used in thousands of apps and developers since it brings modern web techniques that are not available in native development of mobile which comes without any side effect. Having JavaScript as its foundation, React Native provides much shorter learning time since it uses a language that is widely known to people. React Native embraced the interpreted paradigm, since all user interfaces in the app were actual native UI components and are not Web Views or HTML5 solutions.

Flutter is a cross-platform framework which is developed and maintained by Google. It is released at 2016 and currently it is one of the cross-platform frameworks that is quite popular in the developer community. Google defines Flutter as a portable UI toolkit for building beautiful natively compiled applications for mobile, web and desktop from a single codebase (“Flutter - Beautiful native apps in record time,” n.d.). Flutter can run on Android, iOS and currently the flutter team are developing this framework to be able to run on the web(beta) as well as desktop. Flutter uses Dart programming as the base of their programming language and Dart is almost looks like JavaScript which provides easy learning for developer who have JavaScript background since it also supports object-oriented programming concept.

Dart is maintained and developed by Google and it was release at 2011 and is widely used in google environment. Furthermore, Dart is actually intended to replace JavaScript, so it also adopts the JavaScript standards. Flutter main principles is that every view rendered inside flutter are called as a widget. A widget is the basic building blocks of the view in flutter user interface. This widget is an immutable declaration of part of the user interface (Flutter, 2019). The widget will use tree structure which mean there will be a widget in a widget that have child-parent relation.

The trends of developing application using cross platform framework have been increasing and there are several open source frameworks that support this. Having 2 different operating system that is widely used in the market which is android and iOS require developer to both develop their application that can support that operating system. Furthermore, the performance of native cross platform framework is still questionable whether it can work as good as the native’s application or even better. This research of performance analysis conducted by the author, he attempts to compare the performance wise of Flutter and React native.

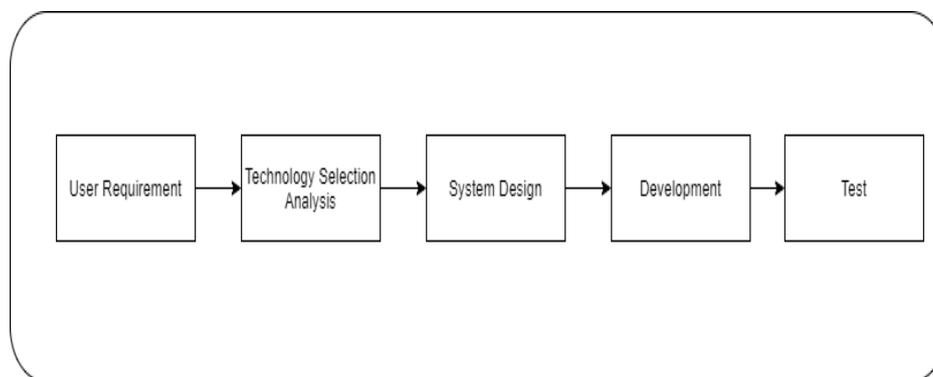
He has summarized the result of his research that compares the performance of running 2 application that have same features and load in both flutter and react native. He mentioned that flutter even before the test should be faster since it does not require a bridge to render the application unlike react native that requires JavaScript. He concluded that Flutter framework performed better than React Native in terms of number of frames dropped, when benchmarked with applications that contained very large lists (Jagie, 2019).

Flutter is a cross platform framework used to create mobile, web and desktop application in a single code base. Designed by Google, flutter have gained a lot of improvement and constant updates which makes it popular nowadays. The author tries to analyze internal and overall architecture of Flutter. He compiled from introduction of Flutter until how to build an app using Flutter. He mentioned that after the end of the research, that flutter internals and the fast development cycle makes flutter become very competitive in cross platform aspect (Dagne, 2019).

React native is a framework that allows the development of mobile application using the concepts that are used in creating a web framework (ReactJS). Some of the language are taken from web development but there is also native specific library of the platforms. The author tries to analyze the performance and looks of application developed using cross-platform React Native and the native language. Ultimately, (Hansson & Vidhall, 2016) believes that by using React Native, with a single development for both Android and iOS the results show similar experiences of design and looks and slight inferiority in terms of performance. The code used was around 75% for all of the platform while the 25% was for app- specific library code.

