DESIGN AND IMPLEMENTATION OF AN ANDROID-BASED MOBILE APPLICATION FOR COURSE LEARNING MATERIALS FOR THE DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF BENIN.

BY

OJADUA BLESSING EBRUBA

PSC1808910

BACHELOR OF COMPUTER SCIENCE(B.Sc) PROGRAMME

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCES

UNIVERSITY OF BENIN

BENIN CITY

EDO STATE, NIGERIA.

SEPTEMBER, 2023

CERTIFICATION

This is to certify that this project work was carried out by **OJADUA BLESSING EBRUBA** with Matriculation Number **PSC1808910** under my supervision and it is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.Sc) Degree in Computer Science from the University of Benin.

_____ _____

Prof. V.V.N. Akwukwuma **DATE**

Project Supervisor

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc) Degree in Computer Science from the University of Benin.

_____ _____

PROF. MRS A.O EGWALI **DATE**

Head Of Department

DEDICATION

I dedicate this project work to God Almighty, who gave us the wisdom, knowledge, and understanding to carry out and complete this work and also for his protection throughout my time in the University of Benin.

I also dedicate this project work to my parents for making this journey as easy as possible and for their guidance, prayers and encouragement.

This work is also dedicated to my project supervisor for the proper guidance and role played in making this work a success

ACKNOWLEDGEMENT

My utmost appreciation goes to God, for seeing me through all my years in school.

To my project supervisor, Professor V.V.N. Akwukwuma. For your patience, and guidance throughout the period of supervision, I am grateful.

Special gratitude goes to the Head of the Department of Computer Science, Prof. Mrs. A.O Egwali, and my project coordinator Dr. Mrs. R.A. Usiobaifo. Also, I want to show my profound appreciation to my highly esteemed lecturers in the Department of Computer Science, for their dedicated tutoring and mentoring. You have all worked hard to set me on the proper road in my professional pursuit and to implant in me sound knowledge about important parts of life. God bless you all.

Immeasurable thanks to my parents, for their constant unwavering support, care, and guidance during the course of this project work and my stay in school. Also, thanks to my siblings and friends who have offered their support and listening ears in my times of weakness. May God, in his infinite mercy, bless you all.

TABLE OF CONTENTS

CERTIFICATION i

APPROVAL ii

DEDICATION iii

ACKNOWLEDGEMENT iv

TABLE OF CONTENTS v

ABSTRACT viii

CHAPTER ONE 1

INTRODUCTION 1

1.1 BACKGROUND OF STUDY 1

1.2 STATEMENT OF PROBLEM 3

1.3 AIMS AND OBJECTIVES OF THE STUDY 4

1.4 PURPOSE OF THE STUDY 4

1.5 SIGNIFICANCE OF THE STUDY 4

1.6 SCOPE OF THE STUDY 5

1.7 LIMITATIONS OF THE STUDY 5

CHAPTER TWO 6

LITERATURE REVIEW 6

2.1 INTRODUCTION 6

2.2 EDUCATIONAL APPS IN LEARNING 7

2.3 IMPACT OF EDUCATIONAL APPS IN LEARNING 7

2.4 ANDROID 7

2.5 THE SOFTWARE TOOLS 11

2.5.1 Dart Programming Language 11

2.5.2 Usage 12

2.5.3 Flutter 13

2.5.4 Why Flutter? 14
2.5.5 Framework architecture 14
2.5.6 Dart platform 15
2.5.7 Flutter engine 15
2.5.8 Foundation library 16
2.5.9 Design-specific widgets 16
2.5.10 IDE Support 16
2.6 BACKEND: FIREBASE 16
2.7 DATABASE MANAGEMENT: CLOUD FIRESTORE 17
2.8 VISUAL STUDIO CODE 17
2.9 ANDROID STUDIO 18
2.14 REVIEW OF RELATED WORKS 19
CHAPTER THREE 25
DESIGN & METHODOLOGY 25
3.0 INTRODUCTION 25
3.2 SOURCES OF DATA 26
3.3 REQUIREMENTS SPECIFICATION 26
3.3.1 Evaluation 26
3.3.2 Development 27
3.4 METHODS 27
3.4.1 Analysis of Existing System 27
3.4.2 Advantages of The Existing System 28
3.4.3 Limitations of The Existing System 28
3.4.4 Analysis of The Proposed System 28
3.4.5 Advantages of The Proposed System 29
3.4.6 Limitations of The Proposed System 30
3. 5 SYSTEM DESIGN 30

3.5.1 Input Design 31
3.5.2 Output Design 32
3.5.3 Database Design 33
3.5.4 Flowchart Design 35
3.5.5 Use Case Diagram 37
CHAPTER FOUR 39
RESULTS AND DISCUSSIONS 39
4.0 INTRODUCTION 39
4.1 PROPOSED SYSTEM FEATURES 39
4.2 HARDWARE AND SOFTWARE REQUIREMENTS 39
4.3 USER GUIDE 40
Sign Up Page 42
CHAPTER FIVE 53
SUMMARY, CONCLUSION AND RECOMMENDATIONS 53
5.0 INTRODUCTION 53
5.1 SUMMARY 53
5.2 CONCLUSION 54
5.3 RECOMMENDATIONS 54
5.4 FURTHER STUDIES 54
REFERENCES 56
APPENDIX 60

ABSTRACT

In recent years, due to the rapidly increasing number of features available for smart mobile devices, mobile applications are playing an important role in higher education systems. A rapid growth in the volume of available learning materials has made quick access to relevant course materials tasking for students. Therefore, it has become inherently necessary to create a system that would adequately address the myriad of issues associated with the ease of access to educational materials as well as the availability of said materials quickly and efficiently. This aim of this project is the design and development of a course material management application on the Android Platform, using Flutter (a cross platform software framework running on Dart Programming Language), to give each student access to relevant and beneficial course materials, as well as all prerequisite resources quickly and efficiently; thereby eradicating stress, limiting cost, and generally enhancing the academic process.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF STUDY

In recent years, the field of education has undergone a transformative shift towards digital learning solutions. Mobile applications, particularly those designed for Android devices, have played a significant role in this transformation. The integration of mobile apps into educational settings has opened up new avenues for delivering course learning materials, enhancing interactivity, and improving overall learning experiences (Ertmer & Ottenbreit-Leftwich, 2013).

Innovative digital solutions that accommodate various learning styles and preferences have replaced traditional classroom arrangements that were dependent on printed textbooks and generalized educational strategies. Mobile applications created for Android devices have emerged as standout options among these remedies for facilitating education (Johnson, Adams, Cummins, & Estrada, 2012).

The integration of mobile apps into educational settings represents a paradigm shift in how students access and engage with course learning materials. Android-based mobile apps offer several advantages, including flexibility, accessibility, and interactivity (Tondeur, van Braak, Ertmer, & Ottenbreit-Leftwich, 2017). These apps enable learners to carry a wealth of educational resources in their pockets, empowering them to explore and access study content at their own pace and convenience.

Ambrose et al. (2010) defines learning as a process that results in

change as a result of experience, increasing the possibility of improved performance and

future learning.

With a well-designed Android app, that student can access course materials, watch instructional videos, or participate in interactive quizzes, transforming otherwise unproductive commuting time into valuable learning experiences (Sharples et al., 2016). Before the invention of computers and computer-related learning, only the traditional classroom norm was used for instruction. During traditional classroom-based instruction, the lecturer delivers the course lectures in a certain location and at a specific time. As a result, it places significant time and space restrictions on both the student and the teacher. Additionally, on the part of the lecturer, because of human factors, the lecturer may not always be able to put the optimum effort towards preparing and delivering course models. Direct interaction might also not always be effective due to the large number of students needing attention while students coming from far distances might not be able to meet up with timing. The conventional method required that all students learn the same information at the same time, and it mandated that failing was the only option for those who did not pick up the subject quickly enough. This approach, which had been imported from Europe, dominated American education until the end of the 19th century, when the education reform movement

imported progressive education techniques from Europe (Dewey, 2008).

Today's culture has accepted the idea of mobile cellphones and applications, which are primarily used by youths and young adults. Since mobile phones have become a regular part of our lives and offer a variety of features including internet navigation, entertainment, taking notes, taking photos, and recording videos, nearly every young adult has access to a mobile smartphone.

Mobile apps have the potential to cater to various learning methods, ensuring that visual, auditory, and kinesthetic learners can all find ways to engage with the content (Joo, Lim, & Kim, 2018). They can provide instant feedback on assessments, offer personalized learning pathways, and form collaboration through discussion forums, promoting a more immersive and effective learning experience (Al-Rahmi, Aldraiweesh, Yahaya, & Alamri, 2019).

The context of this study enables students to study anywhere and anytime. Hence, the proposal to develop an android-based mobile application to aid students in their studies.

The aim of education, in fullest and deepest sense of the word, is to foster natural human psychological growth and development for personal maturity of moral consciousness or conscience in conjunction with the unfolding of human potential in conditions of personal freedom as inner freedom (Toyama, Murphy & Baki, 2013).

1.2 STATEMENT OF PROBLEM

Using the Department of Computer Science, University of Benin as a case study, the current way of obtaining course material requires students to physically make an indication and as a result, buy or create copies of the course materials after lecturers pass it through the class representative. Students can also obtain said materials through social media platforms they use to connect with one another, such as WhatsApp. This approach can cause several problems to arise, such as inefficiency in proper management of course materials. Furthermore, the stress associated with acquiring such content via social media platforms is rather concerning given that students may need to go through multiple chat sessions in order to discover the needed material. Thus, a platform that efficiently provide and catalogue the relevant course materials is essential. By creating a well-designed mobile

app, this project aims to improve the accessibility and effectiveness of course materials, ultimately enhancing the learning experience.

1.3 AIMS AND OBJECTIVES OF THE STUDY

The aim of this study is to design and implement an android-based mobile application in which the course materials for Computer Science department of the University of Benin are catalogued and presented using the advantages of multimedia and hypermedia.

The specific objectives are:

i. To study the existing method of accessing study materials by students in the department of Computer Science, University of Benin.
ii. To highlight the problems with the existing system.
iii. To design and implement an android-based mobile application in which the course materials for the Computer Science Department of the University of Benin are catalogued.

1.4 PURPOSE OF THE STUDY

i. To provide students with the required learning materials and more for the Computer Science department.
ii. To enable students gain access to learning materials at their convenience.
iii. To enable students learn anywhere at any time.

1.5 SIGNIFICANCE OF THE STUDY

The significance of the study is to create a comfortable medium of learning for students. Thus, by using the resources made available on the platform, lecturers and students will be more productive in both academic and non-academic areas. The project will create a stress-free environment which will make students psychologically prepared to learn better because

there will be little to no pressure when learning. It also complies with the advancement of technology brought forth by computerization and its many advantages.

The outcome of this study will be useful for the university..

1.6 SCOPE OF THE STUDY

This project's scope encompasses the design, development, and evaluation of the Android-based mobile app for course learning materials. It includes considerations for content delivery, interactivity and accessibility. However, it is important to note that this project does not cover in-depth pedagogical research, nor does it address the iOS platform.

1.7 LIMITATIONS OF THE STUDY

Given the project’s nature, significant time and financial resources are required. As a result, the specifications of the proposed Android learning material application were adjusted to fit the available financial budget, which resulted in the project design being limited to only course materials for Computer Science students.

CHAPTER TWO
LITERATURE REVIEW

2.1 INTRODUCTION

Picciano (2014) described two broad areas of tutor support academic; "The first is academic (or tutorial) support, which deals with supporting students with the cognitive, intellectual and knowledge issues of specific courses or set of courses. This will include: developing general learning skills, numeracy and literacy". The second is non-academic or counselling support, the support of students in the effective and organizational aspect of their studies. Cope (2016), however provides an excellent summary stating, "a learning system need to have knowledge and a broad conceptual understanding of their field."

Gone are the days of chunky textbooks and snail-paced learning management systems (LMS). Smartphones are now at the center of our lives, and learning is no exception. The use of mobile devices and mobile apps to facilitate learning among individuals introduced a concept of E-learning called m-learning. Mobile learning also referred to as m-learning, is a way of accessing learning content through mobile devices. This method empowers learning at the point of need, enabling users to access content whenever and wherever suits them. The most important element of mobile learning is its focus on the mobility of the learner- giving them the ability to choose when and where they want to access learning means they can go at their own pace, increasing engagement and improving knowledge retention (Brew, 2017).

The rapid progress of information technologies in our age has increased the interest of technology towards people's needs. While technological devices and usage of these were subject to a specific environment or location in the past, environment and location now have their independent specialties in the past recent time. O'malley et al (2003) defined mobile learning as

a type of learning whose learner is determined previously, is not in a specific location, or benefits the opportunities offered by mobile technologies.

2.2 EDUCATIONAL APPS IN LEARNING

Educational apps have emerged as powerful tools in modern education, revolutionizing the way students access and interact with educational content (Laurillard, 2012). Unlike traditional learning materials, educational apps offer dynamic and interactive features that can significantly enhance the learning process.

2.3 IMPACT OF EDUCATIONAL APPS IN LEARNING

i. **Flexibility and Convenience**: Educational apps provide learners with the flexibility to access educational content at their convenience, facilitating self-paced learning (Kukulska-Hulme & Traxler, 2013).

ii. **Data-Driven Insights**: Many educational apps collect and analyze data on learner interactions, enabling educators to make informed decisions about instructional strategies (Picciano, 2017).

2.4 ANDROID

With the advent of new mobile technologies, the mobile application industry is advancing at a rapid state. Consisting of several operating systems like Symbian OS, iOS, etc. The Android operating system, however is recognized as the most widely used, popular and user-friendly mobile platforms. Due to its ability to be customized; this open-source Linux-kernel-based operating system has great flexibility making it a leading mobile operating system (Sarkar et al, 2019). The Android operating system gained popularity among developers for its customizable nature. It is very efficient for building an application in one platform and deploy it to several platforms simultaneously.

Android began in 2003 as a project of the American technology company Android Inc., to develop an operating system for digital cameras. In 2004, the project changed to become an operating system for smartphones. The owning company was later procured by Google Inc., in 2005 and further developments were based on the Linux Open-Source Operating System for personal computers. The first cellular telephone to feature the new operating system was the T-Mobile G1, released on October 22, 2008. In 2012, Android became the most popular operating system for mobile devices, surpassing Apple IOS and Symbian OS, and, as of 2020, about 84.1 percent of mobile devices run Android (*Operating Systems - Statistics & Facts*, 2022).

