Comparative Analysis of Use States of Mobile Application Development Methods

Hakan Ünal

Master Thesis Student, , ISTANBUL AYDIN University Department of Computer Engineering,Florya, Istanbul, Turkey

ABSTRACT: Mobile application development methods are divided into two as Native Application Development and Cross Platform Application Development. Native platform application development environment for Android has Android Studio, and native platform application development environment for iOS has Xcode compilers. Cross application development environments for Android and iOS include environments such as PhoneGap, React Native, Angular, Ionic and Flutter. In this study, Native and Cross Platform Application Development methods were used to adding a simple two numbers, as a more complex application, todo list application working with Firebase Real Time Database was made and comparative analysis of mobile application development methods was made by giving application scenarios.

KEYWORDS: Mobile Application Development, Android Studio,Xcode, Native Platform Application Development,Cross Platform Application Development.

I. INTRODUCTION

Various methods have been developed on Mobile Application Development. The number of these methods is increasing gradually. In addition to methods, new programming languages, new design tools and new compilers are emerging. Mobile application development methods are divided into Native Application Development and Cross Platform Application Development. Different programming languages are used depending on the method chosen. There are Native Platform for Android operating system, Java and Kotlin programming languages for Application Development method [1-2]. Native Platform for iOS operating system, Objective-C and Swift programming languages for Application Development method [3]. There are programming languages such as Cross Platform for Android and iOS operating systems, JavaScript, TypeScript, C#, Html, CSS for the Application Development method.

II. MOBILE APPLICATION DEVELOPMENT ENVIRONMENTS

2.1 Native Platform Application Development Environment for Android

As an Android application development tool, Android SDK (Software Development Kit) was installed on the Eclipse compiler and application developments were made, and Android Studio was started to be used after the Android Studio compiler was released in May 2013 [4]. This compiler is based on Intellij IDEA and designed for Android application development and it has ready-made systems for installing Android SDK. These systems can update and clean up the old SDK when the new SDK is released. In its first release, only Java programming language was used as the programming language, and then the Kotlin programming language was added [2]. With the AVD Manager (Android Virtual Device Manager) in Android Studio, applications can be tested with a phone or tablet running in a virtual environment. With the XML Editor in Android Studio, controls such as textview, button can be added to the application with a drag and drop logic, and events such as clicks and scrolls can be created with these controls. With the Profiler in Android Studio, problems related to the performance of the application on the phone can be seen.

2.1 Native Platform Application Development Environment for iOS

Native platform application development for iOS is carried out using the Xcode compiler. Objective-C is used as the programming language and Swift programming language has also been added [3]. With the iOS Simulator included in Xcode, applications can be tested with iPhone or iPad running in a virtual environment. With the Interface Builder in Xcode, controls such as labels

and buttons can be added to the application with a drag and drop logic, and events such as clicks and scrolls can be created with these controls. With Instruments in Xcode, problems related to the performance of the application on the phone can be seen.

2.2 Cross-Platform Application Development Environment for Android and iOS

PhoneGap is among the first frameworks to come out among cross-platform applications. However, PhoneGap has been deprecated [5]. It is the framework that allows application development using Xamarin C# programming language [6]. React Native is a framework that enables cross-platform mobile applications with the JavaScript programming language developed by Facebook and converted to open source as of March 2015 [7]. Facebook's Facebook Groups and Facebook Ads manager applications are developed with React Native [7]. It is a framework developed by using client-side JavaScript and MV* (Model View Whatever) architectural design pattern developed by Angular Google [8]. Ionic AngularJS is a framework that allows application with JavaScript programming language using MVC (Model View Controller) architecture.Flutter is a framework that enables cross-platform mobile applications using Dart programming language developed by Google [9].

Table 1: Native Platform Application Development vs Cross Platform Application Development [8]

Native	Cross
Animations, slides are faster.	It is slower than native platform applications.
The development process is slow.	The development process is faster.
Resource cost is high.	The resource cost is minimal.
It is difficult to update.	It is easy to update.
Each one requires a different development environment.	One development environment is sufficient.

2.3 About Cross Platform Application Development

The reason why cross-platform application development methods are needed is understood from the features such as being fast and low in cost as indicated in Table 1 [8]. In order to develop native platform applications, it is necessary to learn the programming language and compiler of that platform. It is preferred as it is sufficient to learn a language and a compiler for cross application development. The working logic of cross applications usually runs the application in a Web Browser (WebView) [10]. This Web Browser works in a Chrome-based structure on Android and a Safari-based structure on iOS. Since the application is a Web Browser, changes can be made within the application without installing a new version of the application, but it is necessary to install a new version for changes such as application permissions. Also, when new versions of Android and iOS are released, Cross platforms may not adapt immediately. In order to develop the application related to the native platform, it is necessary to know the programming language and the compiler of that platform. When new versions of Android and iOS are released, native platforms adapt immediately. Because at the same time, compilers are updated.

In this study, using mobile application development methods, a simple mobile application showing the sum of two numbers and a more complex application to do list application were developed, and mobile application development methods were examined and comparative analysis was made. Data in todo list application is stored on the Firebase Real Time Database. Firebase Real-Time Database offers dynamic, extensible functions and real-time data insertion, update and deletion [11]. Login and sign-up screens in the todo list application are made using Firebase Authentication. Firebase Authentication provides secure login by providing multiple authentication methods such as e-mail / password, social login with major providers (Google, Facebook, Twitter…) and also provides anonymous login [11].

III. MOBILE APPLICATION DEVELOPMENT METHODS

3.1 Native Platform Mobile Application Development Method with Android Studio

Figure 1 shows the design and code side of the twonumber addition mobile application on Android Studio. Designs are kept in files with .xml

extension. Codes are kept in files with .java extension.It is necessary to name the components in thedesignsasfirstNumberEditText, secondNumberEditText and match the code side with these names. After this matching, actions such as clicking, sliding and holding can be performed with these components. Application permissions are written into the AndroidManifest.xml file.Figure 2 shows the libraries used and practical screen usage in the code section of the todo list mobile application in Android Studio. Unlike adding two numbers, it seems that more lines of code and more libraries need to be added to the project. Firebase has a direct library for Android Studio and is shown in Figure 2.

