VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTSE@ 2022, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 10, Number 2, April-June 2022 *pp:79-93*

Cross-Platform Mobile Development Approaches and Frameworks

Muhammad Shoaib Farooq, Shamyla Riaz, Atif Alvi, Asghar Ali, Ibtesam U Rehman

Department of Department of Computer Science, University of Management and Technology, Lahore, Punjab, Pakistan
Corresponding author: Shoaib.farooq@umt.edu.pk

ABSTRACT

The use of mobile applications is rapidly increasing, due to rapid development of smartphones. Mobile apps, on the other hand, are platform-specific, making development more difficult and expensive. Cross platform or multiplatform application development is a fairly new idea in which developers apply a single code to create apps for several platforms like Android, BlackBerry, Windows Mobile, iOS etc. With the rising usage of these frameworks, it is important to comprehend both contributions and limitations in this emerging field. This paper presents a systematic literature review (SLR) of the research studies in the field of cross platform mobile app Development including approaches and tools. The SLR has been compiled by reviewing the research studies published between 2012-2022 in reputed venues. A Total of 22 studies has been selected and classified by using systematic process. The review has been presented on cross platform approaches, tools and challenges. Further an approach has been proposed for cross platform mobile application development. Finally, open issues and challenges in the field of cross platform have been presented to provide future direction to the researchers.

KEYWORDS

App, cross platform mobile application, multi-platform mobile application, react native

JOURNAL INFO

HISTORY: Received: March 17, 2022 Accepted: May 10, 2022 Published:May 20, 2022

I. INTRODUCTION

Smartphones have been more demanded as business machine in today's world. several sectors of life, such as business, education, and to name a few. Furthermore, leading influences such as banking, education, agriculture and healthcare are incorporating mobile applications [1]. Currently there are 7.1 Billion smart phone Users, by 2025, the number of smartphone users is anticipated to reach 2.87 billion [2]. The need of developing mobile app is increased incredibly with the number of mobile devices increased [3].

Mobile application development is one of the extremely fast fields, and mobile phones are constantly improving in terms of both hardware and software. People utilize a wide range of cellphones, resulting in a wide range of platforms and devices [4]. Four major platforms are iPhone, Android, BlackBerry, and Windows Phone. The most difficult task for programmers is to create apps for all of these platforms and operating systems [5]. The great majority of mobile apps in the app store are created utilising the native development technique. The native App development strategy is a methodology for developing apps utilizing the programming language and tools unique to the platform [29]. Furthermore, native application only work on the platform for which they were designed, which implies that building native apps for multiple platforms necessitates the hiring of numerous teams, each focusing in a different technology [2]. The native app development techniques with many of the great features in not attracting attention due to the cost, many resources and time required each platform app development. The developers must create and maintain the identical app for each platform [6]. Because of the prevalence of apps and the intricacy of mobile app development, a variety of technological development methodologies and tools have been emerged to augment or partially replace native app development [7][8]. Those alternative development approaches in mobile industry and research has been termed as multiplatform or cross platform. The area of cross platform is fast paced and more demanding form research and industry point of view. The adoption of multiplatform development techniques, which allow developers to design their apps with a range of features, is a perfect answer to this problem. Different techniques to cross-platform mobile app development include hybrid, interpreted, cross-complied, and other techniques [9]. The standard SDL (software Development cycle) cycle is show in part (a) of figure 1. The cross platform technique extends the support different platforms as shown in part (b) of fig 1. This technique offers solutions that make it SDL by building the app once and deploys it several times to easier to create an app once and then distribute it across a variety of platforms [10]. The aim of this study has been to answer the question of all those developers interested in entering the field of multi-platform mobile app development.

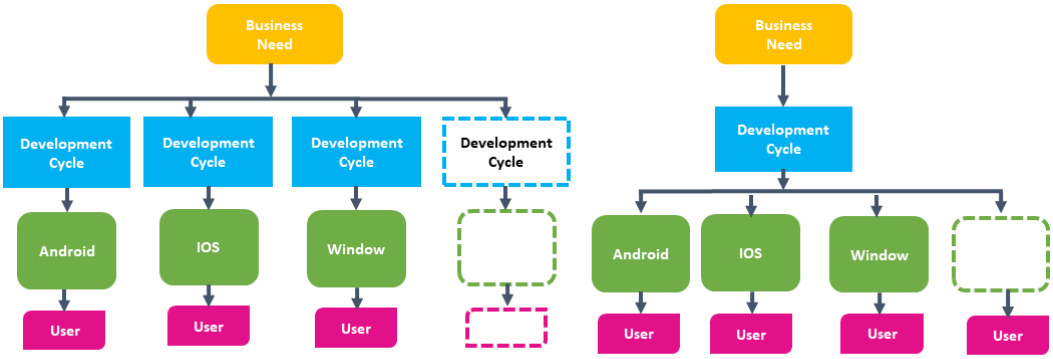

Figure 1 Native and cross platform mobile app development methods

The primary goal and contribution of this research was to provide answers to those questions via a systematic review of the literature. The secondary Goal of this research is to identity the research gaps, challenges and possibilities for future research.

The next section of this paper is formulated as: In Section II related work and main motivation is presented. Section III contains the research Methodology followed in this review, research Questions, motivations, search string, data collection strategy, Quality assessment of selected studies are presented. In section IV, the analysis and results are given in order to consider the extracted outcomes and to respond to the research questions in a systematic manner. In Section V discussion and proposed approach has been discussed. Lastly in section VI SLR has been concluded.

II. RELATED WORK

Appears to be the only study that provides a review of cross-platform development patterns that we are aware of. The authors concentrated their research on the emerging development challenges in the cross-platform field [11]. They also didn't include studies that looked at development tools or processes. Furthermore, they excluded research that looked at development tools or methodologies. Also, their study can't conduct a critical assessment of the papers that were included instead, they simply provided the discovered issues along with the offered remedies in each study. Likewise, we reviewed all relevant papers in various areas of interest and provided a review fo each one. An overview of newly approved developmental approaches and technologies was provided by the authors of another study [10]. They offered a quick overview of cross-platform platforms and strategies. Among the strategies discussed are web, hybrid, interpretative, cross-compiled, and model-driven techniques. They compared the strategies based on how they were used and the most popular platforms. They discovered that the hybrid strategy works well for producing low-complexity features. They go on to argue that cross-compiled platforms are advantageous for enterprise development because they just have to be written once and can then be deployed as native apps anywhere. The key problems in developing multi-platform mobile applications were identified in a study conducted by [11]. They characterized the significant concerns as follows: (a) Access to hardware features is limited. (b) low performance. (c) Poor UI Quality.

Another research [12] took a broad look at cross-platform development technologies and emphasized their benefits and drawbacks. Six distinct tools were studied and compared in their study. Rhodes, Phone Gap, DragonRAD, Titanium, mobile, and mdsl are the tools that were investigated. The study uncovered the strengths and limitations of each tool, as well as the that most of them never use models driven engineering, with just two of them doing so. In the area of mobile development, [12] conducts another rigorous mapping investigation. They provided an overview of the tools and approaches that allow automating the process of mobile app functional testing using a categorization scheme that includes numerous qualities and sub-attributes such as support for test automation, testing inputs, assessment, and others. They included 131 publications that were organised into categories based on the activities they supported, the testing methodologies they used, and the assessment method they used. The findings of this study revealed several research gaps and trends, including a lack of industry participation in the research. There is no comprehensive mapping research in the field of cross-platform development. There was only one literature survey, and it was done at a very early stage of cross-platform development. We found publications that provided an excellent summary of current development tools (e.g. [10], [11]and others). Other articles, such as [11], [8], explored the issues in this field. As a result, we conducted a detailed systematic mapping study in order to provide a full overview of current and recent research in this burgeoning field.

III. RESEARCH METHODOLOGY

For impartial data gathering and validation of studied and retrieved outcomes, the criteria for SLR established by [14]-[44] were followed in this work. The core purpose of Systematic Literature Review is to review the literature by systematic and step-by-step manners. For Successfully accomplishment of literature review SLR practices different processes.

