

# Intelligent Analysis on Frameworks for Mobile App Development

Gowri S  
*School of Computing, Sathyabama  
Institute of Science and  
Technology,  
Chennai, India*  
gowri.it@sathyabama.ac.in

Kanmani Pappa C  
*Department of Electronics and  
Communication Engineering  
Veltech Rangarajan Dr  
Sagunthala R&D Institute of  
Science and Technology  
Chennai, India*  
kanmanipappa.phd@gmail.com

Tamilvizhi T  
*Department of Computer Science  
and Engineering  
Panimalar Engineering College  
Chennai, India*  
tamilvizhi.phd.it@gmail.com

Leema Nelson  
*Department of Computer Science  
and Engineering  
Chitkara University Institute of  
Engineering & technology,  
Chitkara University,  
Punjab, India*  
leema.nelson@chitkara.edu.in

Surendran R  
*Department of Computer Science  
and Engineering  
Saveetha School of Engineering  
Saveetha Institute of Medical and  
Technical Sciences,  
Chennai, India*  
surendran.phd.it@gmail.com

**Abstract-** Establishing new mobile applications that are adaptable across multiple cross platforms is a strenuous task. There are several frameworks for mobile application development existing in the market which has got its own tremendous features and benefits, but choosing the right framework for app development is very crucial for any developer. This study compares the benefits and efficiency of the three most popular and preferred frameworks by the end-users; React-Native, Flutter, and Ionic Frameworks and choose the most optimized and suitable framework for mobile app development. These designed frameworks are platform independent, and the yield was evaluated and appeared to be satisfactory by the users in the respective platform. These methodologies have been implemented using JavaScript, Dart, HTML, and CSS.

**Key Words:** Mobile Application Development, designed frameworks, React-Native, Flutter, Ionic

## I. INTRODUCTION

Developing a mobile app is a quite a challenging task as it has a lot of challenges associated with it and it is not an easy job for an app developer. The multiple stages involved in developing a mobile application include the following : Analysis of the app design, user interface design, usage of the right programming language for the targeted platform, testing the developed application, launching and maintaining the application with patches and version updates [1]. Thus, the developer needs to consider a few important aspects while choosing a framework for building an app. Comprehending a solution to this, more discussion has been initiated on the famous frameworks and perform a comparative study and eventually determine the most suitable option.

Ionic framework is specially designed for building hybrid applications using the knowledge of HTML, CSS, and JavaScript. Ionic is very famous due to its new user interface components, gestures, and animations [2].

Furthermore, this report illustrates the distinct features of these three mobile frameworks, its application design, performance, user interface design, development cost, and most importantly security factor [3]. The main objective of this paper is to determine the best efficient framework to design a mobile application. React-Native is one of the most popular framework for mobile app development invented by Facebook developer's in 2015 utilized for building mobile applications for cross platforms. React-Native framework is designed using JavaScript as the programming language and designing application for both Android and iOS with the same code utilized on both platforms [4]. "Flutter for mobile application development aims at distributing tools to software developers that can have a look and feel of native applications". This also provides the same code base for multiple platforms.

## II. EXISTING SYSTEM

In the React-Native framework, the application concept is written and run in JavaScript. However, the application UI is fully native [5]. React-Native is designed using JavaScript. But rendered with native code. Here is a sample program designed using this framework. It just renders the native platform user interface (using the same native platform application interfaces which other applications do). Using one react, many platforms which means single code shared across various platforms is one of the benefits of using this framework for mobile application development.

The React-Native framework design structure using JavaScript is shown in Fig. 1.

```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () => {
  <View>
    <Header title="Welcome to React Native"/>
    <Text style={heading}/>Step One</Text>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text style={heading}/>See Your Changes</Text>
    <Text>
      Press Cmd + R inside the simulator to reload
      your app's code.
    </Text>
    <Text style={heading}/>Debug</Text>
  </View>
}

<Text>
  Press Cmd + R inside the simulator to reload
  your app's code.
</Text>
<Text style={heading}/>Debug</Text>
</Text>
<Text>
  Press Cmd + R or Shake your device to open the
  React Native Debug Menu.
</Text>
<Text style={heading}/>Learn</Text>
</Text>
<Text>
  Read the docs to discover what to do next:
</Text>
</View>
```

