Technoscience Journal for Community Development in Africa 4 (2025) 211–225
ISSN 2659-0573

Research Article

Cross-platform mobile application development using the low code technology and free and open-source technology

Fatima Enehezei Usman-Hamza[1*], Oluwafemi Ayobami Olutuase[1], Abdullateef Oluwagbemiga Balogun[2], Hammed Adeleye Mojeed[1,3], Shakirat Aderonke Salihu[1], Abimbola Ganiyat Akintola[1], Amos Orenyi Bajeh[1], Ghaniyyat B. Balogun[1], Modinat A. Mabayoje[1], Peter O. Sadiku[1], Ikeola Suhurat Olatinwo[1], Muhammed Jamiu Abdulrafiu[1], Adeleke Raheem Ajiboye[1]

[1]Department of Computer Science, University of Ilorin, Ilorin, Nigeria
[2]Department of Computer and Information Science, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia
[3]Department of Technical Informatics and Telecommunications, Gdansk University of Technology, Gabriela Narutowicza, Gdansk, Poland

Abstract: Cross-platform mobile application development has become a crucial tactic for organizations to reach a wider audience in today's fast-paced digital ecosystems. To maximize user engagement developers must create applications that work flawlessly across several platforms, such as ios and android. The use of low-code and free and open-source technologies in cross-platform mobile application development is explored in this research. This project makes use of the alpha anywhere community edition software as the low code tool which offers x-basic coding system, pre-built components, and easy visual interfaces while the mysql serves as the free and open-source database which is strong in security, reliability and can help contain scalable applications at minimal cost. The project development took advantage of the benefits of both methods by integrating the low code platform and free and open-source technology. rapid application assembly was made possible by the low-code platform, while free and open-source framework offers the performance and flexibility needed for cross-platform development. In conclusion, cross-platform mobile application development using low code and free and open-source technologies is an appealing option for organizations looking for effective, affordable, and scalable solutions.

Keywords: Keywords: Cross platform, Application, Open Source, Low Code, development

1. Introduction

Cross-platform mobile application development refers to the process of creating mobile applications that can run on multiple platforms, such as iOS and Android, using a single codebase (Kuitunen, 2019). Low-code technologies are tools that can simplify the process of mobile app development by allowing developers to create applications without having to write complex code. Low code development platforms (LCDPs) allow the design and execution of applications by utilizing visual abstraction and graphical user interfaces that automates each stage of the application development lifecycle (Sanchis et al., 2019). Computer software that is both Free software as well as Open-source software is referred to as Free and Open-Source Software (FOSS). Software with a free software license, which permits users to use, modify, and distribute it with little or no restrictions, and an open-source license, which permits users to read and alter the software's source code, is also referred to as FOSS (Stallman, 2022). This study intends to develop a mobile application that will operate on Android and iOS mobile operating systems with the usage of Low-Code and the FOSS technologies tools. The FOSS tool to be used is MySQL database management system while the Low Code software tool

* Corresponding author
Email: usman-hamza.fe@unilorin.edu.ng

KWASU PRESS AND PUBLISHING

is the Alpha Anywhere Community Edition mobile application development software. Mobile Applications or simply Mobile Apps are used for mobile devices, such as smartphones and tablets. These apps are available for download and installation from a number of app stores, including the Google Play Store and the Apple iOS App Store, and they can be utilized for a range of tasks, including communication, productivity, entertainment, and more. Our daily lives are filled with mobile applications, and their demand has been rising quickly. The market for mobile applications is anticipated to increase from $106.27 billion in 2018 to $407.31 billion in 2026, according to (Syeedet al., 2021). There are over 2.2 million apps on the Apple iOS App store while the Google Play store has over 3.4 million apps(Rouse, 2020).

Mobile Apps are categorized into three basic types namely: Native, Web and Hybrid Apps (Kathuria & Gupta, 2015). These are highlighted below: **Native Apps**: Native apps were created especially for a specific mobile operating system, such as Android or iOS, and are intended to make use of that platform's features and functionalities.

Web Apps: These apps are accessed through the web browser on a mobile device as opposed to being downloaded and installed on the device itself. Usually, web technologies like HTML, CSS, and JavaScript are used to create them.

Hybrid Apps: Apps that integrate features from both native and web-based applications are known as hybrid apps. They are created with web-based tools like HTML, CSS, and JavaScript but are put inside a native container so they can operate on mobile devices.

The Uses of Mobile apps are enumerated below according to (Rohan Vaidya, 2023) who described that the uses are vast and in various ways in our daily activities. Some of the uses are: **Entertainment:** Mobile apps can be used for leisure activities including playing games, watching films and listening to music, and visiting social media sites. **Productivity:** By letting users manage their schedules, make to-do lists, and access tools and resources for their jobs, mobile apps can help users become more productive. There are several productivity-focused applications available, including learning and informational apps that can help you develop your abilities and discover your hobbies.

Communication: Mobile apps, such as messaging, email, and video conferencing ones, can be used to facilitate the passing of information between people and groups. Applications exist that make communicating simple. With the use of a mobile app, you may communicate with individuals in any part of the world via voice conversations, video calls, texts, and conference calls. **Ease of usage:** It more fully guarantees comfort and ease of which one can buy, sell, survey, etc. It gets rid of the lengthy line for ticket purchasing and for instance arranges for a taxi to arrive at the doorstep e.g., Uber Transportation System.

M-Commerce: Mobile applications with complete security make it easier to earn and spend money. It can obtain a loan for you, calculate interest, list the best insurance options for you, choose the financially soundest businesses for you, and provide other crucial information.

The development of mobile apps can take three approaches which are Conventional Coding approach, Low Coding approach and the third approach being the No Coding approach. Conventional software development techniques for mobile applications need substantial coding expertise, which lengthens and increases the cost of the development process. Developers are therefore looking for new and more effective ways to creating mobile applications. Alternatives to Conventional software development are emerging, including platforms with No Code and Low Code technologies. This project will focus on the Low Coding Approach.

Due to the complexity of their internal technical operations and the volume and intensity of their contacts with the entities in their supply network, businesses or organizations today are forced to deal with more challenging issues. Additionally, because of the dynamic nature of the market, businesses must be quick to adapt and flexible in order to meet ever changing technological requirements. Companies must have the ability to withstand the strains of environmental loading because of this. Enterprise resilience is the term used to describe this capability. According to Sanchis and Poler (Sanchis & Poler, 2019), it is the ability to avoid problems before they arise, to plan ahead, to alter an organization's nature to better fit a changing environment, and to satisfy changing customer demands. As a result, many businesses choose Free and Open-Source Software as an affordable alternative. Additionally, they are using software development tools to quickly design and construct mobile software without the need for software developer specialists, which lowers costs and shortens the time needed to develop a mobile app using traditional methods. These programs have graphical user interfaces and are simple to

comprehend and operate. Thus, the use of No Code or Low Code applications is required.