### 3. Methodology

The Kunyahku online catering platform prototype will be using the pattern of standard development process which are be able to cope with the required timeline and the minimum viable product that will be generated using this process. Figure 1 illustrates the chart explaining about the development process that are held in Kunyahku.



**Figure 1.** Development Process Flow Chart

#### 3.1 System Development

##### 3.1.1 User Requirement

The requirements needed for Kunyahku case are collected from the case where there are many local caterers that is scattered and does not have a hub that could connect them to the market and likewise the user also could not find caterer near them. The system requirements are listed as below:

**Table 1.** Development Process Flow Chart

Function Group	Description
Login Module	Users are able to login with their own account
Home Module	Users are able to view the food available from various vendors
Cart Module	Users are able to add, view, remove, edit the cart items that have been made
Order Module	Users are able to add, view, remove, edit the order that have been made
Settings Module	Users are able to view its profile and history of orders and also manage their addresses for the order
Payment Module	Users are able to complete the order by paying their order and show the status of the payment itself

### 3.1.2 Technology Selection Analysis

Based on the user requirement elaborated previously, the development of the application need to solve those problems, and there are two cross-platform frameworks that is suitable for it which is React Native developed by Facebook and Flutter developed by Google. These frameworks are a cross-platform app development language that are popular in the market.

In choosing the platform that will be used, the amount of time needed in the developing the system should be fast and efficient and can output a quality code that can work perfectly in the market. Since the developers in the team are limited, it also needs to be a platform that is easy to learn in order to minimize learning time and allow for developer to directly jump into creating the application without having to learn all the hierarchy and new language and so on.

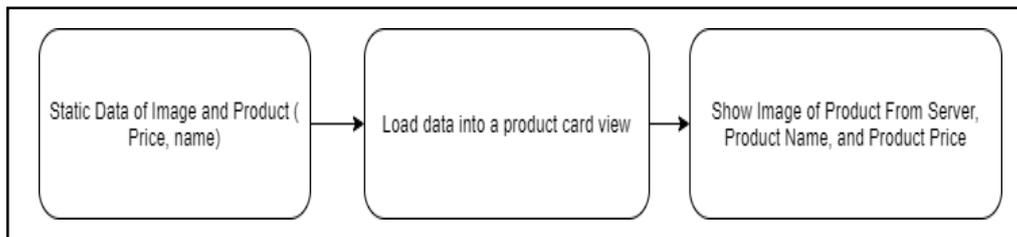
Both Flutter and React Native are similar in terms of the end product, both platforms can produce both Android and iOS application, but the application needed for Kunyahku are the platform that user can experience better and feels comfortable to use. Provided with good backers of community both of the framework have its own advantages and disadvantages but Flutter hold a major key where its engine are more modern and does not need a middle man while React uses a middle man to bridge its communication to its native modules thus create inferiority in performance compared to Flutter.

In terms of code reusability, both platforms do support a good system where the code can be referred by another module and in terms of cross-platform supports, both provide the ability to deploy the application into Android, iOS, and also Web. The documentation provided by flutter is very detailed and specific while in React Native, most of the time it requires additional research on how to do something. Flutter provides a beautiful ready to use templates while React Native only provide base template that requires additional styling and more dependent into third party libraries. Since flutter provides more ready to use components the time required to build a working app is faster than react.

The comparison between these 2 platform are very close due to the fact that both can provide the requirements needed in building the Kunyahku application but in terms of performance wise and development time it is shown that Flutter are more up for the task due to its bottom-to-top architecture, it is more efficient and less resource-demanding (Wu, 2018) and it also provide higher scalability and less time required to build it which is very important.

### 3.1.3 Performance Test of Cross-platform Framework

In this section, the performance of flutter and react native will be compared side by side in terms of performance (CPU, RAM Usage, and render time). This performance test will be conducted using a same scenario where each of the platform will use its own library then the result will be generated after successfully rendering and building the app then monitored. The scenario used for the testing are as described in the figure 2.