Fig. 1. Design Structure using JavaScript

Flutter framework is designed using DART language and can be compiled to JavaScript. It's SDK offered by Google that contributes to a wonderful experience with prebuilt widgets. Same code base utilized for both Android as well as iOS which is reusable. Any application designed using Flutter is customizable according to the user's needs. Beautiful designs can be constructed using widgets in flutter which is discussed later in this paper. Below is a simple example code that displays "Hello World" on the screen. The Sample Code for Flutter is shown in Fig. 2.

```
import 'package:flutter/material.dart';

void main() {
  runApp(
    Center(
      child: Text(
        'Hello, world!',
        textDirection: TextDirection.ltr,
      ),
    ),
  );
}
```

Fig. 2. Sample Code for Flutter

The runApp () method accepts the given widget and makes it the root of the widget tree. Here in the above sample, there are 2 widgets: Center widget and Text widget. Text widget is the child Center widget. The framework forcibly makes the root widget to cover the entire screen, i.e., "Hello World" appears on the screen centre. The direction of the text, its alignment should be mentioned in this materialapp which is also another widget.

There are two sub classes of widgets – State, full widgets and stateless widgets. The basic task of a widget is to implement a build(). This method build() defines the widgets. Flutter has a few good packages of basic widgets. Most of them that are used frequently are: Text, Row, column, stack, containers.

Ionic is an open-source HTML mobile application development framework for constructing hybrid mobile apps. Hybrid apps have various benefits like speeding up the application and supporting the platform.

The React-Native framework is completely loaded with numerous components. Amongst which, Flex box is very helpful in building better layouts in React-Native. This is a layout module based on cascading style sheet which basically comprises of two components called containers and items. They are used to align items and justify content and set layout rules for the items inside the container. React-Native has about 3000 icons which can be accessed from "react-native-vector-icons" package. Custom shapes can be designed using "react-native-svg" package in React-Native.

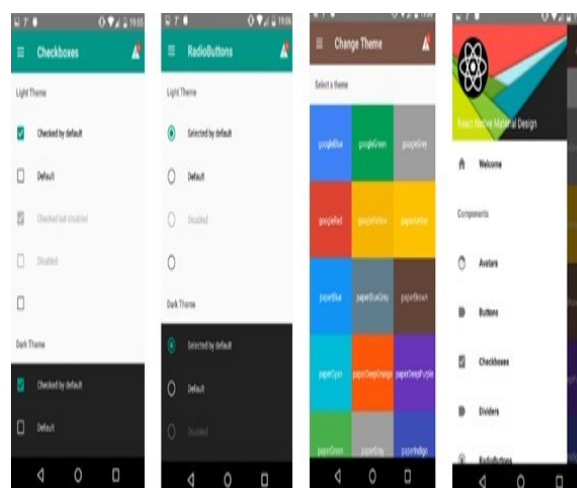


Fig. 3. Implementing the Components in React-Native Framework

Implementing the Components in React-Native Framework is displayed in Fig. 3. The Sample Code for generating an icon is shown in Fig. 4.

```
import Icon from "react-native-vector-icons/Ionicons";

function ExampleView(props) {
  return <Icon name="ios-person" />;
}
```

Fig. 4. Sample Code for generating an icon

The main primitive building blocks in Flutter are the widgets. The basic function of the widget is to define the elements in the user interface when its state and its configuration are known. They provide an API to developers to configure widgets visually. It is stated that Flutter developers are

inspired by the React-Native component technology while constructing Flutter. Since they share a single code base, mobile applications behave and appear similarly in both Android and iOS.

Ionic is an open-source HTML mobile application development framework for constructing hybrid mobile apps. The Widgets in Flutter designed Mobile App is displayed in Fig. 5. Hybrid apps have various benefits like speeding up the application and supporting the platform. The Components usage in Ionic Framework is shown in Fig. 6.



Fig. 5. Widgets in Flutter designed Mobile App



Fig. 6. Components usage in Ionic Framework

### III. PERFORMANCE AND EFFICIENCY

#### A. React-Native

One of the most fascinating reasons for opting React-Native instead of making use of several web-based tools available is to get a native feel and look of the applications designed. Ever since

2018, this framework has the largest number of contributors in GitHub. Today, React-Native framework is supported by several multinational companies all around the world including Microsoft. Evaluating the performance of the applications designed using React-Native; it is observed that it has one of the most common issues with memory. This is due to the native applications running continuously with numerous processes running in the background. Eskola and Rasmus has executed a scientific study on the development process of React-Native and also evaluated the user's experience of using an app designed using this framework and other native applications [6]. Users could determine very little dissimilarities when comparing with these applications side-to-side but not when these applications were tested in isolation. According to another research, the author concluded that React-Native Framework requires more resources in comparison with other native Android applications. Yet another examination indicates that the development process of a React-Native application was done by Nilsson and Anders [7]. In this experiment, the development processes of React-Native apps and a few hybrid frameworks were considered into account. It was reported that React-Native is well packed with features as its competitors and this has got the largest community when compared with the other evaluated frameworks.

