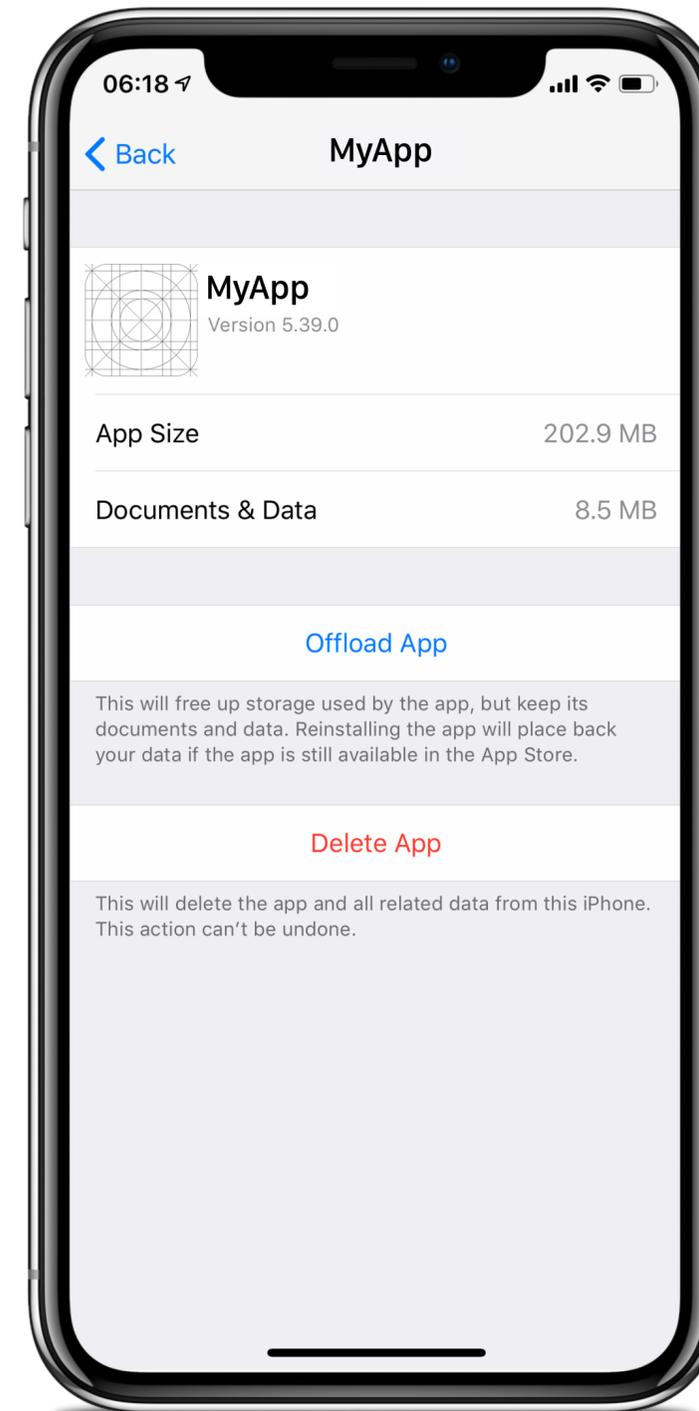
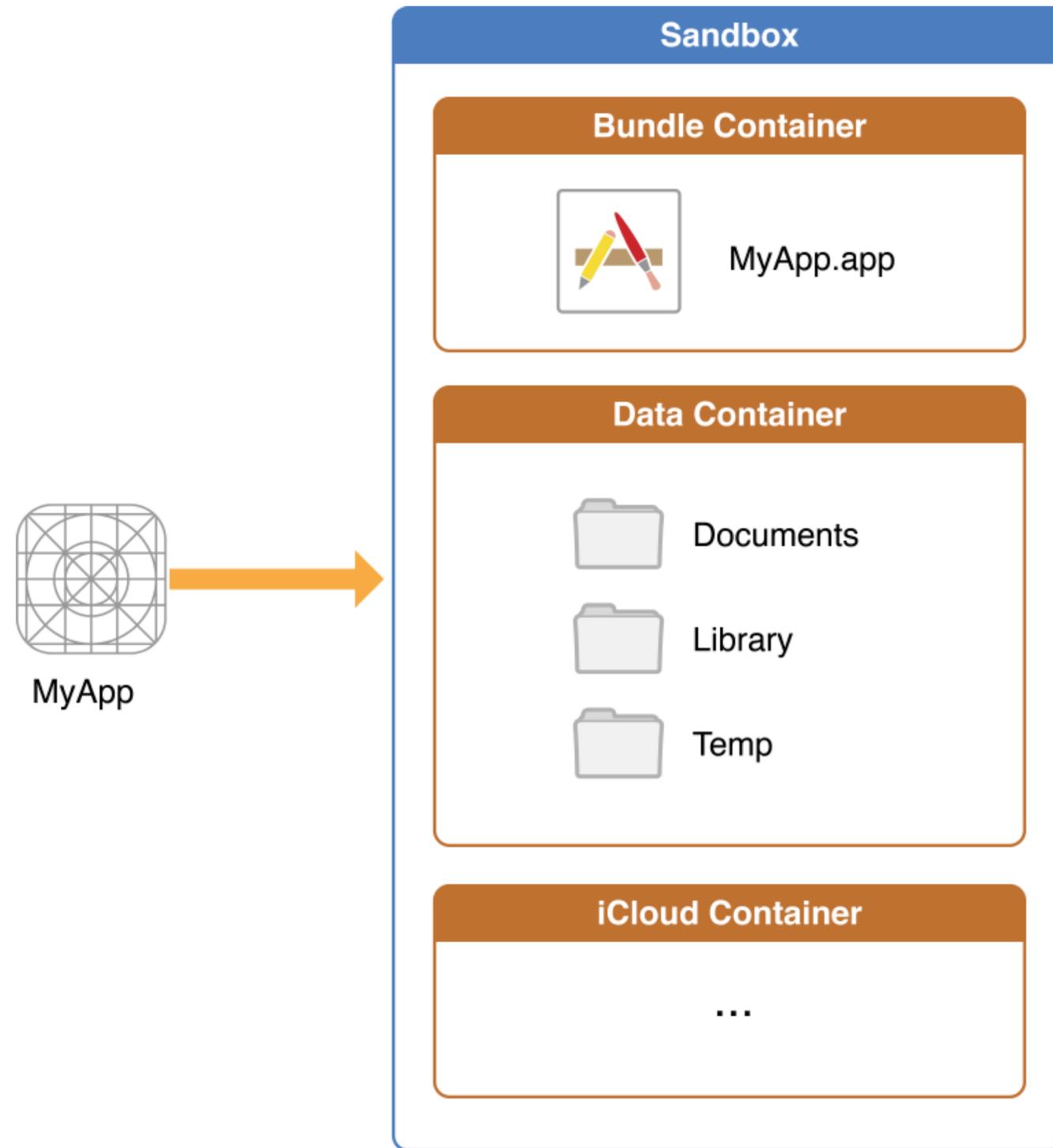


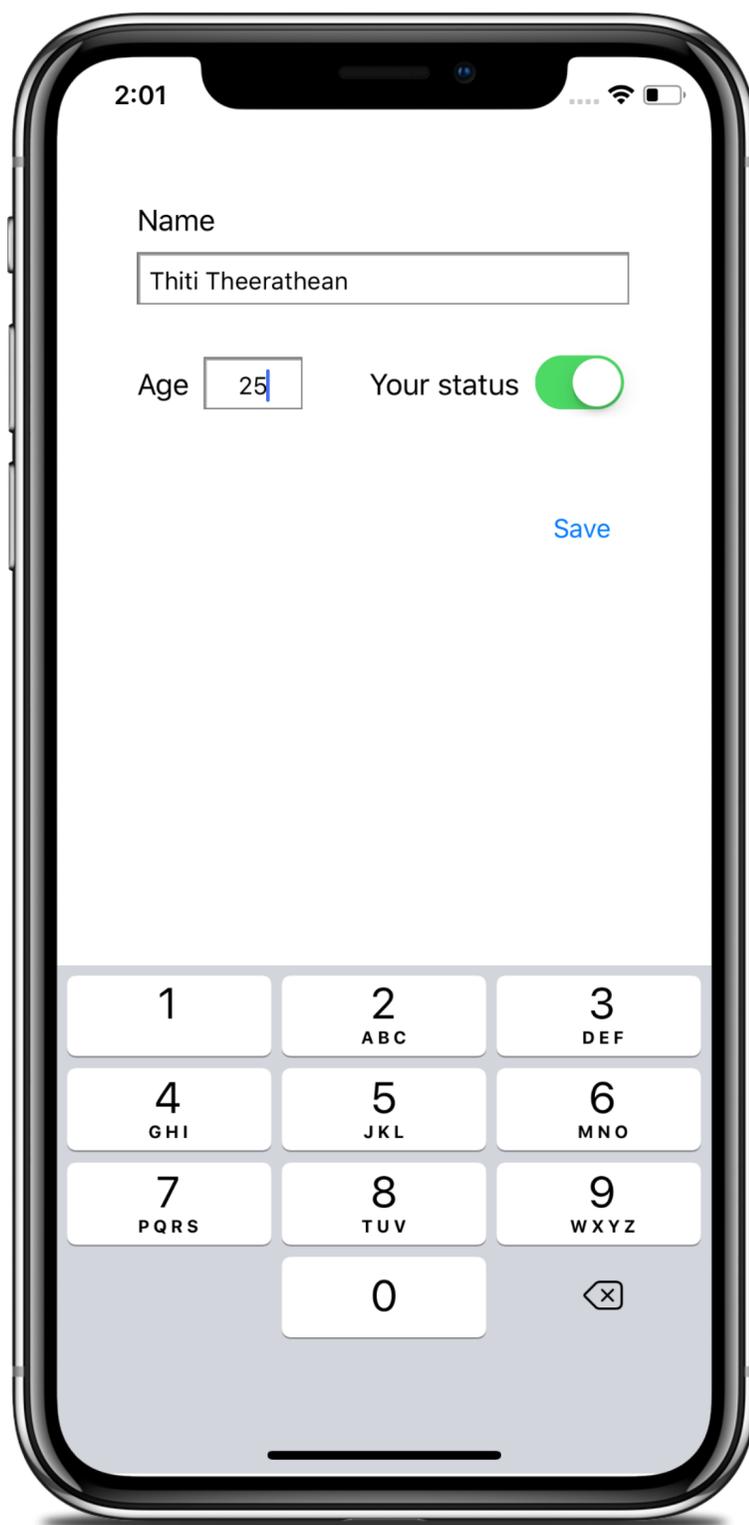


UserDefaults

โดย ธิติ ธีระเรียม

Apple Certified Trainer (App Development with Swift)





