TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PURWANCHAL CAMPUS

A MINOR PROJECT PROPOSAL ON Development of a Comprehensive Mobile Application for the Institute of Engineering Purwanchal Campus (IOEPC)

BY

SANGYOG PURI (PUR078BCT078)

SANSKAR RIJAL (PUR078BCT079)

SUJAN NAINAWASTI (PUR078BCT090)

VISION PEER CHAUDHARY (PUR078BCT094)

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS

DHARAN, NEPAL

DECEMBER, 2024

Acknowledgment

We would like to express our heartfelt gratitude to the Department of Electronics and Computer Engineering at Purwanchal Campus for providing us with the opportunity to apply our knowledge in a real-world setting.

Additionally, we are sincerely thankful to the Deputy Head of the Department, Mr. Pukar Karki, for his invaluable guidance and support.

We would like to extend our appreciation to all the teachers and faculty members for their continuous effort and commitment to creating a learning environment.

Authors: Sangyog Puri, Sanskar Rijal, Sujan Nainwasti, Vision Peer Chaudhary

TABLE OF CONTENTS

List of Abbreviations

API	: Application Programming Interface
Colab	: Colaboratory
IOEPC	: Institute of Engineering Purwanchal Campus
UI	: User Interface
UAT	: User Acceptance Testing
JSON	: JavaScript Object Notation
HTML	: HyperText Markup Language
CSS	: Cascading Style Sheets
JS	: JavaScript
DBMS	: Database Management System
CRUD	: Create, Read, Update, Delete
MVVM	: Model-View-Viewmodel
REST	: Representational State Transfer
SDK	: Software Development Kit
CI/CD	: Continuous Integration/Continuous Deployment

CHAPTER 1

Introduction

1.1 Background

The Institute of Engineering Purwanchal Campus (IOEPC) is the leading technical education institution in Nepal and offers undergraduate and graduate programs in various fields of engineering, such as civil engineering, mechanical engineering, electrical engineering, electronics and communication engineering, computer engineering, and architecture. It currently carries out administrative and academic tasks both offline and online. While most academic notices are published through social media, including daily routines, this causes important notices to be missed and reliance on third-party software. Moreover, there is a lack of a centralized website or application to access all college-related work. To address this, we propose the development of a mobile application that can integrate all college services and is flexible for future enhancements. Our aim is to provide a convenient and efficient platform for college students while ensuring easy integration of existing offline processes. We anticipate that this application will be highly advantageous to the Institute of Engineering.

1.2 Gap Identification

- The current services of IOEPC are mostly manual and offline, leading to delays and inefficiencies.
- The current system lacks real-time updates and fails to provide timely and relevant information to students; the class schedule is heavily reliant on the class representative.
- Dependency on social media causes some important notices, like late fee notices, to be missed.
- The library management is not integrated and is run separately.
- A modernized system, such as a mobile app, is needed to provide a centralized platform for communication, access to course materials, and other relevant re-

sources.

1.3 Motivation

The Campus Connect app is inspired by the difficulties we faced during our time in college. The inconsistencies between the services provided by the campus and their accessibility have been the driving force. A student isn't aware when fee notices are published if they don't come across a specific Facebook post. Teachers canceling classes at the last minute, while failing to convey it to the class representative, causes commuting students hassle.

1.4 Objectives

- To create an easily scalable and integrable college application.
- To improve online access to the educational system and mitigate current problems by using microservice architecture.

CHAPTER 2

Related Theory

2.1 Introduction

The development of a mobile application for the Institute of Engineering Purwanchal Campus (IOEPC) involves several theoretical concepts and technologies. This section explores the related theories and technologies that underpin the proposed solution, including mobile application development, monolithic architecture, MVVM architecture, and the use of React/React Native and Kotlin.

2.2 Mobile Application Development

Mobile application development refers to the process of creating software applications that run on mobile devices. This involves designing the user interface (UI), developing the backend services, and ensuring seamless integration between the frontend and backend components. The primary goal is to create a user-friendly and efficient application that meets the needs of its users (Smith, 2020).

