

Department of Computer  
Engineering

Faculty of engineering and  
technology

University of Buea



Département de Génie  
Informatique

Faculté de Génie et de  
Technologie

Université de Buea

---

COURSE CODE: CEF 440

COURSE TITLE: INTERNET AND MOBILE PROGRAMMING

**TASK 1: MOBILE APP DEVELOPMENT PROCESS: ARCHITECTURE,  
FRAMEWORKS, AND COST ANALYSIS**

Presented by:  
**GROUP 16**

NAME	MATRICULE
BELLAH LOVETTE MANYI	FE22A172
TABOT ALISTAR GIFT	FE22A301
NGEIMASHUNG NJUZE RAVINE	FE22A260
ANNE BENITA NKENG FUA	FE22A149
ETA INDIRA	FE22A209

Course instructor:

**Dr Nkemeni Valery**

Date: 30/03/2025

# ABSTRACT

Mobile application development is a rapidly evolving field that requires an understanding of various technologies, frameworks, and methodologies. This task explores the fundamental aspects of mobile app development, starting with a comparison of major app types—native, progressive web apps (PWAs), and hybrid apps—highlighting their strengths, weaknesses, and use cases. It also examines the most commonly used programming languages and development frameworks based on key factors such as performance, complexity, cost, and community support.

Additionally, mobile application architectures and design patterns are analyzed to understand how they impact scalability and maintainability. The study also covers requirement engineering, focusing on techniques for collecting and analyzing user needs to build effective applications.

Finally, the task delves into cost estimation, exploring factors that influence development costs and strategies for budgeting efficiently. This study provides a solid foundation for making informed decisions in mobile app development.

## TABLE OF CONTENT

<b>ABSTRACT .....</b>	<b>2</b>
<b>INTRODUCTION .....</b>	<b>4</b>
<b>METHODOLOGY .....</b>	<b>5</b>
1.Major Types of Mobile Apps and Their Differences .....	5
2. Comparison of Mobile App Programming Languages .....	6
<b>3. Comparison of Mobile App Development Frameworks .....</b>	<b>7</b>
<b>4.Mobile Application Architectures and Design Patterns .....</b>	<b>8</b>
5.Requirement Engineering for Mobile Apps .....	9
Steps: .....	9
<b>6. Estimating Mobile App Development Cost .....</b>	<b>10</b>
<b>CONCLUSION .....</b>	<b>11</b>
<b>REFERENCES .....</b>	<b>11</b>

# INTRODUCTION

In today's digital landscape, mobile applications have become an integral part of daily life, serving various purposes from communication and entertainment to business and productivity. Developing a mobile application requires a thorough understanding of different app types, programming languages, frameworks, architectures, and development methodologies.

This task explores the key foundations of mobile app development, starting with a comparison of major app types—native, progressive web apps (PWAs), and hybrid apps—highlighting their advantages, limitations, and ideal use cases. Finally, we examine mobile app development cost estimation, exploring factors that influence development costs and strategies for budgeting efficiently.

# METHODOLOGY

## 1. Major Types of Mobile Apps and Their Differences

### Native Apps

- Built for specific platforms (iOS or Android) using platform-specific languages (Swift for iOS, Kotlin/Java for Android).
- It provides higher performance, better UX, full access to device features.
- Costly and time-consuming due to separate code-bases for each platform.

### Progressive Web Apps (PWAs)

- These are applications that use modern web capabilities to deliver an app-like experience through browsers. They are built using web technologies (HTML, CSS, JavaScript) with offline capabilities.
- Its Cost-effective but limited access to device features.
- it offer the advantage of a single code base compared to native apps
- They may have a limitation in accessing advanced device functionalities

### Hybrid Apps

- A mix of web and native elements, built using web technologies and wrapped in a native container via frameworks like Ionic
- It provides a faster development, single codebase, but may have performance limitations.
- May have limitations especially in resource intensive task

## 2. Comparison of Mobile App Programming Languages

- **Swift (iOS):** it's an apple device. It offers modern syntax, strong type safety and performance optimizations that make it well suited for building responsive high quality apps. Because it's designed specifically iOS swift applications can fully exploit device capabilities and integrating seamlessly with apple's ecosystem.
- **Kotlin (Android):** it is the preferred language for android development endorsed by google as an official language. It offers concise syntax and null safety features that reduce common programming errors, additionally Kotlin is fully interoperable with java.
- **Java (Android):** it has long been a step for android development. Its mature libraries made it a reliable choice for building android apps. However, compared to Kotlin, java turns to be more bulky and lacks modern features that enhance developer productivity.
- **JavaScript (Hybrid/PWA):** it's a popular choice for developing hybrid and cross-platform mobile apps. It enables developers to leverage a single code base for multiple platforms. It has a vast community and a rich ecosystem which contributes to its appeal for rapid development cycle
- **Dart (Flutter):** it is the language behind the google flutter framework, which is used to build natively compiled for mobile web and desktop from a single code-base. That is ideal for creating virtually rich apps due to its high performance and smooth UI that can perform well across different platforms

### 3. Comparison of Mobile App Development Frameworks

Frameworks are structured platforms or environment that provide a foundation for building applications.