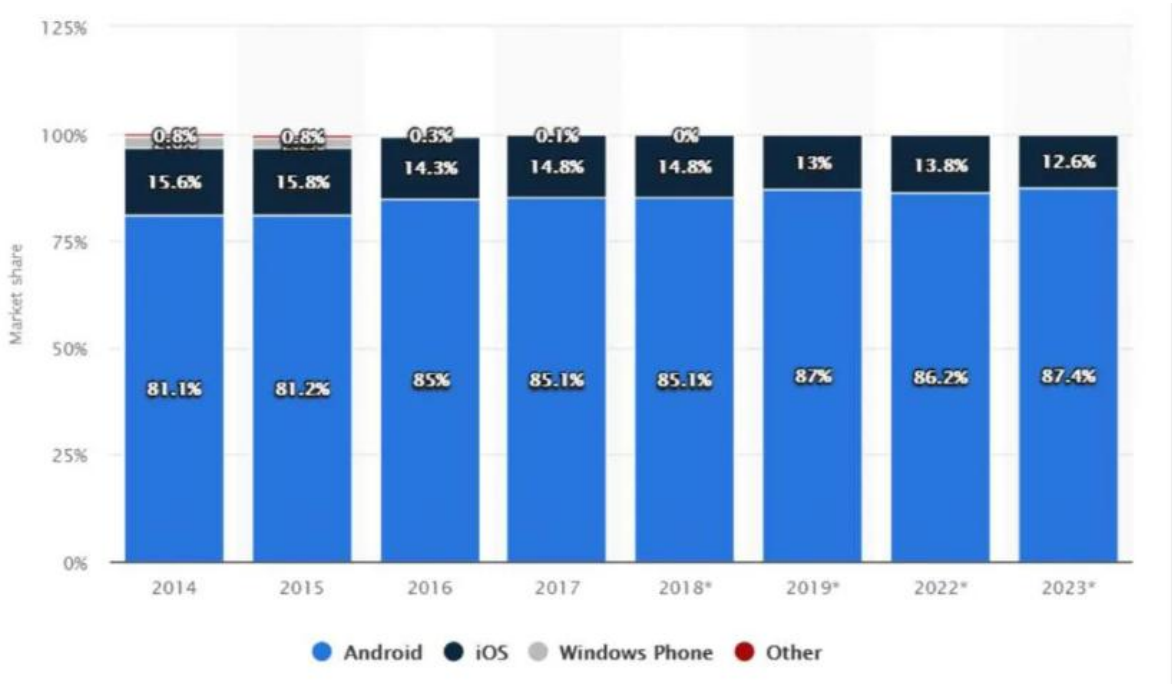

Figure 2.1 A graphical representation of the statistical analysis of the Android operating system usage against others

An operating system is a set of programs that control the execution of application programs and acts as an intermediary between a user and a computer and the computer hardware. It is a

software that manages the computer as well as providing an environment for application programs to run in. The Dalvik VM, a Java virtual machine, is used by the Android operating system to run Java apps that were created in the Java programming language. Google has updated and improved Dalvik in order to better fit the hardware requirements of mobile devices. A utility called *.dx* is included in the smart phone software development kit that converts Java files that have been compiled using a standard Java compiler into the *.dex* formats, joining all of the Java class files and removing any extraneous data (Haris et al., 2017). The android operating system since its inception, has evolved to meet new demands of the users as demands keep changing and technology keeps advancing. The table below shows the evolution of the android versions since its inception till date:

Table1: Evolution of the android operating system

Name	**Version number(s)**	**Initial stable release date**
No official codename	1.0 1.1	September 23, 2008 February 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Éclair	2.0 - 2.1	October 26, 2009
Froyo	2.2 - 2.2.3	May 20,2010
Gingerbread	2.3 - 2.3.7	December 6, 2010
Honeycomb	3.0 – 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011
Jelly Bean	4.1 – 4.3.1	July 9, 2012

Kitkat	4.4 – 4.4.4	October 31, 2013
Lollipop	5.0 – 5.1.1	November 12, 2014
Marshmallow	6.0 – 6.0.1	October 5, 2015
Nougat	7.0 – 7.1.2	August 22, 2016
Oreo	8.0 – 8.1	August 21, 2017
Pie	9	August 6, 2018
Android 10	10	September 3, 2019
Android 11	11	September 8, 2020

According to statistics, the number of people who own mobile devices has expanded over time, with the majority of individuals now owning several devices. The highest age group for mobile device users is between the ages of 18 and 29, which is also the typical age for college students (Jay, 2023).

In 2021, the number of mobile devices operating worldwide stood at almost 15 billion, up from just over 14 billion in the previous year. The number of mobile devices is expected to reach 18.22 billion by 2025, an increase of 4.2 billion devices compared to 2020 levels (Statista, 2022).

By the end of 2023, according to Cisco's Annual Internet Report (2018-2023), smartphones will have had the second-fastest growth. Mobile subscriptions will expand at a 2% yearly growth rate, as shown in Figure 2. This implies that by 2023, more than 70% of the world's population will possess a mobile phone (Cisco Annual Internet Report, 2020).

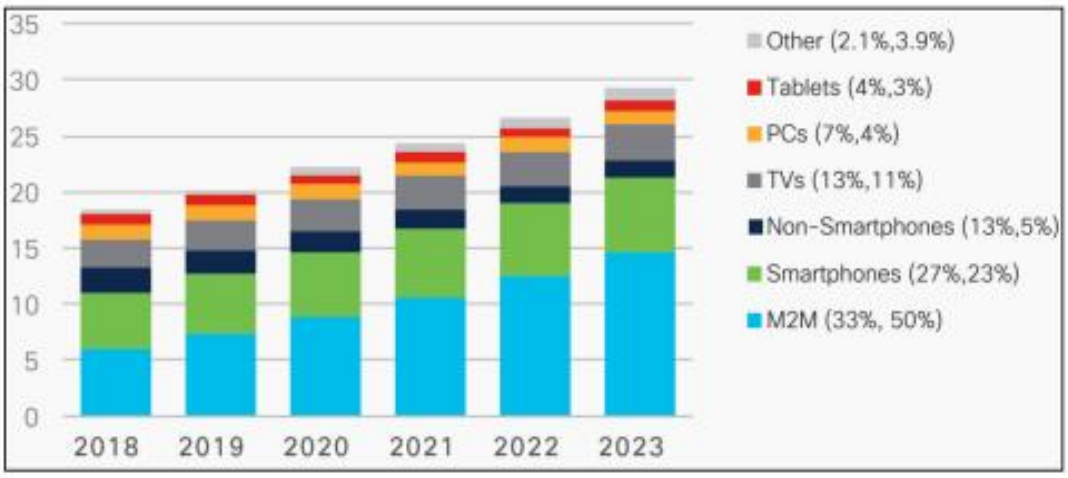

Figure 2.2: Global device and connection growth (Source: Cisco Annual Report, 2018-2023)

2.5 THE SOFTWARE TOOLS

This section aims to provide a brief overview and information about the software to be used in the development of the mobile application such as the programming language, the development environment, the text editors and backend servers.

2.5.1 Dart Programming Language

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is a client-optimized language for developing fast apps on any platform. With a configurable execution runtime platform for app frameworks, it aims to provide the most productive programming language for cross-platform development.

It is an object-oriented, class-based, garbage-collected language with C-style syntax. It can compile to either native code or JavaScript, and supports interfaces, mix-ins, abstract classes, reified generics and type inference.

Dart was unveiled at the GOTO conference in Aarhus, Denmark, October 10–12, 2011. The project was founded by Lars Bak and Kasper Lund. Dart 1.0 was released on November 14, 2013 (Walrath & Ladd, 2014).

Dart initially had a mixed reception and the Dart initiative has been criticized by some for fragmenting the web, due to the original plans to include a Dart VM in Chrome. Those plans were dropped in 2015 with the 1.9 release of Dart to focus instead on compiling Dart to JavaScript. Dart 2.0 was released in August 2018, with language changes including a sound type system.

Dart 2.6 introduced a new extension, dart2native, which extends native compilation to the Linux, macOS, and Windows desktop platforms. Earlier developers could create new tools using only Android or iOS devices. With this extension it also becomes possible to compose a program into self-contained executables. According to company representatives, it's no longer necessary to have the Dart SDK installed, as the self-contained executables can now start running in a few seconds. The new extension is also integrated with the **Flutter** toolkit, making it possible to use the compiler on small services (for example, backend support) (Krill, 2019).

2.5.2 Usage

Dart code can be run in four different ways:

i. **Web:** To run in mainstream web browsers, Dart relies on a source-to-source compiler to JavaScript. According to the project site, Dart was "designed to be easy to write development tools for, well-suited to modern app development, and capable of high-performance implementations." In a web browser, the code is precompiled into JavaScript using the dart2js compiler, making it compatible with all major browsers with no need for browsers to adopt it. By optimizing the compiled JavaScript output

to avoid expensive checks and operations, code written in Dart can, in some cases, run faster than equivalent code handwritten in JavaScript idioms.

ii. **Stand-alone:** The Dart software development kit (SDK) ships with a stand-alone Dart VM, allowing Dart code to run in a command-line interface environment. As the language tools included in the SDK are written mostly in Dart, the Dart VM is a critical part of it. These tools include the dart2js compiler and a package manager called pub. Dart ships with a complete standard library allowing users to write fully working system apps, such as custom web servers.

iii. **Ahead-of-time compiled:** Dart code can be AOT-compiled into machine code (native instruction sets). Apps built with Flutter, a mobile app SDK built with Dart, are deployed to app stores as AOT-compiled Dart code.

iv. **Native:** Dart 2.6 includes the dart2native compiler to compile to self-contained, native executable code. Before Dart 2.6, this feature exposed this capability only on iOS and Android mobile devices via Flutter.

2.5.3 Flutter

For the development of native mobile applications, Google offers the free and open-source Flutter UI framework. Flutter, which was introduced in 2017, enables programmers to create mobile applications using a single codebase and programming language. Building apps for both iOS and Android is now easier and quicker.

A software development kit (SDK) and their widget-based UI library make up the Flutter framework. The reusable UI components in this package include sliders, buttons, and text inputs.

Using the Flutter framework, developers will create mobile applications using the Dart programming language. Dart is a typed object programming language that focuses on front-end development and has a syntax similar to JavaScript.

2.5.4 Why Flutter?

i. Increased productivity as a result of the shared codebase used by iOS and Android, which saves time and resources.
ii. Flutter is simple to understand since it enables developers to create native mobile applications without having to use a lot of code or access OEM widgets.
iii. An appealing user interface simplifies the production of mobile apps.
iv. Excellent performance because it meets all requirements just like native apps would.
v. Cost-effective since creating apps for both iOS and Android using the same codebase essentially results in the creation of two apps for the price of one.
vi. Accessible on several IDEs as developers can choose between VSCode and Android Studio to edit their code.
vii. Excellent documentation and community support.

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia and the web from a single codebase.

2.5.5 Framework architecture

The main components of Flutter are as follows:

i. Dart platform
ii. Flutter engine (Skia Graphics Engine)
iii. Foundation library

iv. Design-specific widgets

v. Flutter Development Tools (DevTools)

2.5.6 Dart platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

While writing and debugging an application, Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. This allows for fast compilation times as well as "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this further with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

For better performance, release versions of Flutter apps on all platforms use ahead-of-time (AOT) compilation.

2.5.7 Flutter engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS. The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

2.5.8 Foundation library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

2.5.9 Design-specific widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and Cupertino widgets implement Apple's iOS Human interface guidelines.

2.5.10 IDE Support

Flutter maintains official support for the following IDEs and editors via plugins:

i. IntelliJ IDEA
ii. Android Studio
iii. Visual Studio Code
iv. Emacs

Other IDEs can be used with community-supported plugins, or by using Flutter tools from the command line.

2.6 BACKEND: FIREBASE

Firebase is a set of hosting services for any type of application (Android, IOS, JavaScript, Node.js, Java, Unity, php, C++…). It offers NoSQL and real time hosting of database content, social authentication (Google, Facebook, Twitter, and GitHub), and notifications, or services, such as real-time communication server. Firebase evolved from Envolve, a prior startup founded by James Templin and Andrew Lee in 2011. Firebase's first product was the firebase Realtime Database, an API that synchronizes application data across IOS, Android, and Web devices, and stores it on a Firebase's cloud. The product assists software developers in building real-time collaborative applications. In 2014, Firebase launched two products: Firebase Hosting and

Firebase authentication. This positioned the company as a mobile backend as a service (AMT, 2021).

The app will utilize firebase for the backend services.

2.7 DATABASE MANAGEMENT: CLOUD FIRESTORE

Cloud Firestore is a NoSQL document database that simplifies storing, syncing, and querying data for apps on a global scale. It is a great way to put the backend of a flutter app without the hassle of having a server. Cloud Firestore enables complex querying structures when compared to Realtime databases. Because our data is structured when stored (as documents), we can perform more cumbersome or impossible queries compared to Realtime Database.

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google cloud. Like firebase Realtime database, it keeps your data in sync across client apps through real time listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or internet connectivity. Cloud Firestore also offers seamless integration with other firebase and google Cloud products, including Cloud functions (Goyal, 2020).

2.8 VISUAL STUDIO CODE

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter.

On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web. Microsoft has released most of Visual Studio Code's source code on GitHub under the permissive MIT License, while the releases by Microsoft are proprietary freeware (Kumar, 2022).

2.9 ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0 (Akhtar & Sahani, 2016).

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++ (Patel, 2022).

