PRESENTED BY

EBAN BORIS MANYO
IKOME M.B JOHNSON
AKWUNG GABRIEL
NYENTY OBI RAMSON

Course: Internet programming and mobile programming
CEB402

Instructor: Dr Valery Nkemini

Contents

Report on Mobile Applications Development Process 3
1. Introduction 3
2. Mobile App Programming Languages 3
3. Mobile App Development Frameworks 3
4. Native vs. Cross-Platform Development 4
5. Mobile App Architectures 4
6. Requirement Engineering Process 5
7. Mobile App Design Patterns 5
8.App Development Cost Estimation 5
9 Factors Affecting Cost 6
10. Emerging Trends in Mobile Development 6
11. Conclusion 6

Report on Mobile Applications Development Process

1. Introduction

The mobile app market is experiencing exponential growth, valued at $208.8 billion in 2024, with 255 billion downloads worldwide. This surge is driven by 6.3 billion smartphone users, making mobile applications an integral part of modern life. From entertainment to productivity, apps connect people globally, enhancing convenience and efficiency.

2. Mobile App Programming Languages

Native Development
- iOS: Swift, Objective-C
- Android: Kotlin, Java, react

Advantages:
- Platform-specific optimization
- High performance
- Full access to device features

Cross-Platform Development
- JavaScript (React Native)
- Dart (Flutter)
- C# (Xamarin)

Advantages:
- Single codebase for multiple platforms
- Faster development
- Cost-effective

Trade-off: Cross-platform apps may sacrifice some performance compared to native apps.

3. Mobile App Development Frameworks

Flutter (Google)
- Uses Dart
- Fast development with expressive UI
- Hot reload feature for quick iterations

React Native (Facebook)
- Based on JavaScript
- Near-native performance
- Large community support

Xamarin (Microsoft)
- Uses C#
- Strong integration with Microsoft tools
- Robust performance

Key Benefit: These frameworks reduce development time and promote code reuse.

4. Native vs. Cross-Platform Development

Aspect	Native	Cross-Platform
- Performance -	- High (optimized for platform)	- Slightly lower (due to abstraction)
- Development Time	- Longer (separate codebases)	- Faster (single codebase)
- Cost	-Higher (platform-specific skills)	-Lower (shared resources)
- Device Access	- Full hardware integration	- Limited by framework capabilities
- Device Access	Android IOS	IOS Android

5. Mobile App Architectures

MVC (Model-View-Controller)

- Separates data (Model), UI (View), and logic (Controller).
- Simple but can lead to bloated controllers.

MVVM (Model-View-ViewModel)
- Enhances testability with data binding.
- Popular in Android (Jetpack Compose) and iOS (SwiftUI).

Clean Architecture
- Focuses on separation of concerns.
- Independent layers for scalability.

6. Requirement Engineering Process

1. Research – Identify target audience needs.
2. Define Requirements – Functional (features) & non-functional (performance, security).
3. Prototyping – Create mockups for feedback.
4. Iterative Refinement– Adjust based on user testing.

7. Mobile App Design Patterns

- A single tone should be use
- Simplify user interface
- User can upload information

8.App Development Cost Estimation

App Type	Cost Range	Features
Mobile disaster (protect)	300,000fca	- Connected to fire sensor - earthquake sensors - Flood sensor - weather forecast

9 Factors Affecting Cost

- Platform (iOS, Android, or both): IOS applications are complex build and time wasting
- Complexity (AI, AR, IoT integrations): some AI features will not work properly with the app
-Accurate planning and budget analysis: This is crucial, to avoid overspending.

10. Emerging Trends in Mobile Development

1. AI Integration – Enhances personalization (e.g., chatbots, recommendation engines).
2. IoT Connectivity – Smart home apps, wearables.
3. Performance Optimization – Faster load times, smoother UX.

Future Outlook: The mobile app market is expected to grow at 7.84% annually, with user experience and security being top priorities.

11. Conclusion

The mobile app industry is evolving rapidly, driven by technological advancements and user demand.

- Native development offers superior performance, while cross-platform is cost-efficient.
- Frameworks like Flutter and React Native accelerate development.
- Proper architecture and design patterns ensure scalability.
- Requirement engineering is critical for success.
- Emerging trends (AI, AR, IoT) will shape the future of mobile apps.