Forrester Research, based in Cambridge, Massachusetts, USA, first used the term "Low-Code" in 2014 (Richardsonet al., 2014), stating that businesses prefer to use Low-Code alternatives for quick, continuous, and test-and-learn delivery. Low-Code development platforms are integrated development environment that make it easier to create apps because the software infrastructure is already in place and pre-configured. Low-Code development platforms put an emphasis on visual interfaces to make it relatively simple for those with limited background in technology to construct and deploy commercial apps. The fundamental goal of the Low-Code development platforms is to enable businesses to create apps without the need for complicated engineering, making it easier to configure them and achieve rapidity and agility. Additionally, these platforms give businesses a more affordable approach to meet their internal and/or external customer needs. Enterprises or organizations can construct multipurpose, highly information-managing programme and apps for desktop or mobile devices using Low-Code development platforms.

Free and Open-source software is software that is made available under license conditions that let it to be freely used, modified, and distributed. It might be constrained, for instance by mandating that copies of the license language, attribution notices, disclaimers, notice files, and notice files be kept when the software is distributed. The source code, including that of any connected software, may occasionally also need to be made available to any recipient of the code (often under the same license terms). Technically, any program licensed in accordance with the Open-Source Definition (Perens, 1999), which is managed by the Open-Source Initiative (OSI, 2007), is considered to be Free and Open-Source software.

According to World Intellectual Property Organization (Andrew Katz, 2021) access to the source code is only one aspect of FOSS. The following requirements must be met by Open-Source Software distribution terms: Free Redistribution, Source Code, Derived Works, Integrity of the Source Code of the Author, No Discrimination Against Individuals or Groups, No Discrimination Against Professions, License Distribution, License must not be tied to a particular product and cannot impose restrictions on other software. and License Must Be Technology-Neutral.

Due to the increasing use of FOSS code in software development, including the creation of mobile applications, FOSS is crucial to the development of mobile apps, this is prevalent in the Android environment which is powered by Open Source. Due to the evident advantages that FOSS components, frameworks, tools, and modules provide; this trend is probably going to continue. Reduced development costs, time, and go-to-market times are all benefits of using FOSS software. Access to and involvement in communities, the advantages of influencing code creation, and the ease with which FOSS -friendly businesses can acquire qualified staff are further perks in mobile application development.

1.1. Mobile applications

A mobile application is a software program made specifically to run on portable electronics like smartphones and tablets. It is the outcome of current technological advancements. The combination of media, information technology, the Internet, and cutting-edge technologies has given rise to mobile applications. Additionally, mobile device makers, mobile service providers, application developers, and several researchers in the fields of information technology (IT) and information systems (IS) have been studying mobile telecommunications for many years. However, the emergence of mobile applications is the most fascinating field of study (Phongtraychack, & Darya, 2018).

1.2. The history of mobile applications

The first mobile phone call was made on April 3rd, 1973, between Dr. Joel S. Engel of Bell Labs and Martin Cooper of Motorola. The object or instrument had dimensions of (23 x 13 x 4.45) cm and weighed 1.1 kg. And it took two decades of research and development (R&D) to create the first mobile app for smartphones; IBM Simon deserves credit for introducing the world to these applications. He devised the first smartphone to support a third-party application. From their early days on what is regarded as the first Personal Digital Assistant (PDA), mobile applications have gone a long way. The Psion Organizer I model, marketed as the "World's First

Practical Pocket Computer" in the 1980s, was created by Psion and featured a calculator, a clock, and other well-known applications (McGerty, 2021).

The world's first smartphone, released by IBM in 1993, arrived about ten years later and featured many of the same practical apps as the Psion. When the BlackBerry Smartphone was announced in 2002 and the firm added email capabilities to its phone, the next advancement in app technology occurred after another ten years. Some of the earliest mobile apps are Psion EPOC (operating system) which was developed by Psion for portable devices, primarily PDA in the early 1990s, Palm OS also known as the Garnet OS, which was developed by Palm, Inc in 1996, WML (Wireless Markup Language) developed by the Wireless Application Protocol (WAP) Forum for use in Wireless Application Protocol enabled devices and it is based on Extended Markup Language (XML), The Psion EPOC mobile operating System served as the foundation for Symbian, which was created by Symbian Ltd., a partnership between Psion, Ericsson, Nokia, and Motorola. In 2009, Symbian was running on 250 million devices, making it practically ubiquitous. Symbian OS was developed by Nokia, and almost all Nokia mobile devices, along with select LG and Samsung models, utilized the S60 platform. Then, in 2010, when Apple unveiled its controversial app store, which opened a year later, mobile apps were thrust into the spotlight (Rajput, 2021).

The advent of Mobile applications, or mobile apps, have become a vital part of our daily lives. There appears to be an app for almost everything, from social media apps to fitness monitors. The mobile apps originated from the earliest mobile devices, such as PDAs (Personal Digital Assistants) and feature phones. Over the past ten years, the mobile application revolution has been the engine behind the smartphone revolution. Smart phones' success has spread to a variety of gadgets, including tablets, wearables, and sensors, which are all now acknowledged as being a part of the mobile device platform. There is still a long way to go before we reach a degree of standardization, despite the enormous success and significant progress in regard to software platforms, hardware standards, development processes, and use. Presently numerous app stores, such as the Apple App Store, Google Play Store, and Amazon Appstore, already offer millions of mobile apps. People use mobile applications for a variety of purposes, from communication and entertainment to productivity and health, and they have become an indispensable part of our life.

1.3. The usage goals and categories of mobile applications development

Creating software environment where the most consumers can utilize applications for the longest period is the usage goal of mobile app development. This means that the mobile software should be of a high standard and serve a useful purpose for the majority of users worldwide. Applications should, in this regard, conform to the following requirements (De, 2018):

1. Conformity with the goals of the users.
2. Ability to connect with the bulk of users.
3. Ability to maintain security.
4. Being accessible and user friendly.
5. Supporting involvement and ongoing improvement.

According to (Islam, Mazumder, & Technology, 2010), mobile apps usage can be categorized into the following based on their application area:

1. Productivity: Spreadsheets, Word Processors, Notepads, Calculators, Diaries, and Calendars
2. Games: Card/Casino, Puzzle/Strategy, and Adventure or Action games
3. Communication: Social networking, email, and instant messaging clients
4. Multimedia, including viewers for graphics and images, presentations, video players, and audio players.
5. Utilities: File manager, Address book, Task manager, Idle screen/Screensaver, and Profile manager
6. Travel: Weather, Global Positioning System (GPS)/Maps, Translators, City Guides, Currency Converters, and Itineraries / Schedules.

1.4. Low code technology

Low code development is a technique for creating software applications without the use of traditional programming languages, instead using visual interfaces and drag-and-drop tools. This method makes it quick and simple for less-technical people to construct complicated apps. By automating many of the procedures that developers traditionally do manually, low code development technologies help make the software development process simpler (Chang & Ko, 2017).

1.4.1. Features of low code development.

Low code app development is a visual-based methodology that enables programmers to rapidly and simply design applications. Some characteristics of low code app development include the following(Sahay, Indamutsa, Di Ruscio, & Pierantonio, 2020):

1.4.1.1. Visual interface

Low code development tools have a visual interface that lets developers build apps by dragging and dropping various components.