2.10 REVIEW OF RELATED WORKS

S/N	AUTHORS & YEAR	ARTICLE TITLE	METHODOLOGY USED	RESULTS OBTAINED	RESEARCH GAP
1	Kocakoyun & Bicen (2017)	Development and Evaluation of Educational Android Application	Qualitative Research	It aimed to develop a mobile lesson application for undergraduate students.	The application did not provide proper explanations of the sharing structure of the application and did not communicate how video sharing using the platform would be achieved.
2	Abildinova et al (2016)	Developing a Mobile Application "Educational Process Remote Management	Qualitative Research	A mobile application based on the android operating system with the capability	The application uses a local language which is not very common and so cannot be particularly useful to a very wide

		System" on the Android Operating System		to assist teachers and students in the learning and organizational process was developed.	range of students, it also lacks features like additional authorization portals and a news board for communicating with students.
3	Lu'mu (2017)	Learning Media of Applications Design Based Android Mobile Smartphone	Qualitative Research	A Learning media mobile application developed using E-learning Moodle site and able to run on the android operating system was developed.	The software however tested and confirmed in areas of validity and practicability is limited to only the android operating system..
4	Okosun & Inyang	Android Based	Qualitative Research	In this paper, a course material	The application is limited to only the

	(2021)	Course Material Management Application Using University of Benin as a Case Study.		management application that made it possible for students to have easy access to their course materials at their fingertips.	android operating system and for subsequent work, can be made available to other platform outside the Android OS.
5	Doaa, Elsaeed & Abeer (2021)	Designing an Educational Android application to Improve Learning Quality and Students' attitudes towards it.	Qualitative Research	A mobile app aimed at helping students in studies relating to Boolean logic and logic gates was developed.	Lack of in-app features to communicate between tutors and students in any form. The application also lacks file sharing capabilities can be enabled using draw-over sharing services like Bluetooth and is

					limited to the android operating system only.
6	Kilmova (2019)	Impact of Mobile Learning on Students' Achievement Results	Qualitative Research	Its aim is to help develop a mobile learning platform that can serve as an appropriate complementary method to other forms of course delivery.	Lack of extensive/adequate subject samples and also, inability to function on other operating systems.
7	Pedro, Barbosa & Santos (2018)	A critical review of mobile learning integration in formal educational contexts	Qualitative Research	The paper aimed to review and present a critical analysis of a reality that is pretty much in	Further research should investigate the interference of technology in the classroom (i.e on how multitasking and orchestration affects the

				existence in every education context nowadays related with the use of mobile devices in the classroom.	students' and teachers' roles).
8	Qureshi, Khan, Gillani and Raza (2020)	A systematic Review of Past Decade of Mobile Learning: What we Learned and Where to go.	Qualitative Research	The research work reviews the concept of mobile learning and its inculcation into the educational system so far and possible improvements for future purposes. It aims to educate	Should work about application importance and value in mobile learning that will be helpful for both students and lecturers

				using social media medium and learn new information.	
9	Hartley, Andujar & (2022)	Smartphones and Learning: An Extension of M-Learning or a Distinct Area of Inquiry	Qualitative Research	The research aims to better contextualize smartphone and learning research in the broader ICT research agenda.	The research lacks detailed characteristics of work to support and identify categories.
10	Mamolo (2022)	Students' evaluation and learning experience on the utilization of digital interactive math comics (DIMaC)	Qualitative Research	The research work's aim is to ascertain the student's evaluation and learning experience based on use of the developed interactive	The study covers a limited field of application using a small sample study and hence, cannot be stated as very efficient.

		mobile app.		mobile application in mathematics.	

CHAPTER THREE
DESIGN & METHODOLOGY

3.0 INTRODUCTION

The purpose of this chapter is to examine the current system for providing students with access to course learning materials and to explore ways to improve access and overall student learning ability.

Methodology is the main plan, rationale and justification of a research project. In order to create a strategy that meets the objectives, it entails analyzing the approaches currently being utilized in the field as well as the ideas or guiding concepts in order to build an approach that support them.

The methods utilized for data gathering and analysis are discussed and explained in research methodology. This chapter discusses what was done and how it was done, enabling readers to assess the validity and dependability of our research.

In this chapter, the goals and issues with the current system are identified, and a proper analysis of the information gathered is done. Additionally, this study's design aims to describe the technique used to ensure the project's success.

The successful integration of the hardware and software is crucial to the project's success. Significant techniques must be used to build and develop a solution that is capable of meeting the undertaking requirements in order to improve ease of access to course learning resources.

3.2 SOURCES OF DATA

The source of our primary data for this study are textbooks, pdf documents, journals and information gotten from surfing the internet.

3.3 REQUIREMENTS SPECIFICATION

A mobile platform where students can conveniently access required course materials. In addition, which also enables them to evaluate and improve their learning strategies.

3.3.1 Evaluation

Students will have convenient and simple access to course learning materials in the form of PDFs and other learning formats owing to our solution's integration with a mobile platform, specifically an Android smartphone. They will be given the chance to take practice tests and

evaluate their performances in order to analyze what they have learned. They have sufficient access to previous exams past questions and papers for the courses they are studying.

In summary, the platform gives students the chance to take lessons, evaluate their progress, and consider their results.

3.3.2 Development

This procedure entails designing, outlining, planning, and developing the solution. In order to ensure ease of use for students and maintain their engagement, an attractive, friendly, accessible, and easy to use interface is a necessity.

The frontend must be connected during development to the database, which holds all the data supplied and received by the app, and the backend (server-side), which is in charge of storing and organizing data and ensuring that everything on the client-side (frontend) actually works. The frontend, backend, and database must all be correctly integrated and functioning for the solution to be successful.

In this project's development, the software is the core. The frontend, backend, and database designs, as well as the documentation are managed by the software. For the application development, we use Google's cross platform flutter software development framework, which runs on Dart. The app will consume a REST API from a Firebase backend server and the API will expose data from a Cloud Firestore database.

3.4 METHODS

3.4.1 Analysis of Existing System

It is crucial for students to adequately prepare for their courses by studying and practicing using the course curricula and resources supplied by the course instructors if they want to ace them.

Students can currently obtain course materials by making copies of them or by downloading them from social media sites like WhatsApp, which they use to connect with one another. This system has a variety of drawbacks, including ineffective course material management. Furthermore, the stress associated with acquiring such content via social media platforms is rather concerning, given that students may need to go through multiple chat sessions in order to discover the needed paper. Some students lack access to certain study resources since they overlooked it among other things.

3.4.2 Advantages of The Existing System

Students are able to share resources across many platforms because of the current system. Additionally, students could have access to as many materials as possible.

3.4.3 Limitations of The Existing System

There are a number of limitations and challenges with the method that digital course materials are currently accessed, including:

1. Inefficient organization and management of the course materials.
2. Finding the appropriate course learning resources requires a lot of effort and tension while you search through chats.
3. Duplicates of course learning materials on students' device.
4. Loss of access to educational resources; for instance, after 21 days, you can no longer download on WhatsApp. It might also be the result of a lost or damaged gadget.

3.4.4 Analysis of The Proposed System

The proposed system is an android-based mobile platform for smartphones, designed to collate the courses and course content offered in the Department of Computer Science, University of Benin, in order to eliminate various issues associated with the department’s currently adopted

conventional method of tutoring such as student-lecturer ratio, student to facility ratio, difficulty in access to materials and a high population of students present in classes amongst others.

The system will include catalogued course materials from all courses offered by students admitted into the Department of Computer Science that can be viewed in pdf formats, as well as hyperlinks to relevant external course materials, such as instructional videos on YouTube or other relevant sites. The student will also have access to the most current course curriculum, which can be updated by an admin in the event of a change, and past questions, which are compiled in a different category under each course section and can also be updated by an admin. The admin will be able to communicate important information such as assignments, changes in class times and whatever relevant information via a message box embedded in the platform, perform update and delete functions and as well, participate as an end user.

For the application development, we use Google's cross platform flutter software development framework, which runs on Dart. The app will consume a REST API from a Firebase backend server and the API will expose data from a Cloud Firestore database.

The proposed system, however, is not intended to eradicate the current working system but rather to enhance it.

3.4.5 Advantages of The Proposed System

1. Self-paced learning modules allow students to work at their own pace.
2. Less stress on the student from the rigorous work of carrying numerous textbooks to class each day as such books will now be accessible from one mobile device.

3. The users over time get accustomed to using mobile learning applications.
4. Access to catalogued past questions from previous years.
5. This system would also benefit lecturers who have E-books because purchase referral links could be added to the specific course section.

3.4.6 Limitations of The Proposed System

1. Without the routine structures of a traditional class, student may get lost or confused about course activities and its deadline.
2. Students with low motivation or bad learning habits may slack in learning.
3. Using the mobile app system for teaching certain technical courses like compiler construction might prove difficult without the combination of a physical tutor for explanations.
4. The mobile application is limited to only students admitted in the Department of Computer Science, University of Benin.
5. The mobile app is only limited to smartphones running on the android operating system/android users.

3. 5 SYSTEM DESIGN

A system is made up of several interacting parts that behave in a predetermined way to become a coherent whole.

A system's limits, structure, and purpose, as well as how it functions, all depend on and are influenced by its surroundings.

A functioning system has the following qualities: Integration, Organization, Interaction, Interdependence, and Central Objective.

System design is the process of organizing a system by defining its modules or components to meet the necessary requirements. It involves defining the architecture, interfaces, and data for a system that satisfies particular requirements.

The requirements for the system are as follows:

i. Input Design
ii. Output Design
iii. Database Design
iv. Flowchart Design

3.5.1 Input Design

The process of translating user-generated inputs to a computer-based representation is known as input design.

The input design allows users (administrators and students) to communicate with the system.

A system's input design is crucial since every piece of data entered determines the effectiveness of the system.

This can be accomplished by putting in place a user interface that is accessible, easy to understand and user-friendly.

The input design aims to make data entry simple and confirms that data are accurate and error-free.

With the proposed solution, users can enter information into the proposed system's forms. The user-provided data are considered and verified by the input design. The system database also benefits from input data.

Considerations for input design include:

1. What kind of data should be provided as input?

2. How to organize or categorize the data.

3. The informational dialog to assist users in entering data.

4. Procedures for creating input validations and what to do in the event of an error.

The system would perform the following input operations:

i. Admin Login

ii. Student Registration

iii. Student Login

3.5.2 Output Design

The information that a system or process generates from a certain input is known as the output.

The output design shows the results of system users' actions.

The user can understand, navigate, and carry out activities across the system with ease due to the output design, which is the component of the system that is visible to the user.

The output design is crucial, because system outputs constitute the basic foundation of the system and the output obtained from the system is used to evaluate the effectiveness and accuracy of the system.

The objectives of the output design:

1. Fulfil the intended purpose by giving users the right information.
2. Providing the proper output volume.

3. Making sure the output is provided where it is needed.
4. Delivering the output result on schedule as needed.
5. Selecting the appropriate output format or method.

The output design includes:

3.5.2.1 **LOGIN PAGE:** A login page or entry page requests user identification and authentication, which is often done by providing a username and password combination. To give administrators and students access to the system, this page shows their login options.

3.5.2.2 **PROFILE PAGE**: This page contains details about the logged in user. It displays details as names, matriculation number, and levels.

3.5.2.3 **DASHBOARD PAGE:** A dashboard is a visual representation of the most crucial data required to accomplish one or more goals, gathered and arranged on a single screen for easy monitoring.

After users log in to the system, they arrive at the dashboard page. Users can navigate to any other pages on the system from this dashboard page.

3.5.3 Database Design

Data storage, organization, and proper relationship maintenance are all facilitated by database design.

This system makes use of MySQL as its database management system.

Table 3.1: Admin Login Page

FIELD NAME	FIELD TYPE (FIELD SIZE)
Admin Username	VarChar (25)
Password	VarChar (25)

Table 3.2: Student Registration Page

FIELD NAME	FIELD TYPE (FIELD SIZE)
First Name	VarChar (30)
Last Name	VarChar (30)
Matriculation Number	VarChar (20)
Gender	VarChar (15)
Email Address	VarChar (30)
Password	VarChar (25)
Level	VarChar (25)
Programme	VarChar (25)

Table 3.3: Student Login Page

FIELD NAME	FIELD TYPE (FIELD SIZE)
Username	VarChar (20)
Password	VarChar (25)

3.5.4 Flowchart Design

Flowchart is a diagrammatic description of an algorithm or a step-by-step process for solving a problem. It is a diagram that depicts a workflow or process. The flowchart displays the stages as a series of boxes of varying sizes, with arrows joining the boxes in the correct order.

A flowchart is used to develop and record complicated processes and programs. They aid in visualizing what is happening, assisting the audience in comprehending the process and possibly identifying its shortcomings.

System flowcharts are a means to show how decisions are made to regulate events and how data flows in a system. The two flowcharts below depict activities of admin and user of the proposed system.