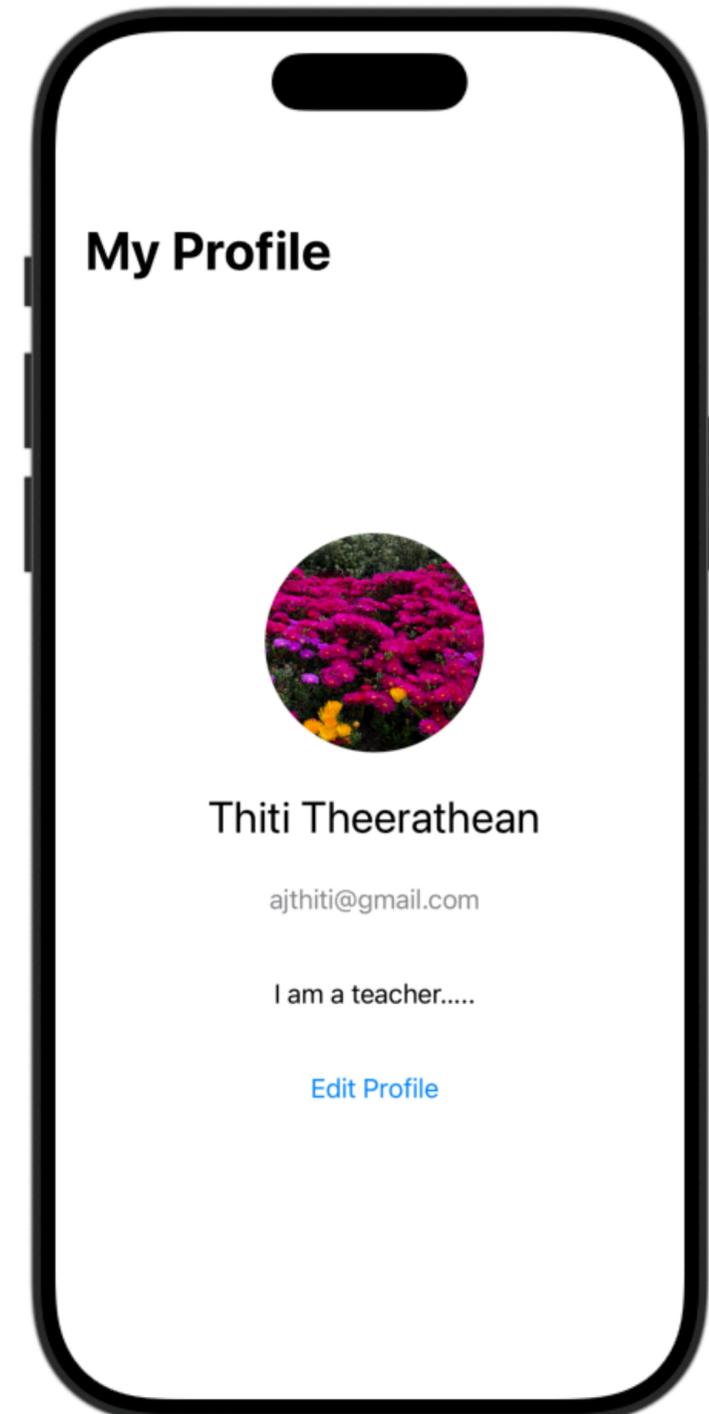
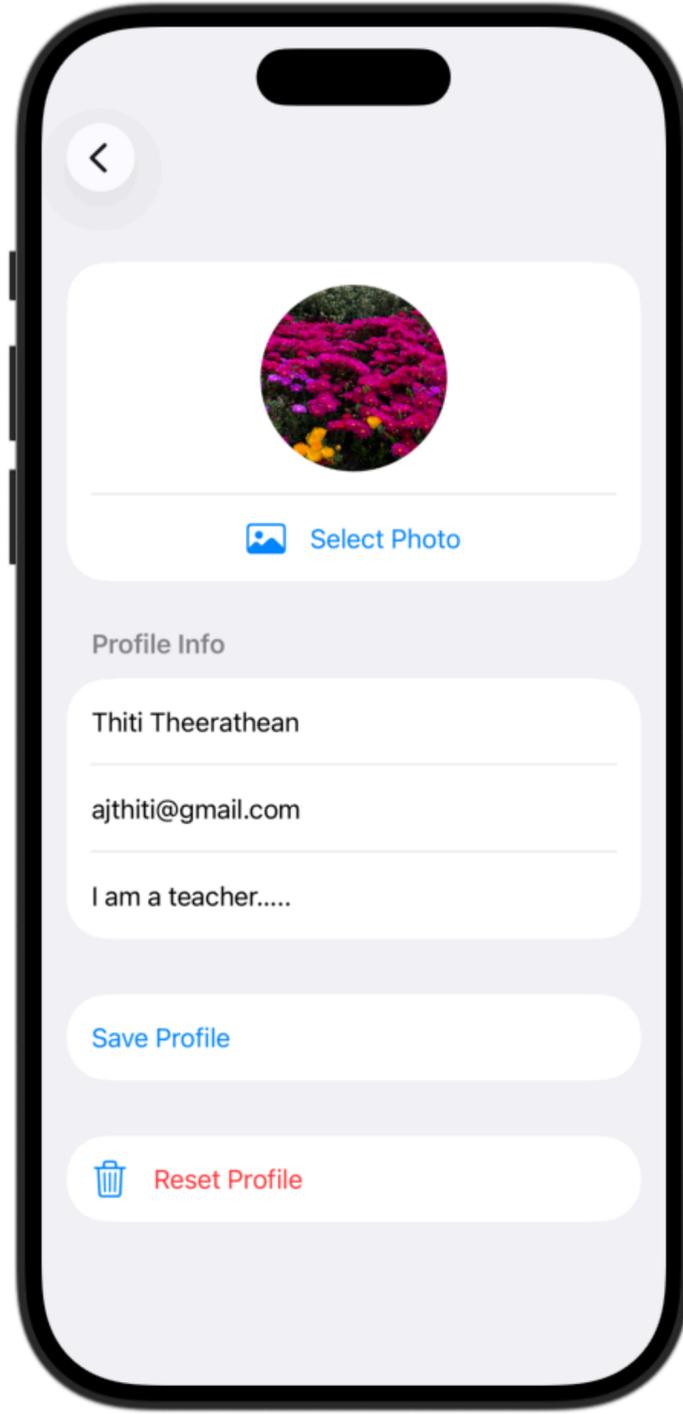
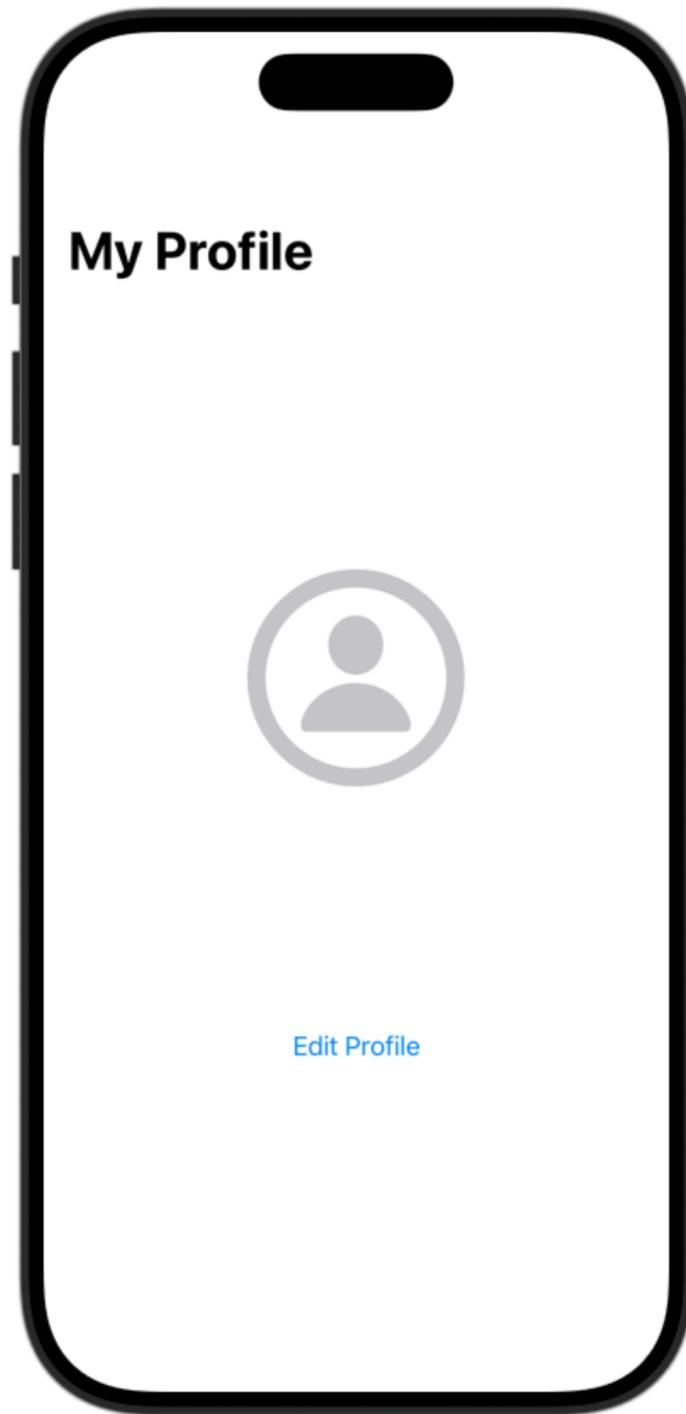
```
// Create an instance of UserDefaults
let myDataStore = UserDefaults.standard

// Storing values
myDataStore.setValue("Thiti", forKey: "myName")
myDataStore.setValue(39, forKey: "myAge")
myDataStore.setValue(true, forKey: "myBool")

let myFriends = ["Bob", "John", "Anny"]
myDataStore.setValue(myFriends, forKey: "myFriends")

// Retrieving Data
let getName = myDataStore.value(forKey: "myName") as! String
print(getName)
let getAge = myDataStore.value(forKey: "myAge") as! Int
print(getAge)
let getBool = myDataStore.value(forKey: "myBool") as! Bool
print(getBool)

let getFriends = myDataStore.value(forKey: "myFriends") as! [String]
print(getFriends)
```



UserProfile.swift

```
import Foundation

struct UserProfile: Codable {
    var name: String
    var email: String
    var bio: String
    var avatarData: Data?