1.4.1.2. Pre-built components

Tools for low-code development include pre-built elements like buttons, forms, and templates that may be altered to create individualized programs.

1.4.1.3. Ease of third-party integration and interoperability:

Low code development solutions interact with third-party services including databases, APIs, and web services, which are simple to access and utilize in apps.

1.4.1.4. Auto code generation

Low code development solutions automatically produce code, which cuts down on the time and labor needed to develop apps.

1.4.2. Advantages of low code

The Low code platform has its advantages and disadvantages (Bock, Frank, & Engineering, 2021) as highlighted thus.

1.4.2.1. Faster application development

Low code tools for app development make the software development process simpler and make it possible for less-technical individuals to rapidly and easily create complicated applications.

1.4.2.2. Reduced costs

By automating many of the manual processes that developers usually take, low code development solutions bring down the price of software development.

1.4.2.3 Improved collaboration

By enabling developers, designers, and business analysts to work together more efficiently, low code development technologies enhance communication and lead to better software development outcomes.

1.4.2.4High Scalability: Low code development solutions have a high degree of scalability, allowing systems to be quickly scaled up or down in response to customer demand.

1.4.3. Limitation of low code

1.4.3.1. Limited customization

Pre-built components in low-code development tools may restrict the options for customization, making it difficult to construct original applications.

1.4.3.2. Limited Control

Low code development systems automatically produce code, which may restrict the amount of control developers have over the code.

1.4.3.3. Limited Capabilities

The range of applications that can be created may be constrained by the fact that low code development tools may not have the same amount of capability as conventional programming languages.

1.4.3.4. Steep learning curve

Low code development tools have a learning curve that can be challenging to master and may take some time.

1.5. Free and open-source software (FOSS)

Software classified as Free and Open Source (FOSS) is freely available for download, use, and modification under Open-Source licenses. Due to its adaptability, affordability, and security, FOSS-based mobile app development is becoming more and more popular among developers in order to construct cutting-edge and long-lasting mobile applications (Crowstonet al., 2008).

1.5.1. Benefits of FOSS-based mobile app development

FOSS offers both individuals and organizations a number of advantages, which has helped FOSS gain popularity and establish itself as a competitive alternative to proprietary software. FOSS which is based on Community-centered approach (AlMarzouqet al., 2005) has subsequently grown to play a significant role in the open-source movement and software development as a whole.

1.5.1.1. Cost-Effective

Since FOSS-based mobile app development makes use of already-existing open source libraries, tools, and frameworks, it is affordable. This eliminates the need for pricey proprietary software, which can dramatically reduce development costs.

1.5.1.2. Flexibility

FOSS-based mobile app development enables creators to modify the program to meet their unique needs. Unlike with proprietary software, developers can access the source code and change it to fit their needs.

1.5.1.3. Security

Because the source code is open for inspection, FOSS-based mobile app development is thought to be more secure. This makes it less susceptible to cyberattacks because security flaws can be promptly found and addressed.

1.5.1.4. Community Support

A sizable community of developers works together to create, maintain, and upgrade the software in FOSS-based mobile app development. This community helps developers by offering them useful resources, support, and updates that make it simpler to create and maintain high-quality mobile applications.

1.5.2. Limitations of FOSS

1.5.2.1 Lack of professional support: Unlike the development of commercial software, FOSS-based mobile apps may not have access to professional support. To solve issues, developers might have to rely on online forums and other sources. The online forums may also not provide sufficient assistance to a novice due to the frustration of some experts who may be irritated at the "stupidity" of some novice questions.

1.5.2.2. Limited documentation

It may be difficult for developers to learn and use FOSS-based mobile app development due to the software's minimal documentation.

1.5.2.3. Integration problems

FOSS-based mobile app development may experience integration problems with proprietary software, which might reduce the functionality and usefulness of the app.

1.5.2.4. Compatibility concerns

The reach and usability of FOSS-based mobile app development may be constrained by compatibility concerns with specific platforms and devices.