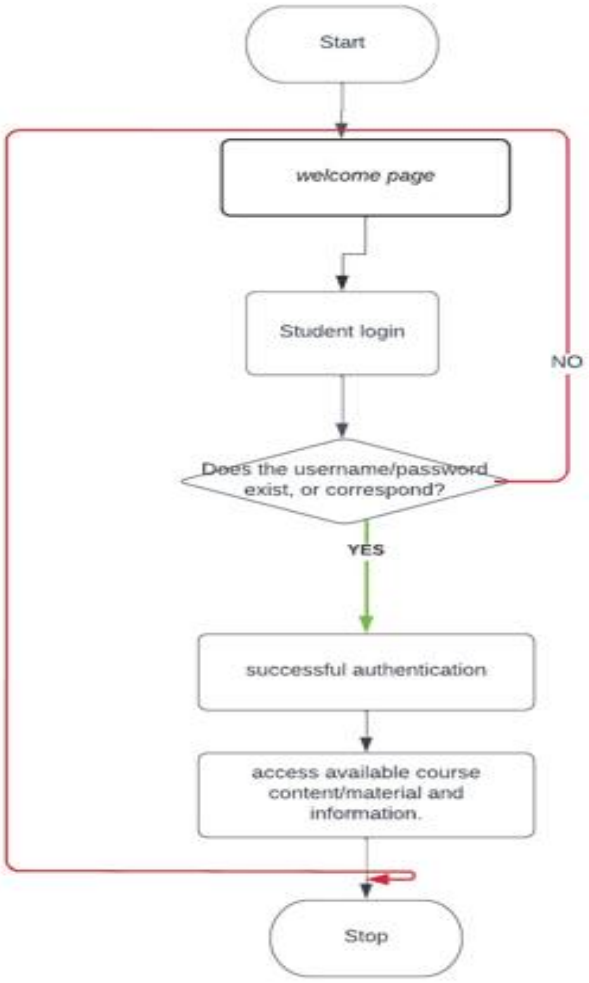

Figure 3.1. Flowchart depicting user activities in the system

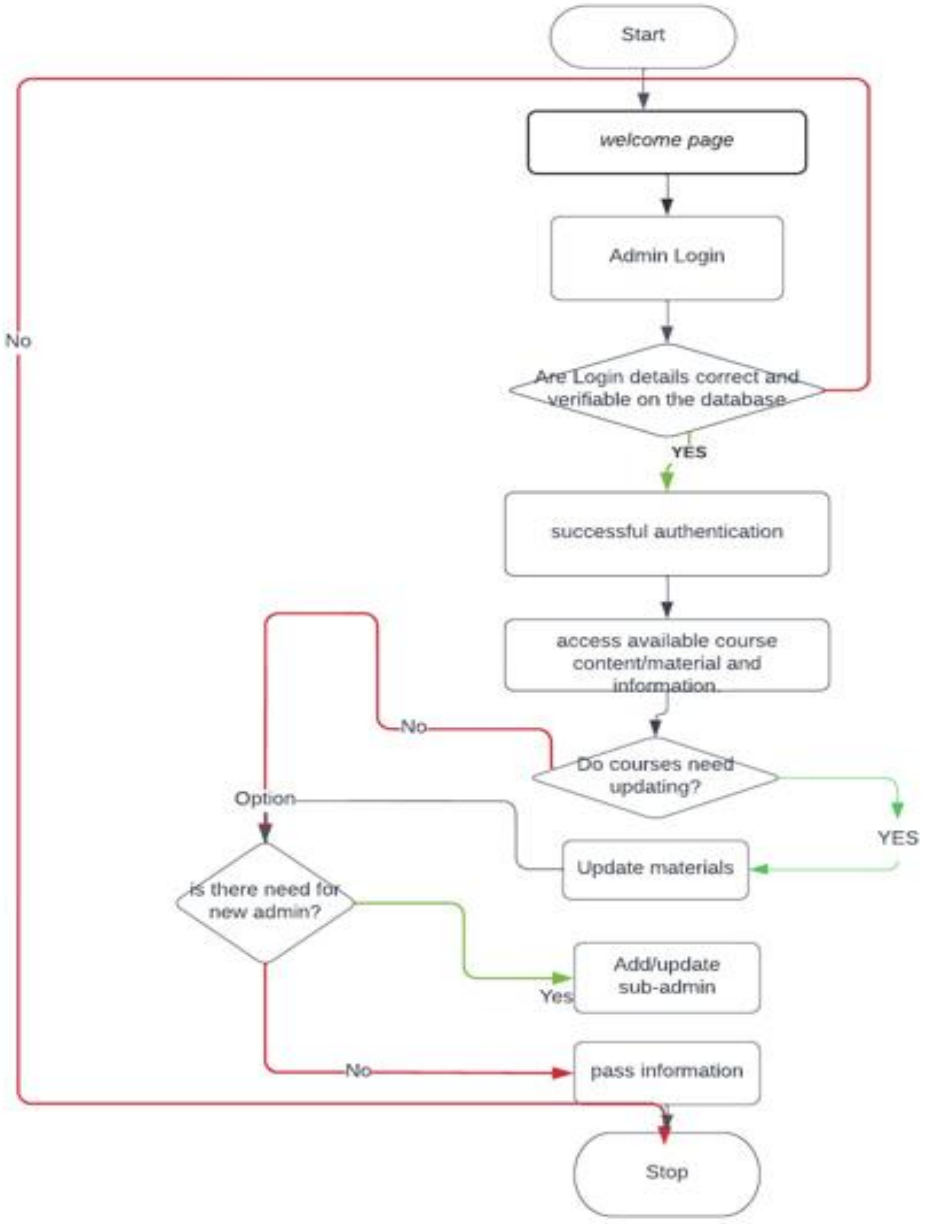

Figure 3.2 Flowchart demonstrating administrator activities in the system

3.5.5 Use Case Diagram

A use case is an activity that the system carries out, usually in response to a request by a user of the system. Think of a use case as a situation in which the system must accomplish some goals of the user. An actor may be a person, a company or organization, a computer program, a computer

hardware, software or both. A use case is another tool for capturing the functional requirements of the system. They define a goal-oriented set of interactions between external actors (parties outside of the system that interact with the system) and the system (*Use Cases,* 2020), as illustrated in figure 3 below:

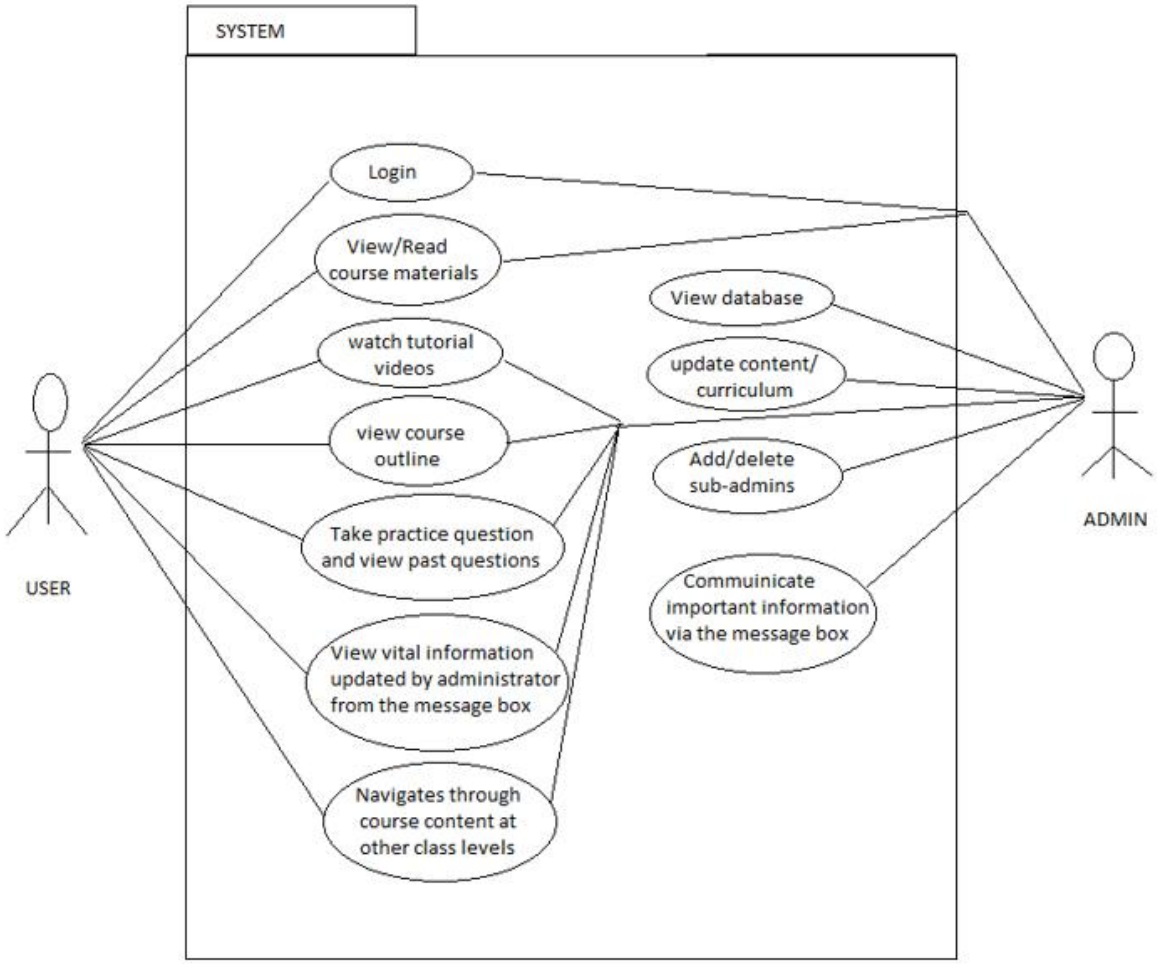

Figure 3.3: A use case diagram showing admin and user interaction with the system.

CHAPTER FOUR
RESULTS AND DISCUSSIONS

4.0 INTRODUCTION

This chapter discusses the procedures for the implementation of the android system developed in the study. This entails the hardware and software requirements and user guide.

4.1 PROPOSED SYSTEM FEATURES

The proposed system is made up of the following components: A homepage, sign in page, sign up page, a landing page, a course section, a notice page, a settings page which gives access to the profile page, the help and support and the privacy policy pages. It also contains a backend database run on firebase which is accessible to the admin through the admin dashboard.

4.2 HARDWARE AND SOFTWARE REQUIREMENTS

All hardware devices used in the development of this project must meet all relevant criteria to achieve the best outcomes.

Since this project is centered on Android, an Android smartphone is essential for testing, as well as a laptop or equivalent for running all codes, the research process, and documentation. Table 4.1 shows the hardware and software specifications.

Table 4.1: Hardware and Software Requirements

	ANDROID SMARTPHONE	**LAPTOP OR EQUIVALENT**
Operating System	Android	Windows, Mac or Linux
RAM (Random Access Memory	4GB or more	8GB or more

	ANDROID SMARTPHONE	LAPTOP OR EQUIVALENT
Processor	Qualcomm Snapdragon (version 630 and higher) or equivalent	3.6GHz or more
Available Storage	Minimum of 8GB	Minimum of 32GB
Resolution	1440 x 2960 or equivalent	1920-by-1080
Display Size	14.732 cm (5.8”) or more	33.02 cm (13") or more
GPU (Graphics Processing Unit)	Mali-G52 MC2 or equivalent	UHD Graphics 620 or 630 or equivalent
OS Version	Android 9.0 minimum	Windows 10/ MacOS 10.14/Linux

4.3 USER GUIDE

The system provides a form that allows users to make inputs during the point of registration/signing up to create user profiles through the use of the created/designed interfaces, the admin also serves as a user.

The system validates the username and password entered by the user against what was created and saved in the backend database during the sign-up, each time the user attempts to login to the system.

If the details come back as correct, the user will gain access to the system and will be able to navigate the system. However, this app does not allow the users to do more than view the course materials, access the past questions and navigate other app features i.e., the users cannot make

changes to the dynamics of the application but can make changes to their profiles through the profile section provided. The users after gaining access, are able to:

- Select program
- Select level
- Select Semester
- View/select/access course material.
- View/select/access past question.
- View/select/access curriculum
- Access the notice board.
- Access the settings page.
- View and/or contact help and support
- View and edit their user profiles through the profile page.
- Upload a profile picture if necessary.

The admin, however, can access the system as a normal user and also, as an administrator. The admin is required to login by providing the login details after which the system checks the details provided against the previously created admin details. If the details are correct, the admin then gains authorized access to the system and is able to make changes to the system from the backend. These changes can be made using the admin dashboard which gives the admin the ability to:

- Update courses
- Delete courses
- View and manage user profiles and activity

- Update the help and support
- Upload information in the notice section
- Add/delete sub-admins

Sign Up Page

The user is required to fill the sign up form by filling in the details in the required fields. This step is a prerequisite to gaining access on using the application.

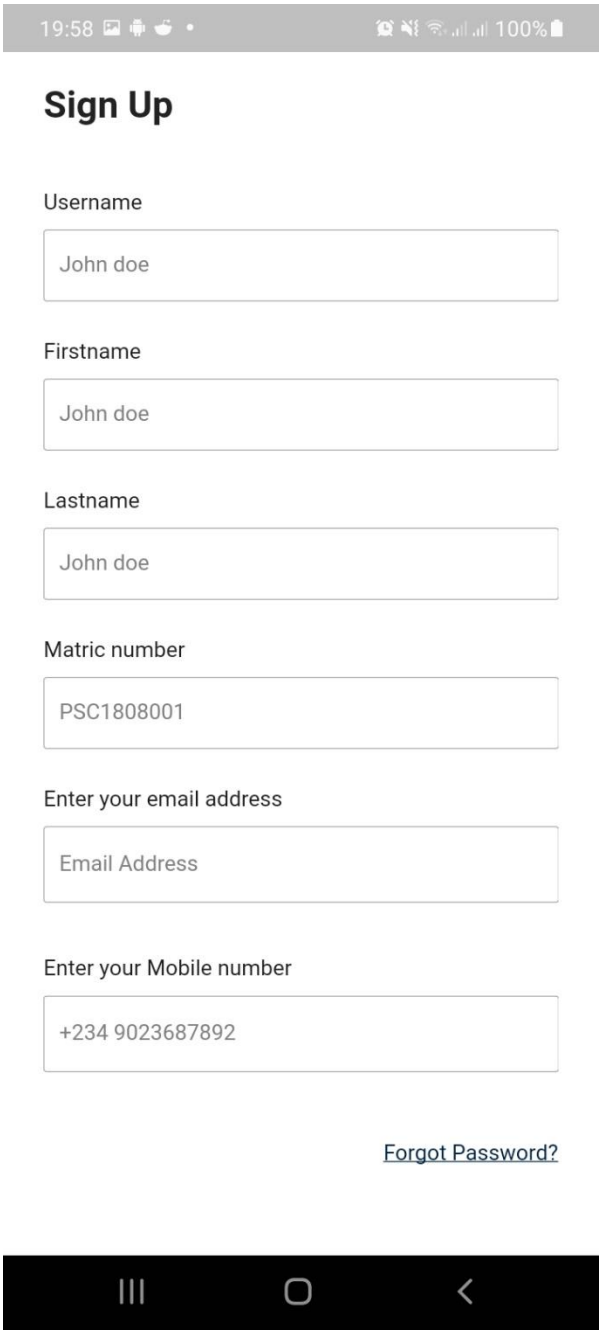

Fig 4.1: Sign up page

Login Page

The user must have registered on the sign up form to be able to login on the app.

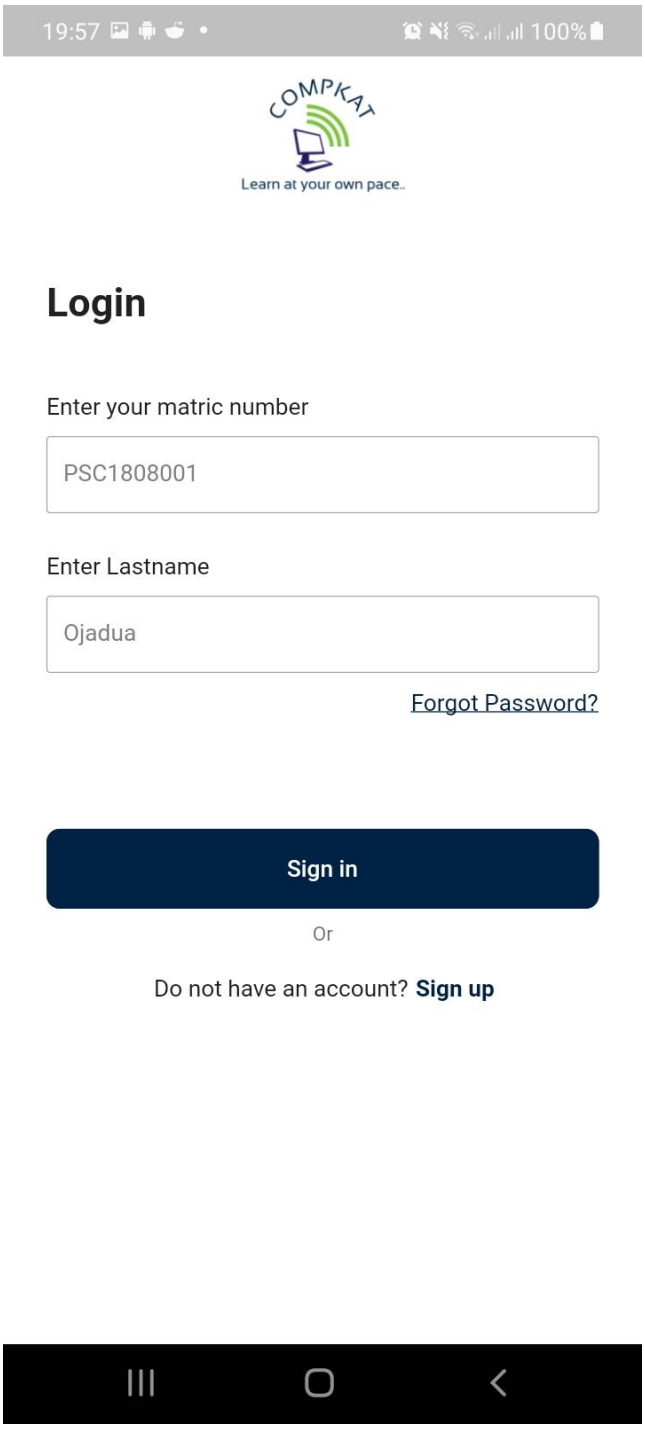

Fig 4.2: Login Page

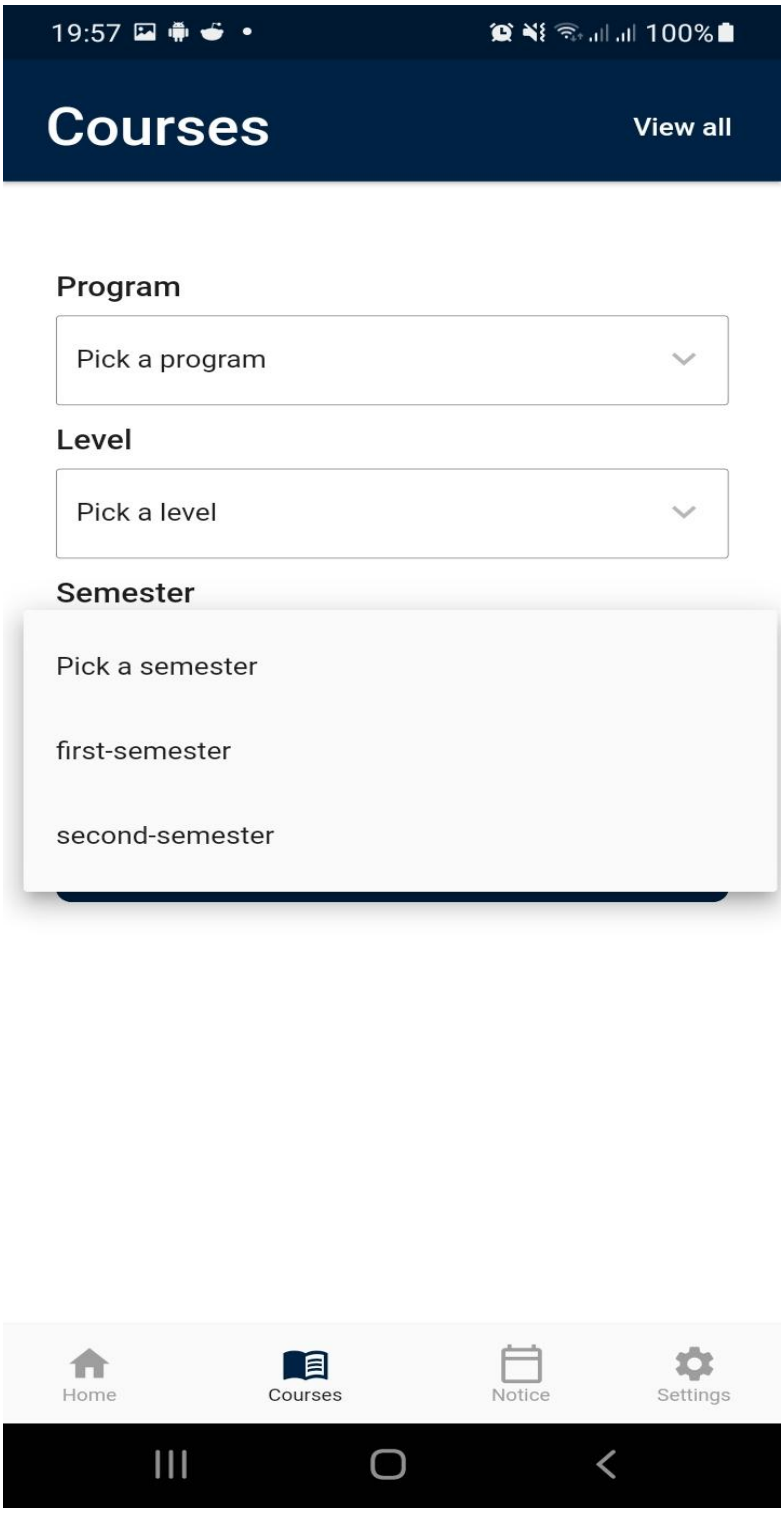

Fig 4.3: Pick a Program Page

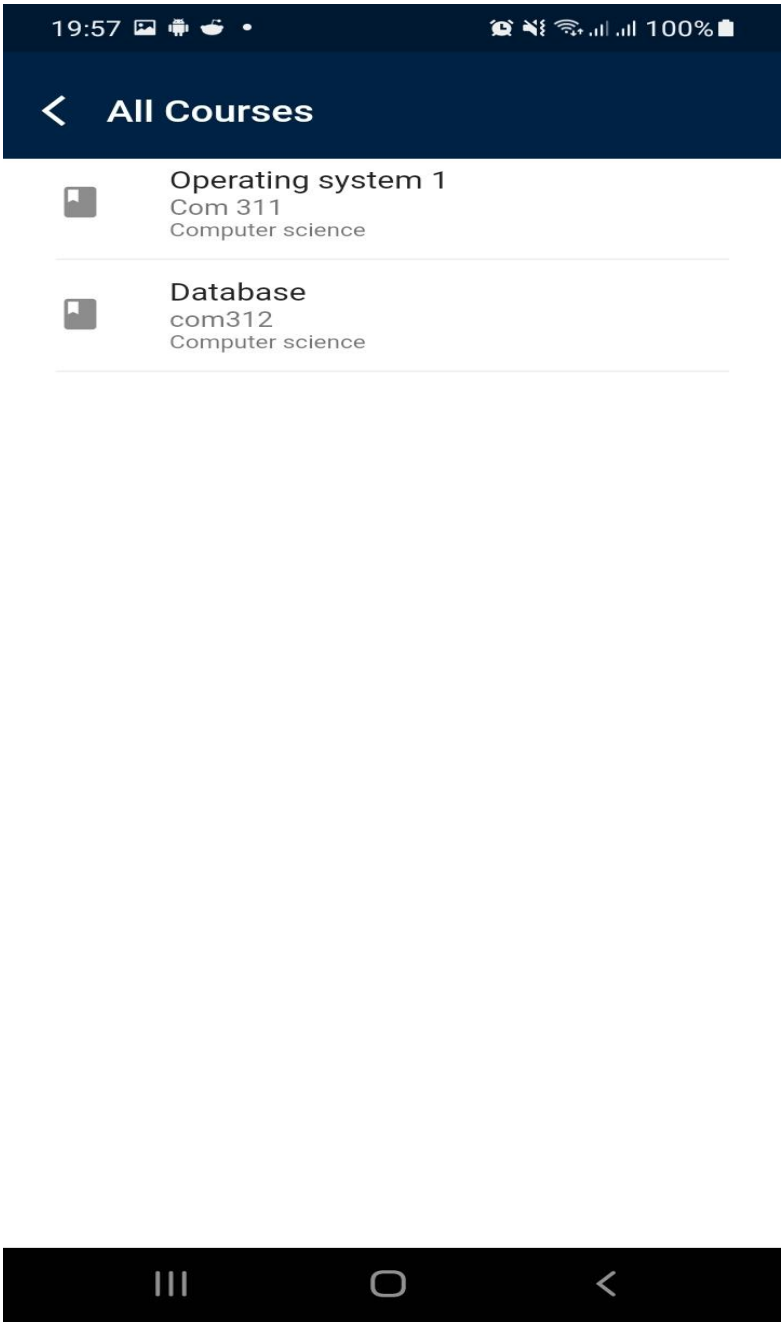

Fig 4.4: Course Page

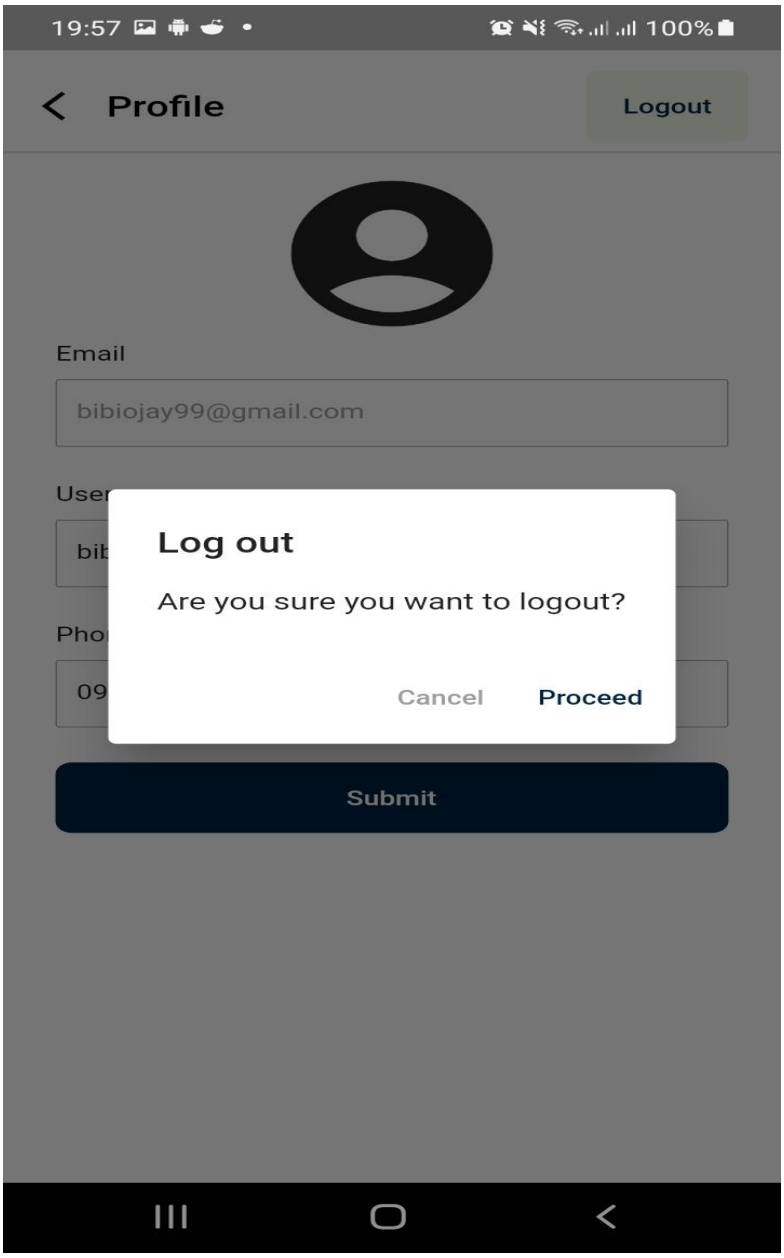

Fig 4.5: Logout Page

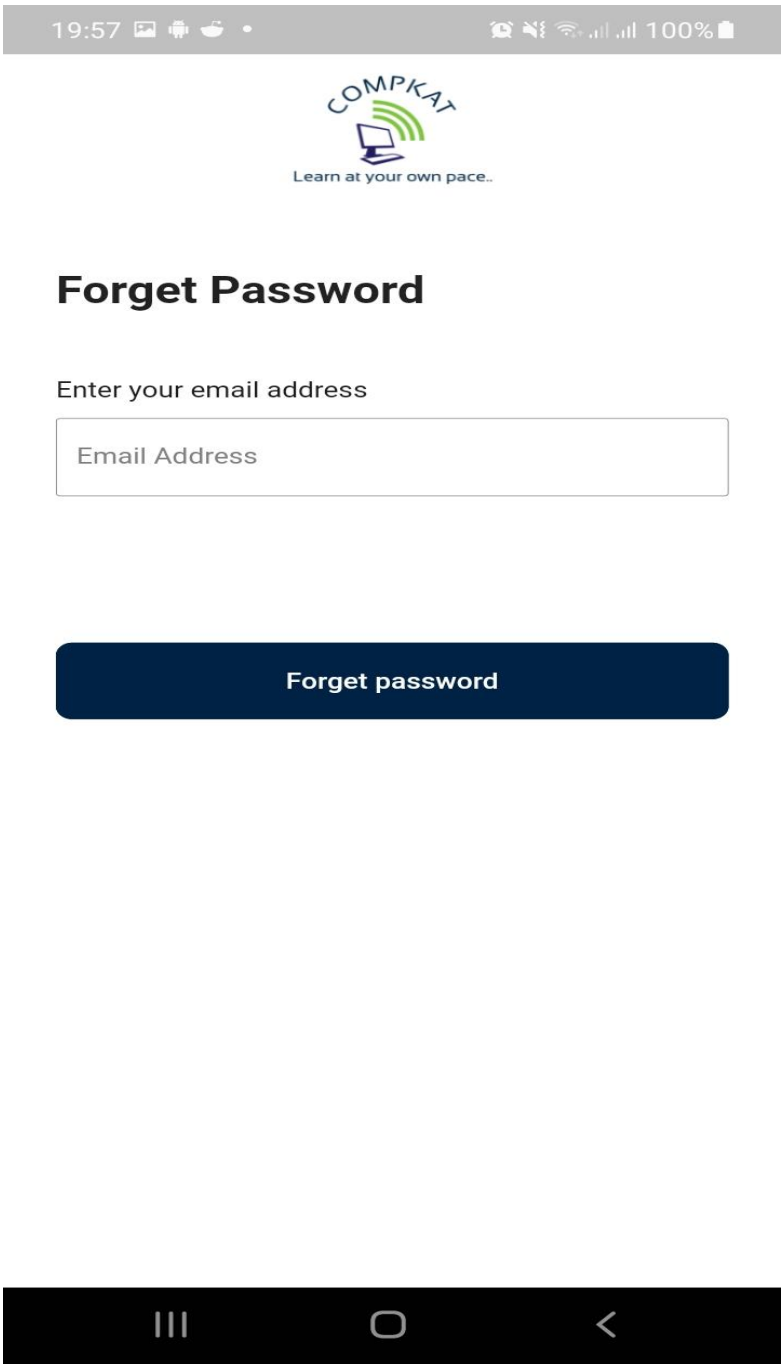

Fig 4.6: Forgot Password

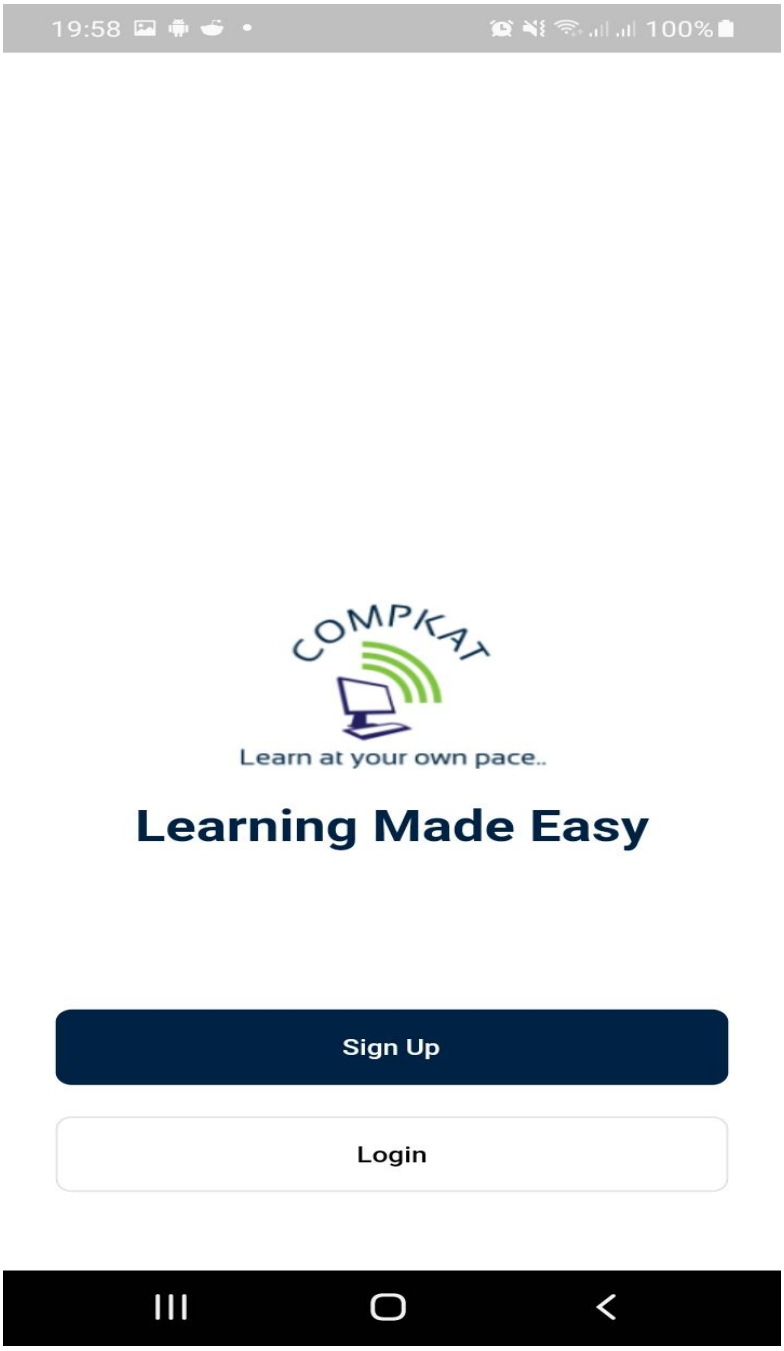

Fig 4.7: Welcome Page

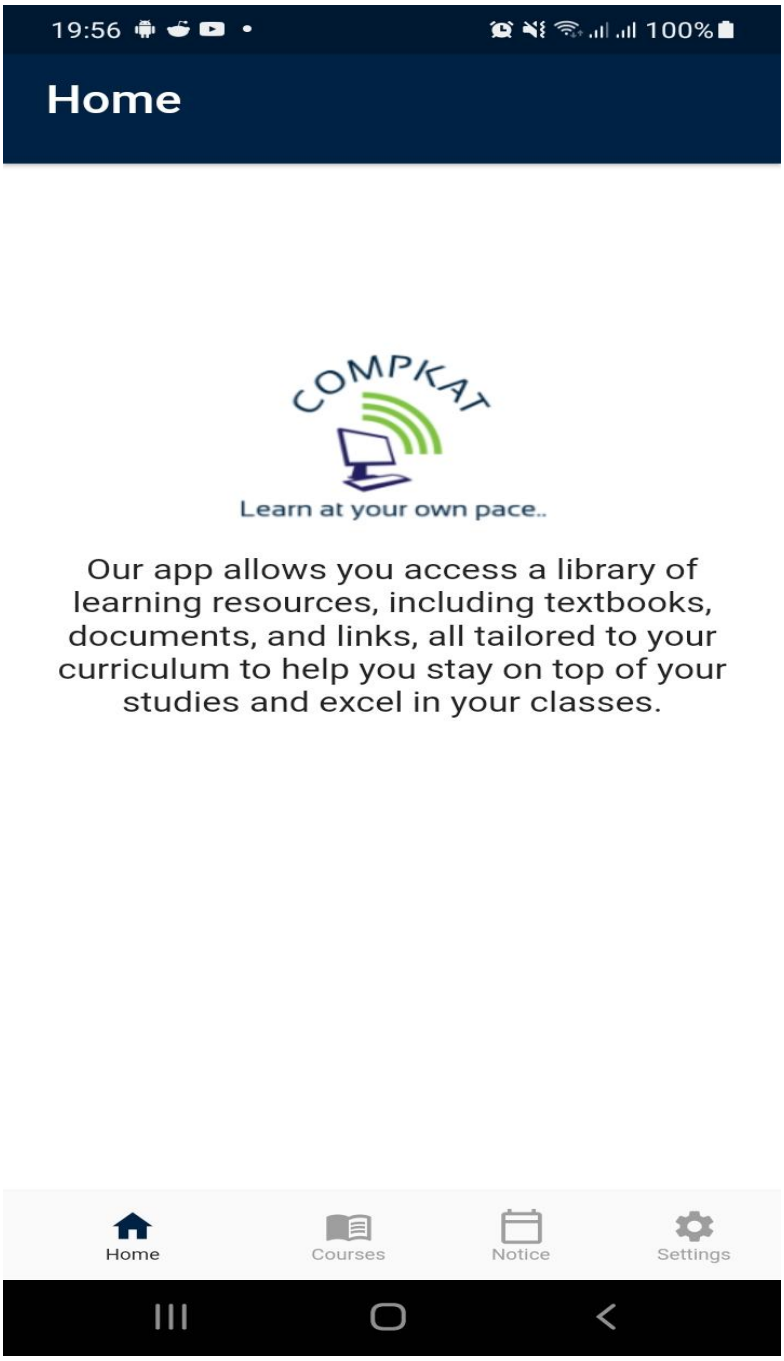

Fig 4.8: Home Page

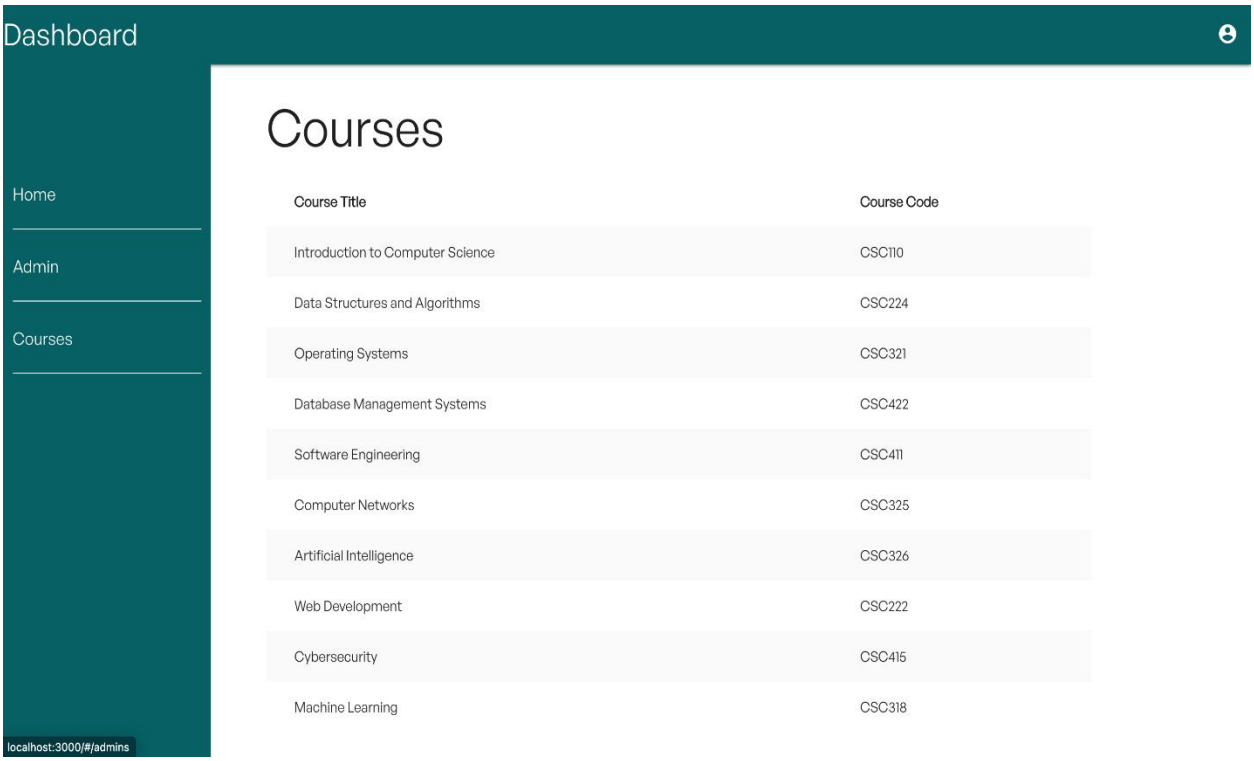

Fig 4.9: Course Page on the Admin Dashboard

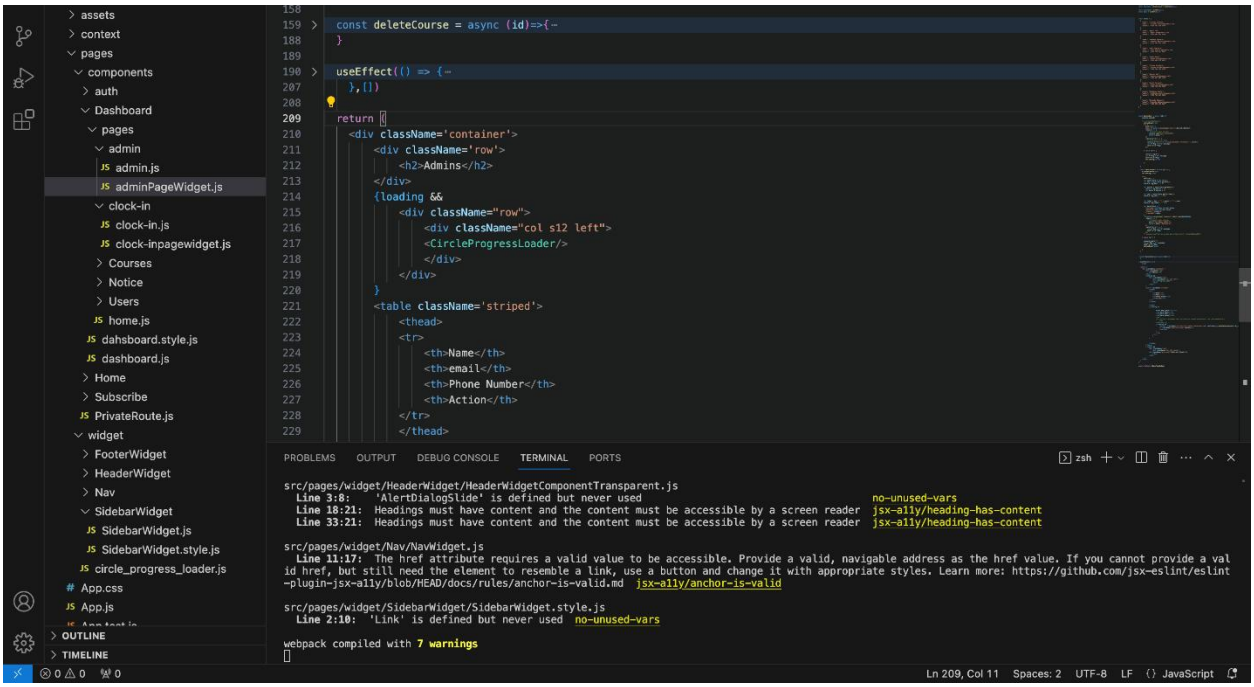

Fig 4.10: Visual Studio Code of the Dashboard

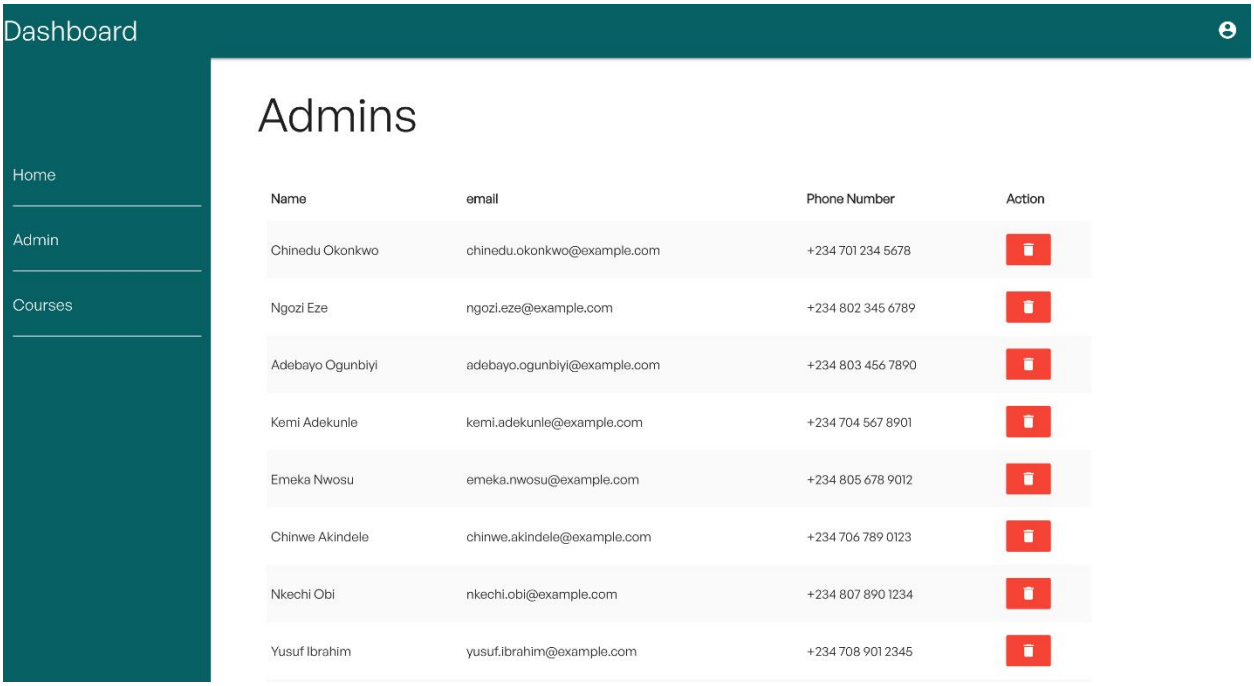

Fig 4.11: Admin Page

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.0 INTRODUCTION

This chapter concludes this study with the following headings: Summary of the study carried out. The conclusion is drawn from the analysis and recommendation to alleviate the problems identified in course of the study. Useful recommendations are also provided.

5.1 SUMMARY

This study focuses on designing an Android-based mobile application for students admitted into the Department of Computer Science, University of Benin. The system is designed to give students an easy, fast and reliable means to access and study course materials for their courses anywhere and at any given time, using Android technology.

Current achievements in the field of Information and Communication Technologies (ICT) have offered tremendous opportunities for learning by electronic means. This electronic means which is an android application for cataloguing course materials is proposed to be used to aid the learning and accessibility of courses. This study showed that the current mode of learning, which is a mixture of the traditional method and partial use of social media platforms like WhatsApp and Telegram for the distribution of materials and communication between course lecturers, representatives and students, is fairly effective but filled with inefficiencies due to reasons associated with using these platforms to aid class activities. This prompted the need for the development of an efficient platform to bridge the gap and reduce the inefficiencies in the process of studying using the aid of mobile devices.

5.2 CONCLUSION

A working mobile application designed specifically to aid students in their studies enhances the ease in accessing and using course materials. The power of ICT to connect lecturers and students is a principle implemented in this project to ease the process of knowledge acquisition and knowledge transfer for students and lecturers respectively.

5.3 RECOMMENDATIONS

The mobile application developed in this project efficiently serves its purpose and the functionalities defined using the resources available (providing a catalogue of course materials including textbooks, past questions, curriculum, and relevant links for each course).

The following recommendations are hereby proposed:

i. The Department of Computer Science may fund the hosting of the mobile application on the Google Play Store server, for unrestricted access to the students and lecturers.
ii. The Department can also make a proposal to the Management for the Information and Communication Technology (ICT) Unit to upscale the development of the application for most or all the departments in the University to use.

5.4 FURTHER STUDIES

This application can be improved by embedding the option of uploading relevant videos and possibly, a means to take assessments which can be graded by the lecturers, and scaled higher to accommodate courses from other departments in the University of Benin. This will serve as a central repository for every course material needed by students of any department in the University.

However, the current working application can be adapted to operate alongside the already existing system, as it will cause a rapid standard shift that will directly have a positive impact on

both students and lecturers of the department, pending further improvements to enhance electronic and mobile learning in the University.

REFERENCES

Akhtar, J. & Sahani, P. K. (2016). *Synopsis on FITNESSBABA android application.* Bachelor of technology, Punjab Technical University Jalandhar. Retrieved from https://www.slideshare.net/jawedakhtar21/synopsis-on-android-application

