CHAPTER THREE

DESIGN OF A VIRTUAL LIBRARY LEARNING PLATFORM FOR 300 LEVEL MANAGEMNET SCIENCE STUDENTS OF UNIVERSITY OF JOS

3.0 INTRODUCTION

This chapter describes the structure and components of the application, programming languages used in development and justification for the use of those languages. The Virtual Library Learning Platform is built to make it possible for both instructors and learners to communicae and share information virtually. It also states the requirement and system design of the platform.

3.1 STRUCTURE OF THE PROPOSED SYSTEM

The Virtual Library Learning Platform is developed following Microservices Artitecture. According to Wolff (2016), Services in a Microservice Architecture are often processes that communicate over a network to fulfill a goal using technology-agnostic protocols such as HTTP. Services can be implemented using different programming languages, databases, hardware and software environments, depending on what fits best (Chen, 2018). The diagram below gives an overview of the Systems Architecture.

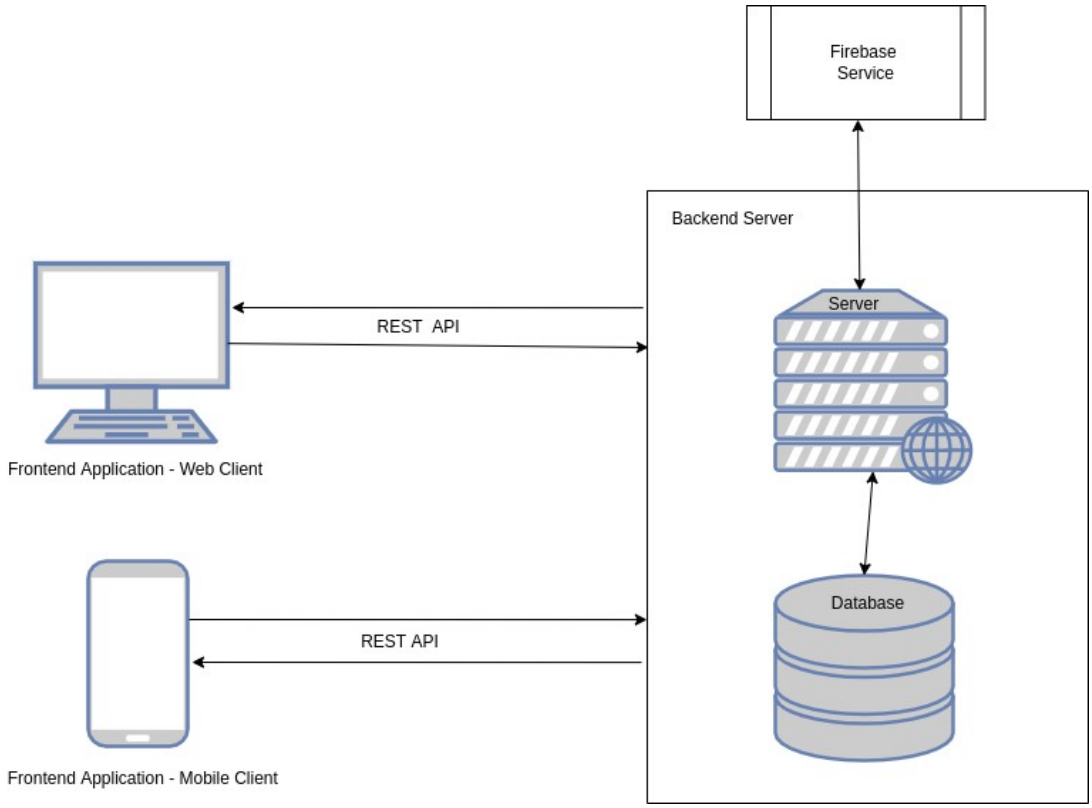


Figure 3.1: System Architecture

Using the Microservices Architecture, the Virtual Library Learning platform was broken down into functional services. For this services to communicate effectively, a Representational State Transfer (REST) Application Programming Interface (API) is used as the basis for communication between these services.

The VLLP mobile application provides an interface for users to interact with the Virtual Learning Platform System seamlessly. The image below shows what users see after successfull installation of the Mobile Application on an Android device.