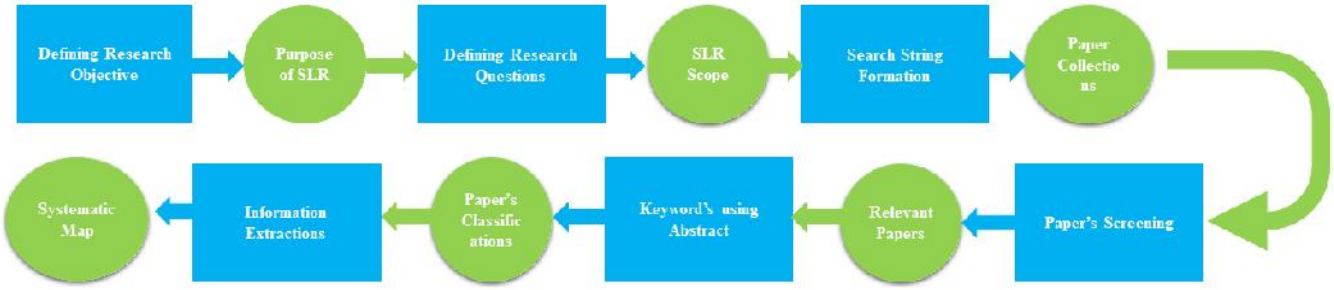

Figure 2 SLR Process Model

The process consists of three stages, includes planning and defining research Questions, conducting review which consists of identifying search string, data source, selecting studies, Quality evaluation and extract the data and at the end reporting review. SLR process model is illustrated in fig 2.

A. RESEARCH OBJECTIVES

The Following are the Core objectives of this review:

i. More focused on latest research work in the field of cross platform mobile application development has been identified.
ii. The prime goal is to investigate the research in the area of Multi-platform mobile app development.
iii. Different cross platform mobile app development approaches and tools has been deliberated to provide better understanding for the developer to choose best fit tool for creating mobiles apps.
iv. The hierarchal taxonomy of Mobile App development was purposed.
v. To identify future research potential, identify the main issues and challenges.

B. RESEARCH QUESTIONS

The primary research questions have been identified in order to properly conduct this SLR. In addition, a rigorous search strategy for identifying and extracting the key papers was devised as part of the evaluation. Table 1 lists research questions that were discussed in this review, as well as the primary motivations. The questions are answered in accordance with the approach in.

C. SEARCH SCHEME

The formulation of a search plan to appropriately identify and gather potentially relevant articles in the targeted subject is a crucial component of SLR. The description of the search string, the literature repositories used to apply the search string, and the inclusion and exclusion criteria used to find the best related articles from the collection are all part of this procedure. To reflect different perspectives related with the research, various aspects of the gathered papers were examined.

1) SEARCH STRING

Using multiple major research repositories, an effective search was conducted by generating a keywords-based search string to collect available works in this topic. The major ideas have been explored in light of research questions to collect relevant keywords and terms used in the given field of study, in order to assure the validity of the search string in terms of the relevancy of its findings.

Table 1 Resaerch Questions

NO:	Research Questions	Motivation
RQ 1	What are the major cross platform Approaches being used in software industry for Mobile Application development?	To Discover cross platform Mobile Application Development Techniques.
RQ 2	Which kinds of key tools/platforms are being used for Multiplatform Apps development?	This question discuss the top level tools or platform, which is used in cross platform mobile application development domain.
RQ 3	What are the substantial challenges or difficulties, which is frequently faced in cross platform mobile application development?	To identify main challenges faced in cross platform mobile Application development.

Table 2 lists the final keywords and synonyms that should be used to solemnise a search string in order to find the most relevant articles.

Table 2 Keywords

Keywords	Synonyms
+ Cross Platform Mobile Applications Development (CPMAD)	Hybrid Mobile Applications Development(HMAD)
+ Techniques(Tech)	Approaches(Ap)
+ Frameworks, Tools	Platforms
+ Challenges	Problems, Limitations
-Native App Development	Platform specific app development

Table 2 uses '+'and '-' signs for inclusion and exclusion of studies having such phrases respectively. We used the approach suggested in the study of to construct the search strategy while considering the research questions. Used AND & OR Logical operators to create search String.

(a) To limit the search by major terms used AND operator.

(b) To expand the search used OR operator

The final search string is consisting of four components. The first component of string is used to limit the results about the terms cross platform mobile app

development. Second component is related to cross app Approaches, and the third and fourth components are Frameworks, tools and challenges respectively. Mathematical representation of the search string is shown in the following equation.

$O = \forall\,[(CPMAD \vee HMAD) \wedge (Tech \vee Ap) \wedge (\text{FmWork} \vee \text{platforms}) \wedge (\text{challanges} \vee \text{problems}) \not\equiv (NMAD)]$

In the above equation 'O' stands for the Output of search result find in contradiction of search string, '∀' is used 'for all', ' ∨' is used for 'OR' Logical Operator, and '∧' used for Logical 'AND' operator for combining the search terms to create the final search string according to each selected digital repository .Keeping in mind the above expression the general search string was as follows:

(Cross Platform Mobile Applications Development OR Hybrid Mobile Applications Development) AND (Techniques OR Approaches) AND (Tools OR Platforms) AND (Challenges OR Limitations)

2) LITERATURE RESOURCES

Top Rated Journal were selected to conduct the filed specific literature search from the online digital repositories. For each selected repository, separate string has been defined. The selected digital repositories, applied data search Query and the finding were shown in Table 3.

Table 3 Publisher-specific search terms

Repositories	Search String
Google Scholar	(Cross Platform Mobile Applications Development OR Hybrid Mobile Applications Development) AND (Techniques OR Approaches) AND (Tools OR Platforms) AND (Challenges OR Limitations)
ACM Digital Repository	(Multiplatform Mobile Applications Development OR Hybrid Mobile Applications Development) AND (Approaches) AND (Tools OR Platforms) AND (Challenges OR Limitations)
Science Direct	(Hybrid Mobile Applications Development) AND (Techniques OR Approaches) AND (Tools OR Platforms) AND (Challenges OR Limitations)
IEEE Xplore	(Cross Platform Mobile Applications Development) AND (Techniques OR Approaches) AND (Tools OR Platforms) AND (Challenges OR Limitations OR Drawbacks)

3) INCLUSION, EXCLUSION CRITERIA

The following are the parameters to inclusion criteria (IC):

i. Include studies which have been conducted specifically on cross-platform mobile applications development.
ii. If the research was done on cross-platform tools and Approaches.

The studies that fulfilled at least one of the following exclusion criteria (EC) were excluded.

i. **EC1** Studies that aren't specifically about mobile app development.
ii. **EC2** Papers that focus on mobile application development in general.
iii. **EC3** Articles that are not about cross platform application Development.

D. STUDY SELECTION

To maintain relevancy, the publishing time for the search was set to 2010 to 2022. The main search String yields a large number of research publications, not all of which are relevant to the study topics, and some of which are duplicated. As a result, the articles that were searched must be re-evaluated and screened in order to obtain genuinely essential publications. For the selection of appropriate studies Inclusion and exclusion terms are defined.

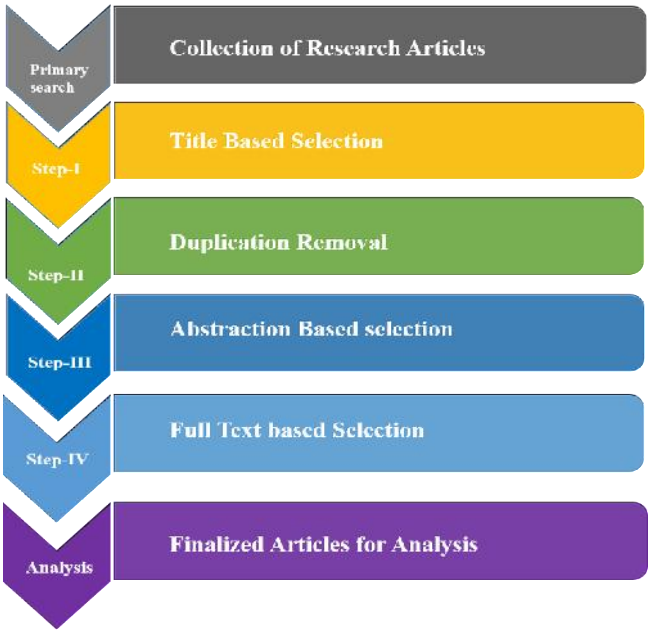

Figure 3 Study Selection Process

When an article appeared in several sources, it was only examined once in accordance with our search criteria. Each article was evaluated to determine whether or not it should be included, taking into account the title, abstract, and keywords. After the articles had been found, the first step was to remove duplicate titles and titles that were plainly unrelated to the review. The studies that fulfilled at least one of the exclusion criteria (EC) were excluded a complete study selection process is shown in Figure 3.

E. ABSTRACT BASED KEYWORDING

The concepts of were followed in this work, and systematic mapping approaches were used. In addition, the organization of our work was influenced by a systematic mapping study conducted by [2]. The first step is to read the

abstracts of the papers you've chosen to see if any key terms that reflect the study's contribution can be found. If the abstracts aren't strong enough to extract keywords, the researcher can concentrate on the introduction and conclusion sections instead. This method's purpose is to create a classification scheme that appropriately represents the research that has been included.