}
```

Data Model กลางของระบบ และใช้การเข้ารหัส/ถอดรหัสเพื่อเก็บใน UserDefaults

UserDefaultsManager ทำหน้าที่เป็น Persistence Layer ของแอป โดยแยกตรรกะการอ่านและเขียนข้อมูลออกจาก ViewModel เพื่อให้โครงสร้างของระบบมีความชัดเจนและลดการพึ่งพาระหว่างส่วนประกอบภายในคลาสรองรับการจัดการข้อมูลพื้นฐาน ได้แก่ การบันทึก (Save) การอ่าน (Load) และการล้างข้อมูล (Reset)

UserDefaultsManager.swift

```
import Foundation

final class UserDefaultsManager {

    private static let key = "USER_PROFILE"

    static func save(profile: UserProfile) {
        if let data = try? JSONEncoder().encode(profile) {
            UserDefaults.standard.set(data, forKey: key)
        }
    }

    static func load() -> UserProfile? {
        guard let data = UserDefaults.standard.data(forKey: key) else { return nil }
        return try? JSONDecoder().decode(UserProfile.self, from: data)
    }

    // เพิ่มการ Reset
    static func reset() {
        UserDefaults.standard.removeObject(forKey: key)
    }
}
```

```
import SwiftUI
import Combine
import PhotosUI
```

```
@MainActor
class UserProfileViewModel: ObservableObject {

    @Published var name = ""
    @Published var email = ""
    @Published var bio = ""
    @Published var avatarImage: UIImage?

    @Published var selectedItem: PhotosPickerItem? {
        didSet { loadImage(from: selectedItem) }
    }

    init() {
        loadProfile()
    }

    private func loadImage(from item: PhotosPickerItem?) {
        guard let item else { return }

        Task {
            if let data = try? await item.loadTransferable(type: Data.self),
                let image = UIImage(data: data) {
                avatarImage = image
            }
        }
    }
}
```