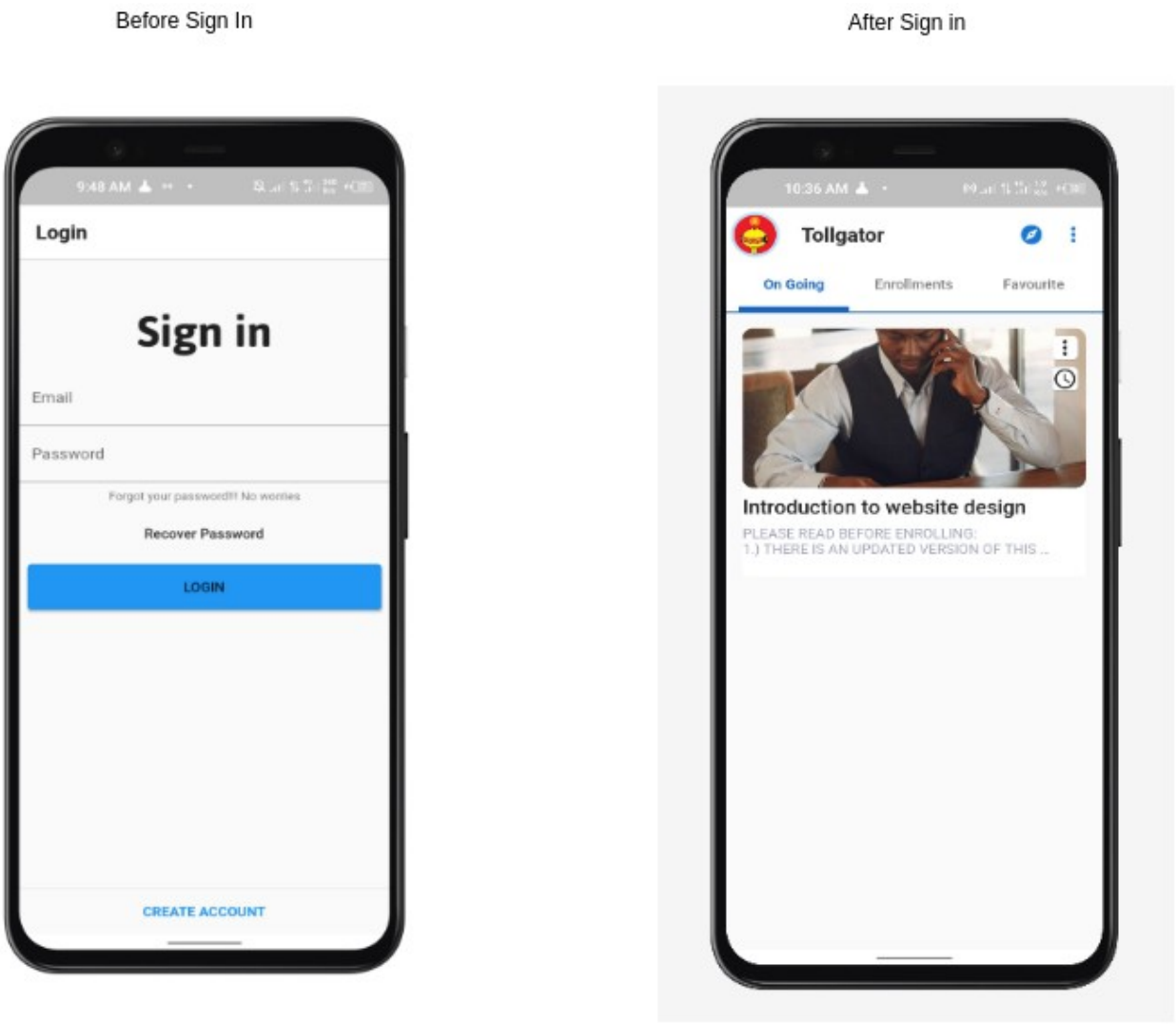


Figure 3.2: Authentication and Home Screens

Interaction between User and Mobile Application Hardware

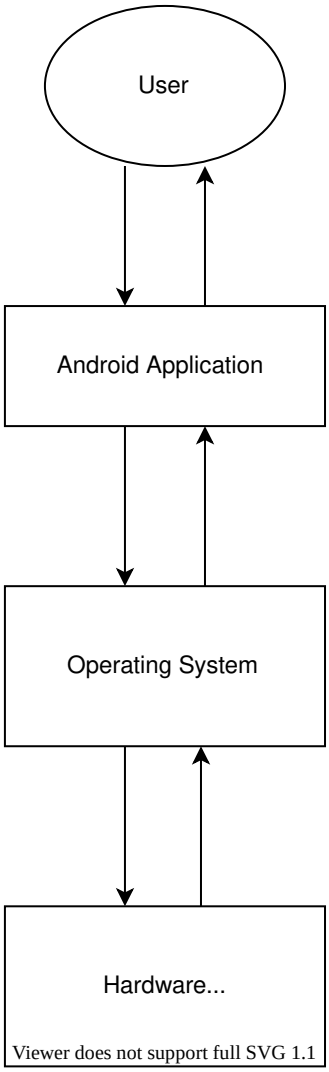


Figure 3.3: Interaction between User and Application Hardware

The major modules of this system are outlined below

1. **Backend module**

   This service serves the purpose of providing all required back-end functionalities for the effective running of the system. This includes; Course management, user data management, user registration, and information resource management. Its functionality is made available through a well-defined API accessible via HTTPS.

2. **Firebase authentication module**

   This service is based on the Google Firebase toolkit for mobile application development. It is used to handle all user authentication data and provide user-specific customization and real-time notifications.

3. **Student Front-end module**

   This module presents an interface users can interact with to access functionalities provided by the platform. This includes browsing a collection of courses uploaded by instructors, accessing freely available information resources uploaded, account creation and login functionalities. The Frontend module is available for desktop and mobile devices.

4. **Instructor Front-end module**

   This module presents an interface instructors interact with to access functionalities provided by the platform. This includes courses management, information resources upload, account creation and login functionalities. The Frontend module is available for desktop and mobile devices.

System Data flowchart

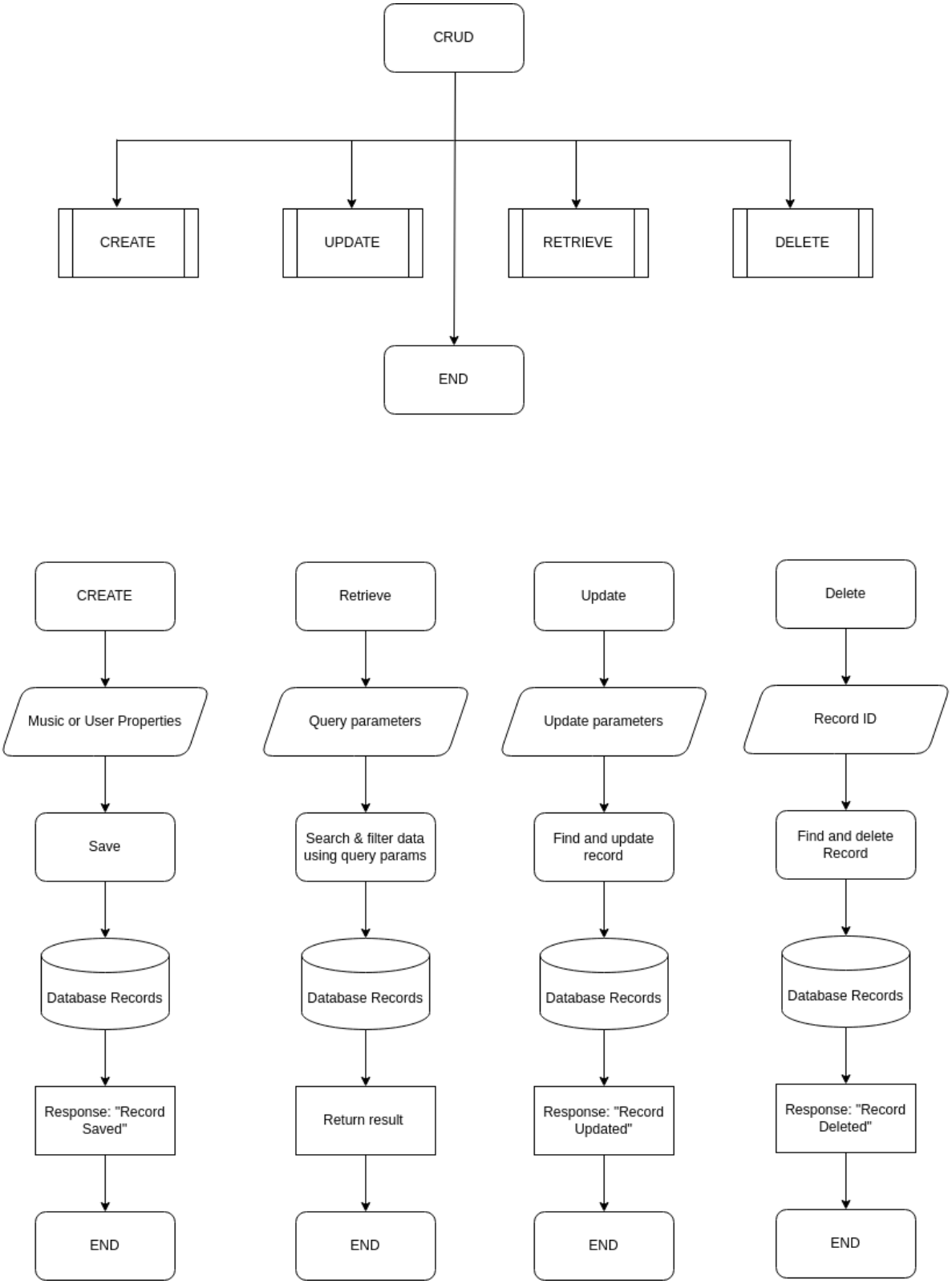


Figure 3.4: System Data flowchart

3.2 HARDWARE DESIGN MODEL

The proposed Virtual Library Learning Platform can be grouped into two parts. The server and the client. The server component of the platform is responsible for handling all business login that makes it possible for users to interact with the platform. The server component of the platform is hosted on an online server accessble via the internet. The client component of the platform are user managed devices which are personal computers and mobile devices. These clients communicate with the server via the internet.