Thematic analysis is the second stage, in which selected keywords from different studies are combined to provide a better comprehension of the articles in issue. A set of key categories for the classification scheme was identified during this step. The keywords created during the first stage led to the discovery of the contribution for articles. Cross-platform approaches, analyzing cross-platform techniques, the impact of using cross-platform, offering new ways, and cross-platform difficulties and opportunities are among the themes we gathered. In the second phase, the results of the previous phase were combined to determine each publication's contributing area. "Framework analysis," "new method," and "comparative analysis" were the three categories used in the categorizing strategy.

F. Quality Assessment

In systematic literature reviews, quality assessment (QA) is frequently performed. We prepared quality questions based on quality criterion to assess the validity of chosen research articles as shown in Table 4. A prior research [5] provided inspiration for the format of this questionnaire. Quality standards are created to assess each of chosen paper and their conclusions, assuring the int egrity of the papers and their results.

Table 4 Quality Assessment Quentionaire

Sr NO.	Assessment Question	Expected Answer	Score
A	Was study provides a contribution toward cross platform Mobile App Development?	a. Yes b. No	a. 2 b. 0
B	Was study presents clearly cross platform mobile application development Approaches, Tools and challenges.	c. Yes d. Partially e. no	a. 2 b. 1 c. 0
C	Was the data collection methods, research methodology were well defined?	a. Yes b. NO	a. 3 b. 0
D	The research was distributed in a renowned and reliable journal. The (CORE) Computer Science Conference Rankings [6] (A, B, and C) and (JCR) the Journal Citation Reports lists were used to mark this question.	a. CORE A ranked b. CORE B ranked c. CORE C Ranked d. If no ranked	a. 3 b. 2 c. 1 d. 0
		a. JCR Q1 b. JCR Q2 c. JCR Q3 & Q3 d. No JCR ranking	a. 3 b. 2 c. 1 d. 0 e.
		other	0

The quality criteria (D) scores indicates that journals are preferable to conferences, seminars, and symposia because authors feel that publishing Paper in a JCR Q1 or JCR Q2 journal may be extra challenging than the other publication channels. Each study's final score was calculated by adding the four closed-question scores (an integer between 0 and 10).

IV. DATA ANALYSIS

This section summarizes the findings and provides a comprehensive assessment of all of the papers. To properly address the research questions, the selected papers were studied. The first section goes through the search results that were acquired using the search string that was specified. The evaluation score is described next, and the final section is devoted to in-depth discussions to address the research topics.

A. SEARCH RESULTS

We tried several times to ensure that we had the most dependable search string. "cross platform mobile application Apps," "approaches," "Tools/frames," and problems are all part of our search query. Finally, after numerous efforts with various pilot search strings, we settled on the very last one to be considered the correct one because it contains all of the critical phrases and nearly all of the retrieved research is a consequence of this Search string, which is:

((Cross Platform Mobile Applications Development OR Hybrid Mobile Applications Development) AND (Techniques OR Approaches) AND (Tools OR Platforms) (Challenges OR Problems OR Limitations)).

Our Search String outputs 64,475 from all digital repositories. On this searched studies, Study selection steps. Described above sections are applied. The different steps involve in selection process are descried in figure 3 and the phase wise outcomes are shown in table 5. The Study Selection Procedure of Title Based was done by two writers in Step-1, resulted with 275 articles. Next, in Step-II duplicate studies were removed, and the domain irrelevant papers were also removed on the bases of inclusion, exclusion criteria specified in above sections.

Table 5 Selected Studies

Online Repository	Primary Search	step-I	step –II	step - III	step - IV
Google scholar	17,900	95	35	20	8
IEEE Xplore	65	30	20	16	7
ACM Digital Library	46,430	100	55	15	5

Science Direct	80	50	18	5	2
Total	64,475	275	128	56	22

The Study Selection Procedure of Title Based was done by two writers in Step-1, resulted with 275 articles. Next, in Step-II duplicate studies were removed, and the domain irrelevant papers were also removed on the bases of inclusion, exclusion criteria specified in above sections. For example, searching procedure has also produced articles that were not targeting the cross platform application development in depth.so those studies were excluded due to their incomplete nature. The interrater agreement between the two writers in their evaluation was calculated using the Cohen's Kappa coefficient. According to [4], the Kappa coefficient was 0.95, indicating nearly complete agreement between the evaluations. The Abstract based procedure in Step-III was applied and resulted with 128 papers acquired from the earlier steps. Finally, in step-IV, a complete text-based analysis was performed on 56 articles, with 22 articles being considered to be the most relevant and selected for inclusion in this SLR for data extraction. According to the search string defined in Search string section, we used highly realized Digital Repositories (DR) to publish studies for different journals, workshops, and conferences to choose studies for this systematic literature review.

The DR-wise distribution percentage of a selected of articles, including Google Scholar as top ranked with 40% share, IEEE XPLORE with 32 % share, ACM Digital Library with 20% share and Science Direct with 8% share. Table 5 has already showed the publisher-based stages wise selection process and distribution percentage of selected studies.

B. QUALITY ASSESSMENT SCORE

According to the scoring system stated above for every selected article score were defined, articles were evaluated as per the quality assessment criteria and mentioned in Table 6

Table 6 Quality Assessment Score

References	**P.Channel**	**A**	**B**	**C**	**D**	**Total Score**
[2]	Journal	2	2	3	1	8
[6]	Journal	2	1	0	0	3
[7]	Conference	2	1	0	0	3
[8]	Journal	2	2	3	2	9
[9]	Conference	2	1	0	0	3
[10]	Conference	2	2	0	0	4
[11]	Journal	2	1	0	0	3
[12]	Conference	2	1	0	0	3
[44]	Journal	2	2	3	1	8
[45]	Conference	2	1	0	0	3
[47]	Journal	2	2	3	1	8
[48]**Error! Reference source not found.**	Conference	2	2	0	0	4
[49]	Other	2	1	0	0	3
[50]	Journal	2	2	3	1	8
[51]	Conference	2	1	3	3	9
[53]	Conference	2	1	0	0	3
[55]	Conference	2	2	0	2	6
[56]	Journal	2	1	3	1	7
[57]	Journal	2	2	3	1	8
[9]	Journal	2	2	3	3	10
[10]	other	2	1	0	0	3
[11]	Conference	2	1	3	3	9

C. ASSESSMENT AND DISCUSSION OF RESEARCH QUESTIONS

This section summarizes the key findings and evaluates all of the articles that were examined. The selected articles were examined in order to answer the study's questions defined in Table 1.

RQ1: *What are the major cross platform Approaches being used in software industry for Mobile Application development?*

RQ2: *Which kinds of key tools/platforms are being used for Multiplatform Apps development.*

RQ1 and RQ2 were addressed collectively with approaches and tools. Figure 4 the cross platform mobile development approaches. To answer these RQ the top research approaches and Tools has been mention in Table 7.

Table 7 Resaerch Approaches and Tools

	Cross platform Development Approaches			
	Hybrid Approaches	***Interpreted Approach***	***Cross-compiled Approach***	***Progressive Web Apps Approach***
Technical Framework's	Cordova	React Native	Flutter by Google	Ionic Framework
	Phone Gap	Native Script	Microsoft Visual Studio Xamarin	Zuix
	Capacitor	Titanium Appcelerator	Codename One	Mithril
	Ionic Framework	Adobe Air	Xojo Mobile	Polymer
	Sencha Touch	Kony	Qt Mobiles	Svelte

	Onsen UI	Weex	RAD Studio	Preact
	Intel XDk	Jasonette	Cross light	Angular
	RhoMobile	Smartface Cloud	RoboVM	React.js
	NSB/AppStudio	Moy Sync	Marmalade	Stencil.js
	Intel App Framework	Tabris.js	Rhodes	Ember.js
	Rho Mobiles	Fuse tools	MoSync	viperHTML
	Cocoon	LuaView	Corona	Moon.js
	Framework 7		MonoCross	Vue.js
	JQuery Mobile		Apporatable	Glimmer.js
	Trigger.io		DragonRAD	

First of all, the related literature to these questions have been presented for review purpose. Then the top cross platform techniques along with applied framework was presented.

Responsive Web Approach, Hybrid Approach, Interpreted Approach, and Cross-Compiled Approach are the four cross-platform mobile development methodologies classified by the authors in [8]. These tactics were investigated, as well as their benefits and drawbacks. According to the authors, developers should select the proper approach based on the App kinds. The authors divide the App into four categories: server data-driven, sensor/IO-based, standalone, and client–server. This study examines only a subset of existing cross-platform mobile development approaches and does not go into detail about how existing solutions apply them.