2. Review of related works

See Table 1 for details

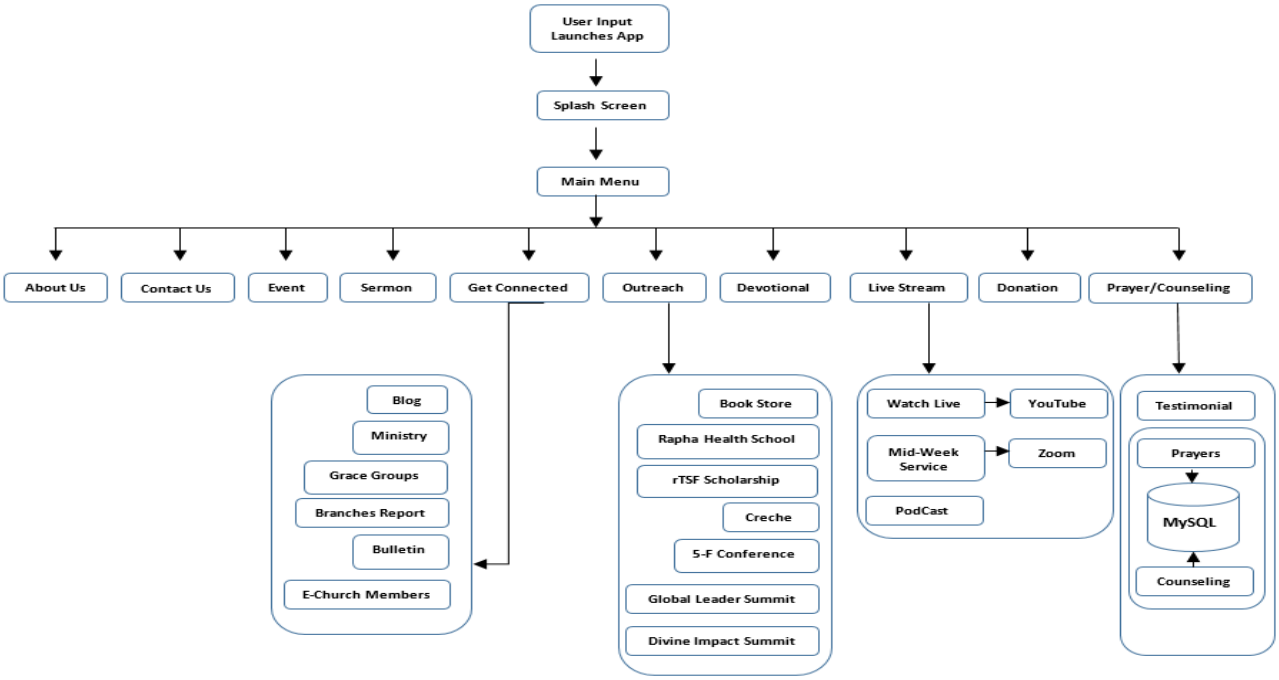

Figure 1: Framework Diagram

Table 1: Review of Related Work

Author/Year	Title	Technology Used	Outcome	Gaps	Solution
Anja Christin Kaiser Akugizibwe Emmanuel, Aligawesa Nathan F, Bamwite Benard (Kaiser, 2016)	Church Management Mobile Application Case Study: Ndejje Christian Union	FOSS used MySQL & PHP Android SDK Integrated Development Environment	Development of a Mobile Church management application that allows users and administrators monitor their contributions and track event attendance.	The application is not cross platformed for it operates on Android. It is based on a Native App development approach. The design and implementation were expensive.	Utilization of Low Code Application Development tools
Stephen J. McNeill (McNeill & Research, 2020)	COVID and Christ: Remote, Faith-Based Activities Using Religious Technology in the Catholic Church. Click to Pray App	FOSS based Native build mobile app. Swift used for Android Java used for iOS	The Multi languages app, was deigned to be user-friendly and available to a global audience. Along with a community prayer wall where users can post their intentions and pray for others, it has a variety of prayer materials. The app also gives users a method to communicate with other Click to Pray community members and get information on international prayer initiatives.	The mobile app does not use a single codebase. Functionality minimal without steady seamless internet connectivity.	Provision of offline usability with limited internet connectivity.
Albertus Alvin Adhitama Susanto (Susanto, 2020)	Android Application to Search for Church Locations in Yogyakarta	Android is a Linux-based (FOSS) operating system GPS Location Service and Google Maps serves as Low Code tools that can be utilized through Google Maps Application Programming Interface Firebase is a backend-as-a-service (BaaS) platform	Church Finder mobile app was designed to find the location of a church in Yogyakarta. This application utilizes the reliability of Firebase to store and manage church data with Google Maps support for maps. The presence of this application increased the effectiveness of locating and obtaining church information in Yogyakarta	Require significant technical expertise to develop, maintain, and update.	Leveraging on the utilisation of open-source libraries and frameworks to reduce development time and cost.

3. Methodology

This section provides a framework for creating cross-platform mobile applications utilizing MySQL, a free and open-source database management system, and the low-code development tool Alpha Anywhere which uses the X-Basic low code. A FOSS database and low-code development work together to build cross-platform mobile applications that are quick to develop and affordable.

3.1. Framework

The Framework used is the Software Structural Framework which offer a methodical process and instructions for planning the architecture and arranging the elements within the TSF Church project.

As shown in the Fig 1 below, the framework depicts how the user gain entrance to the app via the Splash Screen after the launching of the app. The user can click any of the individual pages through the Menu to access the options in the child level layers. This gives access to numerous processes via the pages in the child level layer. The pages are About Us, Contact Us etc. up till the Prayers and Counseling Processes. In this layer, there are further four processes that make up a module each, the pages are the Get Connected, Outreach, Live Stream and Prayers/Counseling modules. The framework shows the relationship of the pages and modules with each other. This framework provides a guide for the development of modular, scalable mobile application project. They specify how various components of the project are organized, related, and interacted with, facilitating effective creation and simple maintenance.

The framework's main purpose in this project work is to offer a clear structure and architecture that supports the TSF Church organization, reuse, and maintainability of the code. The framework also places a strong emphasis on the segregation of concerns, enabling the functionality of this mobile application project to be divided into separate parts or modules. This division makes it simpler to maintain, test, and modify specific components without having an impact on the entire app. External services like MySQL Server database, APIs like Zoom and YouTube, are considered to be incorporated into this mobile app project. The interconnectivity pattern is clearly portrayed in this Framework. The next page shows the Framework utilized for this mobile application development.

3.2. User flow diagram

The user flow diagram adopted in this project shows the graphic representation of the interactions and navigational steps a user takes while using this mobile application. The diagram provides the numerous displays, choices, and actions a user might make when utilizing the TSF Church mobile project. The Entry point is the Splash Screen or the entrance into the application. There are ten pages the user can immediately choose from depending on the task to be done. Note that four of the ten pages further leads to additional sub-pages with specific functions. The User flow diagram for the Mobile application is shown on the next page.

3.3. Iterative waterfall model

The Iterative Waterfall Model was adopted and used for this mobile application project. The reason for this adoption was because of the development flexibility which allows me to return and work on a module that was earlier worked upon. For instance, during the implementation, an unforeseen issue was discovered which necessitated a slight change in the original system design. This makes the Iterative Waterfall model a strong and viable option for the development of this mobile application project, where requirement needs to be adjusted with flexibility. Instead of waiting until the project is finished, it enables enhancements and adjustments to be made at each stage of the development process.

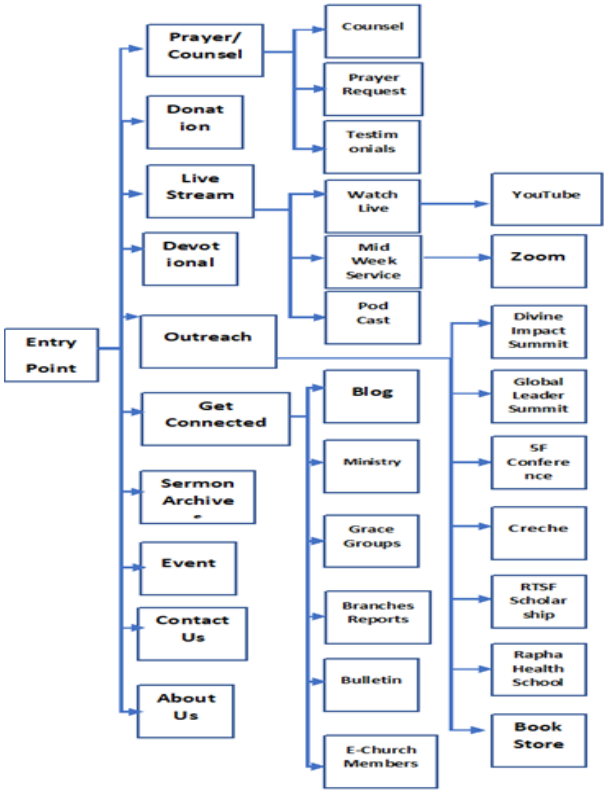

Figure 2: User Flow Diagram

3.4. Activity diagram

The Activity diagram used in this mobile app project shows the sequence in which the users interact with the mobile application right from pressing the button that will launch the app in order to activate the Splash Screen and pass through the Menu options to each given task and process modules. The direction of control and also data flow within the application is also depicted. Below is the Activity diagram of the mobile application project.