Al-Rahmi, W., Aldraiweesh, A. A., Yahaya, N., & Alamri, M. M. (2019). The Role of Social Media for Collaborative Learning to Improve Academic Performance among Students. *Contemporary Engineering Sciences, 12*(29), 1917-1928.

Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons.

Brew, M. (2017). What is mobile learning? Definition, Benefits & top tips. Retrieved from http://www.google.com/amp/s/www.edume.com/blog/mobile learning%3fhs_amp=true

Cisco (2018-2023). Annual Internet Report (2018–2023). Retrieved from https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

Dewey, John (2008). Experience and education. Geneva. Kappa Delta press.

Ertmer, P. A., & Ottenbreit-Leftwich, A. T. (2013). Removing obstacles to the pedagogical changes required by Jonassen's vision of authentic technology-enabled learning. *Computers & Education, 64*, 175-182.

Goyal, S. (2020). *Connecting Cloud Firestore Database to Flutter*. DEV Community. Retrieved from https://dev.to/enappd/connecting-cloud-firestore-database-to-flutter-cfp

Haris, M., Jadoon, B., Yousaf, M., Khan, H.F. (2017). Evolution of Android Operating System: A Review. *2nd international conference on Advanced Research (ICAR).* Retrieved from http://www.apiar.org.au

Jay, A. (2023). *How many smartphone users are there in the world?* Financesonline.com. Retrieved from https://financesonline.com/number-of-smartphone-users-worldwide/

Johnson, L., Brown, S., Cummins, M. & Estrada, V. (2012). *The Technology Outlook for STEM+ Education 2012-2017: An NMC Horizon Report Sector Analysis*. Austin, Texas: The New Media Consortium. Retrieved September 13, 2023 from https://www.learntechlib.org/p/48971/

Joo, Y. J., Lim, K. Y., & Kim, N. H. (2018). The effects of mobile applications in an EFL classroom: Learning motivation, communication apprehension,and digital competence. Educational Technology Research and Development, 66(3), 687-704.

Krill, P. (2019). *Dart 2.6 brings native compilation to the desktop*. InfoWorld. Retrieved from https://www.infoworld.com/article/3454623/dart-26-brings-native-compilation-to-the-desktop.html.

Kukulsa-Hulme, A., & Traxler , J. (2013). Mobile learning as a catalyst fir change. *Open Journal: The Journal of Open and Distance Learning, 28*(3), 201-215.

Kumar, R. (2022). *Best & Most Popular 5 Code Editors for C programming*. DevOpsSchool.com. Retrieved from https://www.devopsschool.com/blog/best-most-popular-5-code-editors-for-c-programming/

Laurillard, D. (2012). *Teaching as a design science: Building pedagogical patterns for learning and technology.* Routledge.

O'Malley, C., Vavoula, G., Glew, J. P., Taylor, J., Sharples, M., Lefrere, P., ... & Waycott, J. (2003). Guidelines for learning/teaching/tutoring in a mobile environment.

Operating systems - Statistics & Facts. (2022). Statista. Retrieved from https://www.statista.com/topics/1003/operating-systems/

Patel, A. (2022). *Google Declares Kotlin as Its Favorite Programming Language for Android App Development*. Mindinventory. Retrieved from https://www.mindinventory.com/blog/kotlin-preferred-language-for-android-app-development/

Picciano, A.G., (2017). Theories and Frameworks for online Education: Seeking anIntegrated Model. *Online Learning, 213*, 166-190.

Sarkar, A., Goyal, A., Hicks, D., Sarkar, D., & Hazra, S. (2019). Android application development: a brief overview of android platforms and evolution of security systems.

In *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 73-79). IEEE.

Sharples, M., Adams, A., Alozie, N., Ferguson, R., FitzGerald, E., Gaved, M., . . . Yarnall, L. (2015). *Innovating Pedagogy 2015: Open University Report 4.* The Open University.

Statista. (2022). *Number of mobile devices worldwide 2020-2025*. Retrieved from https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/

Tondeur, J., Van Braak, J., Ertmer, P. A., & Ottenbreit-Leftwich, A. (2017). Understanding the relationship between teachers' pedagogical beliefs and technology in use in education: A systematic review of qualitative evidence. Educational Technology Research and Development, 65(3), 555-575. Retrieved from https://doi.org/10.1007/s11423-016-9481-2

Use cases. (2020). IST Project Management Office. Retrieved from https://uwaterloo.ca/ist-project-management-office/tools-and-templates/tools/use-cases