To run the backend application on the Server-side. The following are the minimum hardware requirement:

1 RAM: 512 MB
2 NodeJs Server Software
3 Heroku ephemeral storage

To run the frontend application on the Client-side The following are the minimum hardware requirement:

1 RAM: Minimim of 512 MB
2 Android, Windows, or Linux Operating System
3 Mobile or Computer Device

Hardware Diagram

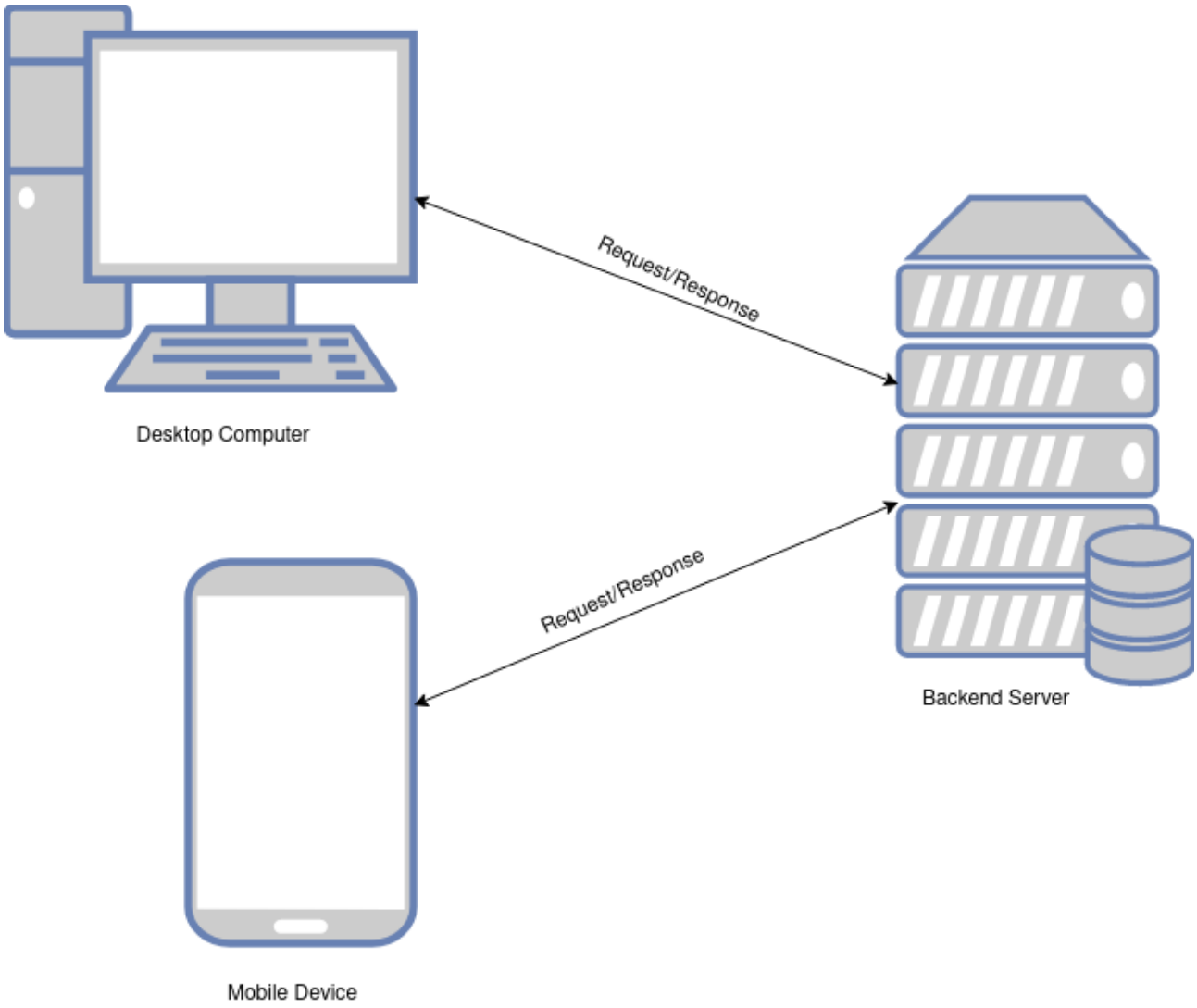


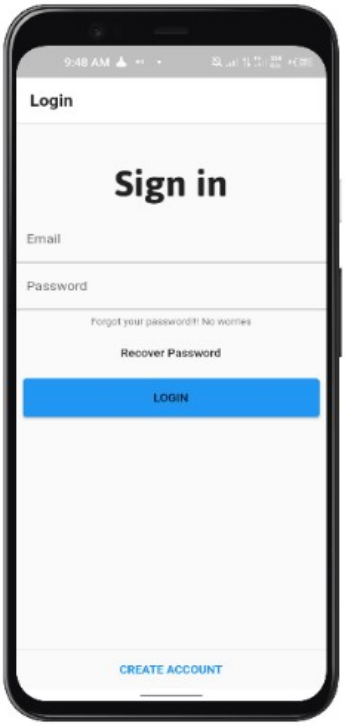
Figure 3.5: Hardware Design Model

3.3 GUI (MOBILE APPLICATION) DESIGN MODEL

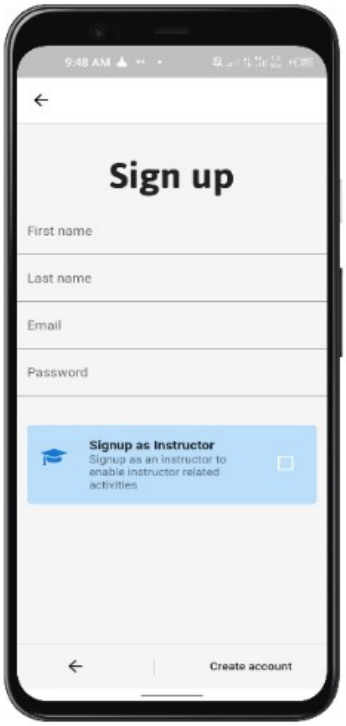
The Model-View-Controller design pattern was used in the development of the mobile application. Model–view–controller (MVC) is a software design pattern; commonly used for developing user interfaces that divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user (Reenskaug, 2019). The VVLP Mobile Application is made up of pages, models and controllers which work together to appropriately represent data and provide a functional interface.