3.2 Native Platform Mobile Application Development Method with Xcode

Figure 3 shows the design and code side of the mobile application on Xcode. Designs are kept in files with .storyboard extension. Codes are kept in files with .swift extension. There is no need to name the components in the designs such as firstNumberTextField, secondNumberTextField and these naming and matching are done on the code side. After this matching, actions such as clicking, sliding and holding can be performed with these components. Application permissions are written in the Info.plist file. Figure 4 shows the code side of the todo list mobile application on Xcode, the libraries used and the screenshot of the applicaiton. Unlike adding two numbers, it seems that more lines of code and more libraries need to be added to the project. Firebase has a direct library for Xcode and is shown in Figure 4.
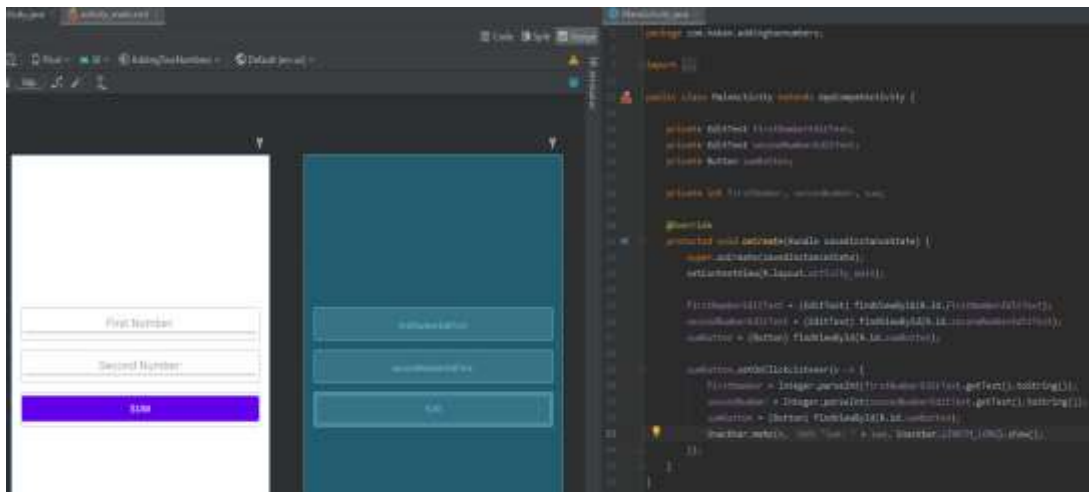

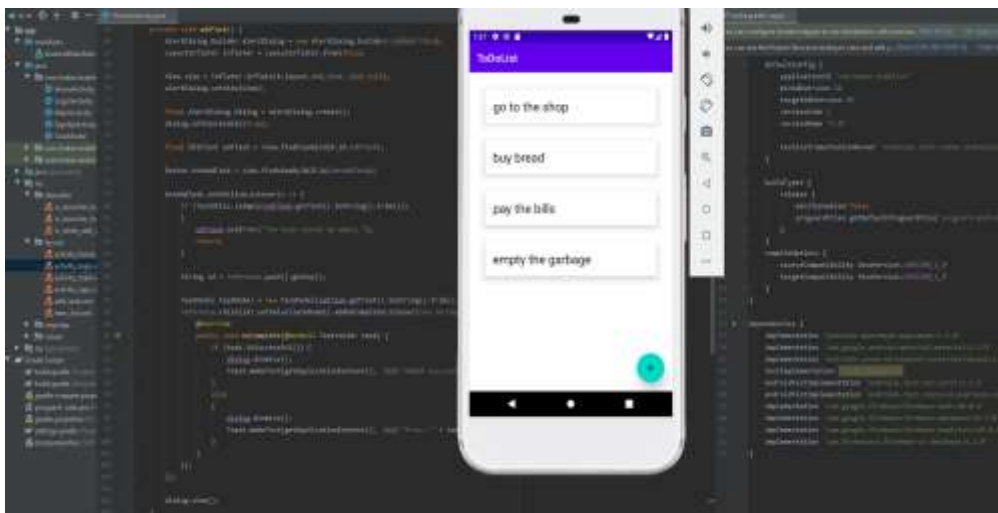**Fig. 1** Screenshot of Android Studio Two Number Addition Application

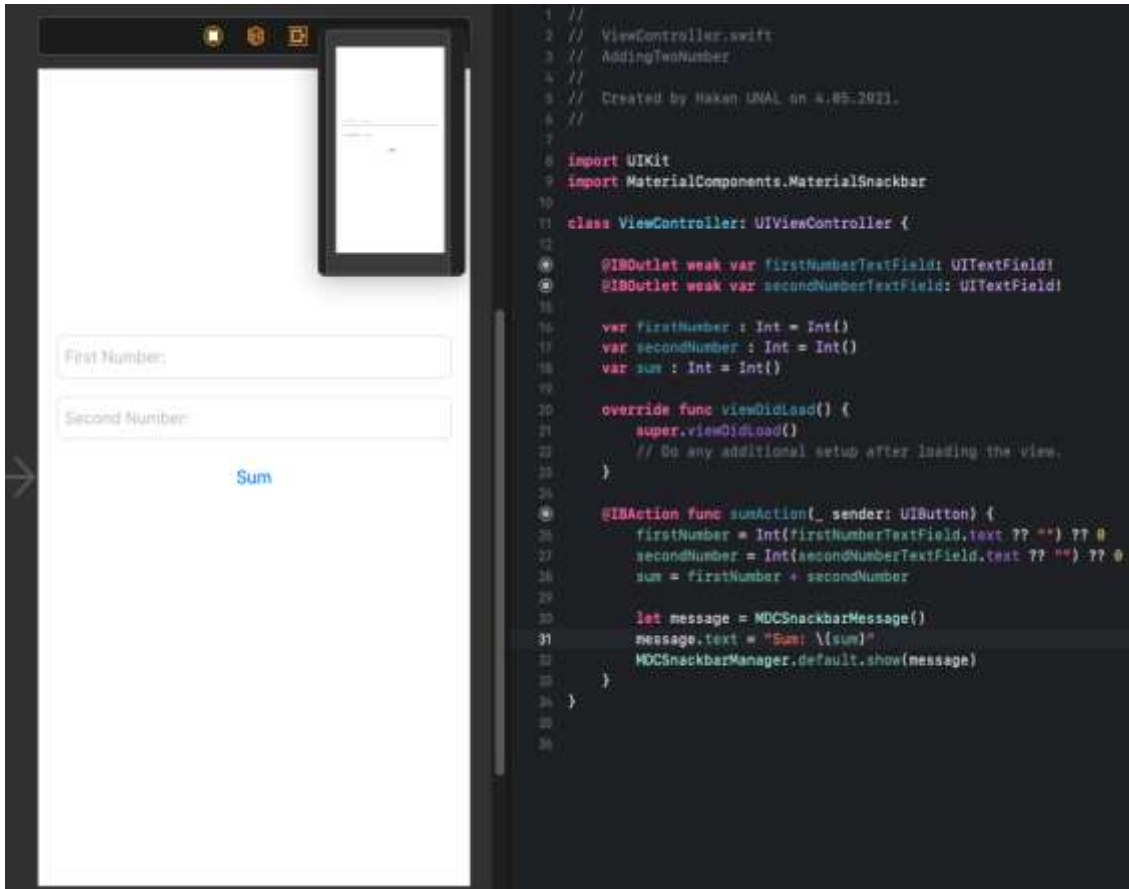**Fig. 2** Screenshot of Android Studio To Do List App

Fig. 3 Screenshot of Xcode Two Number Addition Application
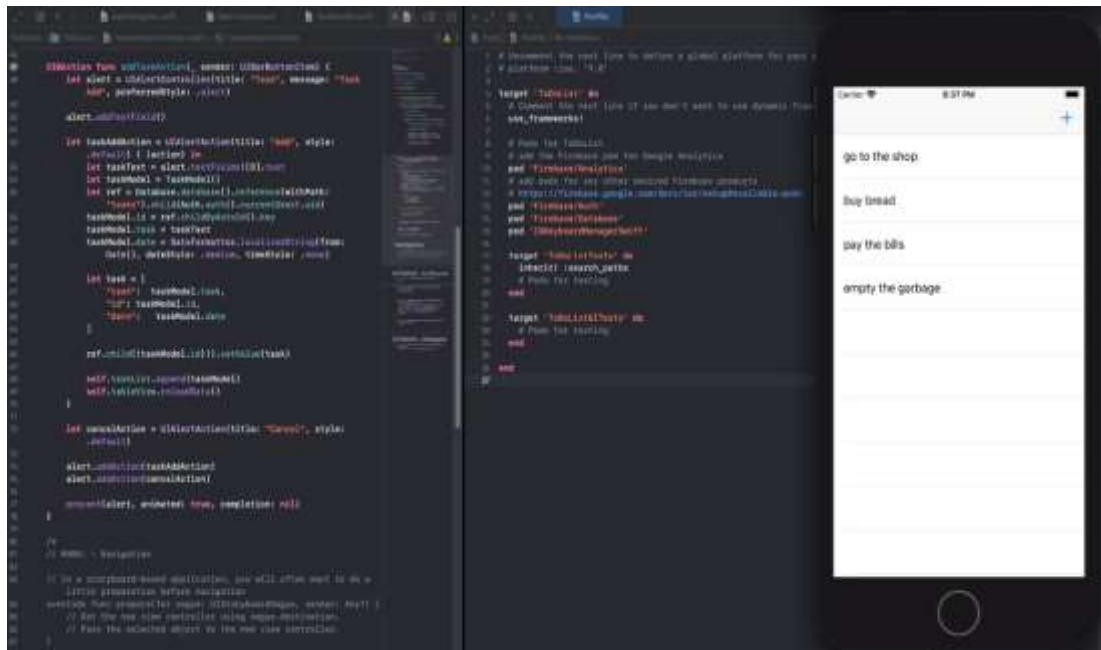

Fig. 4 Screenshot of XcodeTo Do List App

3.3 Native Platform Mobile Application Development Method with Xamarin

Figure 5 shows the design codes and code side of the mobile application on Xamarin. Designs are kept in files with .xaml extension. Codes are kept in files with extension .xaml.cs. It is not necessary to name the components in the designs as firstNumberEditor, secondNumberEditor and to match the code side with these names and can be used directly. Application permissions are written in the AndroidManifest.xml file in Xamarin.Android, in the Info.plist file in Xamarin.iOS. Figure 6 shows the code side of the todo list mobile application on Xamarin, the libraries used and the screenshot of the application. Unlike adding two numbers, it seems that more lines of code and more libraries need to be added to the project. Firebase does not have a direct library for Xamarin and the FirebaseDatabase.net library is used and is shown in Figure 6.
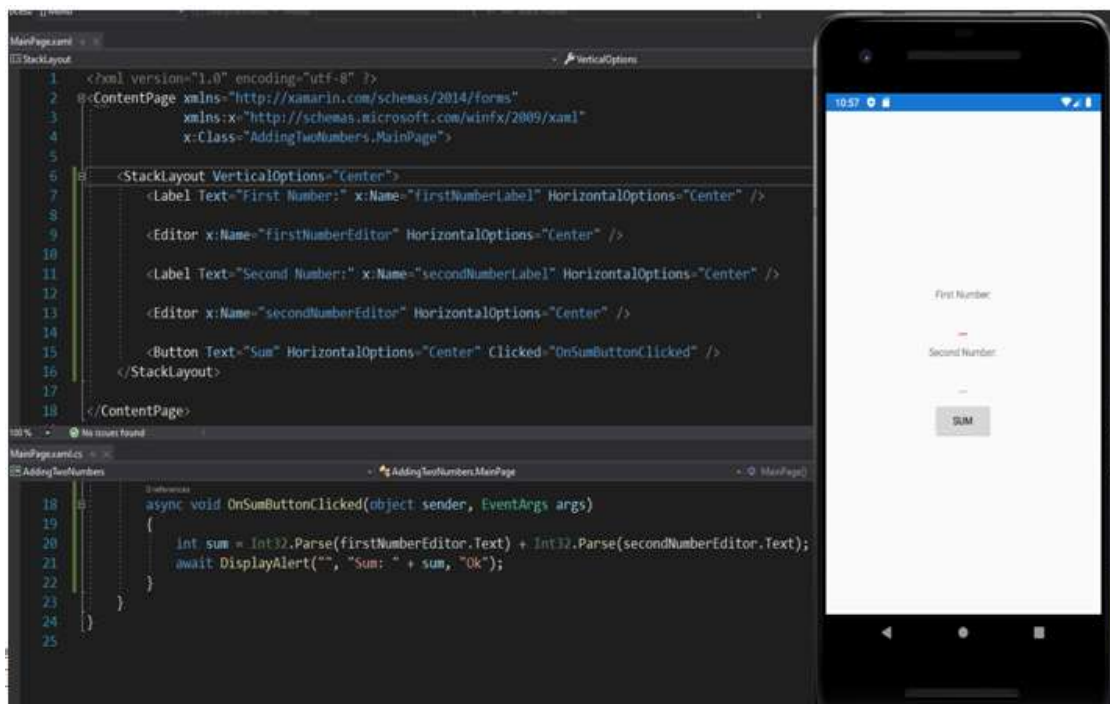

