

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи



Харківський національний університет радіоелектроніки
Кафедра ЕОМ

Модель мобільного застосунку iOS з додаванням
базових сервісів та розширень

Автор:
Оцевик В.А.
студ. гр. КСМм-20-1

Керівник:
Філімончук Т.В.
доц. каф. ЕОМ



Мета кваліфікаційної роботи

- розглянути ряд існуючих архітектур програмного забезпечення, визначити їх переваги та недоліки;
- провести порівняльний аналіз інструментарію для розробки мобільного застосунку iOS;
- сформулювати модель мобільного застосунку iOS, яка буде відповідати вимогам:
 - надійність;
 - масштабованість;
 - швидка розробка;
 - ремонтпридатність.



Вимоги до архітектури програмного забезпечення

В якості архітектури програмного забезпечення для розробки застосунку iOS було обрано MVVM+C (Model-View-ViewModel + Coordinator) через ряд переваг:

підвищення зручності обслуговування: усунення помилок у незв'язаному коді знижує можливість виникнення регресій в іншому коді;

спрощення модульного тестування: блоки коду ізольовані один від одного, їх можна тестувати окремо та поза робочими середовищами;

підтримка спільної роботи команд: незв'язаний код інтерфейсів, які були добре спроектовано може бути розроблений окремими користувачами чи командами та інтегрований в додаток пізніше;

Coordinator – відповідає за навігацію в застосунку, розвантажує ViewController та допомагає загальній архітектурі додатку відповідати SOLID принципам.

3



Інструментарій для написання мобільного застосунку iOS

- середовище розробки – Xcode;
- мова програмування – Swift;
- фреймворк для побудови користувацького інтерфейсу – UIKit;
- менеджер залежностей – SPM.

4

Модель мобільного застосунку iOS

$$M = \{ AD, AC, F, Sc, Ser, CE \}$$

AD (AppDelegate) – об'єкт, який керує загальною поведінкою додатку;
 AC (AppCoordinator) – об'єкт, який відповідає за навігацію додатку, створює та налаштовує необхідний потік;
 F (FlowCoordinator) – об'єкт, який обробляє навігацію в потоці та налаштовує сцени;
 Sc (Scene) – модуль, який відповідає одному екрану;
 Ser (Service) – об'єкти, через які виконується взаємодія сцени з певним інтерфейсом;
 CE (Common Extension) – розширення базових класів UIKit та Foundation.

$$Sc = \{ M, V, VM \}$$

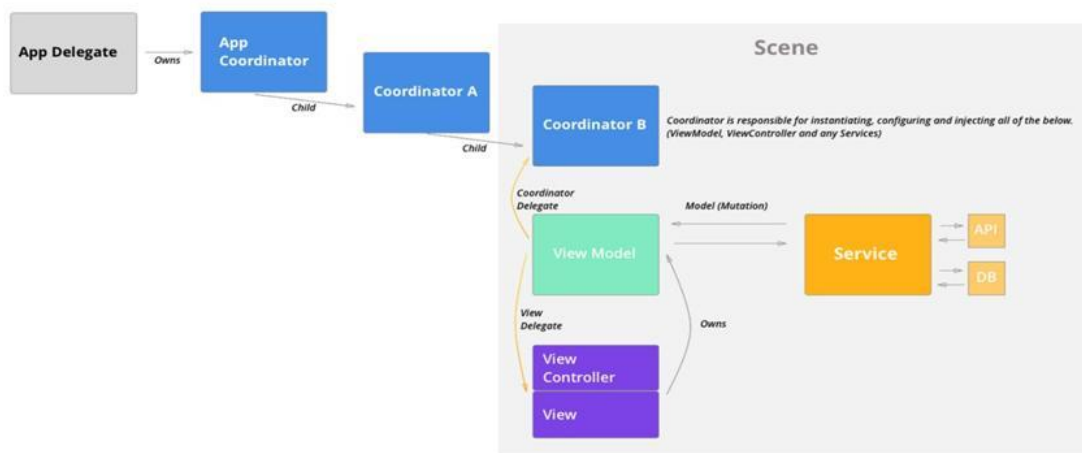
M – дані, з якими працює сцена;
 V – візуальні елементи інтерфейсу користувача;
 VM – дані, з якими працює сцена, та інтерфейс користувача відповідно до них.

$$Ser = \{ NS, KS \}$$

NS (NetworkService) – сервіс, який відповідає за отримання даних з API;
 KS (KingfisherService) – сервіс, який відповідає за завантаження зображення по URL-адресі.