Mobile Application Screens

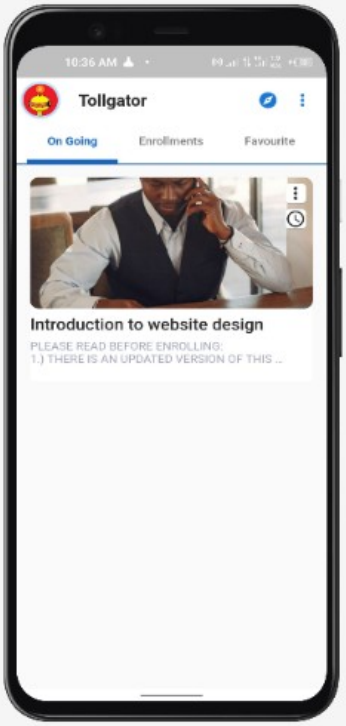
Login Screen



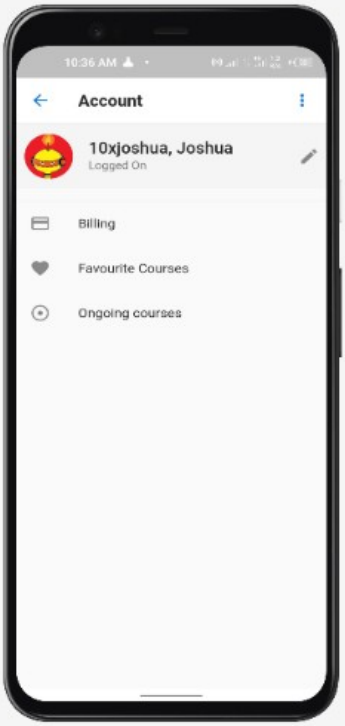
Register



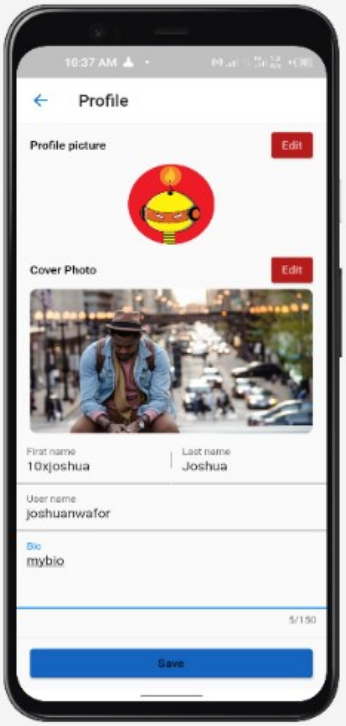
Home Screen



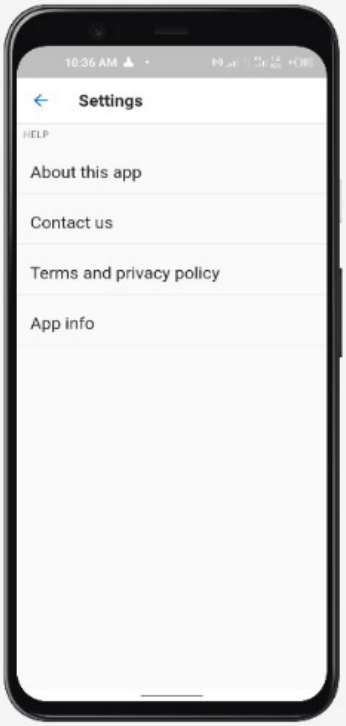
Account Page



Profile Update



Setting Screen



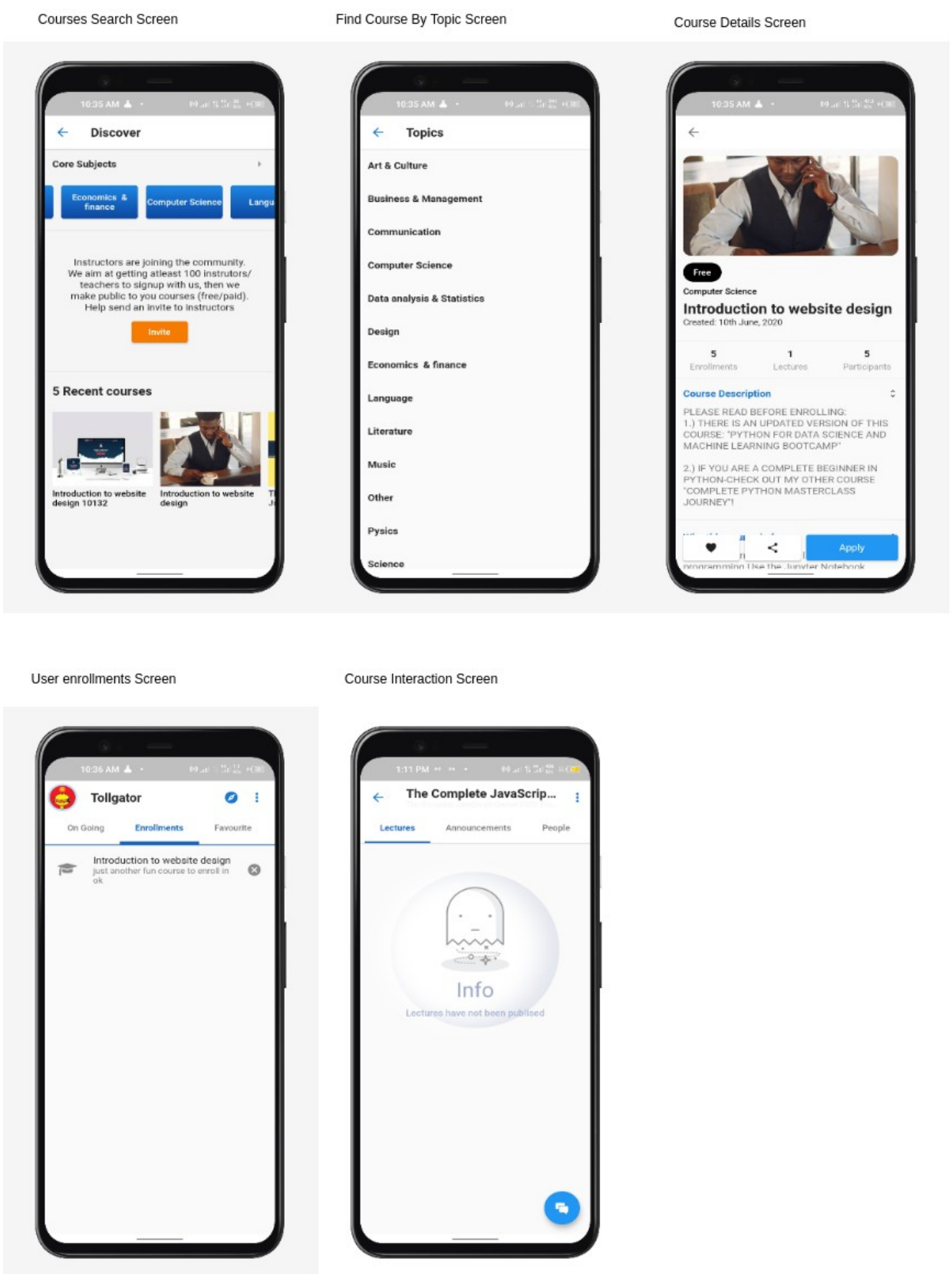


Figure 3.6: Mobile Application Screens

Entity Relationship Diagram

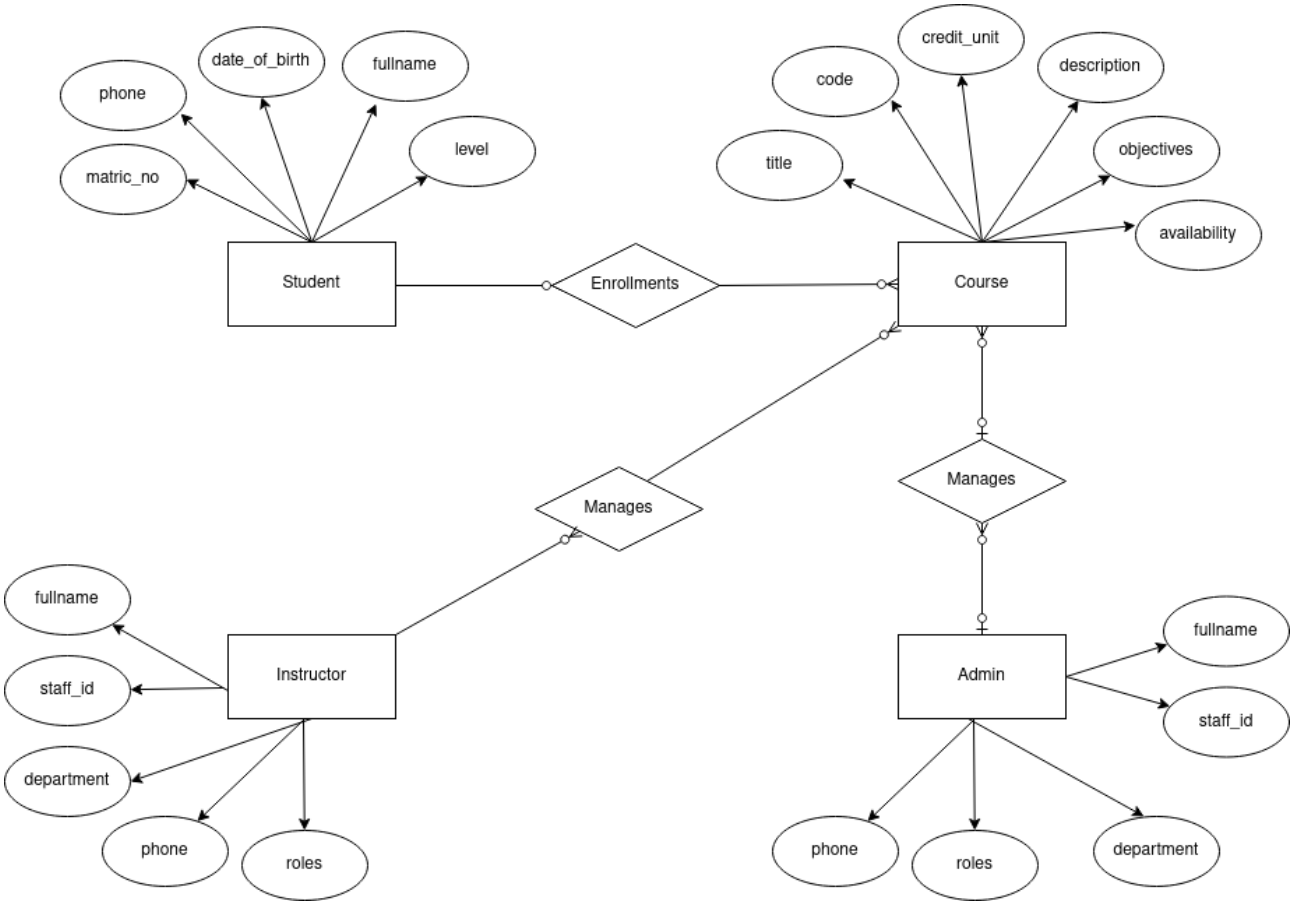


Figure 3.7: Entity Relationship Diagram

3.3.1 MOBILE APPLICATION COMPONENTS DESCRIPTION

The VLLP mobile application is made up of modules that work together to properly represent information to the user.

- **Authentication module**: The authentication module is responsible for authentication users in to the application, for users to be properly authenticated, they must create an account on the application. Account creation and management is handled by the Firebase authentication service. This module provides the login and registeration screen that makes it possible for users to get authenticated.

- **User home module**: The user home view provides a single entry point into other parts of the application. From this view, users can navigate to other routes in the application.

- **User courses module**: The user courses view is responsible for loading all courses which an authenticated user has enrolled in.

- **Course search module**: The course search module provides an interface for authenticated users to search courses available in the system and view information on those course.

- **Course module**: This module is responsible for presenting an interface to user to have access to course resoures, this includes course members, forums, and lessons.

- **Course lesson module**: The presents a view to users to intereact and make use of course resources. This could be a video, audio, or digital file resource.

- **Course forum module:** This module provides a view for users to interact with and access course forum resources. Resources in a course forum are messages published by members of a course.

- **User profile module**: The user profile module provides an interface where users can update and view their profile.