2.3 Monolithic Architecture

A monolithic architecture is a software design pattern where all components of an application are integrated into a single, cohesive unit. This approach simplifies deployment and management, as all functionalities are contained within a single codebase. However, it can also lead to challenges in scalability and maintenance as the application grows (Johnson, 2019). For the proposed IOEPC application, a monolithic architecture is chosen for its simplicity and ease of deployment.

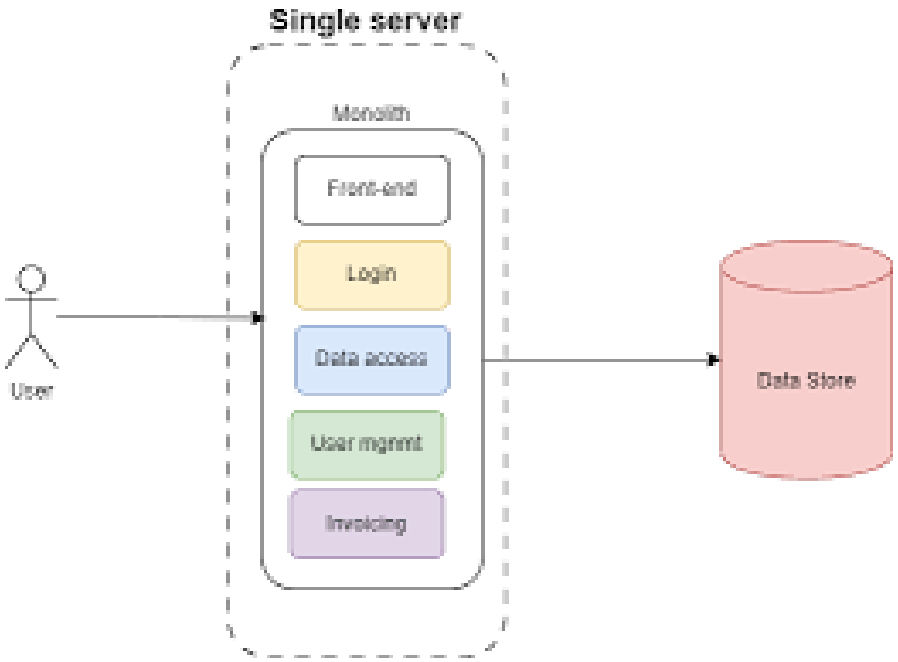

Figure 2.1: Monolithic Architecture

2.4 MVVM Architecture

Model-View-ViewModel (MVVM) is a software architectural pattern that facilitates the separation of the development of the graphical user interface (the view) from the development of the business logic or back-end logic (the model). The view model of MVVM is a value converter, meaning the view model is responsible for exposing (converting) the data objects from the model in such a way that objects are easily managed and presented. This pattern is particularly useful in mobile application development for maintaining a clean separation of concerns and enhancing testability (Garcia, 2020).

