

**Київський національний торговельно-економічний університет**  
**Кафедра інженерії програмного забезпечення та кібербезпеки**

# **ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ**

**на тему:**

*«Проектування та розробка соціальної мережі для бігунів на  
платформі iOS»*

Студента 2 курсу, 23 групи,  
спеціальності 121 «Інженерія  
програмного забезпечення»  
спеціалізації «Інженерія  
програмного забезпечення»

---

підпис студента

**Марценюка Антона  
Сергійовича**

Науковий керівник  
кандидат технічних наук,  
доцент кафедри інженерії  
програмного забезпечення та  
кібербезпеки

---

підпис керівника

**Харченко Олександр  
Анатолійович**

Гарант освітньої програми доктор  
економічних наук, професор  
кафедри інженерії програмного  
забезпечення та кібербезпеки

---

підпис гаранта

**Токар Володимир  
Володимирович**

**КИЇВ – 2021**

# