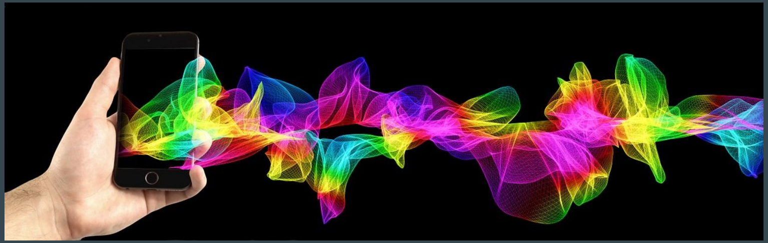
Mobile Application Development

Introduction

Mobile Application Development

- Each day thousands of mobile apps are published to the Google Play and Apple App Stores
- Must understand that app development is not just about coding
- In 2016, mobile and tablet devices accounted for a higher percentage of internet usage than desktop
- Mobile app development process typically includes
  - Idea
  - Strategy
  - Design
  - Development and testing
  - Deployment



Mobile Application Development - Idea

- All great apps has a good idea behind
- What problem exists?
- What is the potential solution?
- Why nobody else has made an app?
- Is it even possible?
- What your app is doing?

Remember that there are thousands and thousands apps in Markets, maybe it is not good idea to create same kind of app!

Mobile Application Development - Strategy

- After idea, you need to plan for your app’s success
- Check similar apps
  - How many
  - Number of installs
  - Ratings and reviews (cons and pros)
- “Just a developer” -> Make money, and how? Free with ads or paid app?
- How marketing your app?
- Plan features and how to update, not all good features at once, people wants a new features time to time

Again: Remember that there is thousands and thousands apps in Markets, maybe it is not good idea to create same kind of app!

Mobile Application Development - Design

- List your app features
- List what will be needed to display in app
- Communicate and share your idea to the client
- Wireframes, Workflows, UI Design
  - Create screens, think functions and data too
  - Think all things what user is able to do in your app
  - Color themes (dark, white, ...), fonts, images, etc...
  - Changes not cost here, later it will!
  - Tools: Sketch, InVision Studio, Adobe XD, Balsamiq, MockFlow, NinjaMock, FluidUI, ...
- Get your app to the market faster

Mobile Application Development - Programming and Testing

- There are many technologies and programing languages to build a mobile app
- Native
  - Apps are written separately for each mobile platform
  - Can’t share code
  - Optimized, work fluidly
  - UI can look entirely native
- Cross-platform Native
  - Some or all code shared
  - Run natively
- Hybrid
  - Usually build with HTML/CSS/JavaScript
  - Installed with native wrapper
  - Usually WebView based

Testing

- Functional
- Usability
- Performance
- Device specific
- End user

Ensure app is bug free and it works in different mobile platforms

Native: Android Application Development

- The platform site for Android: https://www.android.com/
- The official site for Android app developers: https://developer.android.com/
- Applications are created for devices running the Android operating system
- Apps can be written using Kotlin, Java, and C++ languages or using some cross-platform Third-party development tools
- Apps are created with Android Studio
- Target: Phone, Tablet, Wear OS, TV, Auto, Things, Chrome OS, ...

Native: iOS Application Development

- The platform site for Apple: https://www.apple.com/
- The official site for iOS app developers: https://developer.apple.com/
- Applications are created for devices running the iOS operating system
- Apps can be written using Objective-C and Swift languages or using some cross-platform Third-party development tools
- Apps are created with Xcode
- Target: iPhone, iPad, Apple Watch, Apple TV, ...

Cross-platform Application Development

- Cross-platform development tools “promise” that developer can build codebase once, and then run the app on any platform
- Developers get to use the tools and languages they know to build apps for platforms they are not familiar with
- Well-know tools: React Native, NativeScript, PhoneGap, Qt, PWA, Flutter, Xamarin, Ionic, Framework 7, Mobile Angular UI, etc...
- Can be native or hybrid based
- The cost of developing native apps for both (iOS and Android) platforms is rising, so there is place for Cross-platform tools (they have now become mainstream)

Native vs Cross-Platform

Native	Cross-Platform
+ High performance	+ Shorter development time
+ Excellent user experience	+ Code reusability
+ Broad functionality	+ Cost effectiveness
+ Better/familiar UI	+ UI components
+ App store visibility	+ Easy learn languages
- Development cost, slow	- Limited feature support
- Target only one store	- Performance issues
- ...	- UI might be different
	- ...

Mobile Application Development - Deployment

- Most mobile apps require a server back-end to function
  - Transfer data to/from app
  - Remember that your app wont work, if server is not working!
- App Stores
  - Google Play, App Store
  - You will need a lot of different marketing materials (text, images, videos, etc...)
  - Publish updates
- Monitoring
  - Crashes
  - Feedbacks
  - Analytics
  - Performance