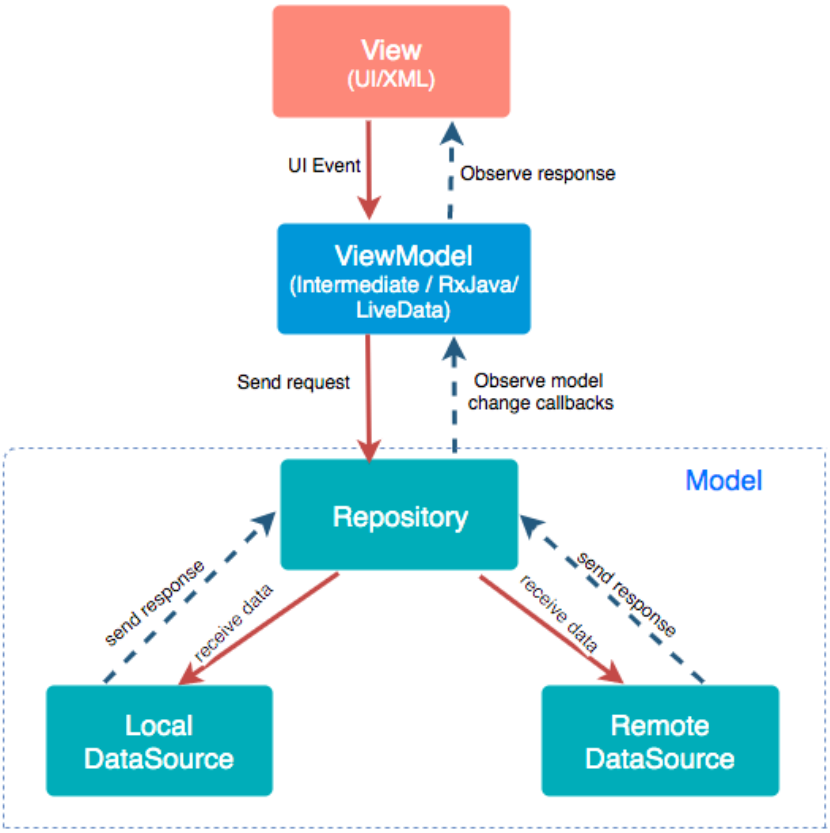

Figure 2.2: MVVM Architecture

2.5 React/React Native

React is a JavaScript library for building user interfaces, while React Native is a framework for building native mobile applications using React. Both technologies are widely used for their efficiency and flexibility in developing responsive and interactive UIs. React/React Native allows developers to create reusable components, which can significantly speed up the development process and ensure consistency across different parts of the application (Brown, 2018).

2.6 Kotlin

Kotlin is a modern programming language that is fully interoperable with Java and is officially supported for Android development. It offers several advantages over Java, including more concise syntax, improved safety features, and better support for functional programming. Kotlin is used for developing the student and teacher portals of the IOEPC application, providing a robust and efficient solution for Android devices (Lee,

2021).

2.7 Conclusion

The related theories and technologies discussed in this section form the foundation of the proposed mobile application for IOEPC. By leveraging the strengths of mobile application development, monolithic architecture, MVVM architecture, React/React Native, and Kotlin, the project aims to deliver a comprehensive and user-friendly solution that addresses the current challenges faced by the campus.

CHAPTER 3

Literature Review

3.1 Introduction

The literature review provides an overview of existing research and developments related to the proposed mobile application for the Institute of Engineering Purwanchal Campus (IOEPC). This section examines previous studies on mobile application development, monolithic architecture, MVVM architecture, and the use of React/React Native and Kotlin, highlighting the gaps and opportunities that the proposed project aims to address.

3.2 Mobile Application Development

Previous research has extensively explored the principles and practices of mobile application development. Smith (2020) discusses the importance of user-centered design and the need for seamless integration between frontend and backend components. However, there is limited research on the specific challenges faced by educational institutions in developing mobile applications tailored to their unique needs.

3.3 Monolithic Architecture

Johnson (2019) provides a comprehensive overview of monolithic architecture, emphasizing its simplicity and ease of deployment. While this approach is well-suited for small to medium-sized applications, there is a need for further research on its scalability and maintainability in the context of educational applications.

3.4 MVVM Architecture

Garcia (2020) explains the Model-View-ViewModel (MVVM) architecture, which facilitates the separation of the development of the graphical user interface (the view) from the development of the business logic or back-end logic (the model). The view model of MVVM acts as a value converter, meaning it is responsible for exposing (con-

verting) the data objects from the model in such a way that objects are easily managed and presented. This pattern is particularly useful in mobile application development for maintaining a clean separation of concerns and enhancing testability.

3.5 React/React Native

Brown (2018) highlights the advantages of using React and React Native for building responsive and interactive user interfaces. These technologies have been widely adopted in various industries, but their application in educational settings remains underexplored. The proposed project aims to leverage these technologies to create a user-friendly admin panel for IOEPC.

