Lecture 2 - Mobile Platforms and Application Development fundamentals

Mobile Application Development

This divide into 4 types

1. Native Mobile Application

   This is an application developed using platform-specific development tools
   We are going to design applications for android, apple devices or windows devices
   This is focus on particular type of platform/OS
   We have to implement the code for numbers of devices

   Example: - Pinterest (This is designed for android and ios different way), ios/android calculator

   Android

   - This is now owned by google (after 2005)
   - Android applications are basically written upon the Linux kernel
   - It is an open source software
   - All most all the android applications are coming under pre –installed applications

   Android Architecture

   - Basically all the applications are running on the linux program
   - Upon linux kernel there are libraries, application framework, applications

   Android versions

   - Android 1.0/1.1: has basic applications like Gmail, google maps (2008)
   - Android 1.5 (cupcake) (2009)
     - This is the beginning of naming pattern
     - Introduce 1st on screen keyboard
   - Android 1.6 (Donut)
     - Supported different screen sizes and different resolutions
   - Android 2.0/2.1 (Eclair)
     - Introduce live wall papers
     - This reach to take functional requirements
   - Android 2.2 (Froyo)
     -
   - Android 2.3 (Gingerbread)
   - Android 3.0/3.1 (Honeycomb)
     - This is coming only for tablets, not for mobiles
     - It has space like holographic design

- Android 4.0 (Ice cream sandwich)
    - Introduce notifications
    - Suggested some reason apps

- Android 4.1/4.2/4.3 (Jelly beans)
    - This had some interact to notifications
    - They introduce videos perspective

- Android 4.4 (KitKat)
    -
- Android 5.0 (Lollipop)
- Android 6.0 (Marshmallow)
- Android 7.0 (Nougat)
    - They introduce an app for organize notification

- 8.0 Orio
    - Introduce picture in picture mode
    - Introduce notification channels (when we typing something they were able to suggest words)

- 9.0 Pie
    - Introduce power management systems
    - And brightness management system (smart phones automatically able to handle brightness)

10.0 (Q)

- Dark mode
- Shortcut facility

11.0 (beta version)

- Enable screen technology
- Pop up videos

Android Development Environments

- Android Studio (modern we use this (latest version is 4.2))
- Eclipse (before using android studio we use this)
- Apache Cordova
- App Inventor for Android
- C++ Builder
- Blue J
- FlashDevelop
- Titanium

Languages

- Java
- Kordin

IOS

- Founders are steve jobs, steve woznik and Ronald Wayne
- Used in apple devices

Development Environments

- Xcode
- AppCode
- Apache Cordova

Windows

- Discontinued family
- Because of people don't like windows smartphones, people like more towards android smart phones encourage windows to discontinued

Development environment

- Visual studio
- Apache Cordova

2. Hybrid Apps

- Usually they are running in web browsers
- We can reuse the code
- Can be share among OS
- Created using codebase using standard web technologies (HTML, CSS, JavaScript)

Examples

- Just watch
- NHS
- Airbus helicopters

Tools

- Ionic
- Visual studio
- Apache Cordova

3. Cross-platform

- It is focus on different types of mobile platforms
- Used to development of mobile apps that can be used on multiple mobile platforms
- We have only one application

- If we write a code using cross-platform way we can apply it in android, ios as well as windows

Development Environments

- Apache Cordova
- PhoneGap
- Xamarine
- Ionic
- Framework 7
- React Native
- Jasonette

Advantages

Native Mobile Application	Hybrid	Cross-platform
Highly perform well		Control cost
User friendly		Can reuse the code
Platform integration problems will not occur		We have only one application
UIs' are common		No Need to higher many no of employees
Uniqueness is there		Quicker development time
		Easy to implement
		Easy to implement the update
		User can easily operate the device