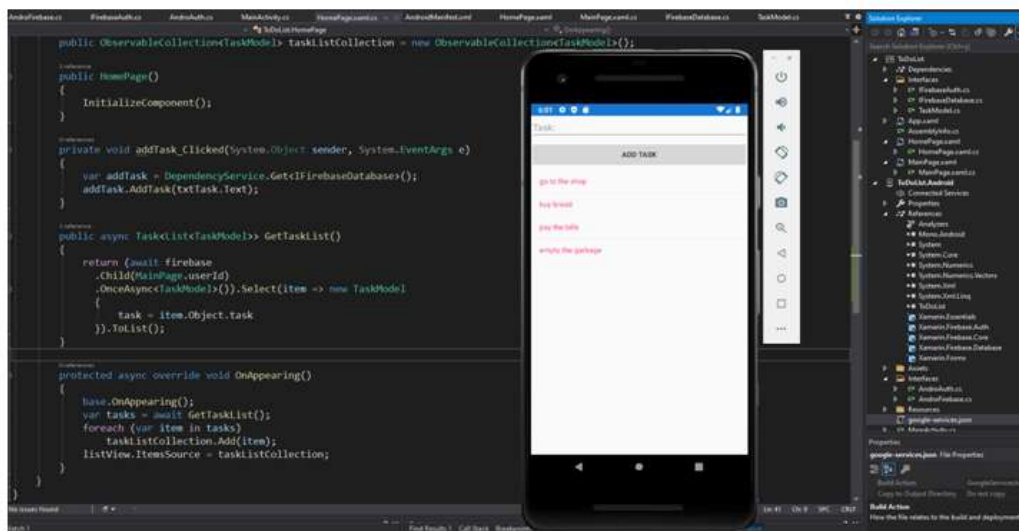Fig. 5 Screenshot of Xamarin Two Number Addition Application

Fig. 6 Screenshot of Xamarin To Do List Application

3.4 Native Platform Mobile Application Development Method with React Native

Figure 7 shows the code side of the mobile application's design codes on React Native. Designs are made with CSS programming language. The codes are written in the JavaScript language. CSS codes and JavaScript codes are kept in the same file and must be associated with each other. The transfer of the codes written to Android Emulator or phone is done with Node.js. Application permissions are written in the AndroidManifest.xml file for Android and in the Info.plist file for iOS. Figure 8 shows the code side of the todo list mobile application on React Native, the libraries used and the screenshot of the application. Unlike the two number addition applications, it seems that more lines of code and more libraries need to be added to the project. Firebase does not have a direct library for React Native and the React Native Firebase library is used.
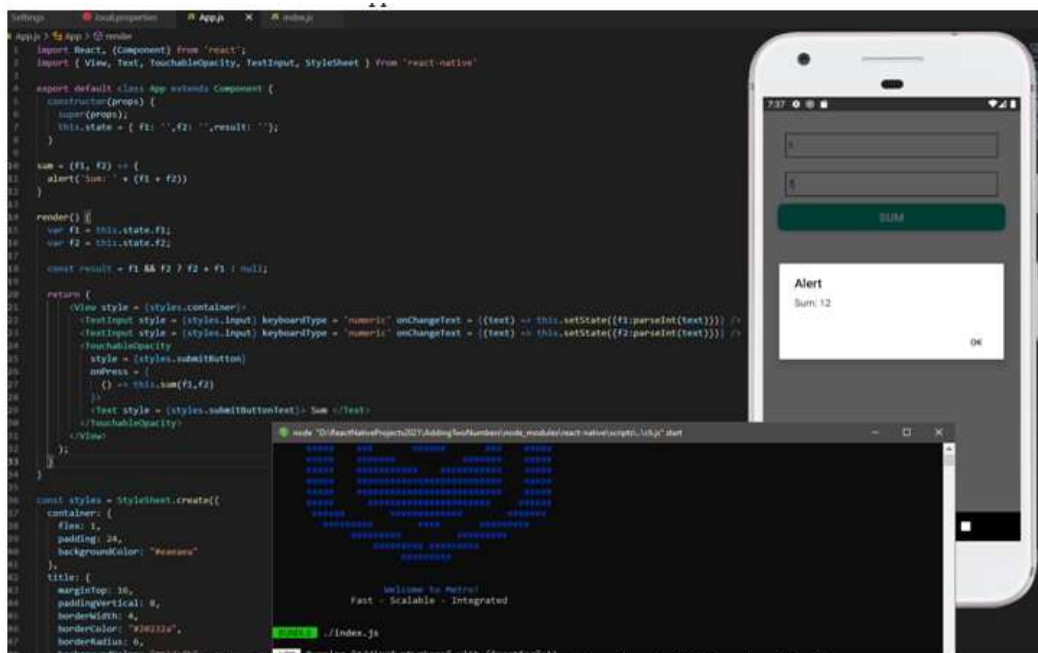