Entity Relationship Diagram

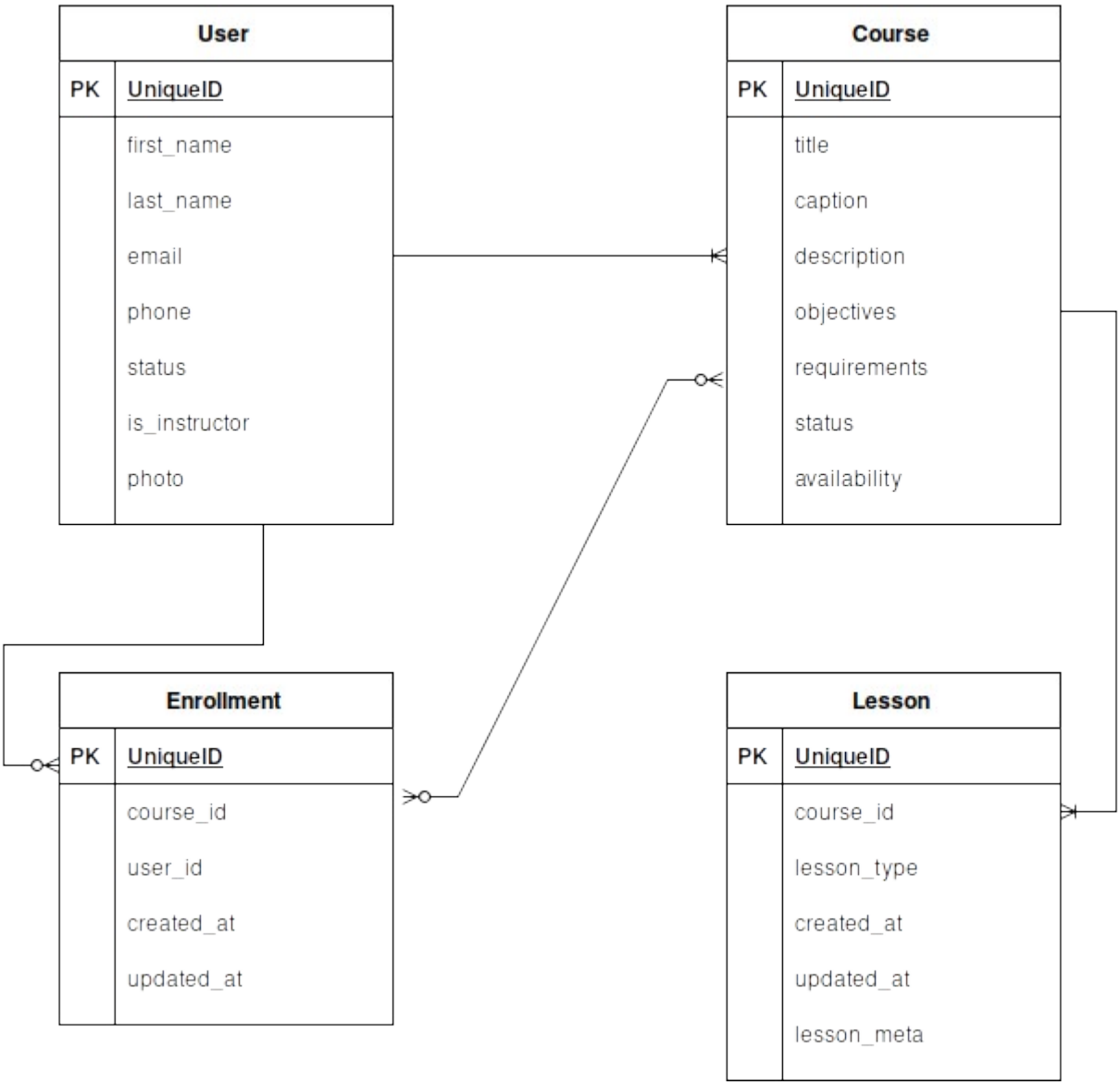


Figure 3.8: Entity Relationship Diagram

3.4 OPERATION OF THE PROPOSED SYSTEM

Representation of User roles and actions interacting with the system

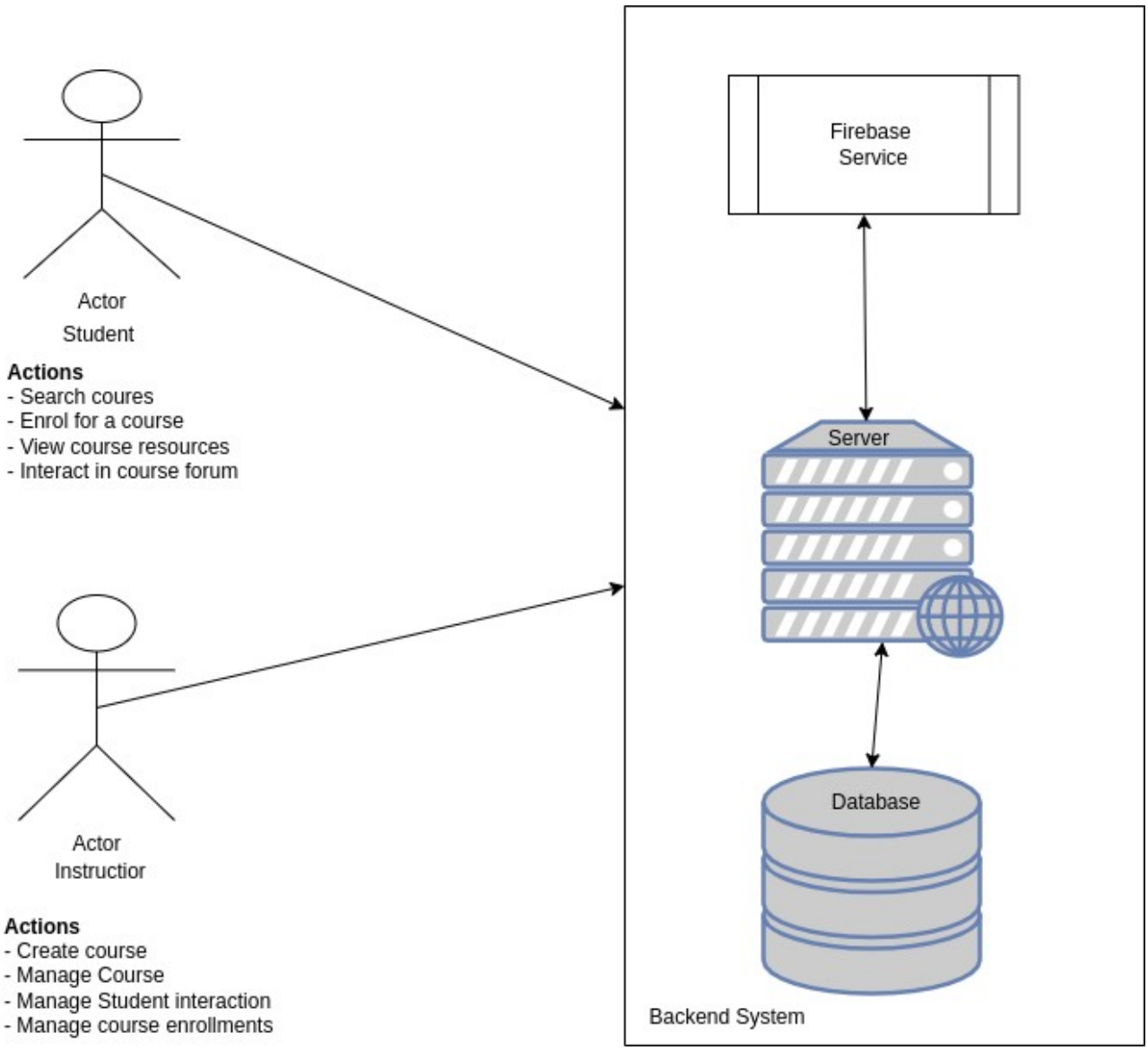


Figure 3.9: User interaction with system

Student Mobile Application Authentication Flow

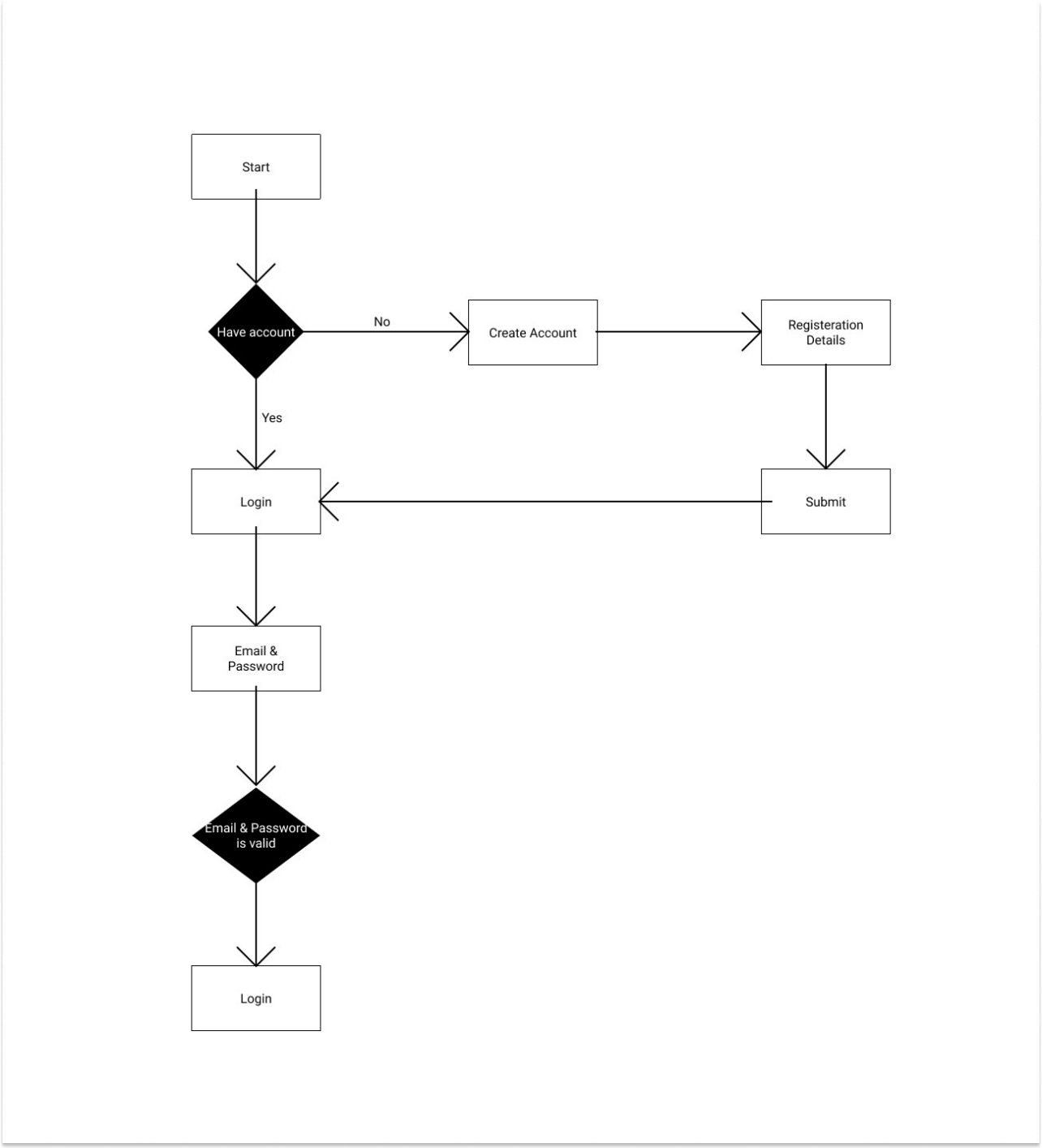


Figure 3.10: Student Authentication Flow

Instructor Mobile Application Authentication Flow

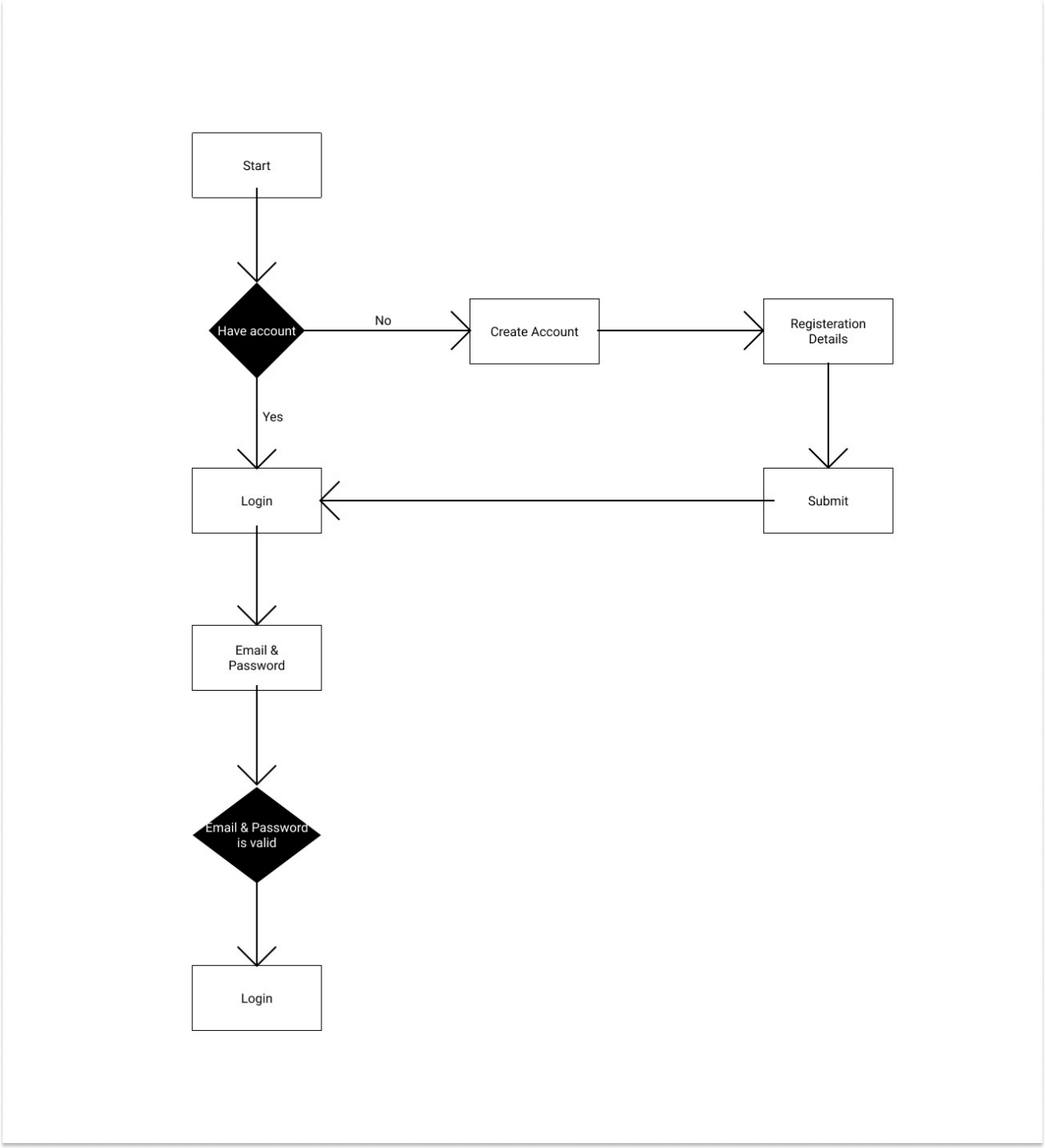


Figure 3.11: Instructor Authentication Flow

Representation of User interaction with mobile application

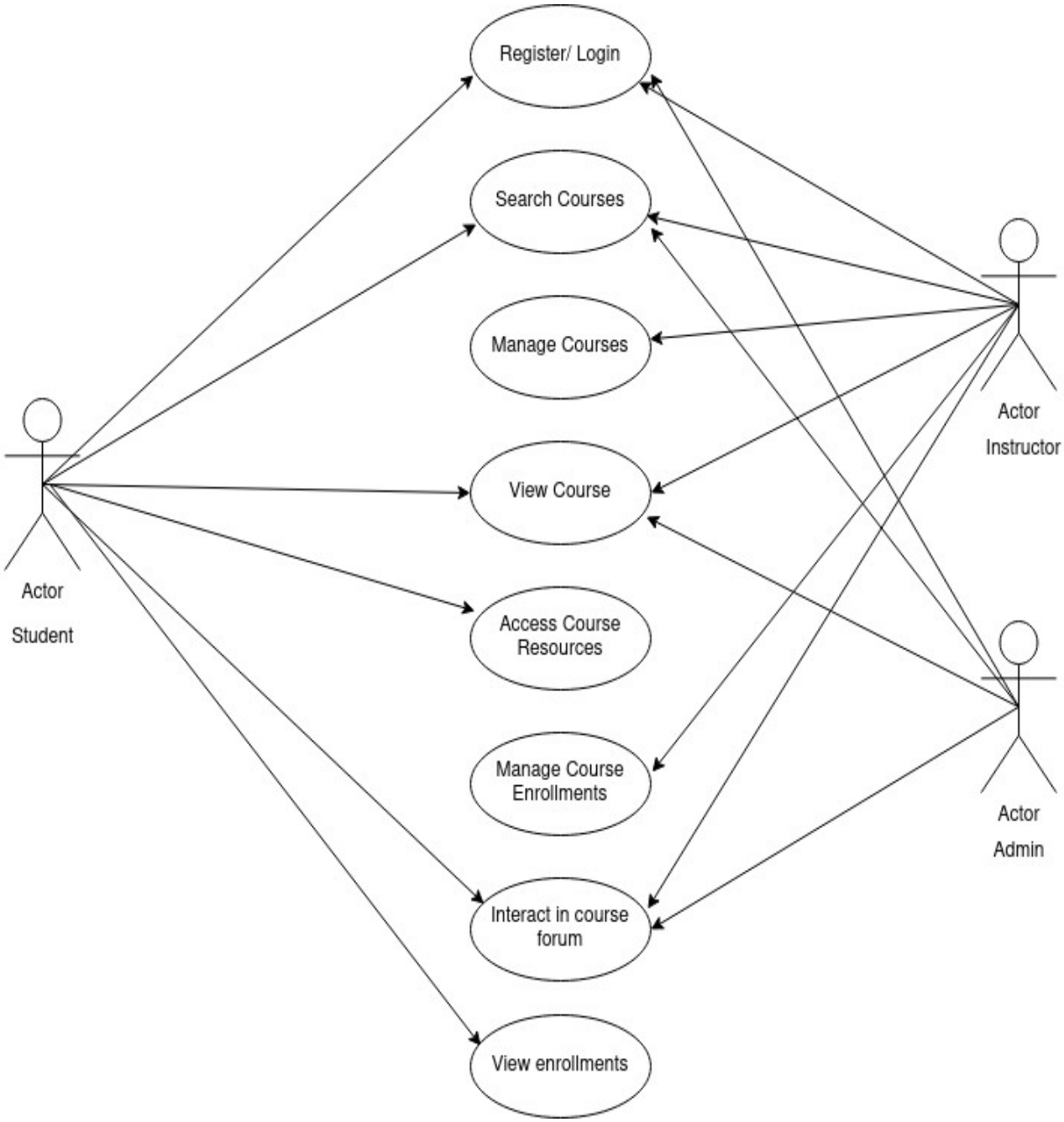


Figure 3.12: User interaction with mobile application

Authenticated User Dashboard flow

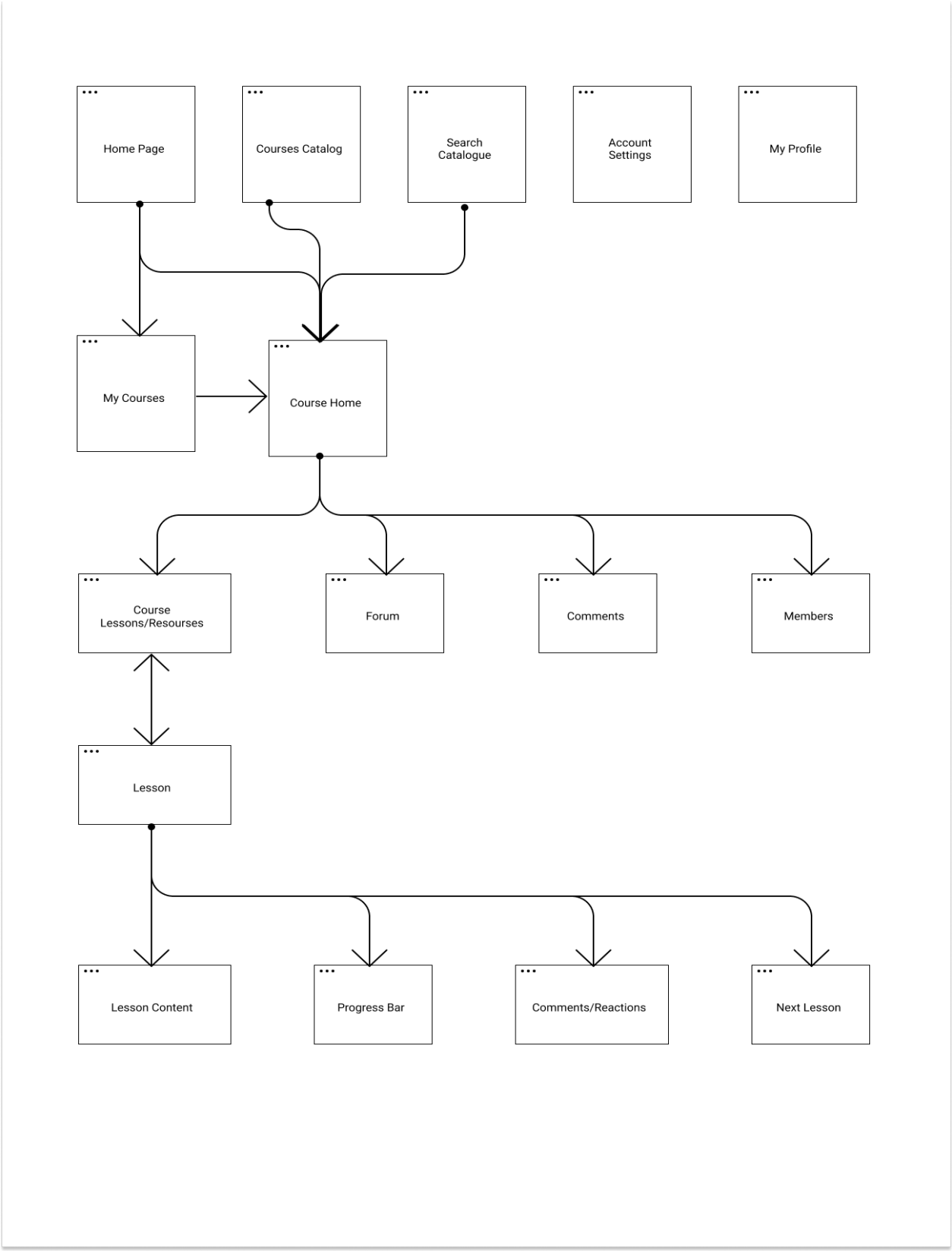


Figure 3.13: Authenticated user flowchart

3. ADVANTAGES OF THE PROPOSED PLATFORM

1. Learners can easily access course resources from the comfort of their phones.
2. Course forum makes it possible for learners to as questions virtually and have answered buy fellow learners or the instructor realtime
3. Instructors can easily manage resources and course enrollments.
4. Instructors can reuse course resources for other sessions they would handle in the feature, this saves them the stress of publishing resources at the beginning of semesters
5. Instructors can track the knowledge of learners by monitoring interaction on the course forum.
6. The proposed system makes it possible for lecture sessions to hold irrective of wheather there's an unforseen circumstance that hinders learners and instructors from being available physically.

3.6 PROGRAMMING LANGUAGE

Programming languages used are listed below;