#### B. Flutter

Performance in flutter is characterized by Google. There was an experiment conducted by constructing the same mobile application in five various platforms (native iOS, Android, React-Native). This study keenly focused on the usage of the CPU of the devices [8]. Here, it is observed that flutter behaves natively and economically when compared to React-Native framework. Second experiment focused on the usage of resources in both frameworks- React-Native and Flutter. This study constructed a timer application for Flutter, native and React-Native and was examined on different devices. Flutter excels with its CPU usage while the React-Native struggled [9]. Another important aspect – Memory was considered and again flutter exceeded the expectations with a positive response.

#### C. Ionic

Ionic, being a mobile framework for building hybrid applications with ease is used with the native applications of the mobile in Apache Cordova (PhoneGap) for a massive performance

in comparison with other existing hybrid applications. Ionic makes use of CSS to expand the tome of the processor [10].

#### IV. DEVELOPMENT DURATION

##### A. *React-Native*

One of the best features of React-Native is “Flexibility”. The application developer can choose any editor according to the requirement. Using an editor would rise up the productivity aswell. There exist a number of editors for text and IDEs which would make our work a loteasier. Few of them are Atom, Visual Studio, etc. Such applications contain new tools and features that help you write the code quicker and are not more efficient for the current day scenario. The author have stated post so much analysis that the time taken to createan app can beat that of one native framework’s associated application [11].

##### B. *Flutter*

Using Flutter framework not just enhances the speed of the application development process but also exceed in productivity performance. The SDK contributes prebuilt packages, tools and there is no requirement of too much configuration [12]. But on the other side, itlets the developers to construct the libraries on their own, which is quite time consuming and challenging.

##### C. *Ionic*

The time taken to build an application using Ionic Framework took almost a month [13]. While, applications designed using React-Native literally took a month and a half duration to complete the build, which is quite long.

#### V. BENEFITS OF FRAMEWORKS

##### A. *React-Native*

React-Native is well known for it is an attractive user interface and various components that are utilized to build them up. Moreover, with the boost of JavaScript, iterations could be performed extremely fast and there is absolutely no need to wait for any modifications . Yet another feature called “Refreshing” allows us to view the changes immediately as save. Single code base can be utilized on several platforms like Android, iOS, Windows, etc [14,15]. Performance is satisfactory according to multiple research conducted and demonstrated.

Power consumption is another important aspect that is evaluated. The usage of the battery

by applications designed using React-Native framework is less when compared to another framework here (Ionic framework used to design hybrid applications). The below graph shows the difference in battery usage by these frameworks [16,17]. The Evaluation of the Battery Consumption is displayed in Fig. 7.

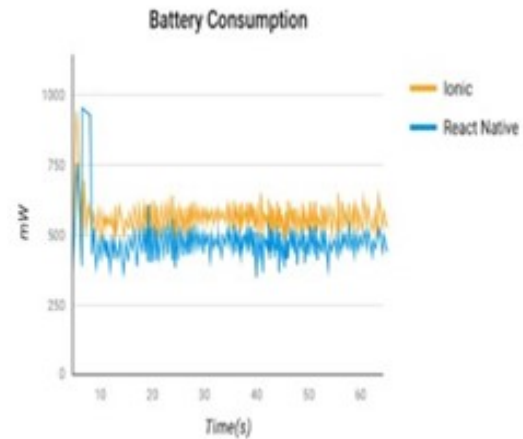


Fig. 7. Evaluation of the Battery Consumption

##### B. *Flutter*

Flutter framework comprises of several advantages as it attracts several mobile application developers. The most important feature that Flutter offers is “Hot Reloading”, which enables the application developers to view the modifications made in the application at the very next moment, which reduces the complexity of debugging. It also enables better association between the developers and application designers when they need to enhance the look and effects quickly. In contrast with the native applications, it takes time to view the modifications made.

Flutter uses DART programming language which is object oriented. It’s quite similar to React-Native as the style of programming is declarative as well as reactive. The performance of Flutter is great as it does not use a JavaScript bridge that enhances the start-up time of the application. Hot Reload also refreshes the user interface during the mobile application development without completely building a new one. The most Interesting feature of Flutter is the usage of widgets which enhances an application’s view. There are fast and customizable according to the user’s requirement. It has a large set of widgets for creating custom widgets. Here widgets as an element (Menu/ buttons, etc) or a layout similar to padding or style elements like fonts, color schemes, etc. They are very easy to set up.



### C. Ionic

Applications can be easily built using Ionic with the knowledge of HTML, CSS, JavaScript. It is a more suitable framework when you want to construct hybrid applications or to be run on different operating systems and built apps with native functionality. Ionic framework uses AngularJS. This framework is said to be a cost cutting framework due to its efficiency, its ease of use, and lesser maintenance requirements. The user interface layout is so eye-catching that it provides cool themes, interactive paradigms. The applications designed using this framework are built using the CLI and require Cordova to run as a native application.

## VI. DRAWBACKS OF FRAMEWORKS

### A. React-Native

Navigation is not so smooth. This is evaluated during its testing in the production environment by various giant companies like Facebook, Skype, and Bloomberg. There is always a change in the tools and their dependencies between versions. Developers often face difficulties with the compatibilities of packages, tools used for debugging, and failures in hot reloading and causes very slow development of applications. Although it has many components, it still lacks a few of them. This is not an issue when there are sufficiently available modules required for the developer's requirement and working properly. However, the problem arises when the developer has to build the application from scratch. In performance, React-Native stays a step behind optimized native applications.

In fact, there has been an argument that React-Native hinders the performance of an application and JavaScript will not be faster when compared to the native codes, but we do not see any dissimilarity in most of the evaluated cases. Evaluation was performed on the versions of two sample applications, but both of them achieved similar results in their performance. Dissimilarity in performance are likely negligible and not an issue anymore.

### B. Flutter

Flutter Framework is fresh and completely new in the market. It has very limited libraries, so that the developers would have to construct the libraries on their own, which is quite complex and time consuming. Earlier, separate codes have to be written for Android and iOS. But now, with the invention of Flutter framework, it introduces a bridge that would enable the developer to construct

quality applications for both platforms at the same time without any anomalies.

It is observed that there was a bit struggle in product view where the working was denied. The development team tried to figure out the reason for this and eventually determined it's due to the lack of cluster in the documentation of the Hot Reload feature. The error was found to be in the implementation of view. Therefore, we can say that hot reloading is good under high care and monitoring but not always.

### C. Ionic

Debugging is quite complicated when an application is designed using Ionic as it occupies longer time due to error messages. It is unsure to determine the origin of the errors and even causes the structure to break down. For instance, if any destruction happens in any of the folders, it allows the developer to clone the repository into a new folder. The major disadvantage with Ionic is that the build can collapse at any moment. The Evaluation of the CPU Usage is shown in Fig. 8.

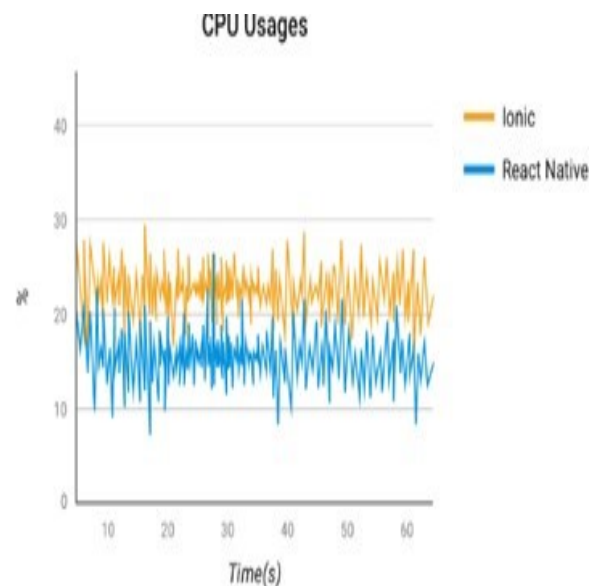


Fig. 8. Evaluation of the CPU Usage

The CPU usage is compared to React-Native and the results are shown below: Ionic occupies 21.3. Another factor is examined, which is the memory allocation of both frameworks. Here, it is noted that Ionic occupies more memory than React-Native as shown in the graph below. A study is performed to measure the framework's performance, where the developer was asked to build a 3D video game (Subway Surfers/ Pokémon Go/Temple Run/etc) ionic is highly not recommended. Moreover, another observation was found in terms of security where there were few gaps during the analysis when worked with Mc Donald's and kindle communication and later it was fixed, as stated by

Bouyer. The evaluation of the Memory Usage is displayed in Fig. 9.