In [8], the author distinguishes between two sorts of cross-platform application development approaches. (1) methods for generating platform-specific Apps at compile time, such as Web Apps, Hybrid approaches, or Self-contained environment approaches, and (2) methods for generating platform-specific Apps at runtime, such as Model-Driven solutions and Cross-Compiling. The first method is the focus of the authors.

The authors defined three scenarios to evaluate native App development, one for each type of runtime environment. Mobile web apps, Phone Gap as a hybrid framework, and Titanium Mobile as a standalone solution are examples of this. From an infrastructure standpoint (licence and costs, supported platforms, access to platform-specific features, long-term feasibility, look and feel, application speed, and distribution) and from a development standpoint (development environment, Graphical User Interface (GUI) design, ease of development, sustainability, scalability, opportunities for further development, and pace and cost of development), the authors define a list of 14d criterion elements. Compilation, Component-Based, Interpretation, Modeling, Cloud Based, and Merged are the six primary techniques proposed in this study for cross-platform mobile development, according to study [8]. Sub-categorization techniques such as compilation, interpretation, and modelling are also recommended. The techniques to cross-platform application development explored in this study are depicted in Figure 4. Cross-platform programmes are divided into four types by the authors of [7]: web, hybrid, interpreted, and created.

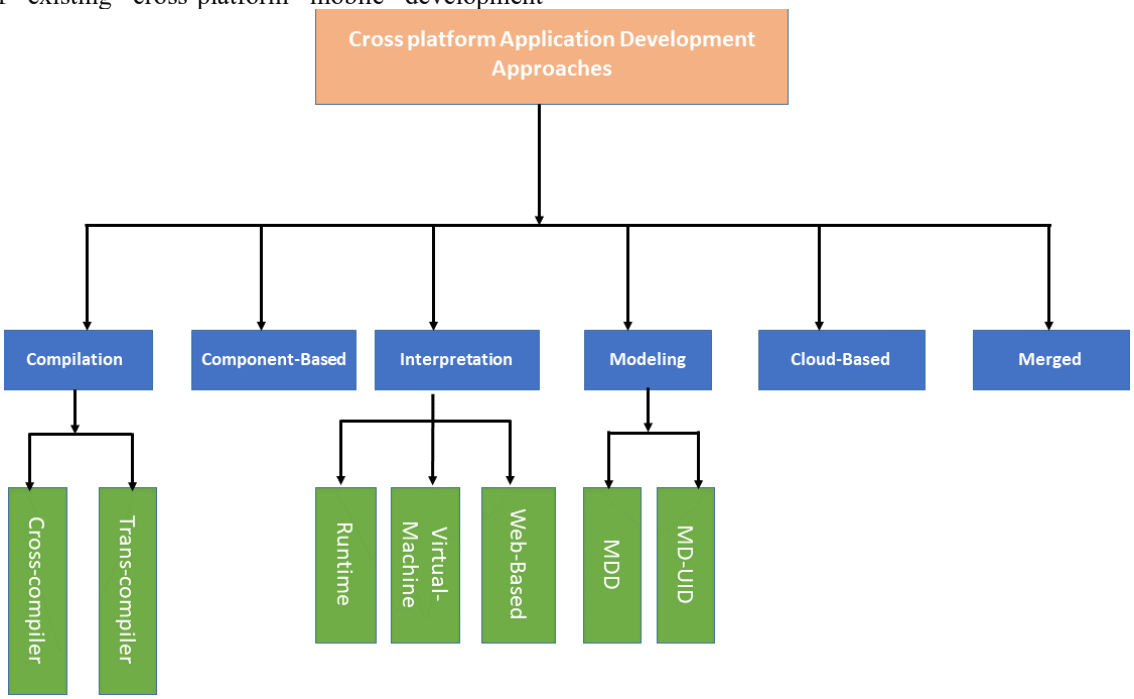

Figure 4 Cross Platform Mobile Development Approaches

The writers provide a comparative analysis to show the benefits and drawbacks of each type. Market place deployment (distribution) capability, generally used technologies for developing Apps, hardware and data access, user interface and look and feel, and user-perceived performance are among the comparing features (i.e. loading time and execution speed). The paper concludes with a case study that demonstrates how to build a mobile App (RSS Reader) using the cross-platform mobile programming tool "Titanium" and JavaScript without knowing anything about the target Android and iOS platforms.

ImagIngDev, an innovative solution for autonomously developing cross-platform mobile apps using image processing and pattern recognition techniques, was introduced by the authors in [12]. The major processes for automatic cross-platform app development were image analysis, configuration, and source code generation. In[13], a Model Driven Development (MDD) technique based on Domain Specific Language (DSL) was introduced. For each native platform, a cross platform code is built using this method. [7] presented a library to building of cross-platform apps incorporating acoustic sensors easier. In this study, the solution package LibAS was assessed by creating three apps with it. These apps addressed the most important features of acoustic sensing, including sound fingerprinting, cross interaction, and more. The results of the evaluation revealed that LibAS lowered the number of lines of code necessary, resulting in less work and development time.
(DSL) was presented in. In this approach a cross platform code is created for each native platform. Presented a library to building of cross-platform apps incorporating acoustic sensors easier. In this study, the solution package LibAS was assessed by creating three apps with it. These apps addressed the most important features of acoustic sensing, including sound fingerprinting, cross interaction, and more. The results of the evaluation revealed that LibAS lowered the number of lines of code necessary, resulting in less work and development time. Cross Platform Development proposed a static code approach, which is a tool for detecting data-flows in hybrid mobile apps by creating calls graphs. Apps extracted through Google Play were also used to test this method; those apps were written and used the Cordova framework as well as listed as highest Cordova apps throughout the store.

The findings demonstrated that the cross-platform language produced calls graphs are extremely exact. The performance is measured by the programming language employed, which in this case is JavaScript. proposed a novel code conversion strategy in which a method was proposed for locating a matching set of code patterns within the input source programmed in order to generate platform-specific code. The study compared the efficiency of ICPMD applications created with previous cross-platform solutions to ICPMD applications created with the new version. As a result of the experiment, performance, memory usage, and application size all improved dramatically. The above-mentioned cross platform mobile application development approaches have been discussed form the literature. All of the above-mentioned approaches were providing good understanding to the developer and researcher to choose best fit approach for their need. The top used cross platform approaches have been discussed below.
The relevant tools for specific approach is also presented.

1. Web Approach

The cross-platform web strategy is built on the web browser on the mobile device. In the development of all web-based apps, HTML, CSS, and JS are employed. Because it is multiplatform, the browser is employed as the runtime environment for these apps. This strategy is used to make the application available as a single mobile-friendly webpage is shown in figure. 5. This optimization must take into account the varying screen sizes of the devices as well as their usage patterns.

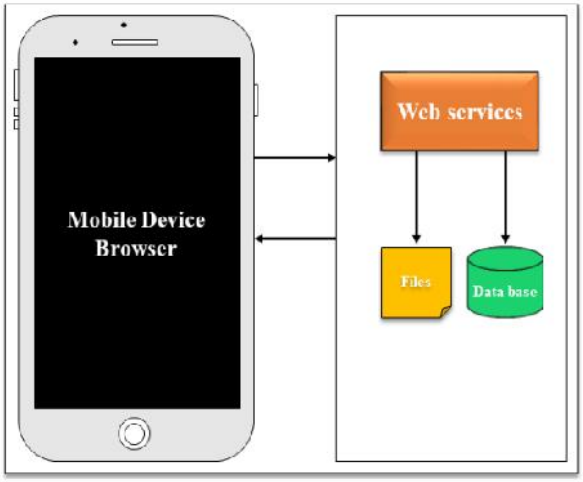

Figure 5 Mobile Friendly Web Page

Mobile web applications have the advantage of working consistently across all mobile web browsers and devices. As a result, there is no need to upgrade the mobile app. The web technique has the disadvantage of restricting access to the phone's native features (such as notifications, Geolocation, and mobile number). The time it takes to render web pages by downloading them from the internet is also longer than the time it takes to produce the native mobile user experience.

Furthermore, web apps are only accessible via a URL and are not generally available on mobile app stores. As a result, the approach's effectiveness would suffer. As shown in table 5, there are numerous platforms that support web-based programming. The most common are JQuery mobile, Bootstrap, and Sencha touch. The following is a comparison of these three choices.

(a) JQuery

JQuery Mobile is an open source platform that creates a highly responsive website that works on all major smartphones, tablets, and computers using a cross-platform web approach. The jQuery and jQuery UI frameworks are

used to create JQuery Mobile, an HTML5-based app. A large range of touch-friendly components and widgets are included, as well as page transitions, touch events, Ajax navigation, and a wide range of touch-friendly components and widgets. It's a simple code that makes use of progressive improvement to create a theme-able design. However, the application can take a long time to launch, and changing the theme involves some knowledge and work.

(b) Bootstrap

Bootstrap is a free and open-source framework for creating fully responsive web applications. Because we can design bootstrap components using only CSS class names or HTML5 elements, Bootstrap has a significant advantage over other frameworks in terms of ease of use. On the one hand, Bootstrap allows for the creation of additional extensions based on the framework's core functionality. The current version V5.X, which is still in development, is expected to add new features and improve optimization.

(c) Sencha Touch

Sencha Touch is a JavaScript user interface (UI) package or framework that is specifically designed for mobile websites. It can be used to create UI for web applications that looks like native apps on mobile devices. Sencha Touch is a library of controls (or components) for mobile web apps that are based on the graphical user interface (GUI). These components are intended for touch input and may be tailored to the target mobile device.

2. Hybrid Approach

The features of web applications are combined with those of native functions with in hybrid technique. This method makes advantage of the device's browsers engine and integrate HTML content in a native web container like Web View in Android and WKWebView in iOS. Through the usage of an API, native functionality may be accessed. Figure 6 shows an abstract JS bridge. Hybrid apps, unlike web applications, are distributed through apps store, and native functionalities are accessible via abstract layer. Hybrid interfaces, on either hand, perform worse than native interfaces since they are executed inside the web engine. Furthermore, the interface is not able to use the native appearance and feel. It becomes important to employ particular development libraries to do this. There was many framework that's works on the hybrid based development approaches. The popular frameworks of the hybrid approach were presented below like phone Gap, OnsebUI framework and Ionic Framework.

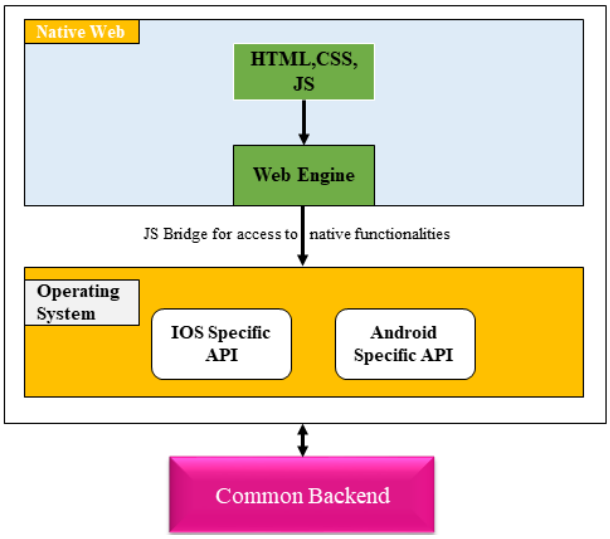

Figure 6 JS Bridge

(a) Phone Gap

The oldest and more famous Hybrid approach development framework is Phone Gap, that was introduced in 2011. It uses HTM5, CSS3 and JavaScript to generate an efficient web view and may be used in conjunction with other web frameworks. Phone gap just released a new Multi View plugin that allows you to start numerous Cordova Web views from within a single Phone Gap app. The Cordova plugin ensures native functionalities.

(b) Ionic Framework

Ionic Framework [34] is HTML5-based hybrid solution for creating native-feeling mobile apps. The current version of Ionic framework is Ionic v6.0.0. Cordova plugins [12] are used by Ionic to gain mobile natively capabilities like alerts and data storage. It has more performance characteristics than other hybrid frameworks, such as Phone gap, which is the most popular hybrid framework because to its optimized modern web technologies like Angular JS.

3. Interpreted Approach

Using a common language such as JS or others, the interpretative approach provides user interface code and native components for all platforms. The native APIs were accessed through an abstraction layer that, at runtime, converts code between platforms. This method has the advantage of allowing native user interfaces to be used. A downside is the reliance on the development platform. Unless the application framework supported new platform-specific features, such as new user interface enhancements, apps would not have had access to them. The call to the abstraction layer during runtime also degrades the application's performance. Figure 7 showing the most native components.

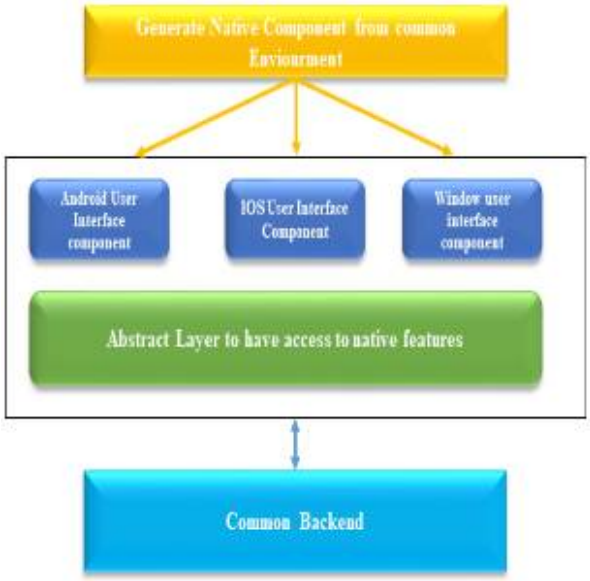

Figure 7 Native Components

(a) React Native

The React Native framework is based on React. Js, which was developed by Facebook to improve their chat. The business translated React to mobile devices after the interface built using JavaScript and the adaptive design approach adopted from the web proved to be so effective. React Native is now the most fascinating and popular framework, because it combines the best of the web with React.js. The JS-engine gives great performance that is equivalent to native. React Native, which is based on the notion of constructing an interface from blocks, does not require a browser or a Web View, instead relying on the JavaScript API. As a result, programmers create JavaScript code to interacts with native platform components, moving the benefits and usability of React.js from the browser to mobile applications. One of the biggest benefits of React Native it has a high percentage of shared code (up to 90%), which makes it easier to develop contemporary apps that seem native. At the same time, the development is easier.

(b) Appcelerator Titanium

Appcelerator Titanium [18] is free framework that enables for the building of mobile apps using an interpreted approach. It includes the alloy framework, which is an MVC (Model View Controller) development kit. Appcelerator Titanium, on the other hand, comes with an API for accessing local UI components like navigation bars, menus, and dialogue boxes, as well as native device functionality like the file, network, sensors, and geolocation.

4. Cross-Compiled Approach

In cross complied approach the developer's writer the code using a common programing language. Then these code are transformed by using the cross compiler for every single native platform. As shown in figure 8.

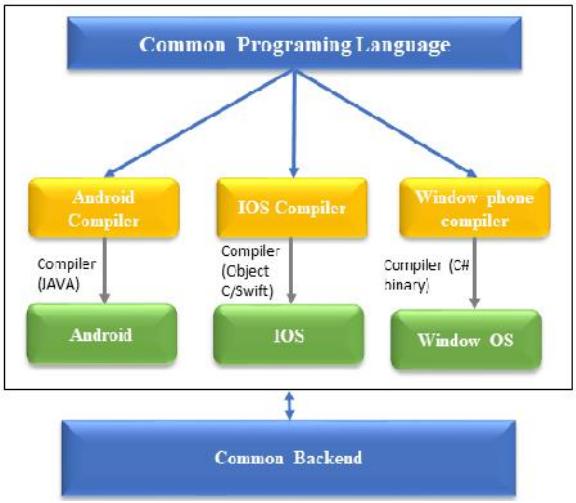

Figure 8 Cross Platform Approaches

The key advantage of this method is that the apps may operate at native rates and include all of the capabilities of native apps, such as native UI elements. This strategy has the drawback of preventing the use of a few functionalities. Access to webcams and geolocation services are two examples of such functionality. This is due to the fact that some functionalities are platform-specific, and access varies from platform to platform.

Many of the platforms has been mentioned in table 5 that support cross platform approach. The top rated and most demanding framework is Flutter. The reason behind the popularity of flutter was its offered by Google.Xamarine is also a good interpret tools.

(a) Flutter

Flutter is a relatively new cross-platform free software platform solution from Google aimed at making it easier to create beautiful, high-performance native applications for Android, desktop, web and IOS. The framework uses the Dart programming language, which was designed by Google and has a lot of similarities to Java and JavaScript.

The Flutter, like some other famous cross-platform frameworks, has a lot of helpful options like Hot reload, which UI development, implementation, and App testing

faster and easier. Third-party SDKs, APIs for 2D, animations, and native Material Design widgets are included in Flutter, as well as the possibility to existing facilities Java, Kotline, Swift, and Objective-C code.

Flutter is relatively new and free cross platform Software Development KIT(SDK) developed by Google. The Objective of Flutter is to release high performance mobile application for Android, IOS and Desktop platform more fastly and smoothly. Flutter neither used web View nor Java Script for their implementation, it implements UI components by itself [17].