Fig. 7 Screenshot of React Native Two Number Addition Application

3.5 Native Platform Mobile Application Development Method with Angular

Figure 9 shows the design codes and code side of the mobile application on Angular. Designs are made with HTML programming language. The codes are written in TypeScript language and are kept in files with .ts extension and must be associated with each other. In Figure 9, the image of the mobile application is shown on Chrome and it is necessary to export it as an Android Studio or Xcode project in order to be shown on the phone. Apk file can be created directly for Android and .ipa file cannot be created directly for iOS. The filled .apk file for Android can be uploaded to be published in the Google Play Store. It can be sent to publish in the App Store via the project file created for iOS. Application permissions are written into the AndroidManifest.xml file for Android, into the Info.plist file for iOS. Figure 10 shows the code side of the todo list mobile application on Angular, the libraries used and the screenshot of the application. Unlike adding two numbers, it seems that more lines of code and more libraries need to be added to the project. Firebase does not have a direct library for Angular and the AngularFire library was used.
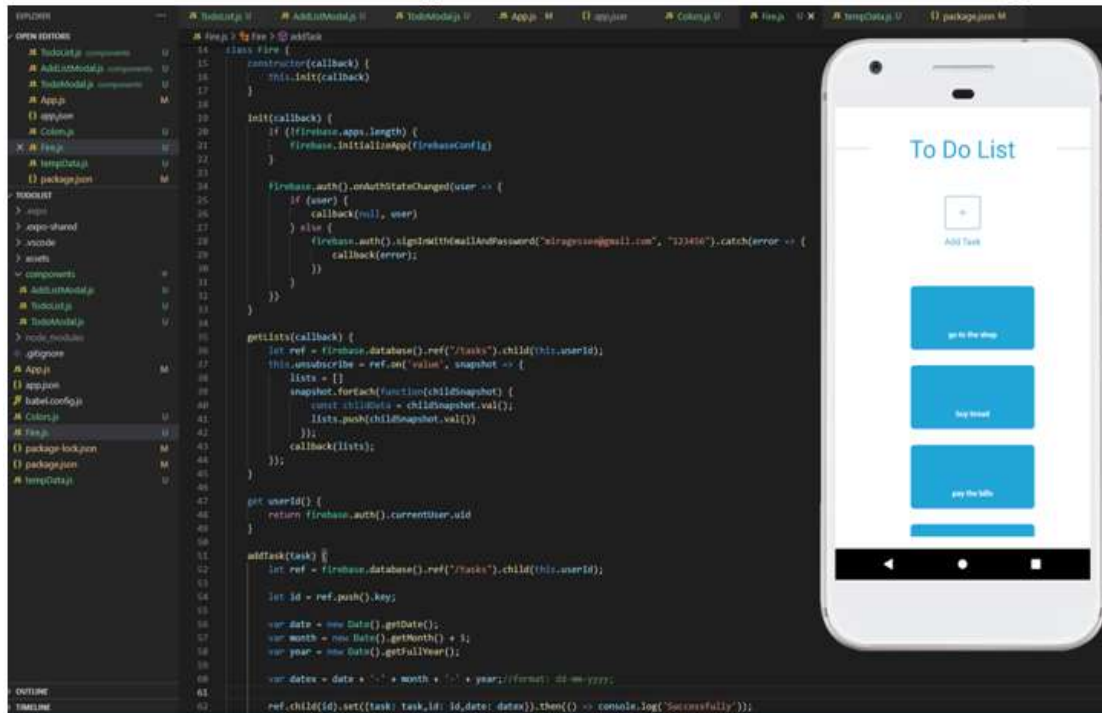
Fig. 8 Screenshot of React Native To Do List Application

3.6 Native Platform Mobile Application Development Method with Ionic

Figure 11 shows the design codes and code side of the mobile application on Ionic. Ionic can be made with Angular, React, and Vue, in this example it is made with Angular. Designs are made with HTML programming language. The codes are written in TypeScript language and are kept in files with .ts extension and must be associated with each other. In Figure 11, the image of the mobile application is shown on Chrome and it is necessary to export it as an Android Studio or Xcode project in order to be shown on the phone.Apk file can be created directly for Android and .ipa file cannot be created directly for iOS. The .apk file created for Android can be uploaded to be published in the Google Play Store. It can be sent to publish in the App Store via the project file created for iOS. Application permissions are written into the AndroidManifest.xml file for Android, into the Info.plist file for iOS. Figure 12 shows the code side of the todo list mobile application on Ionic, the libraries used and the screenshot of the application. Unlike adding two numbers, it seems that more lines of code and more libraries need to be added to the project. Firebase does not have a direct library for Ionic and the AngularFire library was used.
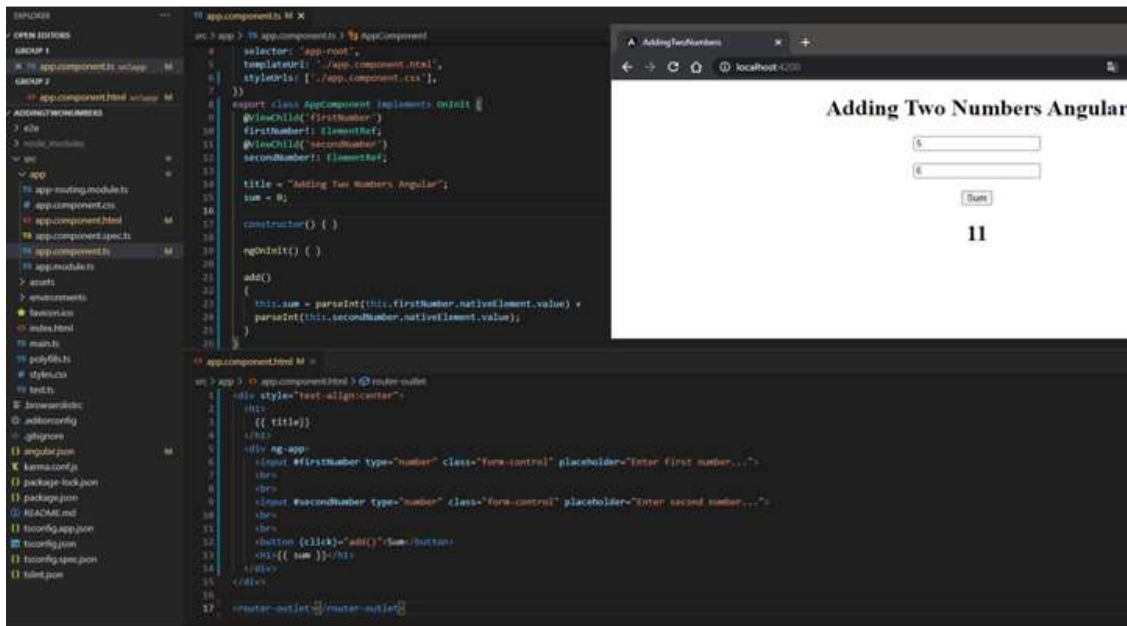
Fig. 9 Screenshot of Angular Two Number Addition Application