ตัวแปร `selectedItem` สำหรับรองรับการเลือกภาพจาก Photo Library ผ่าน `PhotosPicker` โดยเมื่อผู้ใช้เลือกภาพ ระบบจะเรียกเมธอดภายในเพื่อโหลดข้อมูลภาพแบบ asynchronous แล้วแปลงเป็น `UIImage` เพื่อแสดงผลใน UI

เมื่อ `ViewModel` ถูกสร้างขึ้น ตัวสร้างเริ่มต้น (`init`) จะเรียกเมธอด `loadProfile()` เพื่ออ่านข้อมูลที่บันทึกไว้ใน `UserDefaults`

```

func saveProfile() {
    let profile = UserProfile(
        name: name,
        email: email,
        bio: bio,
        avatarData: avatarImage?.jpegData(compressionQuality: 0.8)
    )
    UserDefaultsManager.save(profile: profile)
}

func loadProfile() {
    guard let profile = UserDefaultsManager.load() else { return }
    name = profile.name
    email = profile.email
    bio = profile.bio

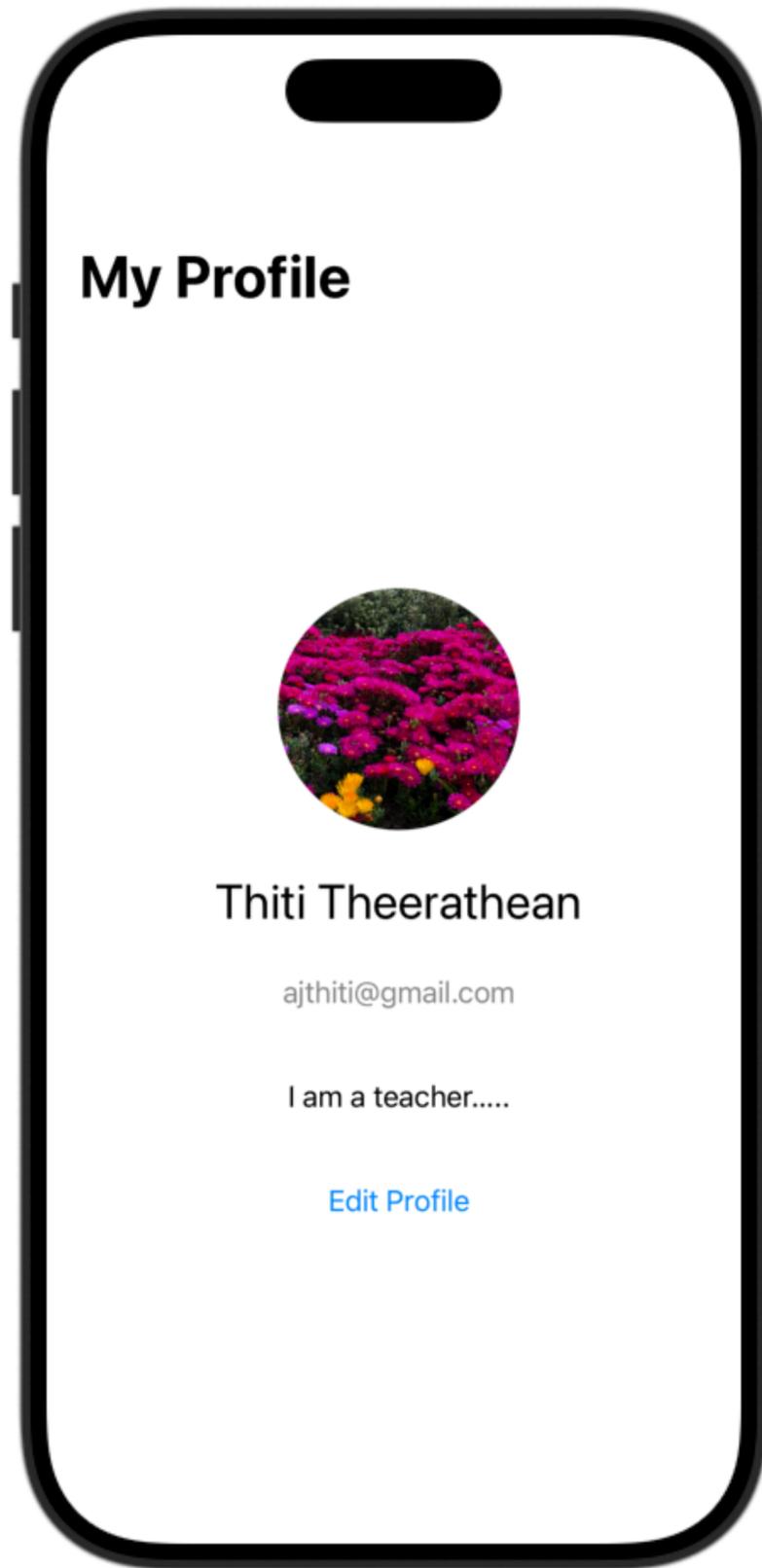
    if let data = profile.avatarData {
        avatarImage = UIImage(data: data)
    }
}

// เพิ่มการ Reset
func resetProfile() {
    UserDefaultsManager.reset()
    name = ""
    email = ""
    bio = ""
    avatarImage = nil
}
}

```

saveProfile() ทำหน้าที่รวบรวมข้อมูลจากตัวแปรสถานะทั้งหมด สร้างเป็นอ็อบเจกต์ UserProfile และแปลงภาพโปรไฟล์เป็นข้อมูลแบบ Data ก่อนส่งไปยัง UserDefaultsManager เพื่อบันทึกลง Sandbox ของแอป

resetProfile() สำหรับล้างข้อมูลผู้ใช้ ซึ่งจะสั่งให้ Persistence Layer ลบข้อมูลใน UserDefaults และโหลดสถานะเริ่มต้นกลับเข้าสู่ระบบ



```
import SwiftUI

struct ProfileView: View {

    @StateObject private var vm = UserProfileViewModel()

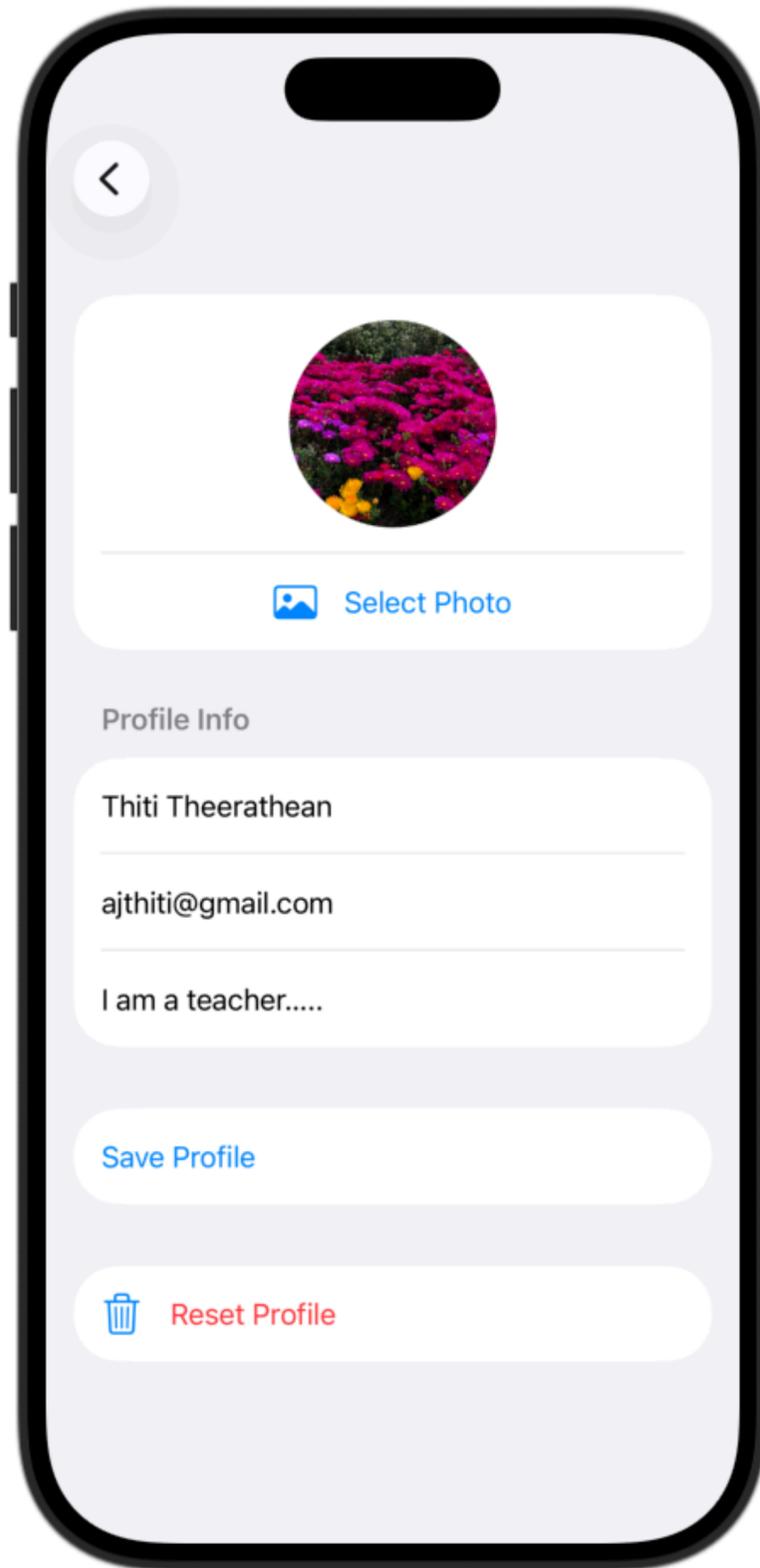
    var body: some View {
        NavigationStack {
            VStack(spacing: 20) {

                if let avatar = vm.avatarImage {
                    Image(uiImage: avatar)
                        .resizable()
                        .scaledToFill()