(b) Xamarin

Xamarin Is open source platform, which makes use of a common C# codebase. Using the Xamarin platform, developers can create native Android, iOS, and Windows

apps that share code across platforms and native user interfaces. Xamarin apps are compiled for native performance and take use of platform-specific hardware acceleration. This is impossible to do using runtime code interpretation techniques. However, in order to take use of some platform-specific capabilities, Xamarin needs you to write custom code.

RQ3 *What are the substantial challenges or difficulties, which is frequently faced in cross platform mobile application development?*

Although multi-platform mobile application development has a lot of attractive advantages, it also has certain challenge's. Some of the major challenges faced by cross platform mobile app are following.

(a) Low Performance

The performance is one the most demanding feature of the mobile application. It depends upon many factors. If two applications were compare than the application developed by using native approach is faster than that of cross platform. However, in context of small application theses differences are very small.

(b) Switching platform

Many cross-platform tools employ JavaScript as its subject, which causes problems while using reusable code. As a result, locating the problem among all of the coding will become a time-consuming effort, increasing both the time and expense of android app development.

(c) Integration Process

Another challenge for cross platform is while integration, it takes more time. The integration with local setting becomes lengthier.

(d) Limited Update

The operating system may not always support all of the functionality that a framework requires. If the iOS platform delivers a new update or adds a new element, you will be needed to upgrade the iOS version of the application. However, you will not be able to do so until Google provides the same upgrade for Android.

(e) Sluggish Code

Cross-platform application development is not as simple as it sounds, especially for new programmers. As a result, developers employ cross-compliance during development phase, leading to sluggish code. Therefore, a slow application resulted.

V. DISCUSSIONS

Most of the cross platform mobile application development solutions used only single approach for development. That's why they are limited. The limitations of the current cross platform mobile application development solutions are as follows:

i. Each existing cross platform solution required to learn new programming language to use it, like Flutter required to learn Dart programming language to build cross platform app. Typically, the developer will design the mobile App using his or her chosen programming languages (e.g. Java for Android, Objective C for iOS, etc.).
ii. Only limited types of apps are supported, such as MDP [8], which only enables database-based Application development.
iii. Existing approaches concentrate only the Developing of the mobile app from scratch rather than utilizing existing Apps. For example, Flutter, React Native and MD^2 approaches build mobile app from start and not support code reuse. As a result, these produced Apps will be maintained or updated manually or by regenerating them.
iv. Static solutions are unable to keep up with platform providers' modifications to their tools and programming languages. For example, at Apple's Worldwide Developers Conference 2014, apple announced SWIFT [44], a new programming language for iOS and OSX development that would substitute Objective C, with both languages coexisting until developers are accustomed to the new language. As a result, J20bjC [45] will not be used if Apple discontinues support for Objective C development.
v. In different cross-platform strategies, the challenge of supporting new systems varies.

Due to the limitations of the existing solutions a new approach is proposed. The major objective for proposing the new solution Unified cross platform mobile application development Solution (UCPMADS) was to address the gaps in previous systems. The mobile developers need an integrated development solution that facilitates the development process, supports many mobile app features, and reduces the time and effort required to create a fully functional mobile app. The proposed (UCPMADS) solution (architecture is intended to provide a new integrated cross-platform mobile application development solution that overcomes the majority of the existing solutions' limitations. The following are some of the conditions that must be met by the proposed solution:

i. The develop mobile app type is hybrid to support the web and native platforms
ii. This solution does not need the developer to learn the new programming language. The programmer should utilize his or her favorite native app programming languages (e.g., Java, Swift, Dart, C#, etc.).
iii. The proposed solution handles both the developing of both the user interfaces and full source code.
iv. The solution should reuse the existing mobile application source code.
v. The solution will support the different types of mobile apps.
vi. The solution will make it as easy as possible to support new platforms.

The methodology of proposing the solution is based on classify the current tools according to their corresponding approach as mentioned. The study technique is based on classifying current solutions and technologies into

appropriate methods. The new propose solution architecture is then designed using a set of existing approaches to leverage the benefits of most of the approaches while overcoming their limitations. Finally, additional modules are added to help in the integration of the platform. Cross-compilation, Component-Based and Model Driven Development are the three approaches that have been chosen. Because the developer must learn a new programming language, other alternatives are ignored.

In the purposed solution different approaches are merged to produce “UCPMADS”. As a result, you'll have the following strengths: code analysis, code generation, code reuse, reverse engineering, code reuse, code analysis, reverse engineering, code creation, the capacity to concentrate on the application's functionality rather than the technical implementations, and the simplicity with which additional platforms may be supported. The architecture of the proposed solution “UCPMADS” shown in figure 9.

The suggested solution "UCPMADS" aims to reduce the time and effort required to build, update, manage, and document mobile applications that run on several platforms. The UCPMADS solution presents the developer with three possible scenarios(**SCE**) depending on his inputs:

a) **E1:** The application developer has a source code of the project and wants to produce a new project for another platform like android, IOS, blackberry.

b) **SCE2:** The developer has some requirements and wants to build a new project for the specific target platform.

c) **SCE3:** when the Developer has an abstract model type project and the developer wants to upgrade, store and turn into to application for a specific target platform.

The main building blocks of suggested solution “UCPMADS” has been discussed below.

i. Abstract Model

The abstract representation of the app which is independent from any platform. The model contains the XML files of manifest, UI layouts and resource draw able.

ii. App Modular

This module converts the sources mobile app project consisting of Resource files and XML (Hypertext Markup Language) into Abstract model app project. The idea of this module is similar to the [46] solution. But this module is different from because cross compiles the java classes byte code to XML files, While the UCPMADS provide converters for converting the manifest, resource files, code and source files of the source application code.

d) SCap Producer

In this module the abstract model project is transformed to target mobile app project like android, iOS. Each supported platform has its own implementation of this module. This module is similar in ideas to the solution mentioned in [48] and [27] but the implementation our module is different. UCPMADS, XMLVM [43], and MD^2 [49] all produce specific project files based upon the languages they provide. However, as would be explained later, UCPMADS employs a component-based architecture, with unified interfaces specified in the XML language of Abstract Model.

iii. App Reader

The contents of abstract model project are read by this module. This model can be sent into one or more of the following modules by the developer: App Producer, App Updater, and App Saver.

iv. AppWizard

This module provide assistance to the developer’s implement the user requirement to abstract module level project. This module does not need the developer to learn any additional languages. The developer may reuse predetermined template pages or create new ones, which can then be saved in the database and reused in future projects. This solution primarily assists the developer in focusing just on application's functionality instead of the underlying implementations.

v. App Updater

This app updater module is used if is there any need to update the abstract module project.

vi. App Saver

This app saver module is used to save the abstract model as project along document files for abstract model.

vii. Data bases

The databases are used for all modules of UCPMADS App modeler, app wizard, app producers, app updater and app saver. The databases are of following Template, UI definitions, permissions, class definition, folder structure and device profile.

viii. Platform Components

Every supported platform can have its own number of components to execute the mobile application's specified unified interfaces. These interfaces present in abstract model.

ix. Web based Contribution manager(WBCM)

The WBCM is used to update the contents of the UCPMADS database. Using a browser interface, the volunteer developers may contribute to the database enrichment. Before these contributions may be used by the UCPMADS solution modules, they must first be approved by the system Admin.

The ICPMADS approach provide 3 execution path depending on the input of the developer’s as fellows.

a) The developer offers a source app project like video downloader app code and wishes to create identical application projects for the other platforms supported (e.g. Android iOS). In this situation, the source platform's App Modeler will transform the source project to an Abstract model. Each targeted platform's App Producer will turn the Abstract Model into a platform-specific project (for example, iOS and Android apps). These projects may be built using platform-specific tools (e.g., Android Studio or Xcode) to build fully prepared Apps.

