

Chương 1

IOS

1 Nền tảng iOS và các thành phần cơ bản

1.1 Kiến trúc tầng của iOS

Chèn cái ảnh vào vào iOS có kiến trúc nhiều tầng, mỗi tầng cung cấp các framework và dịch vụ khác nhau:

Cocoa Touch: Tầng cao nhất, cung cấp các framework cốt lõi như UIKit và SwiftUI cho phát triển giao diện người dùng.

Media: Chứa các công nghệ đồ họa, âm thanh và video như Core Graphics, Core Animation, AVFoundation.

Core Services: Cung cấp các dịch vụ cơ bản như Core Data, Core Location, và Foundation framework.

Core OS: Tầng thấp nhất, bao gồm kernel, file system, bảo mật, và các dịch vụ hệ thống cấp thấp.

1.2 Vòng đời ứng dụng IOS

Khi user mở điện thoại của mình lên thì không có ứng dụng nào chạy cả ngoài những thứ nằm trong Operation system . Application của bạn cũng sẽ không chạy. Sau khi user nhấn vào icon của app, Springboard sẽ kích hoạt application của bạn. Application cùng với các thư viện của nó sẽ được thực thi và được tải vào bộ nhớ, trong khi đó thì Springboard sẽ nhận nhiệm vụ hiển thị màn hình launch screen của ứng dụng. Sau cùng thì ứng dụng của bạn bắt đầu được chạy và application delegate sẽ nhận được các notification. Các iOS app chạy trên các thiết bị đều có các trạng thái chuyển đổi như: Not running, In active, Active, Background,

Suspended. Tại bất kì thời điểm nào, app của bạn đều rơi vào các trạng thái trên.
chèn ảnh vào

NotRunning: Ứng dụng chưa được khởi chạy hoặc đã bị hệ thống chấm dứt.

Inactive: Ứng dụng đang chạy ở foreground nhưng không nhận events (ví dụ: khi có cuộc gọi đến).

Active: Trạng thái bình thường khi ứng dụng chạy ở foreground và đang xử lý events.

Background: Ứng dụng không hiển thị nhưng vẫn chạy và thực thi mã.

Suspended: Ứng dụng ở background nhưng không chạy mã, có thể bị hệ thống chấm dứt để giải phóng tài nguyên.

- 1.3 App Delegate và Scene Delegate
- 1.4 Luồng làm việc của người dùng và Storyboards
- 2 Mô hình kiến trúc ứng dụng iOS**
 - 2.1 Model-View-Controller (MVC)
 - 2.2 Model-View-Presenter (MVP)
 - 2.3 Model-View-ViewModel (MVVM)
 - 2.4 Model-View-Controller-ViewState (MVCVS)
 - 2.5 Viper
 - 2.6 The ELM Architecture (TEA)
 - 2.7 Coordinator Patterns
- 3 Quản lý trạng thái trong ứng dụng iOS**
 - 3.1 Cách tiếp cận quản lý trạng thái
 - 3.2 State Containers
- 4 Quản lý phụ thuộc**
 - 4.1 Các loại Dependency Injection
 - 4.2 DI Containers
- 5 Cách thiết kế giao diện người dùng**
 - 5.1 UIKit
 - 5.2 SwiftUI
 - 5.3 So sánh UIKit và SwiftUI
- 6 Xử lý dữ liệu và lưu trữ**
 - 6.1 Core Data
 - 6.2 Realm
 - 6.3 SQLite
 - 6.4 UserDefaults