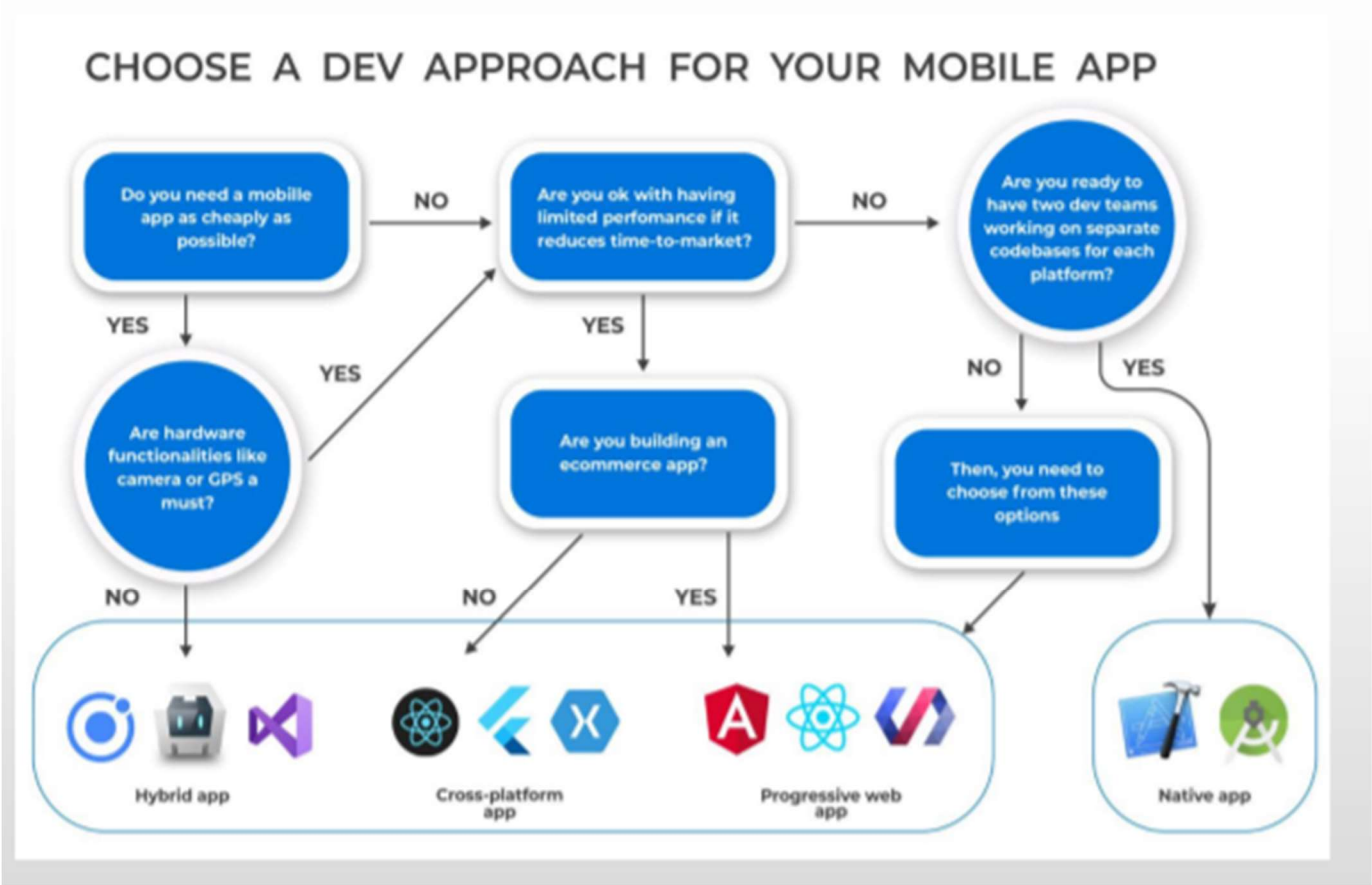
Disadvantages

Native Mobile Application	Hybrid	Cross-platform
High cost		Not much perform well

We have to implement the code for numbers of devices		Loss flexibility
Cannot reuse the code		Problems in platform integration
Need to higher many no of employees		UI may change in each platform
		May not be user friendly
		Uniqueness of each platforms are different
		Difficult to satisfy users

App Type	Native	Hybrid	Cross-platform
Pros	• Full access to device's/ OS's features • Powerful performance • Native UI (updating along with the OS) • Efficient App Running • High-quality functionality and UX • Access to all native APIs and the platform-specific functionality	• Lower development cost • Different OS support • Code reuse • Cost effective development • Big customization capabilities	• Different OS support • UI performance is almost as fast as native • Code reuse • Cost-effective development
Cons	• No multi-platform support • High dev cost if different OS support is needed • No code reuse	• Slower performance • Limited access to OS features • No interaction with other native apps	• *Slower performance • Limited access to OS features • Poor interaction with other native apps

App Type	Native	Hybrid	Cross-platform
Tools	• XCode • AppCode • Android Studio	• Ionic • Apache Cordova • Visual Studio	• React Native • Xamarin • Flutter
Rendering Engine	Native	Browser	Native
Libraries	Not much dependency on open-source libraries and platforms	Highly dependent on different libraries and frameworks	Highly dependent on different libraries and frameworks
Debugging	Native debugging tools	Native + web development debugging tools	Depends on the framework
Codebase	Separate codebase – one per platform	Single codebase with potential platform-specific abilities	Single codebase with potential platform-specific abilities


CHOOSE A DEV APPROACH FOR YOUR MOBILE APP
Do you need a mobille app as cheaply as possible?
NO
Are you ok with having limited perfomance if it reduces time-to-market?
NO
Are you ready to have two dev teams working on separate codebases for each platform?
YES
YES
YES
NO
YES
Are hardware functionalities like camera or GPS a must?
Are you building an ecommerce app?
Then, you need to choose from these options
NO
NO
YES
Hybrid app
Cross-platform app
Progressive web app
Native app