3.5. Database structure

The MySQL database Server is used in the project. It consists of the creation of a single database called Mobile_Project which has two tables named Prayer_Req and Counseling_Req. These are the table structures of the tables in the database (Table 2 and Table 3)

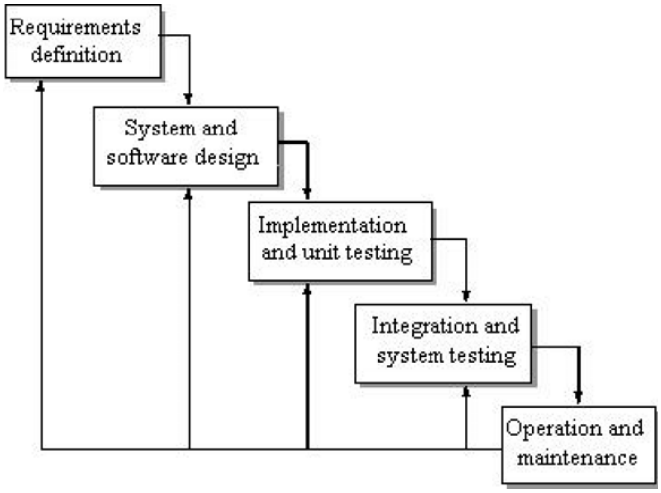

Figure 3: Iterative Waterfall Model

Table 2: Prayer_Req Table

Column Name	Data Type	Data Size
Id	Int (Primary Key)	3
FirstName	Varchar	50
Surname	Varchar	50
Gender	Varchar	6
Mobile_Number	Int	15
Prayer_Request	Varchar	250

Table 3: Counseling_Req Table

Column Name	Data Type	Data Size
Id	Int (Primary Key)	3
FirstName	Varchar	50
Surname	Varchar	50
Gender	Varchar	6
Mobile_Number	Int	15
Counseling_Request	Varchar	250

4. Implementation

This section discusses mobile application system implementation, testing and results. The system implementation and testing phase of this mobile application is the process of testing and deployment of the mobile application's functionality, performance, and usability before the application is released to end users, this step usually comes after the design in the development phase.

This system was implemented and tested on the iOS and Android platforms to evaluate its cross-platform compatibility and working performance. The process of implementation will be discussed here.

The procedural stages of implementation are outlined thus.

4.1. The Integrated Development Environment (IDE)

The development environment was set up which includes installation of the necessary software and frameworks such as the Alpha Anywhere Software and the MySQL database required for the mobile app development. The Alpha Anywhere integrated development environment (IDE), includes the software development kits (SDK), emulators used for virtual testing.

4.2. System evaluation

The mobile application as stated earlier was developed using the Alpha Anywhere Software which uses the X-Basic low code. Being a low coding system with efficiency and optimum performance with a lot of integrated auto debugging tools, the system was tested to be of high performance. The User Experience was assessed through the User Interface responsiveness and smooth transition of panels within the mobile application.

Due to its cross platform combabilities, the two range of platforms supported by the cross-platform framework was evaluated. The framework therefore supports the required features and APIs for both the iOS and Android platform. The system was designed as a Hybrid mobile application system that also accesses the Native Functionality of each platform. It has the native GUI interface which is Web enabled and it uses the ability of the cross-platform framework to access native device functionalities and APIs. The ease of accessing the device specific features was implemented.

The Alpha Anywhere version used is the Community Edition, this also applies to the MySQL which is FOSS based. Hence, the online community are available to provide help to technical issues by providing community

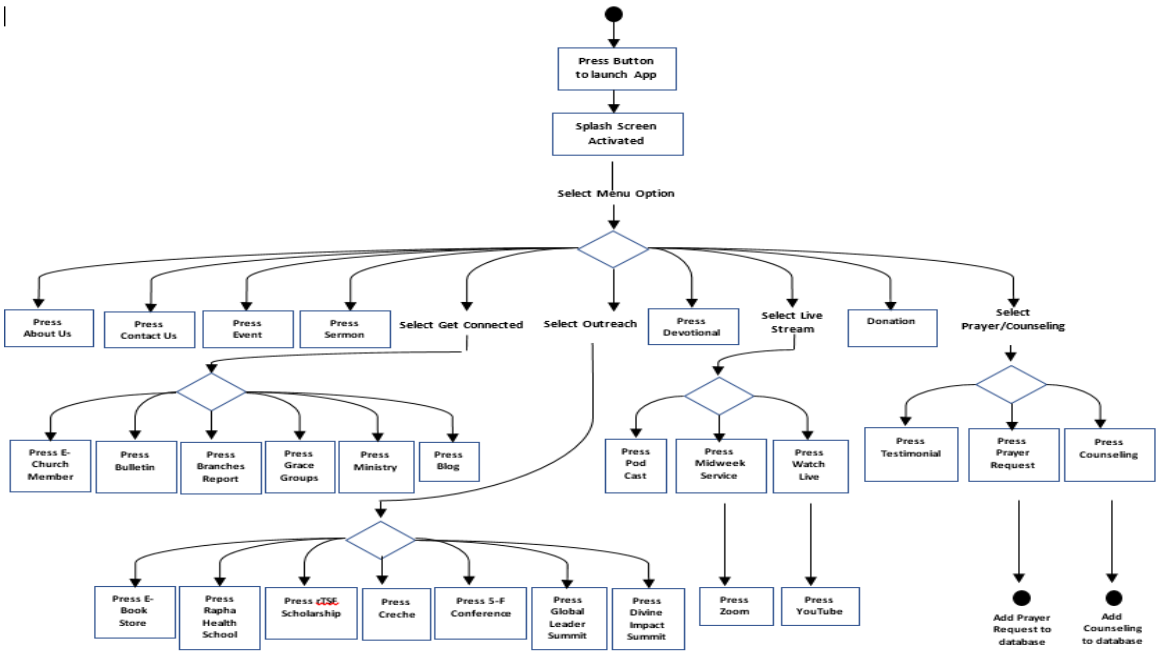

Figure 4: Activity Diagram

support and documentation. This assist in assessing the size and activity of the cross-platform framework's community. The availability of documentation, tutorials, forums, and community support was taken into consideration. The community also provides valuable resources and assistance during the development process. This enhances future maintenance and updates of the application.

4.3. System requirement

The following requirement should be met in order for the mobile application to run on the system. The requirements are in two phases namely:

4.3.1. Hardware Requirement

1. For iOS: The device should be an iPhone, iPad, or iPod touch. Most especially the iPhone 5s or later due to the test carried out on iPhone 5s.