5

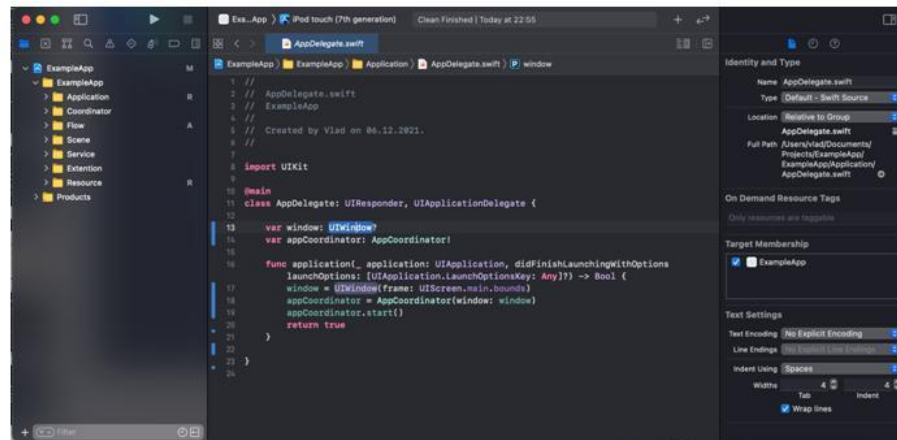
Схема моделі



6



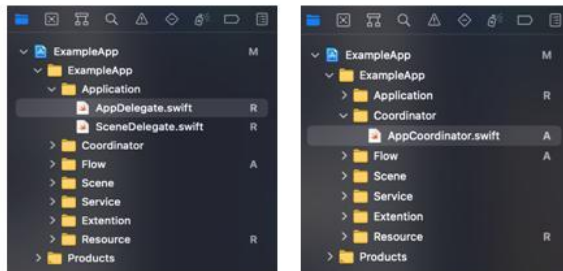
Загальна структура проекту



7



Навігація по проекту



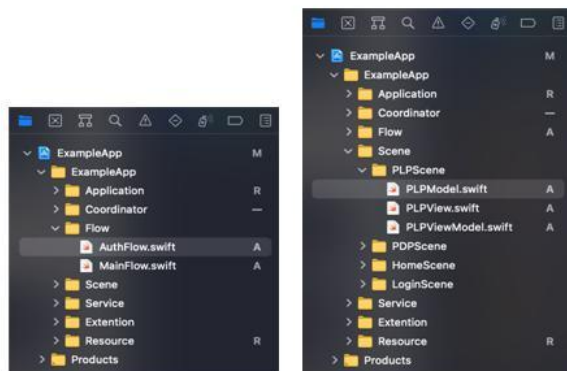
Application – містить в собі делегати додатку, які керують загальною поведінкою додатку та є вихідною точкою в нього;

Coordinator – містить в собі об'єкт `AppCoordinator` який керує загальною навігацією застосунку.

8



Навігація по проєкту



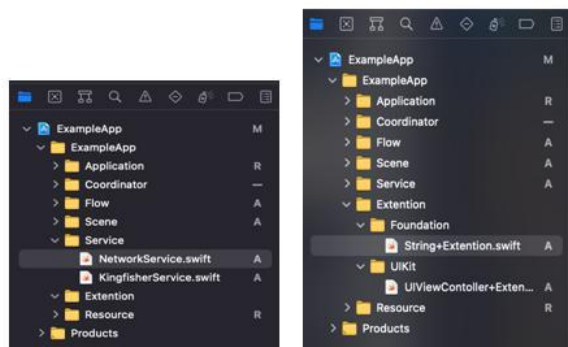
Flow – містить в собі координатори потоку, об'єкти які відповідають за навігацію потоку який складається з декількох сцен;

Scene – містить в собі сцени. Сцена – модуль що відповідає одному екрану.

9



Навігація по проєкту



Service – містить в собі сервіси;

Extention – містить в собі розширення бібліотек Foundation та UIKit.

10



Висновок

- розглянуто ряд існуючих архітектур програмного забезпечення, визначити їх переваги та недоліки;
- провести порівняльний аналіз інструментарію для розробки мобільного застосунку iOS;
- сформулювати модель мобільного застосунку iOS, яка буде відповідати вимогам:
 - надійність;
 - масштабованість;
 - швидка розробка;
 - ремонтпригодність.