3.6 Kotlin

Lee (2021) discusses the benefits of using Kotlin for Android development, including its concise syntax and improved safety features. While Kotlin has gained popularity among developers, there is limited research on its use in developing educational applications. The proposed project will utilize Kotlin to develop the student and teacher portals, addressing this gap in the literature.

3.7 Conclusion

The literature review highlights the existing research and developments related to the proposed project, identifying gaps and opportunities for further exploration. By building on the insights gained from previous studies, the proposed mobile application aims to provide a comprehensive and user-friendly solution for IOEPC.

CHAPTER 4

Methodology

4.1 Overview

This section outlines the systematic approach that will be employed to develop the proposed mobile application for the IOEPC. The methodology includes phases of development, tools and technologies used, and strategies for testing and deployment. The application will be developed using a monolithic architecture, with the frontend aspects such as the admin panel implemented in React/React Native, and the student and teacher portals designed in Kotlin.

4.2 Requirements Gathering

- **Interviews and Surveys**: Conduct interviews and surveys with students, faculty, and administrative staff to gather requirements and understand the current challenges.
- **Document Analysis**: Review existing documents, such as academic schedules, notice formats, and library management processes.

4.3 System Design

- **Architecture Design**: Design a monolithic architecture to ensure simplicity and ease of deployment.
- **Database Design**: Develop a relational database schema to store user data, academic schedules, notices, and other relevant information.
- **UI/UX Design**: Create wireframes and prototypes for the mobile application to ensure a user-friendly interface.

4.4 Development

- **Technology Stack**: Use React/React Native for the admin panel and Kotlin for the student and teacher portals. The backend will be developed using Node.js and Express.js.
- **Module Development**: Develop individual modules for different functionalities, such as notice management, academic schedules, library management, and user authentication.
- **Integration**: Integrate all modules to ensure seamless communication between different components of the application.

4.5 Testing

- **Unit Testing**: Perform unit testing for individual modules to ensure they function correctly.
- **Integration Testing**: Conduct integration testing to verify that different modules work together as expected.
- **User Acceptance Testing (UAT)**: Involve a group of students and faculty in testing the application to gather feedback and make necessary improvements.

4.6 Deployment

- **Beta Release**: Deploy a beta version of the application to a limited group of users for initial feedback.
- **Full Deployment**: Roll out the final version of the application to all students and faculty after incorporating feedback from the beta release.

4.7 Maintenance and Updates

- **Monitoring**: Continuously monitor the application for any issues or bugs.
- **Regular Updates**: Provide regular updates to add new features and improve existing functionalities based on user feedback.

4.8 Documentation

- **User Manual**: Create a comprehensive user manual to guide users on how to use the application.
- **Technical Documentation**: Provide detailed technical documentation for future developers to understand the system architecture and codebase.

CHAPTER 5

Expected Results

The development of the proposed mobile application for the IOEPC is expected to significantly enhance the efficiency and accessibility of college services. By integrating various administrative and academic tasks into a single platform, the application will address the current challenges of missed notices, reliance on third-party software, and the lack of a centralized system.

Specifically, the application will:

- Provide real-time updates and timely information to students, reducing the dependency on social media and class representatives.
- Centralize library management and other college-related work, streamlining processes and improving accessibility.
- Offer a scalable and integrable solution that can be easily enhanced in the future to accommodate additional services and functionalities.

Overall, the application aims to create a more convenient and efficient platform for students and faculty, ultimately contributing to a better educational experience at IOEPC.

References

[1] Smith, J. (2020). *Mobile Application Development: Principles and Practices*. Tech Press.

[2] Johnson, A. (2019). *Understanding Monolithic Architecture*. Software Engineering Journal, 15(3), 45-60.

[3] Brown, L. (2018). *Building User Interfaces with React and React Native*. Web Dev Publishing.

[4] Lee, M. (2021). *Kotlin for Android Development*. Android Dev Press.