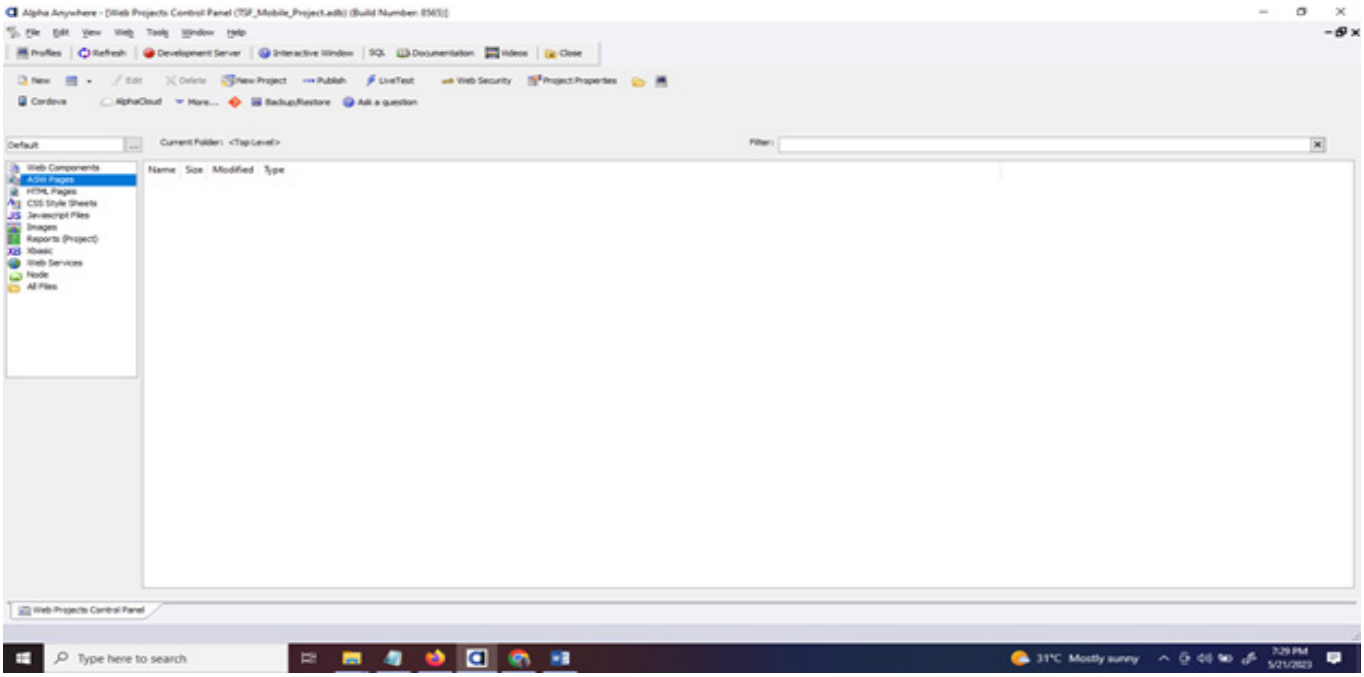

Figure 5: Alpha Anywhere IDE and Control Panel

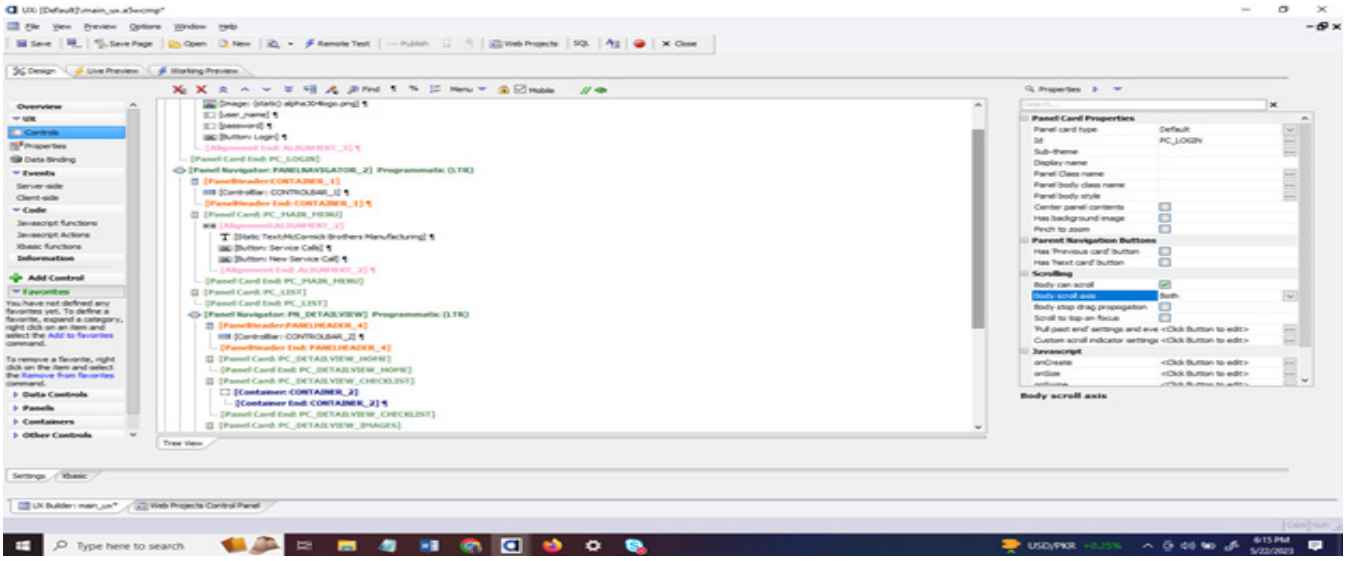

Figure 6: Alpha Anywhere UX Builder

2. Android: The application can work on Android based mobile phones such as Samsung Galaxy S5.
3. An Internet connection
4. A minimum of 1 GB of RAM
5. A minimum of 8 GB of storage space

4.3.2. Software Requirement

1. A compatible operating system such as iOS 12 or later or Android 11 or later
2. A web browser like Google Chrome, Firefox, Opera Mini etc.
3. Stable Internet connectivity.

4.4. Installation procedure

To eventually install the application, access will be needed for the apk. download on iOS Store, Google Playstore or Alpha Cloud. However, the project is in its Alpha Version hence it is not available for use by the general public for now.

5. Results and discussion

The following screen shots in Figure 5 shows the IDE consisting of the Project Control Panel of the Alpha Anywhere software after installation and Figure 6 shows the UX Builder of the Alpha Anywhere during the software of the mobile application during the development.

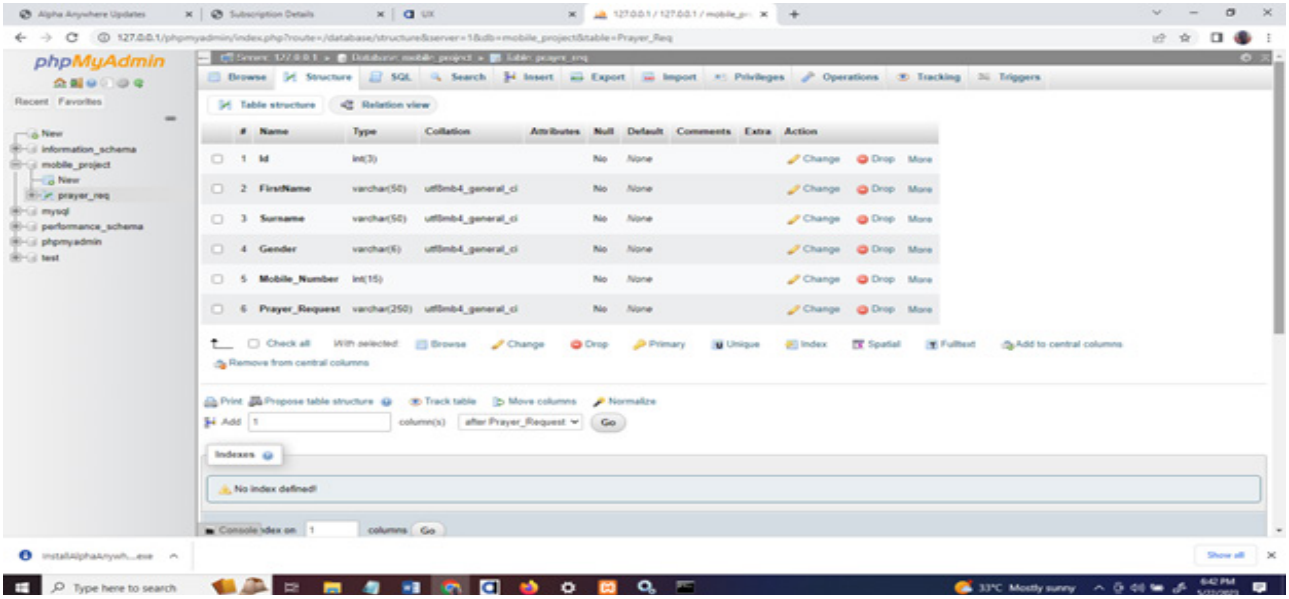

Figure 7: Prayer Req Table

Figure 5 shows FOSS (Free and OpenSource Software) with the IDE and Control panel which let developers write a single codebase that compiles to native binaries for iOS, Android, Windows, etc. The main advantage of this system is full control over the stack and because the code is open, you can fix bugs or add features yourself, which is valuable when a platform updates faster than a proprietary tool.