3.7 Native Platform Mobile Application Development Method with Flutter

Figure 13 shows the design codes and code side of the mobile application on Flutter. Designs and codes are made in Dart Programming language and must be associated with control codes such as TextEditingController to control components. Android Emulators can be used to test the mobile application. Apk file can be created directly for Android and .ipa file cannot be created directly for iOS. Application permissions are written into the AndroidManifest.xml file for Android into the Info.plist file for iOS. Figure 14 shows the code side of the todo list mobile application on Flutter, the libraries used and the screenshot of the application. Unlike adding two numbers, it seems that more lines of code and more libraries need to be added to the project. Firebase does not have a direct library for Android Studio and the FlutterFire library has been used.
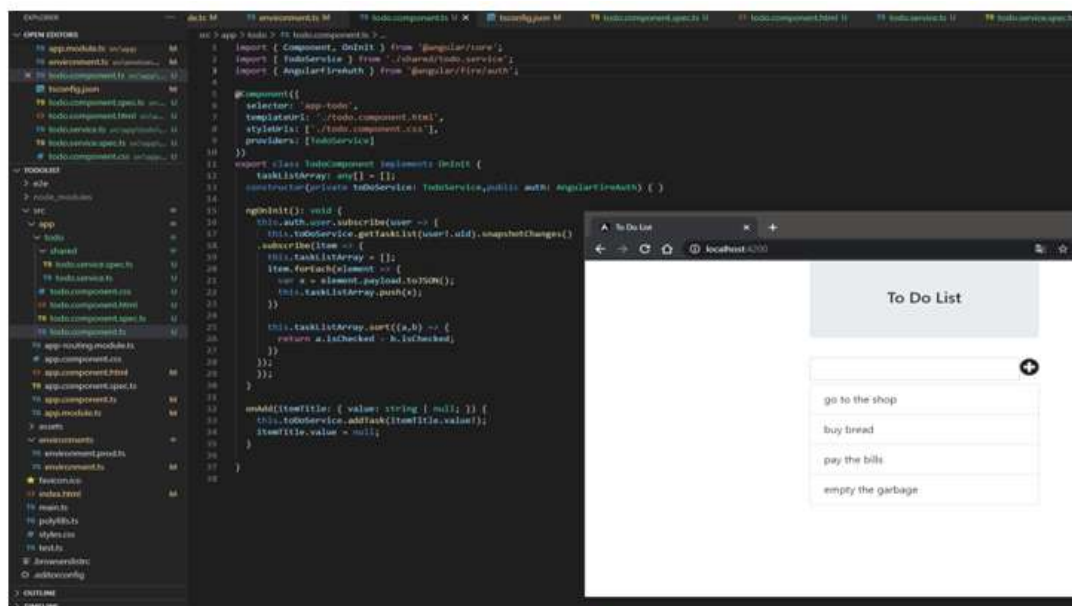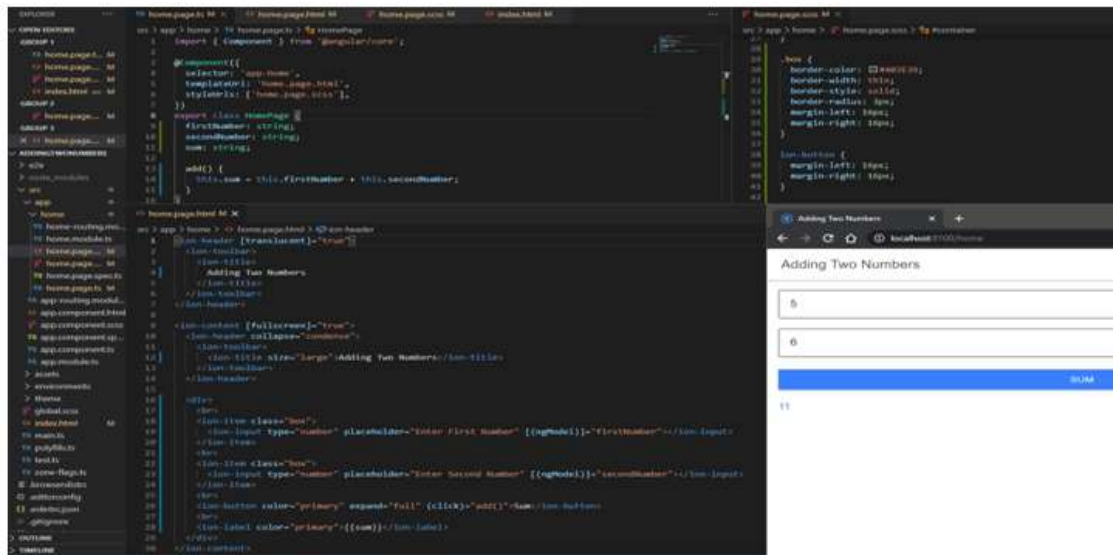

Fig. 10 Screenshot of Angular To Do List Application

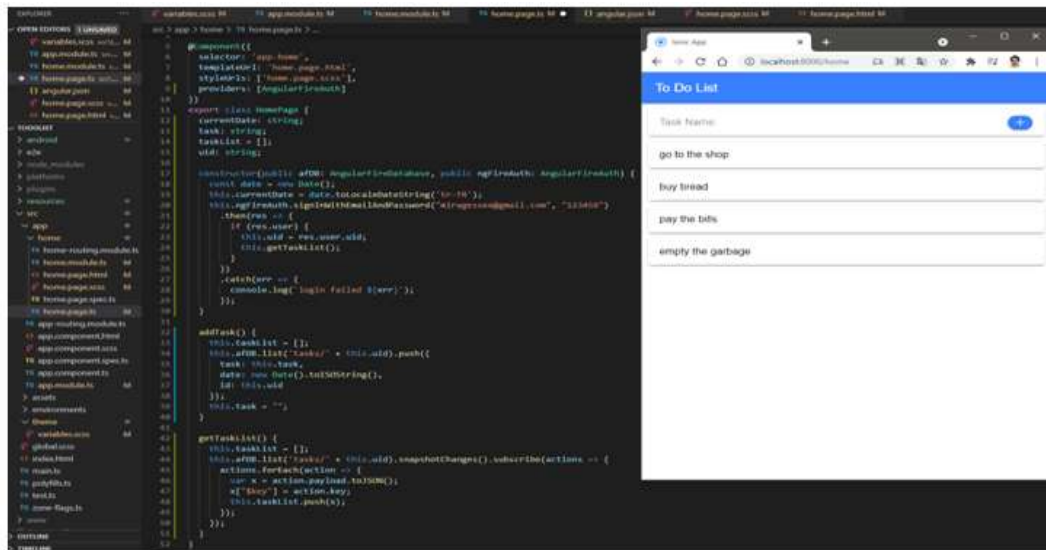**Fig. 11** Screenshot of Ionic Two Number Addition Application