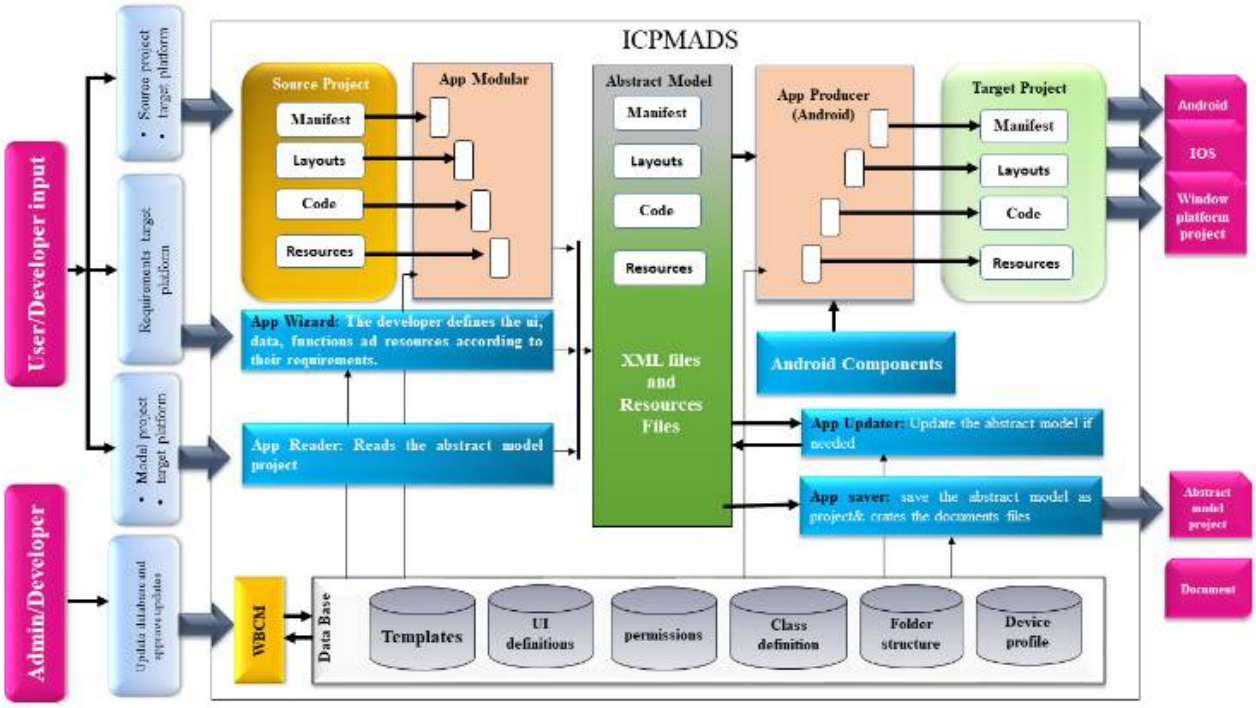

Figure 9 The architecture model of proposed approach UCPMA

The developer has been the list of user requirements then in this scenario, the AppWizard would be used to design the user interface, activities, methods, and resource files for the needed mobile app's Abstract model.

b) The developer will next provide the Abstract model to App Producers, who will create the targeted projects. The abstract model would be supplied as an entry to a App Updater, which will update any model changes, or to the App Saver, which will generate the Abstract model project and documentation files.

c) In this path the developer has been the abstract model type project, then in this case the App reader will used to produce the abstract model. Then this abstract model project is passed as an input to the further modules.

This proposed solution "UCPMADS" is the combination of the previous approaches and overcomes the issue of the previous solution with new options.

VI. CONCLUSION

This study has presented the systematics literature review on different cross platform mobile application development approaches, Frameworks and the challenge faced by using cross platform development. This research has been conducted by employing the systematic methodology to select 22 articles. Different approaches and tools has been discussed and proper understanding is provided to the readers. Although there are several techniques for developing cross platform mobile applications, each strategy has its own set of restrictions. To overcome these restrictions and allow a variety of usage situations, a new generic architecture (UCPMADS) for cross platform mobile app development is proposed. The future research directions to evaluate the performance of tools or frameworks of cross application development and to overcome the issues of existing solutions proposed better options.

REFERENCES

[1] M. Nagappan and E. Shihab, "Future Trends in Software Engineering Research for Mobile Apps," in 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), 2016.

[2] T. Zohud and S. Zein, "A systematic mapping study of cross-platform mobile apps," J. Comput. Sci., vol. 15, no. 4, pp. 519–536, 2019.

[3] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," J. Syst. Softw., vol. 80, no. 4, pp. 571–583, 2007.

[4] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," Biometrics, vol. 33, no. 1, pp. 159–174, 1977.

[5] S. Ouhbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, "Requirements engineering education: a systematic mapping study," Requir. Eng., vol. 20, no. 2, pp. 119–138, 2015.

[6] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Evaluating cross-platform development approaches for mobile applications," in Lecture Notes in Business Information Processing, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 120–138.

[7] S.Xanthopoulos and S. Xinogalos, "A comparative analysis of cross-platform development approaches for mobile applications," in Proceedings of the 6th Balkan Conference in Informatics on - BCI '13, 2013.

[8] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, "Taxonomy of cross-platform mobile applications

development approaches," Ain Shams Eng. J., vol. 8, no. 2, pp. 163–190, 2017.

[9] S. Amatya and A. Kurti, "Cross-platform mobile development: Challenges and opportunities," in ICT Innovations 2013, Heidelberg: Springer International Publishing, 2014, pp. 219–229.

[10] M. Latif, Y. Lakhrissi, E. H. Nfaoui, and N. Es-Sbai, "Review of mobile cross platform and research orientations," in 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), 2017.

[11] Alamri, H. S., & Mustafa, B. A. Software engineering challenges in multi platform mobile application development. Advanced Science Letters, vol. 20, no. 9, pp. 2115-2118 2014.

[12] A. Ribeiro and A. R. da Silva, "Survey on cross-platforms and languages for mobile apps," in 2012 Eighth International Conference on the Quality of Information and Communications Technology, 2012.

[13] P. Tramontana, D. Amalfitano, N. Amatucci, and A. R. Fasolino, "Automated functional testing of mobile applications: a systematic mapping study," Softw. Qual. J., vol. 27, no. 1, pp. 149–201, 2019.

[14] Farooq, M. S., Khan, S. A., Abid, K., Ahmad, F., Naeem, M. A., Shafiq, M., & Abid, A. (2015). Taxonomy and design considerations for comments in programming languages: a quality perspective. Journal of Quality and Technology Management., vol. 10, no. 2, pp. 167-182, 2015.

[15] M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Azam, and U. Farooq, "Resource allocation techniques in cloud computing: A review and future directions," Elektron. ir elektrotech., vol. 26, no. 6, pp. 40–51, 2020.

[16] I. A. Khawaja, A. Abid, M. S. Farooq, A. Shahzada, U. Farooq, and K. Abid, "Ad-hoc collaboration space for distributed cross device mobile application development," IEEE Access, vol. 8, pp. 62800–62814, 2020.

[17] M. S. Farooq, Z. Kalim, J. N. Qureshi, S. Rasheed, and A. Abid, "A blockchain-based framework for distributed agile software development," IEEE Access, vol. 10, pp. 17977–17995, 2022.

[18] A. Abid, M. S. Farooq, and U. Farooq, "A strategy for the design of introductory computer programming course in high school," Journal of Elementary Education, vol. 25, no. 1, pp. 145–165, 2015.

[19] R. Tehseen, M. S. Farooq, and A. Abid, "Fuzzy Expert System for earthquake prediction in western Himalayan range," Elektron. ir elektrotech., vol. 26, no. 3, pp. 4–12, 2020.

[20] A. H. A. Zahid, M.Waji Haider, M. S. Farooq, A. Abid, and A. Ali, "A critical analysis of software failure causes from project management perspectives," VFAST trans. softw. eng., pp. 113–119, 2018.

[21] M. S. Farooq, M. Khan, and A. Abid, "A framework to make charity collection transparent and auditable using blockchain technology," Comput. Electr. Eng., vol. 83, no. 106588, p. 106588, 2020.

[22] Farooq, M. S., Khan, S. A., & Abid, A. A framework for the assessment of a first programming language. *Journal of Basic and Applied Scientific Research*, Vol. *2, no.* 8, pp. 8144-8149, 2012.

[23] Farooq, M. S., Abid, A., Khan, S. A., Naeem, M. A., Farooq, A., Abid, K., & Shafiq, M. (2012). A Qualitative Framework for Introducing Programming Language at High School. *Journal of Quality and Technology Management*, Vol. *8,* no. 2, pp. 135-151.

[24] M. Ramzan, M. S. Farooq, A. Zamir, W. Akhtar, M. Ilyas, and H. U. Khan, "An analysis of issues for adoption of cloud computing in telecom industries," Eng. technol. Appl. sci. res., vol. 8, no. 4, pp. 3157–3161, 2018.

[25] A. A. Shah, M. Khurram Ehsan, K. Ishaq, Z. Ali, and M. S. Farooq. (2018). An efficient hybrid classifier model for anomaly intrusion detection system. *IJCSNS*, vol. *18 no.* 11, pp. 127-136.

[26] A. Naeem, M. S. Farooq, A. Khelifi, and A. Abid, "Malignant melanoma classification using deep learning: Datasets, performance measurements, challenges and opportunities," IEEE Access, vol. 8, pp. 110575–110597, 2020.

[27] I. Obaid, M. S. Farooq, and A. Abid, "Gamification for recruitment and job training: Model, taxonomy, and challenges," IEEE Access, vol. 8, pp. 65164–65178, 2020.