When compared with the work done by Susanto (2020) in which the performance was not really efficient, the present system cuts development time by 3050 % for standard apps and providse nearnative access to device hardware (camera, sensors) through pluginsThe implication of this new system is that it is deeply integrated with emerging native API and avoids vendor lockin rapid delivery of internal tools.

In Figure 6, the embedded UX builder lets designers iterate without writing code,interfaces can be built without deep programming skills and the drag and drop components allows prototype. The UX builder embedded in Low-code reduced technical barriers and development time compared to traditional methods, addressing gaps noted in Kaiser (2016) regarding cost and complexity. Kaiser highlighted high cost and complexity as major hurdles for traditional development, these problems have been solved in the present system by lowering the need for skilled programmers and offering draganddrop UI assembly, the UX builder directly tackles those costandcomplexity concerns, aligning with Kaiser's call for more affordable, streamlined solutions.The implications of these findings are designers become productive contributors and there is reduced risk of lock-in and Quality trade- offs.

5.1. Database

The MySQL Server is utilized for the database administration and implementation. The database name is Mobile_Project while it consists of two tables, the Prayer_Req table and the Counseling_Req Table. The Prayer_Req table handles all the Prayer Request from the mobile application interface while the Counseling_ Req table handles the counseling section. They are connected to the Mobile application interface through a Connection string. Below is the data structure for the two tables.

Figure 7 presents the database management interface of the developed system, implemented using MySQL and accessed through phpMyAdmin.The figure illustrates the structured organization of relational tables, user privileges, and database schemas that underpin the backend of the cross-platform mobile application developed using low-code and free and open-source technologies. The results demonstrate that the system adopts a modular and normalized database structure, which supports efficient data handling and ease of maintenance. The presence of clearly defined tables, indexed fields, and role-based access control (as shown in Figure 7) suggests that the system was designed with scalability and security considerations in mind. Unlike monolithic database designs, this architecture allows the system to accommodate future growth in user base and data volume without requiring extensive restructuring.
When compared with earlier implementations such as that reported by Kaiser (2016), which utilized MySQL primarily for basic data storage without explicit emphasis on scalability, the present system shows notable

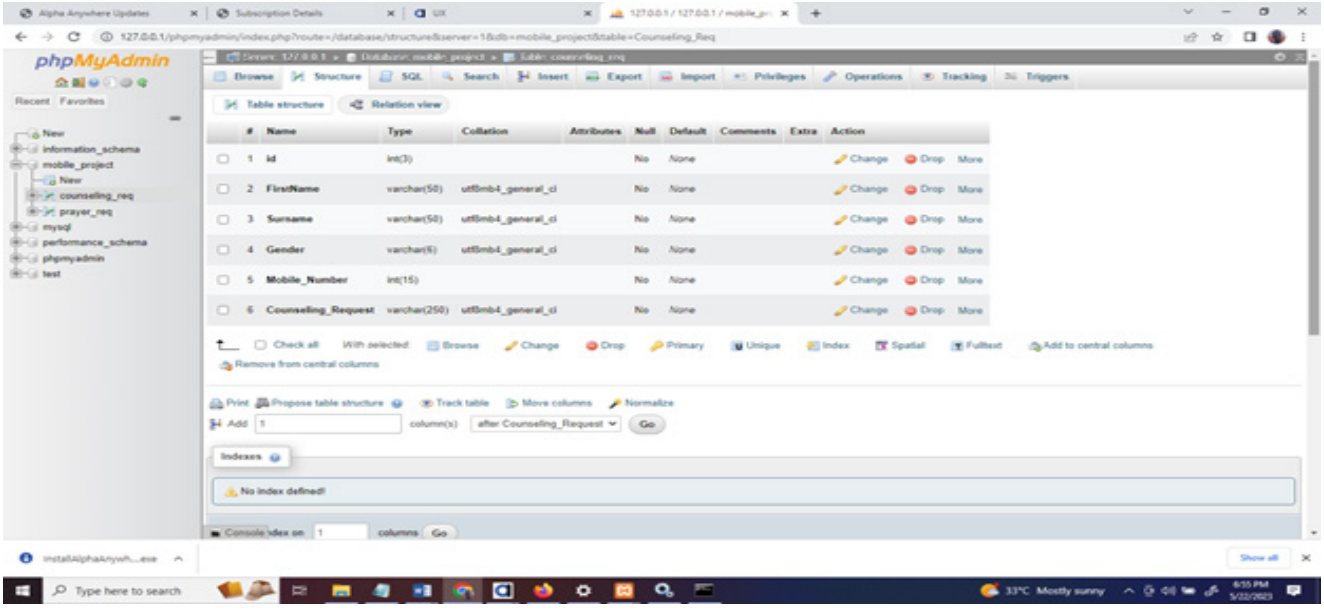

Figure 8: Counseling Req Table

improvements. Kaiser's approach focused largely on functionality, with limited discussion of database optimization, access control, or extensibility. In contrast, the current system integrates database normalization, structured schema design, and administrative control mechanisms, which are recognized in the literature as critical components for scalable and secure application development (Elmasri & Navathe, 2016; Silberschatz et al., 2020). The implications of these findings are significant for developers and institutions operating in resource-constrained environments. The demonstrated backend architecture suggests that scalable and secure mobile applications can be developed without reliance on expensive proprietary tools. This approach enhances accessibility, promotes sustainability in software development, and supports wider adoption of cross-platform mobile solutions in educational, governmental, and small-scale enterprise contexts.

Figure 8 above presents a more conventional approcah of coding with full control in which one can tweak every UI element, access the latest OS APIs, and architect the app exactly as one wants it to be.

When compared with McNeill (2020) that had the limitation of long-term maintainability, the present system is highly consistent because the generated code follows a standard template, making maintenance easier. The implication of this findings is that the present system is cost effective, very flexible and the system delivers comparable performance with far less overhead.

5.2. System testing

The system was tested on both the Android and iOS mobile operating systems to ensure its cross-platform compatibility. The resulted generated through the Emulator portrayed in the diagram below is the testing result of the Splash Screen on iPhone 5 which runs iOS platform and the Samsung Galaxy S5 which is Android based.