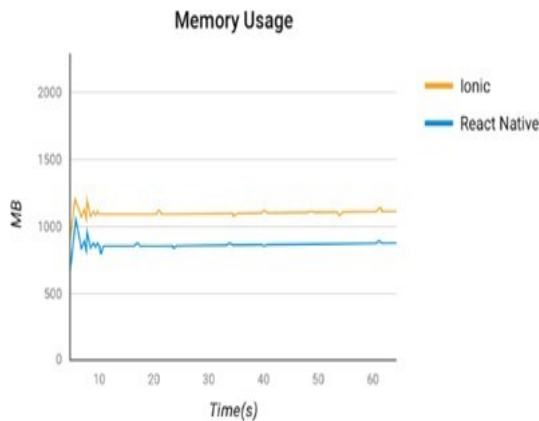


Fig. 9. Evaluation of the Memory Usage

Around 1865, one of the very first walking models appeared. Attempts have been made for the past 150 years to shape walkable technologies based on various kinematic cables that were made up to create an anticipated movement while in action. Several models were proposed during that time, but the execution of many of them was hampered by cyclic rambling and the incapability to adjust a movement form towards the scene. Technologies should just not be predicated on dynamic systems machineries that deliver concurrent actions, it became clear in the early 1950s, and surveillance and organizing processes should be integrated. The main job of the robot's computer has been to decipher the mathematical formulas and control the hybrid engines which managed to drive the left leg so that the device could keep moving forward and remaining balanced. Thanks to the advancements in theoretical and systematic technologies, computers and distinct systems with exceptionalities have indeed been constructed since then. Running, walking placed above a white rough terrain, jumping over barriers, climbing, and other skills are examples of these abilities. Following the industrial revolution, robots have a long history of use. As evidenced by the few records, there has been and keeps going to be advancement in the field. In 1967-68, the first wheeled to walk device with perception and other detectors was discovered. In 1974, NASA launched the first affect approximately microcomputer-controlled robots, which were used to collect data from the surface of the planet. A computer chip mechanically operated pick and place robot was invented by S.R.DEB in the Manufacturing Electrician department at Jadavpur University in 1981. In 1984, the Bhabha Atomic Research Centre developed a 6-axis multifunction robot that weighed around 300 kilograms and could move a 10 kg side

along with the end effector. There may be presently a large number of active programmers in the arena of legged motion.

Ionic is preferred amongst these three frameworks because building applications using this model requires skilled professionals in Angular.Js, which is a very steep learning curve. However, while designing the app for the first time, it is quite challenging. Secondly, AngularJs debugging is very much complicated. Lastly, it does not have a smoother performance like a native application does. Flutter and React-Native has also got its own pros and cons, but React-Native has more popularity among developers as flutter is still new in the productivity environment. Flutter although looks attractive, still many stages have to be refined and it would take some duration to show its potential and settle down in the market. So far, React-Native has paved its way to receive a great audience on stage.

TABLE I. SUMMARY OF FRAMEWORKS

FRAMEWORKS	REACT NATIVE	FLUTIER	IONIC
<b>Programming Language</b>	Java script and React	Dart	Html, Css, Java Script, Angular
<b>Platforms</b>	Android, iOS, Web, Windows	Android, iOS, MacOS, Windows	Android, iOS, WebApp,
<b>Benefits</b>	Live and hot reloading features	Hot reloading features	One codebase, any platform
<b>Code Reuseability</b>	Reusable	Reusable Widgets	Reusable
<b>Open Source</b>	Yes	Yes	Open Source and Paid
<b>Used By</b>	Facebook, Ubereats, Instagram, Skype	Google Ads,	Accenture

## VII. CONCLUSION

All three mobile application development frameworks are great as they give they offer privileges to the application developers to prototype and launch applications faster on various targeted platforms with a single code base. Security is another main essential factor to be evaluated while deciding the framework. Confidential data, biometric information, and end user password details need to be secure. Several encryption and security mechanisms are implemented in each of these frameworks. The React-Native contains a 'Key Chain' library which provides key chain access to the designed application and secures it intensively. Flutter, on the other hand, although new in the market, also possess few security features. "Square has developed a Flutter plugin for In-App Payments SDK that utilizes the native Android and iOS SDKs to make secure payments". Analysing the Ionic model, it provides an encryption mechanism on the device by providing a secure storage with an additional plugin called Cordova.