**Figure 2.** Test Scenario

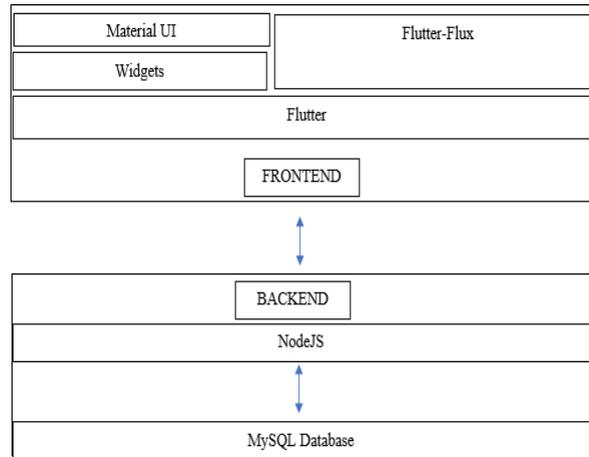
In each of the platform ( Flutter and React ), both needed to use this test scenario where first it should fetch the product list from Kunyahku backend then create a listing card for each of the products and then in each listing card it should consist of the product name, image and also the price. After each of the platform have followed the test scenario then result will be gathered using android profiling monitor from Android Studio. Render time from clicking the application and into the state all the frames are rendered is collected. The test are conducted using a physical device with an Android 9.0, 8GB of RAM, and Snapdragon 855 Chipset.

The performance of flutter when the product is successfully built and rendered it shows that the CPU usage is around 5% and the memory usage is 205.1 MB. The render time of flutter application and it is calculated from the point of the program was clicked until all the frames are rendered. The time was recorded using stopwatch for 10 times and the average time of flutter render time is around 1.87 second which is quite fast. The performance of react native resulted from rendering the view in the physical device and it takes 9% of CPU usage and the memory usage is 187.7 MB. The render time from using react native from the point of the app is launched until the frame is rendered. There are 10 trials and the average resulted from the test is it take 3.395 second to fully launch and render the view in react native

## 3.2 Design

The design in this section are created to cope with the user requirements. The design also follows the system selected in the technology and selection analysis and includes architecture diagram and use case diagram of the feature that should be included base on the user requirements.

### 3.2.1 Architecture Design

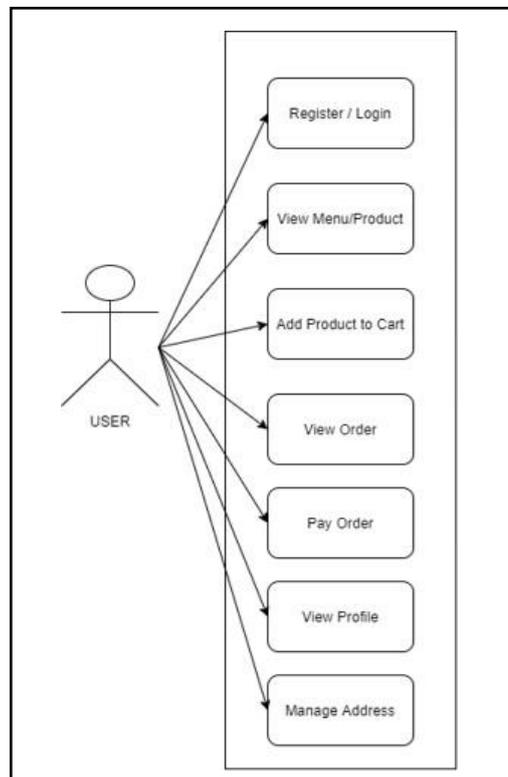


**Figure 3.** Architecture Design

As illustrated in Figure 3, Kuyahku application will run on the Flutter framework then using widgets to render the UI and styled using material UI then the state that controls the flow of the data will be managed by the flutter-flux. The data will be provided from the backend system using NodeJS and MySQL database that will store the data. The backend system is managed by the backend team of Kuyahku.

### 3.2.2 Use Case Diagram

The feature that will be developed using the selected technology should include the features mentioned in the figure 4.