Figures 9,10 and 11 above show that the system can run cross platform (IOS and Android), this means one code base, many markets, it saves the stress of maintaining separate native codebases. The system can be iterated in real time, tested on multiple devices simultaneously. Compared with Kathuria et al. (2015) that operated on a single platform and had the problem of limited reach, the presented system has a wider reach to diverse users of different platforms.

The implication of the present system is that it involves performance trade-offs, while offering advantages such as improved speed of development, cost savings, and broader platform reach.

Figure 9: Horizontal View of Galaxy S5

Figure 10: Vertical View of iPhone 5 (Top view)

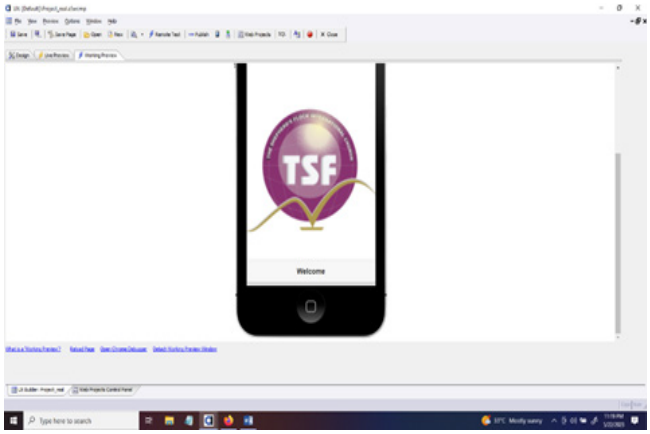

Figure 11: Vertical View of iPhone 5 (Bottom view)

5.3. Summary

In summary, developing a cross-platform mobile application using MySQL as FOSS and Alpha Anywhere as low code software is a practical solution for organizations of all sizes. Without being an expert in coding knowledge, creating unique mobile applications is simple with Alpha Anywhere with its Graphical User Interface. MySQL which is a free and open-source database has been proven to be reliable and scalable. Combining these two technologies will enable the development of a robust and reasonably priced mobile application that will aid organizations

especially churches in this context in connecting with their constituents and engaging the public.

Alpha Anywhere allows the development of the mobile application at once and the deployment across multiple platforms, such as iOS and Android. This ensures that the application can reach a wider audience and cater to users on different devices.

The low-code methodology used by Alpha Anywhere facilitates quicker development by offering a visual development environment and pre-built components. This lessens the requirement for intricate code and enables developers to concentrate more on the functionality and features of the program. Hence, with the flexibility and customizability possibilities offered by Alpha Anywhere, mobile application program can be modified to match the particular requirements of a religious setting. Developing features like event registration, prayer requests, and sermon streaming is made simpler by the ability to construct custom forms, workflows, and user interface elements. MySQL is a secured, popular and dependable open-source database system. MySQL server provides scalability, versatility, and strong data management capabilities. Data pertaining to the church management, such as member information, prayers and counseling, events, sermons, and donations, can be easily stored and retrieved using MySQL.

6. Conclusion

In conclusion, the MySQL and Alpha Anywhere software both provide solutions that are affordable. Due to the fact that Alpha Anywhere is low-code, less time and effort are required during development, potentially cutting development expenses. The same applies to MySQL which has a free and open-source database, there are no licensing fees, which makes it an affordable option for handling the application's data.

References

AlMarzouq, M., Zheng, L., Rong, G., & Grover, V. (2005) Open source: Concepts, benefits, and challenges. *Communications of the Association for Information Systems, 16(1), 37–64.*

Katz, A., Guness, U., & Gantchev, D. (2021) Open source for mobile apps. Geneva, Switzerland: World Intellectual Property Organization.

Bock, A. C., Frank, U. J.*, & Information Systems Engineering (2021) Low-code platform. Information Systems, 63, 733–740.*

Chang, Y. -H., & Ko, C. -B. (2017) A study on the design of low-code and no-code platform for mobile application development. *International Journal of Advanced Smart Convergence, 6(4), 50–55.*

Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2008) Free/Libre open-source software development: What we know and what we do not know. *ACM Computing Surveys, 44(2), Article 7, 1–35.*

Elmasri, R., & Navathe, S. B. (2016) Fundamentals of database systems (7th ed.). Boston, MA: Pearson Education.

Islam, R., Islam, R., & Mazumder, T. A. (2010) Mobile application and its global impact. *International Journal of Engineering & Technology, 10(6), 72–78.*

Kaiser, A. C. (2016) Church management mobile application (Undergraduate thesis). Department of Information Technology, Faculty of Basic Sciences.

Kathuria, A., & Gupta, A. (2015) Challenges in Android application development: A case study. *International Journal of Computer Science and Mobile Computing, 4(5), 294–299.*

Kuitunen, M. (2019) Cross-platform mobile application development with React Native (Master's thesis). University of Jyväskylä, Finland.

McGerty, L. (2021) Psion Organiser. Cambridge, UK: Centre for Computing History.

McNeill, S. J. (2020) COVID and Christ: Remote, faith-based activities using religious technology in the Catholic Church. *European Journal of Social Science Education and Research, 7(2), 127–137.*

Open-Source Initiative. (2007) The Open-Source Definition. https://opensource.org/osd

Perens, B. (1999) The open-source definition. In C. DiBona, S. Ockman, & M. Stone (Eds.), Open sources: Voices from the open-source revolution (pp. 171–188). Sebastopol, CA: O'Reilly Media.

Phongtraychack, A., & Darya, D. (2018) Evolution of mobile applications. *MATEC Web of Conferences, 145, 01012.*

Rajput, M. (2021) Tracing the history and evolution of mobile apps. *Journal article.*

Richardson, C., Rymer, J. R., Mines, C., Cullen, A., & Whittaker, D. (2014) New development platforms emerge for customer-facing applications. Cambridge, MA: Forrester Research.

Rouse, M. (2020) Mobile application. Newton, MA: TechTarget.

Sahay, A., Indamutsa, A., Di Ruscio, D., & Pierantonio, A. (2020) Supporting the understanding and comparison of low-code development platforms. *In Proceedings of the 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp.* 200–208). IEEE.

Sanchis, R., & Poler, R. (2019) Enterprise resilience assessment—A quantitative approach. *Sustainability, 11(16), 4327.*

Sanchis, R., García-Perales, Ó., Fraile, F., & Poler, R. (2019) Low-code as enabler of digital transformation in manufacturing industry. *Applied Sciences, 10(1), 12.*

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020) Database system concepts (7th ed.). New York, NY: McGraw-Hill Education.

Stallman, R. (2022) What is free software? Free Software Foundation.https://www.gnu.org/philosophy/free-sw.en.html

Susanto, A. A. A. (2020) Android application to search for church locations in Yogyakarta (Undergraduate thesis). Universitas Atma Jaya Yogyakarta, Indonesia.

Syeed, A. B., Bhat, S. H., & Kaur, D. (2021) Study of mobile app development industry. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 7(6), 154–170.*

Vaidya, R. (2023) Importance of mobile application in everyday life. *International Journal of Advanced Research in Computer Science, 14(2), 34–42.*