JavaScript Programming language; According to Flanagan (2021), JavaScript is high-level, often just-in-time compiled and multi-paradigm. It has dynamic typing, prototype-based object-orientation and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Over 97% of websites use it client-side for web page behavior; Javascript was choosen as the primary programming language for both frontend and backend development.

Dart Programming Language is a language developed and maintained by google; this language was created for the pupose of developing mobile friendly applications which can be compiled for both the Android and IOS operating systems. This makes it possible for developers to build applications for several operating systems using just one language; this increases development time and developers efficiency.

HyperText Markup Language (HTML) is the standard language every website is made up of. It is usually used alongside CSS and Javascript progamming language to produce more functionalies.

HTML was choosed as the primary language to be used in building the website aspect of the system as it provides all that is needed to build a website.

Cascading Style Sheets (CSS) is the language used to style web pages; CSS was choosen as the primary language to be used in styleing basic components of the VLLP website.

3.7 LANGUAGE JUSTIFICATION

JavaScrript was choosed as the primary programming language for this project as it brings about somany advantagees. Looking at the fact that JavaScript can be used for backend and frontend development, it makes development faster as one langauge is being used accross major application modules. The Dart programming language was choosed to develop the mobile application as it poses several advanges. According to dart.dev (2021), 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. HTML as the standard language for developing websites was used to build the platforms website.