Walrath, K., & Ladd, S. (2014). *What is dart?* OREILLY Media. Retrieved from https://books.google.com.ng/books?id=8_ccs3hMw1oC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

APPENDIX

Main.dart Page

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:elib/feature/onboarding/services/auth_services.dart';
import 'package:elib/helpers/components/app_icon.dart';
import 'package:elib/helpers/page_layout/page_layout.dart';
import 'package:elib/helpers/theme.dart';
import 'package:elib/screen_loader.dart';
import 'package:elib/welcome.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:sizer/sizer.dart';

void main() async {
WidgetsFlutterBinding.ensureInitialized();
SystemChrome.setPreferredOrientations(
[DeviceOrientation.portraitUp, DeviceOrientation.portraitDown]);
```

```
await Firebase.initializeApp(
options: DefaultFirebaseOptions.currentPlatform,
);
runApp(const MyApp());
}

class MyApp extends StatelessWidget {
const MyApp({Key? key}) : super(key: key);

// This widget is the root of your application.
@override
Widget build(BuildContext context) {
return Sizer(builder: (context, orientation, deviceType) {
return MaterialApp(
title: 'eliby Tv',
debugShowCheckedModeBanner: false,
theme: elibTheme.lightTheme,
home: FutureBuilder(
future: AuthServices().checkIfAuth(),
builder: (BuildContext context, AsyncSnapshot snapshot) {
if (snapshot.connectionState != ConnectionState.done) {
return PageLayout(
```

```
noAppBar: true,
navPop: false,
child: Container(
color: Colors.white,
child: const Center(child: AppIcon())),
);
}

return snapshot.data["token"] != null
? const ScreenLoader()
: const WelcomeOnBoardingPage();
}),
);
});
}
}
```

HomePage

```
import 'package:elib/helpers/components/app_icon.dart';
import 'package:elib/helpers/page_layout/text_formating.dart';
import 'package:flutter/material.dart';
import 'package:elib/helpers/colors.dart';
```

```
import 'package:elib/helpers/page_layout/page_layout.dart';

class Home extends StatefulWidget {
const Home({super.key});

@override
_HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
@override
Widget build(BuildContext context) {
return PageLayout(
// noAppBar: true,
navPop: false,
title: "Home",
fontSize: 25,
appBarColor: bluePrimaryColor,
titleTextColor: Colors.white,
appBarElevation: 2.0,
child: SingleChildScrollView(
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
```

```
mainAxisAlignment: MainAxisAlignment.center,
children: [
const SizedBox(
height: 100.0,
),
const Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
AppIcon(
width: 200.0,
),
],
),
Row(
children: [
Expanded(
child: Text(
"Our app allows you access a library of learning resources, including textbooks, documents, and links, all tailored to your curriculum to help you stay on top of your studies and excel in your classes.",
textAlign: TextAlign.center,
style: textStyle.copyWith(
fontSize: 18.0,
```

)))

],

)

],

),

));

}

}

SignIn

```
import 'package:elib/feature/onboarding/screens/forget_password/screens/forget_password.dart';

import 'package:elib/helpers/util_helpers.dart';

import 'package:firebase_auth/firebase_auth.dart';

import 'package:flutter/material.dart';

import 'package:elib/feature/dashboard/screens/dashboard.dart';

import 'package:elib/feature/onboarding/screens/sign_up/sign_up.dart';

import 'package:elib/helpers/colors.dart';

import 'package:elib/helpers/components/app_icon.dart';

import 'package:elib/helpers/components/button.dart';

import 'package:elib/helpers/components/input_field.dart';

import 'package:elib/helpers/navigators.dart';

import 'package:elib/helpers/page_layout/page_layout.dart';

import 'package:elib/helpers/page_layout/text_formating.dart';
```

```
import 'package:elib/helpers/snakbars.dart';
import 'package:shared_preferences/shared_preferences.dart';

class SignIn extends StatefulWidget {
const SignIn({super.key});

@override
_SignInState createState() => _SignInState();
}

class _SignInState extends State<SignIn> {
final GlobalKey<FormState> _signInFormkey = GlobalKey<FormState>();

final TextEditingController _matricNumberController = TextEditingController();
final TextEditingController _passWordController = TextEditingController();

final bool _showPassword = true;
bool _loading = false;