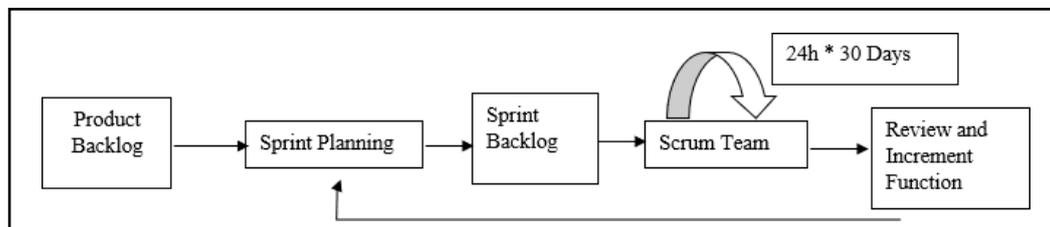


**Figure 4.** Use Case Diagram

The feature will be built into view page where user can navigate through each of it.

### 3.3 Development

Development strategies used is Scrum Agile to cope with changes every day and to better adapt towards the need of the business requirement.



**Figure 5.** Scrum Agile

### 3.4 Software Testing

Before the product can be launched into the market, there are two test that will be conducted in order to make sure the user requirement have been fulfilled which are Alpha testing and Beta testing.

In Alpha test, the internal teams will sit together and evaluate the application and looks for any mistakes, error, bugs and design that is not up to the task and will be improved before launching it in the market. This is to simulate the flow of the application and a chance to ask for improvements and clarify any issues in real time.

In Beta test, the application will be released into a small group of people to be tested. This is intended to see the user reactions towards the application functionality, design, experience and flow.

## 4. Result

### 4.1 Research Resolution

The learning curve to learn this framework is very fast rather than learning Android (Java) or iOS (Swift) framework since the base language of flutter (Dart) is based on the most popular language which is JavaScript and is using OOP (Object Oriented Programming) model. There are a lot of similar pattern of use like JavaScript in Dart language which makes it very easy to comprehend. Unlike other framework, Flutter does not have visual designer which means all the design needs to come from code which is quite hard to manage since it requires the developer to manage all the design from codes unlike android that have drag and drop system for design.

Flutter framework provide the ability to compile its code into three framework which is very beneficial to reduce development time and it really works very smooth. Compared to developing every framework one by one, flutter reduces the development time and cost by 1/3 that are very beneficial for small start-up that have few developers and since they do not have a lot of resources to hire programmers and also the application quality is pretty good as well. Given a lot of ready to use libraries and third-party libraries makes it even faster to just grab anything required for the application and apply it into the code. Customizing the design and managing its style is very flexible because flutter itself providing all these possibilities of customization.

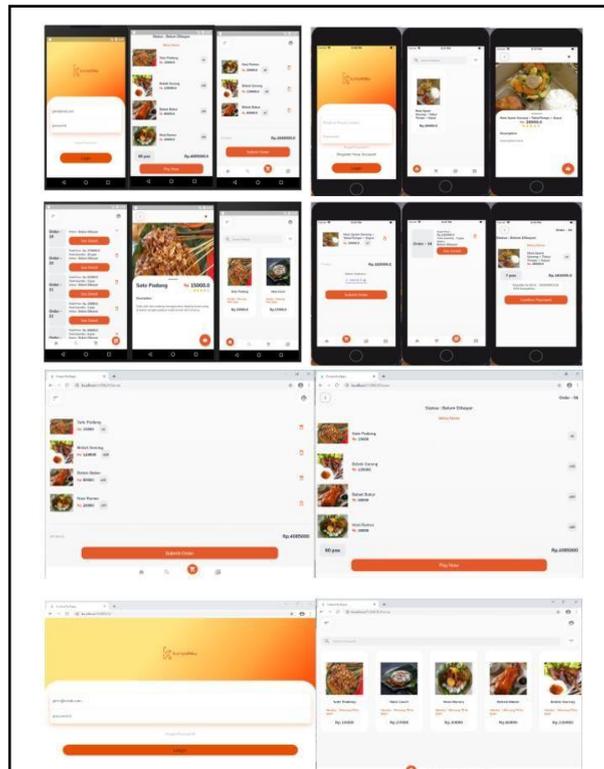
Framework	Flutter (Dart)
Required Time	3 Weeks
Application Quality	Great
Intended Platforms	Android, iOS and Web
Provided Library	Many ready to use library

**Figure 6.** Development Result

In 3 weeks, the minimum viable product of Kunyahku application that already fulfil the user requirement given in the previous chapter is produced whilst if it is developed in each of the framework then it will take much longer time. The Quality of product also is not so inferior compared into the native platform since flutter compiled the code into native code by using their engine unlike other cross platforms such as react that uses wrapper. Even tough, some libraries do not support every platform, the possibilities of changing into other library that have same functionality is provided so it does not affect much of the functionality but the design.