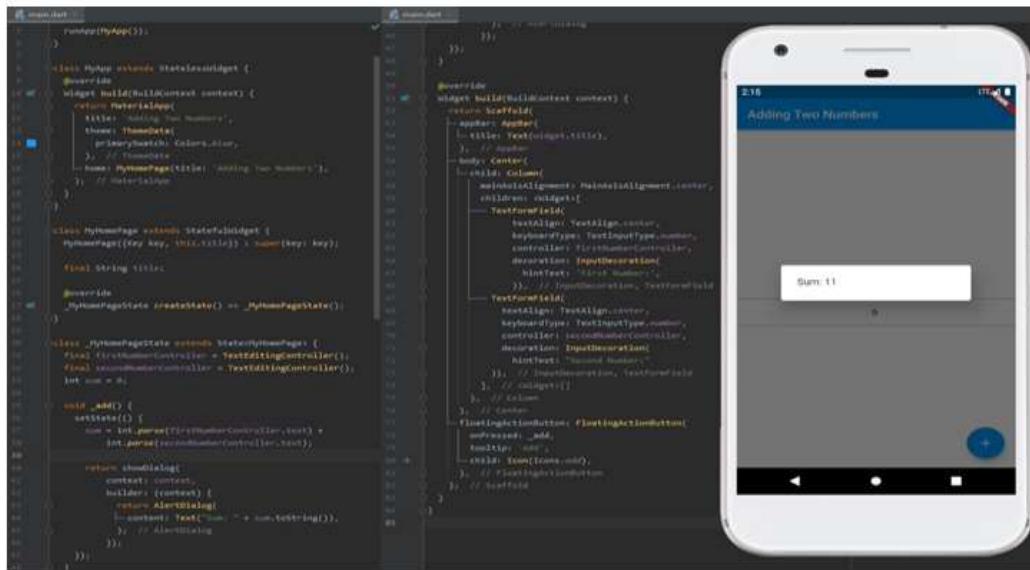**Fig. 12** Screenshot of Ionic To Do List Application

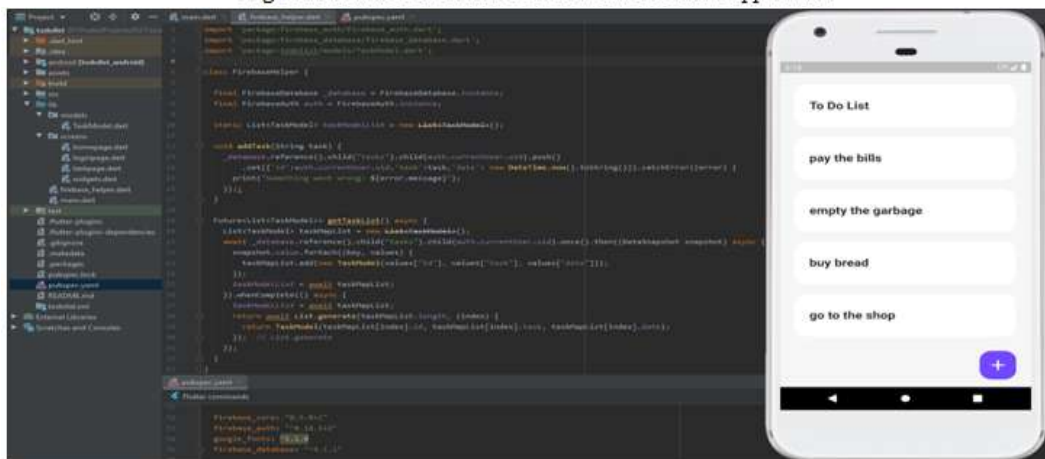**Fig. 13** Screenshot of Flutter Two Number Addition Application

Fig. 14 Screenshot of Flutter To Do List Application

IV. COMPARISON

Native Platform Application Development Methods and Cross Platform Application Development methods are compared in Table 2. As seen in Table 2, Native Platform Application development methods provide a better usage than Cross Platform Application development methods.

Table 2: Comparison of Native Platform Application Development Methods and Cross Platform Application Development Methods

Native Application Development Methods	Cross Application Development Methods
There is no compatibility problem with new and old versions.	There is a compatibility problem with the new and old versions.
The development process is slow. Because it is necessary to develop separate mobile applications for	The development process is faster. Because it is sufficient to develop a single mobile application for

both Android and iOS platforms.	both Android and iOS platforms.
Its performance is faster. Because it uses pure UI designs for Android and iOS.	Its performance is slow. Because it cannot use pure UI designs for Android and iOS.

4.1 Comparison of Xamarin and Native Platform Mobile Application Development Methods

Advantages of mobile application development with Xamarin:
- Applications can be developed with C# and Visual Studio [12].
- Applications developed with Xamarin can perform at the same level with native applications [13].
- Allows full hardware compatibility using plugins and specific APIs for a seamless user experience [13].
- Xamarin comes in a single package complete with all development toolkits, including its own IDE [13].
- Supports TVs, wearable devices and IoT [13].

Disadvantages of mobile application development with Xamarin:
- The application developed with Xamarin takes up more space than the native application [12-13].
- When a new version of the mobile operating system is released, Xamarin may need to be updated according to the new version [12].
- Xamarin is a free open source platform for individual developers but can be expensive for enterprise use [13].
- Xamarin can still be susceptible to stability, bugs and crashes [13].

4.2 Comparison of React Native and Native Platform Mobile Application Development Methods

Advantages of mobile application development with React Native:
- The application looks and behaves the same on Android and iOS platforms [14-15].
- It saves time since it can be developed for both Android and iOS [14-15].
- It can develop an application in a very short time and with minimum cost [14].

Disadvantages of mobile application development with React Native:
- If it is necessary to create an application for only one platform (Android or iOS), it may be better to use native mobile application development methods [14].
- If it is necessary to createa special application that requires the use of code specific to the platform (Android or iOS), native mobile application development methods should be used [14-15].
- If the application needs to support all operating system updates as soon as they are released, native mobile application development methods must be used [14-15].

4.3 Comparison of Angular and Native Platform Mobile Application Development Methods

Advantages of mobile application development with Angular:
- It enables the developed application designs to appear similarly on Android and iOS platforms [8].
- It reduces the cost of the process as it can be printed on both Android and iOS platforms at once [8].
- Since the application can be developed by learning a single programming language instead of learning more than one programming language, the learning process is shorter [8].

Disadvantages of mobile application development with Angular:
- While applications are translated into the language of the machine on which they are running in native platform application development methods, Angular cross-platform mobile application development method works by interpreting the application on the browser so the performance of native platform applications is higher [8].

