

Mobile Application Development with Flutter

Lecture 1: Introduction to Mobile App Development

Introduction

- What is Mobile Application Development?
 - **Mobile Application Development** is the process of designing, building, testing, and deploying software applications that run on **mobile devices** such as smartphones and tablets
- These applications are designed to:
 - Run on **mobile operating systems**
 - Interact with **touch-based interfaces**
 - Use **mobile hardware** features like camera, GPS, sensors, and internet

Why Mobile Applications Matter Today

- “How many hours do you spend daily on mobile apps?”
- Mobile phones are now **primary computing devices**
- Businesses prefer **mobile-first solutions**
- Almost every service has a mobile app:
 - Banking
 - Education
 - E-Commerce
 - Healthcare
 - Entertainment

Examples of Mobile Applications

- **Utility Apps:** Calculator, Flashlight
- **Social Apps:** WhatsApp, Instagram
- **E-commerce Apps:** Amazon, Daraz
- **Educational Apps:** Coursera, Duolingo
- **Business Apps:** POS systems, inventory apps
- **Gaming Apps:** PUBG, Candy Crush
- **Health & Fitness:** MyFitnessPal, Fitbit

Mobile Operating Systems Overview

A **Mobile OS** is system software that:

- Manages mobile hardware
- Provides services to applications
- Acts as a bridge between **user and device**

Android OS

- Developed by **Google**
- Open-source (Linux-based)
- Used by multiple manufacturers (Samsung, Xiaomi, etc.)
- Characteristics:
 - Highly customizable
 - Large user base
 - Wide device variety
 - Apps distributed via **Google Play Store**
- Development Language:
 - Java / Kotlin (native)
 - Dart (Flutter)

iOS

- Developed by **Apple**
- Closed ecosystem
- Runs only on Apple devices (iPhone, iPad)
- Characteristics:
 - High performance
 - Strong security
 - Consistent UI experience
 - Apps distributed via **Apple App Store**
- Development Language:
 - Swift / Objective-C (native)
 - Dart (Flutter)

Android vs iOS

Feature	Android	iOS
Ownership	Google	Apple
Open/Closed	Open-source	Closed
Devices	Multiple brands	Apple only
Market Share	Higher globally	Higher revenue
App Approval	Flexible	Strict

1. Open Source

- Open source refers to software whose source code is publicly available
- Anyone can view, study, modify, and distribute the code according to its license
- Key Characteristics:
 - Source code is open to the public
 - Developers can:
 - Study how it works
 - Modify it
 - Improve it
 - Build their own version
 - Encourages collaboration and innovation

Example: Android (Open Source)

- Android is based on AOSP (Android Open Source Project)
- Manufacturers like:
 - Samsung
 - Xiaomi
 - Oppo
- modify Android for their devices
- “Android is open source, which is why it runs on many different devices.”

Advantages and Disadvantages

- Advantages of Open Source

- ✓ Freedom to modify
- ✓ Large developer community
- ✓ Faster innovation
- ✓ Lower cost
- ✓ Transparency & security (bugs visible)

- Disadvantages of Open Source

- ✗ Less control over quality
- ✗ Fragmentation (many versions)
- ✗ Security depends on community updates

2. Closed Ecosystem

- A **closed ecosystem** is a software environment where:
 - The **source code** is not publicly available
 - Only the **owning company** can modify or control it
 - Hardware, software, and services are **tightly integrated**
- **Key Characteristics**
 - Source code is **private**
 - Strict rules for:
 - App development
 - App distribution
 - Hardware compatibility
 - Strong control over user experience

Example: iOS (Closed Ecosystem)

- iOS is owned and controlled by **Apple**
- Runs only on:
 - iPhone
 - iPad
- Developers must:
 - Follow Apple guidelines
 - Get approval before publishing apps
- “Apple controls everything — hardware, software, and app store.”

Advantages and Disadvantages

- Advantages of Closed Ecosystem

- ✓ Better performance optimization
- ✓ Strong security
- ✓ Consistent user experience
- ✓ Less fragmentation

- Disadvantages of Closed Ecosystem

- ✗ Less flexibility
- ✗ Higher cost
- ✗ Restricted customization
- ✗ Vendor lock-in

Paths to Mobile Development

- There are three path to mobile app development
 1. Native Application Development
 2. Hybrid Application Development
 3. Cross-Platform Application Development

1. Native Applications

- Apps developed **specifically for one platform** using its official tools and languages
- Examples:
 - Android → Java / Kotlin
 - iOS → Swift
- Advantages:
 - ✓ High performance
 - ✓ Full hardware access
 - ✓ Best user experience
- Disadvantages
 - ✗ Separate codebases
 - ✗ Higher development cost
 - ✗ More time required

2. Hybrid Applications

- Apps built using **web technologies** (HTML, CSS, JavaScript) and wrapped inside a mobile container
- Examples:
 - Cordova
 - React Native
 - Ionic
- Advantages:
 - ✓ Single codebase
 - ✓ Faster development
- Disadvantages
 - ✗ Lower performance
 - ✗ Limited native access
 - ✗ UI not truly native

3. Cross-Platform Applications

- Apps developed using a **single codebase** that runs on **multiple platforms**, while still providing near-native performance
- Examples:
 - Flutter
- Advantages:
 - ✓ One codebase → Android + iOS
 - ✓ Faster development
 - ✓ Near-native performance
 - ✓ Lower cost
- Disadvantages
 - ✗ Learning framework required
 - ✗ Slight platform limitations

Comparison Summary Table

Aspect	Native	Hybrid	Cross-Platform
Codebase	Separate	Single	Single
Performance	Best	Low	High
Cost	High	Low	Medium
Industry Use	High	Declining	Rapidly Growing

Flutter gives us the **best balance** between performance and productivity

Industry Trends & Career Paths in Mobile Development

- Cross-platform development is booming
- Flutter is used by:
 - Google
 - Alibaba
 - BMW
 - Startups & freelancers
- Demand for:
 - Mobile UI developers
 - App-based startups
 - Freelance mobile developers

Tools & Technology

- **Framework:** Flutter
- **Language:** Dart
- **Database:** SQLite / Firebase
- **IDE:** VS Code / Android Studio

Prepared by Ehtisham Rasheed