#### 4.1.1 Single Code Base for 3 Platform

Having to only have a single code to manage and compiled into 3 platform is actually a bit hard to achieve because there are some library dependencies that can only work on specific platform, thus limiting the ability of the programmer to only use common library that can work in each platform or separate the different feature using each working library. The flutter web is the most problematic since it does not support many libraries compared to the flutter android and iOS since they have different system hierarchies that uses some of their native functionalities that is different from web hierarchy. In this case, the best way is separation of critical library and apply different measure for it.



**Figure 7.** Application Result in iOS, Android, and Web

Overall, by using the flutter framework, the code that can be used for multiple platform is around 85% and the 15% is the app specific code to access the native library.

#### 4.1.2 Problems in Integrating Libraries of 3 Platform

Some native libraries that are only available such as Camera/Gallery module in mobile while in web it uses File directly from storage causes some integration problem since some library may not support this feature. Each native uses different approach as well when handling some library such as google maps. Even though there are libraries that able to comply with some features but still the difference in native libraries and converted library is that the native library is specifically meant for the native device itself but flutter have converted some of that native library but there are still libraries that are not meant to be used outside of that platform specific target.

### 4.2 Application Test

In order to make sure everything is set in place and in line with the needs of the application, Alpha & Beta test is performed. Alpha test is done by the internal team of Kunyahku and are focused to test the cases that are crucial in the application and satisfy all the requirements needed for the Beta launch.

#### 4.2.1 Alpha Test

Having to check every crucial test case in the application, the user that participate in this test are internals member of Kunyahku. This test makes sure everything is at the point of passable and are working. The test question is consisted of 11 test cases where the users are interviewed and asked to give a feedback from scale 1 to 5. The test is done by 3 internal member team and the result is as following:

User	Kunyahku Application
User 1 Average	5
User 2 Average	4.9
User 3 Average	4.5
Total Average	4.8

**Figure 8.** Alpha Test Result

#### 4.2.2 Beta Test

Additionally, to prepare for the big launch and calculate the readiness of the platform, a Beta test are done in order to make sure the application is in a step where it can satisfy the users and reliable enough to run all the required features. The test question is made from scenarios of user satisfaction and view on the application. There are 5 people that is interviewed and directly asked for feedbacks from scale 1 to 5 towards the experience on using the application.

User	Kunyahku Application
User 1 Average	4.43
User 2 Average	4
User 3 Average	4.57
User 4 Average	3.86
User 5 Average	5
Total Average	4.37

**Figure 9.** Beta Test Result

## 5. Conclusion

Through this thesis, the analysis of how using cross-platform are beneficial in meeting the Kunyahku business process is clear that it is very beneficial and efficient. The result that is being used for Android and iOS platform reduces half of the development time and Flutter is the most suitable framework. Flutter able to provide the ability to further scale the application and the performance is much better compared to the other cross platform. Flutter also show that the quality of code is better since it converts the code into native code.

Furthermore, through the Alpha & Beta testing it is proven that the use case required for Kunyahku case is fulfilled and throughout the development process it is shown that flutter is very customizable and easy to learn thus allow for faster time in developing it. In alpha test the team has concluded the product built using flutter is enough and works well. In beta test, the application also shows a good feedback that meet all the criteria for launching the application thus fulfilling all the user requirements.

## References

- Dagne, L. (2019). *Flutter for cross-platform App and SDK development*. Helsinki, Finland: Metropolia University of Applied Sciences.
- Hansson, N., & Vidhall, T. (2016). *Effects on performance and usability for cross-platform application development using React Native*. Linköping, Sweden: Linköpings Universitet.
- Jagiello, J. (2019). *Performance comparison between react native and flutter*. Umeå, Sweden: Umeå University.
- Wu, W. (2018). *React Native vs Flutter, cross-platform mobile application frameworks*. Helsinki, Finland: Metropolia University of Applied Sciences.