@override
Widget build(BuildContext context) {
return PageLayout(
noAppBar: true,
```

```
child: SingleChildScrollView(
child: Column(
children: [
Column(
children: [
const Row(
// mainAxisAlignment: MainAxisAlignment.spaceBetween,
mainAxisAlignment: MainAxisAlignment.center,
children: [
// Text("Sign In",
// style:textStyle(
// fontSize: 24.0,
// fontWeight: FontWeight.w700,
// )
// ),
AppIcon(
width: 130,
),
],
),
const SizedBox(
height: 40.0,
),
```

```
Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
mainAxisAlignment: MainAxisAlignment.center,
children: [
Row(
children: [
Text("Login",
style: textStyle.copyWith(
fontSize: 24.0,
fontWeight: FontWeight.w700,
))
],
),
const SizedBox(
height: 40.0,
),
Form(
key: _signInFormkey,
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
InputField(
title: "Enter your matric number",
```

```
controller: _matricNumberController,
// hintText: "E.g F/ND/20/3210001",
hintText: "PSC1808001",
),
const SizedBox(
height: 24.0,
),
InputField(
title: "Enter Lastname",
// passwordInput: _showPassword,
controller: _passWordController,
hintText: "Ojadua",
// suffix: InkWell(
// onTap: () {
// setState(() {
// _showPassword = !_showPassword;
// });
// },
// child: SizedBox(
// width: 20.0,
// child: Image.asset(
// _showPassword
// ? "assets/icons/show_pass.png"
```

```
// : "assets/icons/hide_pass.png",
// width: 12.0,
// height: 12.0,
// ),
// ),
// ),
),
const SizedBox(
height: 10.0,
),
Row(
mainAxisAlignment: MainAxisAlignment.end,
children: [
SizedBox(
child: InkWell(
onTap: () => nextPage(context,
(context) => const ForgetPassword()),
child: Text("Forgot Password?",
style: textStyle.copyWith(
decoration: TextDecoration.underline,
color: bluePrimaryColor,
fontSize: 14.0,
fontWeight: FontWeight.w400,
```

```
)),
),
)
],
),
const SizedBox(
height: 68.0,
),
AppButton(
color: bluePrimaryColor,
loading: _loading,
onPress: () {
if (_signInFormkey.currentState!.validate()) {
// signInAction(context);
final matric = _matricNumberController.text
.trim()
.toLowerCase();
final pass = _passWordController.text
.trim()
.toLowerCase();
signInWithMatricNumber(context, matric, pass);
}
},
```

```
title: "Sign in",
),
const SizedBox(
height: 8.0,
),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
"Or",
style: textStyle.copyWith(
fontSize: 12,
fontWeight: FontWeight.w400,
color: const Color.fromRGBO(
30, 30, 30, 0.64)),
textAlign: TextAlign.center,
),
],
),
SizedBox(
height: 50.0,
child: Stack(
children: [
```

```
Container(
margin: const EdgeInsets.only(top: 18.0),
child: Row(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
Text(
"Do not have an account? ",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: textColor,
fontSize: 14,
fontWeight: FontWeight.w400),
),
const SizedBox(
width: 2.0,
),
InkWell(
onTap: () => nextPage(context,
(context) => const SignUp()),
child: Text("Sign up",
textAlign: TextAlign.center,
style: textStyle.copyWith(
```

```
color: bluePrimaryColor,
fontSize: 14,
fontWeight: FontWeight.w700,
)),
),
],
),
)
],
),
),
const SizedBox(
height: 80.0,
),
],
),
),
],
),
),
],
),
```

```
));
}

signInWithMatricNumber(context, matric, pass) async {
final pref = await SharedPreferences.getInstance();
setState(() {
_loading = true;
});
try {
final req = await firestore
.collection("users")
.where('matricNumber', isEqualTo: matric)
.where('lastName', isEqualTo: pass)
.get()
.then((value) {
setState(() {
_loading = false;
});
print(value.docs);
final data = value.docs;
if (data.isNotEmpty) {
final user = data[0].data();
print(user);
```

```
pref.setString("token", user['userId']);
pref.setString("email", user['email']);

defaultSnackyBar(context, "login successfull", successColor);
nextPageNoPop(context, (context) => const Dashboard());
} else {
defaultSnackyBar(context, "User not found", dangerColor);
}
}).catchError((err) {
setState(() {
_loading = false;
});
});
} catch (e) {
// print(e);
defaultSnackyBar(context, "An error occured", dangerColor);
setState(() {
_loading = false;
});
}
}

signInAction(context) async {
```

```
final pref = await SharedPreferences.getInstance();
final token = pref.getString("token");
setState(() {
_loading = true;
});

try {
final credential = await FirebaseAuth.instance.signInWithEmailAndPassword(
// email:"fluffydev007@gmail.com",
// password:"090Aai###",
email: _matricNumberController.text.trim(),
password: _passWordController.text.trim());
setState(() {
_loading = false;
});
print(credential.user!.uid);
pref.setString("token", credential.user!.uid);
pref.setString("email", credential.user!.email!);

defaultSnackyBar(context, "login successfull", successColor);
nextPageNoPop(context, (context) => const Dashboard());
} on FirebaseAuthException catch (e) {
setState(() {
```

```
_loading = false;
});
String err = "${e.message}";
if (e.code == 'user-not-found') {
err = 'No user found for that email.';
} else if (e.code == 'wrong-password') {
err = 'Wrong password provided for that user.';
}
defaultSnackyBar(context, err, dangerColor);
print(e);
} catch (e) {
// print(e);
defaultSnackyBar(context, "An error occured", dangerColor);
setState(() {
_loading = false;
});
}
}
}
```

SignUp

```
import 'dart:async';
```

```
import 'package:elib/helpers/snakbars.dart';
import 'package:elib/helpers/util_helpers.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
// import 'package:google_fonts/google_fonts.dart';
import 'package:elib/feature/dashboard/screens/dashboard.dart';
import 'package:elib/feature/onboarding/screens/sign_in/sign_in.dart';
import 'package:elib/helpers/colors.dart';
import 'package:elib/helpers/components/app_icon.dart';
import 'package:elib/helpers/components/button.dart';
import 'package:elib/helpers/components/email_input.dart';
import 'package:elib/helpers/components/input_field.dart';
import 'package:elib/helpers/components/telephone_input.dart';
import 'package:elib/helpers/navigators.dart';
import 'package:elib/helpers/page_layout/page_layout.dart';
import 'package:elib/helpers/page_layout/text_formating.dart';
import 'package:shared_preferences/shared_preferences.dart';

class SignUp extends StatefulWidget {
const SignUp({super.key});

@override
```

```
_SignUpState createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {
final GlobalKey<FormState> _signUpFormkey = GlobalKey<FormState>();
final TextEditingController _emailController = TextEditingController();
final TextEditingController _userNameController = TextEditingController();
final TextEditingController _firstNameController = TextEditingController();
final TextEditingController _lastNameController = TextEditingController();
final TextEditingController _phoneNumeberController = TextEditingController();
final TextEditingController _matricController = TextEditingController();
final TextEditingController _passWordController = TextEditingController();

final bool _showPassword = true;
bool _loading = false;

@override
Widget build(BuildContext context) {
return PageLayout(
noAppBar: true,
child: SingleChildScrollView(
child: Column(
children: [
```

```
Column(
children: [
const SizedBox(
height: 20.0,
),
const Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
AppIcon(
width: 130,
),
],
),
const SizedBox(
height: 40.0,
),
Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
mainAxisAlignment: MainAxisAlignment.center,
children: [
Row(
children: [
Text("Sign Up",
```

```
style: textStyle.copyWith(
fontSize: 24.0,
fontWeight: FontWeight.w700,
)),
],
),
const SizedBox(
height: 40.0,
),
Form(
key: _signUpFormkey,
child: Column(
crossAxisAlignment: CrossAxisAlignment.stretch,
children: [
InputField(
title: "Username",
controller: _userNameController,
hintText: "John doe",
),
const SizedBox(
height: 24.0,
),
InputField(
```

```
title: "Firstname",
controller: _firstNameController,
hintText: "John doe",
),
const SizedBox(
height: 24.0,
),
InputField(
title: "Lastname",
controller: _lastNameController,
hintText: "John doe",
),
const SizedBox(
height: 24.0,
),
InputField(
title: "Matric number",
controller: _matricController,
hintText: "PSC1808001",
),
const SizedBox(
height: 24.0,
),
```

```
EmailInputField(
title: "Enter your email address",
controller: _emailController,
hintText: "Email Address",
),
const SizedBox(
height: 24.0,
),
TelephoneInput(
title: "Enter your Mobile number",
controller: _phoneNumeberController,
hintText: "+234 9023687892",
),
const SizedBox(
height: 24.0,
),
// InputField(
// title: "Enter password",
// passwordInput: _showPassword,
// controller: _passWordController,
// hintText: "Password",
// suffix: InkWell(
// onTap: () {
```

```
// setState(() {
// _showPassword = !_showPassword;
// });
// },
// child: Container(
// width: 20.0,
// child: Image.asset(
// _showPassword
// ? "assets/icons/show_pass.png"
// : "assets/icons/hide_pass.png",
// width: 12.0,
// height: 12.0,
// ),
// ),
// )),
const SizedBox(
height: 10.0,
),
Row(
mainAxisAlignment: MainAxisAlignment.end,
children: [
SizedBox(
child: InkWell(
```

```
onTap: null,
// onTap: () => nextPage(context,
// (context) => ResetPasswordEnterEmail()),
child: Text("Forgot Password?",
style: textStyle.copyWith(
decoration: TextDecoration.underline,
color: bluePrimaryColor,
fontSize: 14.0,
fontWeight: FontWeight.w400,
)),
),
)
],
),
const SizedBox(
height: 68.0,
),
AppButton(
loading: _loading,
onPress: () {
if (_signUpFormkey.currentState!.validate()) {
final data = {
"phoneNumber":
```

```
_phoneNumeberController.text.trim(),
"password": _passWordController.text.trim()
};
signUpAction(context, data);
}
},
title: "Sign Up",
),
const SizedBox(
height: 8.0,
),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
"Or",
style: textStyle.copyWith(
fontSize: 12,
fontWeight: FontWeight.w400,
color: const Color.fromRGBO(
30,
30,
30,
```

```
0.64,
),
),
textAlign: TextAlign.center,
),
],
),

SizedBox(
height: 50.0,
child: Stack(
children: [
Container(
margin: const EdgeInsets.only(top: 18.0),
child: Row(
mainAxisAlignment:
MainAxisAlignment.center,
children: [
Text("Already have an Account? ",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: textColor,
fontSize: 14,
```

```
fontWeight: FontWeight.w400,
)),
const SizedBox(
width: 2.0,
),
InkWell(
onTap: () => nextPage(context,
(context) => const SignIn()),
child: Text(
"Sign in",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: bluePrimaryColor,
fontSize: 14,
fontWeight: FontWeight.w700,
),
),
),
],
),
)
],
),
```

```
),
const SizedBox(
height: 80.0,
),
SizedBox(
height: 50.0,
child: Stack(
children: [
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
"By Signing in or creating an account, you agree with",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: textColor,
fontSize: 13,
fontWeight: FontWeight.w400)),
],
),
Container(
margin: const EdgeInsets.only(top: 18.0),
child: Row(
```

```
mainAxisAlignment:
MainAxisAlignment.center,
children: [
Text(
"our",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: textColor,
fontSize: 13,
fontWeight: FontWeight.w400),
),
const SizedBox(
width: 2.0,
),
Text("Terms & Conditions",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: bluePrimaryColor,
fontSize: 13,
fontWeight: FontWeight.w700)),
const SizedBox(
width: 2.0,
),
```

```
Text("and",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: textColor,
fontSize: 13,
fontWeight: FontWeight.w400,
)),
const SizedBox(
width: 2.0,
),
Text("Privacy Statement",
textAlign: TextAlign.center,
style: textStyle.copyWith(
color: bluePrimaryColor,
fontSize: 13,
fontWeight: FontWeight.w700,
)),
],
),
)
],
),
),
```

```
const SizedBox(
height: 80.0,
),
],
),
),
],
),
],
),
],
),
));
}


signUpAction(context, data) async {
// nextPage(context, (context) => Dashboard());
final pref = await SharedPreferences.getInstance();
final token = pref.getString("token");
setState(() {
_loading = true;
});
```

```
try {
final credential = await FirebaseAuth.instance
.createUserWithEmailAndPassword(
email: _emailController.text.trim(),
password: _userNameController.text.trim());
setState(() {
_loading = false;
});
print(credential.user!.uid);
pref.setString("token", credential.user!.uid);
pref.setString("email", credential.user!.email!);

// defaultSnackyBar(context, "login successfull", successColor);
// nextPageNoPop(context, (context) => Dashboard());
final data = {
"name": _userNameController.text.trim().toLowerCase(),
"firstName": _firstNameController.text.trim().toLowerCase(),
"lastName": _lastNameController.text.trim().toLowerCase(),
"matricNumber": _matricController.text.trim().toLowerCase(),
"email": _emailController.text.trim(),
"phoneNumber": _phoneNumeberController.text.trim(),
"userId": credential.user!.uid
};
```

```
addUser(context, data, credential.user!.uid);
} on FirebaseAuthException catch (e) {
setState(() {
_loading = false;
});
String err = "${e.message}";
if (e.code == 'user-not-found') {
err = 'No user found for that email.';
} else if (e.code == 'wrong-password') {
err = 'Wrong password provided for that user.';
}
defaultSnackyBar(context, err, dangerColor);
print(e);
} catch (e) {
// print(e);
defaultSnackyBar(context, "An error occured", dangerColor);
setState(() {
_loading = false;
});
} // setState(() {
}

addUser(context, userData, uid) async {
```

```
final req = await firestore
.collection("users")
.doc(uid)
.set(userData)
.whenComplete(() {
defaultSnackyBar(context, "Sign up successfull.", successColor);
Timer(
const Duration(seconds: 3),
() => nextPageNoPop(context, (context) => const Dashboard()),
);
}).catchError((err) {
setState(() {
_loading = false;
});
});
}
}
```