4.4 Comparison of Ionic and Native Platform Mobile Application Development Methods

Advantages of mobile application development with Ionic:
- It is open source [16].
- It has ready-made add-ons that can be used to develop applications [16].
- The same code can be run on both Android

and iOS platforms [16-17].

Disadvantages of mobile application development with Ionic:
- The performance of applications developed with Ionic is less than the performance of applications developed natively [16-17].
- It takes more time to develop new plugins separately, as it does not include plugins for every requirement [16].

4.5 Comparison of Flutter and Native Platform Mobile Application Development Methods

Advantages of mobile application development with Flutter:
- It has a very rich documentation [18].
- MaterialApp library for Android designs there is a Cupertino library for iOS designs [18-19].
- There is a central website (pub.dev) with many libraries that can be used [20].

Disadvantages of mobile application development with Flutter:
- Applications for tvOS, watchOS, CarPlay or Android Auto cannot be made with Flutter [21].
- The size of applications developed with Flutter is larger and the size of a Hello World application is 4.7 MB in Flutter and 550 KB in native applications [21].

V. CONCLUSION

It is necessary to choose the appropriate method according to the advantages and disadvantages of mobile application development methods, programming languages, development environments and Xamarin should be selected if C# programming language is desired. While making this selection, it is necessary to make a selection by evaluating the size of the team that will develop the mobile application, the time of development of the mobile application, the operating systems of the computers to be used in mobile application development and the programming language to be used. It is necessary to learn the programming language and compiler of the platform in native platform application development methods. However, since the applications run on their own platform, there is no compatibility problem. In Cross Platform application development methods, both Android and iOS applications can be made on a single platform. However, in some cross-platform application development methods since applications are written on another platform, this platform needs to be updated according to the newly released Android and iOS versions and because of this, there may be compatibility problems in applications made with the cross-platform application development method. On the other hand, when a change is requested in the mobile application, changes are made for both Android and iOS at once. In addition, the mobile application made in some cross-platform application development methods cannot be run directly on the Android Emulator. However, Google solved these problems with Flutter and created a new framework.

REFERENCES
[1]. Sıcak, M. A., 2016, Android Programming and Application Guide, LEVEL, Turkey/Kocaeli
[2]. Sıcak, M. A., 2017, Kotlin Turkey's First and Only Kotlin Book, LEVEL, Turkey/Kocaeli
[3]. Kızılören, T., 2015, iOS SDK with Swift and Objective-C, Kodlab, Turkey/Istanbul
[4]. https://developer.android.com/studio/releases, Date of Access: 11 May 2021, Subject: Android Studio release notes
[5]. https://helpx.adobe.com/experience-manager/kb/adobe-phonegap-end-of-service.html, Date of Access: 11 May 2021, Subject: End of service for Adobe PhoneGap
[6]. Petzold C., 2016, Creating Mobile Apps with Xamarin Forms, Microsoft Press A Division of Microsoft Corporation One Microsoft Way Redmond, Washington 98052-6399, ISBN: 9781-5093-0297-0
[7]. Canker, M., 2015, Web and Mobile Application Development with React and React Native, Kodlab, Turkey/Istanbul
[8]. Canker, M., 2015, Web and Mobile Application Development with Angular, Kodlab, Turkey/Istanbul
[9]. Mainkar P., Giordano S., 2019, Google Flutter Mobile Development Quick Start Guide, Packt Publishing Ltd., UK, ISBN: 978-1-78934-496-7
[10]. Charland, A., Leroux, B., 2011, Mobile Application Development: Web vs. Native, Communications of the ACM, 54, 5, 49-53. DOI:10.1145/1941487.1941504
[11]. Yahiaoui H., 2017, Firebase Cookbook, Packt Publishing, Birmingham
[12]. Jepbarov D., 2017, Subject: Xamarin React Native Ionic,Access Address: https://medium.com/ios-development-turkey/xamarin-react-native-ionic-dbdf18eb9487, Date of Access: 11 May 2021

[13]. Surbhi., 2020, Subject: Frameworks for Cross-Platform App Development: Flutter vs Xamarin vs React Native, Access Address: https://surbhi-technical-blogger.medium.com/frameworks-for-cross-platform-app-development-flutter-vs-xamarin-vs-react-native-systango-d82a427c16b9, Date of Access: 11 May 2021

[14]. Smartym P., 2019, Subject: React Native vs iOS/Android Native: what is the best choice for mobile development today?, Access Address:https://medium.com/@smartym.pro/react-native-vs-ios-android-native-what-is-the-best-choice-for-mobile-development-today-67cde42fc8bc, Date of Access: 11 May 2021

[15]. Trnka D., 2018, Subject: Mobile App Development: React Native vs Native (iOS, Android), Access Address: https://medium.com/mop-developers/mobile-app-development-react-native-vs-native-ios-android-49c5c168045b, Date of Access: 11 May 2021

[16]. Panchal M., 2016, Subject: Ionic Vs Native Apps: Revealing An Outstanding Platform For Your Mobile App, Access Address: https://www.excellentwebworld.com/ionic-vs-native-app-development/, Date of Access: 11 May 2021

[17]. Delaney J., 2018, Subject: Ionic 4… Should you Build a Hybrid App?, Access Address: https://itnext.io/should-you-use-ionic-4-fa04daebaffd?gi=c32d8f478ed9, Date of Access: 11 May 2021

[18]. Çolak E., 2020, Subject: React Native vs Flutter!Which?, Access Address:https://medium.com/defactoteknoloji/react-native-vs-flutter-hangisi-d11ad3582050, Date of Access: 11 May 2021

[19]. Bellinaso M., 2018, Subject: Flutter: the good, the bad and the ugly, Access Address: https://medium.com/asos-techblog/flutter-vs-react-native-for-ios-android-app-development-c41b4e038db9, Date of Access: 11 May 2021

[20]. Decanini E., 2020, Subject: Flutter vs Android/iOS – Hybrid vs Native App Development in 2020, Access Address: https://www.ericdecanini.com/2020/05/25/flutter-vs-android-ios-hybrid-vs-native-app-development-in-2020/, Date of Access: 11 May 2021

[21]. Chang W. S., 2020, Subject: Flutter vs Native for Mobile Development, Access Address: https://medium.com/swlh/flutter-vs-native-for-mobile-development-4f92e4d2203b, Date of Access: 11 May 2021