## References

- [1] Martin Hensher, Paul Cooper, Sithara Wann Arachchige Dona, Mary Rose Angeles, Dieu Nguyen, Natalie Heynsbergh, Mary Lou Chatterton, Anna Peeters, Scoping review: Development and assessment of evaluation frameworks of mobile health apps for recommendations to consumers, *Journal of the American Medical Informatics Association*, vol. 28, no. 6, 2021, pp. 1318–1329, <https://doi.org/10.1093/jamia/ocab041>
- [2] P. Nawrocki, K. Wrona, M. Marczak and B. Sniezynski, "A Comparison of Native and Cross-Platform Frameworks for Mobile Applications," in *Computer*, vol. 54, no. 3, pp. 18-27, March 2021, doi: 10.1109/MC.2020.2983893.
- [3] G. K. J. G. S. S. Rajendran, J. S. Vimali, J. Jabez and S. Srinivasulu, "Identification of Cyber Threats and Parsing of Data," 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 556-564, 2021. doi: 10.1109/ICOEI51242.2021.9452925.
- [4] Lachgar M, Hanine M, Benouda H and Ommane Y. (2022). Decision Framework for Cross-Platform Mobile Development Frameworks Using an Integrated Multi-Criteria Decision-Making Methodology. *International Journal of Mobile Computing and Multimedia Communications*. 10.4018/IJCMC.297928. 13:1. (1-22). Online publication date: 1-Jan-2022.
- [5] Tamilvizhi.T, Surendran.R, Bommi. R. M, "Radio Frequency Identification (RFID) based ubiquitous health care data handling", *IOP Conference Series: Materials Science and Engineering*, vol. 994, no. 1, 012021, 2020.
- [6] Eskola, Rasmus. "React Native Performance Evaluation." (2018).
- [7] Nilsson, Anders. "Performance and feature support of Progressive Web Applications: A performance and available feature comparison between Progressive Web Applications, React Native applications and native iOS applications." (2022).
- [8] R. Surendran and T. Tamilvizhi, "Friendly Online Technology Development Cloud Service for Bahraini Students based on E-Advisor," *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 1-7, 2018. doi: 10.1109/3ICT.2018.8855783
- [9] Hensher, Martin, Paul Cooper, Sithara Wann Arachchige Dona, Mary Rose Angeles, Dieu Nguyen, Natalie Heynsbergh, Mary Lou Chatterton, and Anna Peeters. "Scoping review: development and assessment of evaluation frameworks of mobile health apps for recommendations to consumers." *Journal of the American Medical Informatics Association* 28, no. 6 (2021): 1318-1329.
- [10] <https://ionicframework.com/>
- [11] Singh, Mohit & Shobha, G.. (2021). Comparative Analysis of Hybrid Mobile App Development Frameworks. 10.35940/ijscce.F3518.0710621.
- [12] Gowri, S., Surendran, R., Jabez, J., Srinivasulud, S. "IoT forensics: What kind of personal data can be found on discarded, recycled, or re-sold IoT devices," *Journal of Discrete Mathematical Sciences and Cryptography*, 2022, 25(4), pp. 999–1008.
- [13] Taoufik Rachad and Ali Idri, "Intelligent Mobile Applications: A Systematic Mapping Study," *Mobile Information Systems*, vol. 2020, no. 6715363, 2020. <https://doi.org/10.1155/2020/6715363>.
- [14] K. R. Ramkumar and C.S. Ravichandran, "A Novel node failure Management for Mobile Ad hoc networks using ant agents," *International Journal of Engineering and Technology*, vol. 6, no. 1, pp. 306-314, 2014.
- [15] Zahra, Husam Abu, and Samer Zein. "A Systematic Comparison Between Flutter and React Native from Automation Testing Perspective." 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2022.
- [16] Biørn-Hansen, Andreas, et al. "An empirical investigation of performance overhead in cross-platform mobile development frameworks," *Empirical Software Engineering* vol. 25, no. 4, pp. 2997-3040, 2020.
- [17] N. Tuli, and A. Mantri, "Evaluating usability of mobile-based augmented reality learning environments for early childhood," *International Journal of Human-Computer Interaction*, vol. 37, no. 9, pp. 815-827, 2021.