According to Lozinski (2021), Uber Technologies uses JavaScript for frontend development including other programming languages such as Python, Go and Java. According to Ironhack (2021), LinkedIn relies on NodeJS for its mobile site; LinkedIn switched over to NodeJS to solve its scaling problems as the previous technology stack used (Rails application) was slow and hardly scalable.

3.8 CONCLUSION

This chapter summarised the structure and components of the Virtual Library Learning Platform, described the variouse languages used and justification for use and stated the requirement for systme use.

References

Dart.dev, (2021). Dart overview. Available at: https://dart.dev/overview.html

Chen, L (2018). *Microservices: Architecting for Continuous Delivery and DevOps.* Available at: https://www.researchgate.net/publication/323944215.

Flanagan, D. (2021). *JavaScript the definitive guide.* Available at: https://www.worldcat.org/title/javascript-the-definitive-guide/oclc/64313259

Reenskaug, T. & Coplien, J.O. (20 March 2009). The DCI Architecture: A New Vision of Object-Oriented Programming. Available at: https://web.archive.org/web/20090323032904/https://www.artima.com/articles/dci_vision.html

Wolff, E (2016). Microservices: Flexible Software Architectures. ISBN 978-0134602417. Available at: https://en.wikipedia.org/wiki/Special:BookSources/978-0134602417The Uber

IronHack. (2021). How 10 Major Companies Are Using JavaScript. Available at: https://www.ironhack.com/en/web-development/10-major-companies-avascript

Lozinski, L. (2016). Engineering Tech Stack, Part II: The Edge and Beyond. Available at: https://eng.uber.com/uber-tech-stack-part-two/