Framework	React Native	Flutter	Ionic	Swift (Native iOS)	Kotlin (Native Android)
Language	JavaScript	Dart	JavaScript	Swift	Kotlin
Performance	Near-native	High	Lower	Best	Best
Cost & Time	Fast, single codebase	Fast, but Dart learning curve	Fast, single codebase	Expensive, separate development	Expensive, separate development
UX & UI	Good, needs optimization	Excellent, custom UI rendering	Decent, web-based UI	Best for iOS	Best for Android
Community Support	Large	Growing fast	Moderate	Large	Large
Use case	Social media apps, E-commerce app	High performance UI centric apps	Content-driven, simple applications	Native iOS apps	Native android apps

## 4.Mobile Application Architectures and

### Design Patterns

Mobile application architectures refer to structured frameworks that define the interaction and organization of various components within a mobile application.

Architecture Types:

- **Monolithic architecture:** This is a single-tiered application where all components are tightly coupled. Simple to develop but difficult to maintain as the app grows
- **Layered architecture(N-tier):** Its divides an application into separate layers:presentation, business logic and data.It introduces performance overhead due to multiple layers. This architecture enhances maintainability and modularity.
- **MVVM (Model-View-View-model):**This architecture separates the UI from the business model which interacts with data.Its popular in mobile development for maintaining a clean testable code base Used in Android and Flutter for better separation of concerns. It also facilitates data binding.
- **Micro service architecture:** The architecture breaks down the application into independent services that communicate through APIs. It improves scalability and flexibility but managing multiple services can increase complexity.

Design Patterns:

This selection depend on factors such as project complexity,scalability and maintainability. The various types include;

**Singleton:** Ensures only one instance of a class. Commonly used for database connections or network managers



**Factory Pattern:** Helps in object creation, it dynamically reduces dependencies in the code.

**Observer Pattern:** Helps in real-time updates (e.g., notifications). It is used in event-driven programming to notify components when data changes

**Adapter patterns:** It helps incompatible Api work together useful for integrating third-party APIs

## 5.Requirement Engineering for Mobile Apps

Requirement engineering is a structural process in software engineering that involves identification,documentation,analysis,validation and management of requirements for a software system. This process ensures that the user final products aligns with the user needs and business

### Steps:

**Requirement Gathering:** This initial phase involves gathering user information through interviews surveys and many more. The goal is to uncover needs and latent requirements that users might not articulate.

**Requirement Analysis :** This phase involves classifying requirements into functional and non-functional requirements. It ensures that only relevant and feasible information moves forward.

**Requirement Documentation:** Once the requirements are analyzed they must be documented in a clear and structured manner which leads in the creation of a Software Requirement Specification (SRS). This document acts as a blueprint for developers and serves as a reference point through out the development cycle.

**Requirement Validation:** Validation involves verifying that the documented requirements accurately fit the needs of the user. Techniques such as prototyping, user acceptance test are employed to ensure that the requirements are both correct and complete ensuring clarity and feasibility.

**Requirement Management:**It is the continuous process of monitoring and controlling changes to the requirements. Effective management ensures that the final product adapts to changing environments and user experience

## 6. Estimating Mobile App Development Cost

**1. App Type and Complexity:** The number and complexity features significantly influence cost. Simple apps with basic functionalities require less time and resources ,whereas apps incorporating advanced features such as real-time data processing,artificial intelligence or integration demand more development hours and specialized expertise.

**2. Platform and technology:** Choosing between native, cross-platform or hybrid development affects both performance and cost. Native development generally offers superior performance but may require separate development for each platform, leading to higher cost. In contrast cross-platform frameworks allow for a single code-base for multiple platforms reducing both development time and cost.

**3. Design and user experience (UX/UI):** Custom designs and user interfaces that require high-end graphics and animations can lead to development cost. Investing in a well-designed UI/UX means additional iterations, user testing which all adds to a projects budget.

**4. Development Team and Location:** The cost structure varies for whether you want to use an in-house team, freelancers or outsource to a development agency. Developers in a higher living cost typically charge more influencing the overall project cost.

Some estimation method include:

- Fixed-price model:**Suitable for projects whose requirements are well defined
- Time and Material model:**Charges based on actual development time and resources used.
- **Detailed estimation:** Involves breaking down the project into smaller components, estimating the cost for each and summing them for a comprehensive budget.

# CONCLUSION

Developing a mobile application involves making important decisions from choosing the right type of app and programming language to deciding on the development framework and architectural patterns. Understanding the requirements and accurately understanding the requirements and estimating cost are also critical to the success of a project.

This report helped us see how different approaches have their own pros and cons, and how careful evaluation of all these factors is necessary to make informed decisions in mobile app development.

# REFERENCES

- Google. "Guide to android app architecture." [developer.android.com, developer.android.com/guide/architecture](https://developer.android.com/guide/architecture)
- Wikipedia. "Application mobile." Wikipedia, The Free Encyclopedia, [en.Wikipedia.org/wiki/Application\\_mobile](https://en.wikipedia.org/wiki/Application_mobile)
- ITechcraft. "Best Programming languages for mobile app development."