[28] O. Aziz, M. S. Farooq, A. Abid, R. Saher, and N. Aslam, "Research trends in enterprise service bus (ESB) applications: A systematic mapping study," IEEE Access, vol. 8, pp. 31180–31197, 2020.

[29] A. Arooj, M. S. Farooq, T. Umer, and R. U. Shan, "Cognitive Internet of Vehicles and disaster management: A proposed architecture and future direction," Trans. emerg. telecommun. technol., p. e3625, 2019.

[30] E. Mehmood, A. Abid, M. S. Farooq, and N. A. Nawaz, "Curriculum, teaching and learning, and assessments for introductory programming course," IEEE Access, vol. 8, pp. 125961–125981, 2020.

[31] A. Arooj, M. S. Farooq, A. Akram, R. Iqbal, A. Sharma, and G. Dhiman, "Big data processing and analysis in internet of vehicles: Architecture, taxonomy, and open research challenges," Arch. Comput. Methods Eng., vol. 29, no. 2, pp. 793–829, 2022.

[32] R. Tehseen, M. S. Farooq, and A. Abid, "Earthquake prediction using expert systems: A systematic mapping study," Sustainability, vol. 12, no. 6, p. 2420, 2020.

[33] M. Attique, M. S. Farooq, A. Khelifi, and A. Abid, "Prediction of therapeutic peptides using machine learning: Computational models, datasets, and feature encodings," IEEE Access, vol. 8, pp. 148570–148594, 2020.

[34] M. S. Farooq, R. Tahseen, and U. Omer, "Ethical guidelines for AI: A systematic literature review," VFAST trans. softw. eng., vol. 9, no. 3, pp. 33–47, 2021.

[35] A. Abid, W. Ali, M. S. Farooq, U. Farooq, N. S. Khan, and K. Abid, "Semi-automatic classification and duplicate detection from human loss news corpus," IEEE Access, vol. 8, pp. 97737–97747, 2020.

[36] A. Khelifi, O. Aziz, M. S. Farooq, A. Abid, and F. Bukhari, "Social and economic contribution of 5G and blockchain with green computing: Taxonomy, challenges, and opportunities," IEEE Access, vol. 9, pp. 69082–69099, 2021.

[37] D. M. Vistro, M. S. Farooq, A. U. Rehman, and S. Malik, "Smart application based blockchain consensus protocols: A systematic mapping study," in Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021), 2021.

[38] Farooq, M. S., & Akram, S. IoT IN AGRICULTURE: CHALLENGES AND OPPORTUNITIES. *J. Agric. Res*, vol. *59 no.* 1, pp. 63-87, 2021.

[39] L. A. Haafza, M. J. Awan, A. Abid, A. Yasin, H. Nobanee,

and M. S. Farooq, “Big Data COVID-19 systematic literature review: Pandemic crisis,” Electronics (Basel), vol. 10, no. 24, p. 3125, 2021.

[40] A. Rashid, M. S. Farooq, A. Abid, T. Umer, A. K. Bashir, and Y. B. Zikria, “Social media intention mining for sustainable information systems: categories, taxonomy, datasets and challenges,” Complex intell. syst., 2021.

[41] M. S. Farooq et al., “Untangling computer-aided diagnostic system for screening diabetic Retinopathy based on deep learning techniques,” Sensors (Basel), vol. 22, no. 5, p. 1803, 2022.

[42] M. J. Anjum and M. S. Farooq, “SDN based V2X networks for disaster management: A systematic literature review,” VFAST trans. softw. eng., vol. 9, no. 4, pp. 82–91, 2021.

[43] M. Shaheen, M. S. Farooq, T. Umer, and B.-S. Kim, “Applications of federated learning; Taxonomy, challenges, and research trends,” Electronics (Basel), vol. 11, no. 4, p. 670, 2022.

[44] Latif, M., Lakhrissi, Y., & Es-Sbai, N. (2016, March). Cross platform approach for mobile application development: A survey. In 2016 International Conference on Information Technology for Organizations Development (IT4OD) (pp. 1-5). IEEE.

[45] V. Y. Rosales-Morales, L. N. Sánchez-Morales, G. Alor-Hernández, J. L. Garcia-Alcaraz, J. L. Sánchez-Cervantes, and L. Rodriguez-Mazahua, “ImagIngDev: A new approach for developing automatic cross-platform mobile applications using image processing techniques,” Comput. J., vol. 63, no. 5, pp. 732–757, 2020.

[46] H. Benouda, M. Azizi, M. Moussaoui, and R. Esbai, “Automatic code generation within MDA approach for cross-platform mobiles apps,” in 2017 First International Conference on Embedded & Distributed Systems (EDiS), 2017.

[47] P. Nawrocki, K. Wrona, M. Marczak, and B. Sniezynski, “A comparison of native and cross-platform frameworks for mobile applications,” Computer (Long Beach Calif.), vol. 54, no. 3, pp. 18–27, 2021.

[48] L. Gaouar, A. Benamar, and F. T. Bendimerad, “Model driven approaches to cross platform mobile development,” in Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication, 2015.

[49] Y.-C. Tung, D. Bui, and K. G. Shin, “Cross-platform support for rapid development of mobile acoustic sensing applications,” in Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, 2018.

[50] P. Smutny, “Mobile development tools and cross-platform solutions,” in Proceedings of the 13th International Carpathian Control Conference (ICCC), 2012.

[51] I. Dalmasso, S. K. Datta, C. Bonnet, and N. Nikaein, “Survey, comparison and evaluation of cross platform mobile application development tools,” in 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), 2013, pp. 323–328.

[52] T. Zohud and S. Zein, “Cross-platform mobile app development in industry: A multiple case-study,” Computing, vol. 20, no. 1, pp. 46–54, 2021.

[53] T. F. Bernardes and M. Y. Miyake, “Cross-platform mobile development approaches: A systematic review,” IEEE Lat. Am. Trans., vol. 14, no. 4, pp. 1892–1898, 2016.

[54] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, “Enhanced code conversion approach for the integrated cross-platform mobile development (ICPMD),” IEEE trans. softw. eng., vol. 42, no. 11, pp. 1036–1053, 2016.

[55] C. Escoffier and P. Lalanda, “Managing the heterogeneity and dynamism in hybrid mobile applications,” in 2015 IEEE International Conference on Services Computing, 2015.

[56] JS Foundation-js. foundation, “JQuery mobile,” Jquerymobile.com. [Online]. Available: https://jquerymobile.com/. [Accessed: 15-May-2022].

[57] M. Otto, J. Thornton, and Bootstrap contributors, “Bootstrap,” Getbootstrap.com. [Online]. Available: https://getbootstrap.com/. [Accessed: 15-May-2022].

[58] “Sencha Touch,” Sencha.com, 28-Mar-2015. [Online]. Available: https://www.sencha.com/products/touch/. [Accessed: 15-May-2022].

[59] Ionic, “Cross-platform mobile app development:,” Ionic Framework. [Online]. Available: https://ionicframework.com/. [Accessed: 15-May-2022].

[60] “FAQ,” Flutter.dev. [Online]. Available: https://docs.flutter.dev/resources/faq. [Accessed: 15-May-2022].

[61] B. S.Thakare, D. Shirodkar, N. Parween, and S. Parween, “State of art approaches to build cross platform mobile application,” Int. J. Comput. Appl., vol. 107, no. 20, pp. 22–23, 2014.

[62] K. Verma and D. Maniktala, “Comparative analysis of cross-platform MAD frameworks,” INT. J. OF COMPUT. TECHNOL., vol. 13, no. 11, pp. 5162–5171, 2014.

[63] “Xamarin,” Microsoft. [Online]. Available: https://dotnet.microsoft.com/en-us/apps/xamarin. [Accessed: 15-May-2022].

[64] “Forecast number of mobile users worldwide 2020-2025,” Statista. [Online]. Available: http://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010. [Accessed: 15-May-2022].

[65] Asroni, S. Riyadi, and T. Cahyono, “Information system for providing food services based on mobile application using flutter framework,” in Proceedings of the 4th International Conference on Sustainable Innovation 2020–Technology, Engineering and Agriculture (ICoSITEA 2020), 2021, pp. 164–169.

[66] Apple Inc, “Swift,” Apple.com. [Online]. Available: https://developer.apple.com/swift/. [Accessed: 15-May-2022].

[67] “J2ObjC,” Google Developers. [Online]. Available: https://developers.google.com/j2objc. [Accessed: 15-May-2022].

[68] A. Datta, S. Kajanan, and N. Pervin, “A mobile app search engine,” Mob. Netw. Appl., vol. 18, no. 1, pp. 42–59, 2013.