                        .frame(width: 140, height: 140)
                        .clipShape(Circle())
                } else {
                    Image(systemName: "person.circle")
                        .resizable()
                        .scaledToFit()
                        .frame(width: 140, height: 140)
                        .foregroundColor(.gray.opacity(0.5))
                }

                Text(vm.name).font(.title)
                Text(vm.email).foregroundColor(.secondary)
                Text(vm.bio).padding()

                NavigationLink("Edit Profile") {

                    EditProfileView(vm: vm)
                }
            }
        }.navigationTitle("My Profile")
    }
}
```



```
import SwiftUI
import PhotosUI

struct EditProfileView: View {

    @ObservedObject var vm: UserProfileViewModel
    @State private var showResetConfirm = false

    var body: some View {
        Form {
            Section {
                HStack {
                    Spacer()

                    if let avatar = vm.avatarImage {
                        Image(uiImage: avatar)
                            .resizable()
                            .scaledToFill()
                            .frame(width: 120, height: 120)
                            .clipShape(Circle())
                    } else {
                        Image(systemName: "person.circle.fill")
                            .resizable()
                            .scaledToFit()
                            .frame(width: 120, height: 120)
                            .foregroundColor(.gray.opacity(0.5))
                    }

                    Spacer()
                }
            }
        }
    }
}
```

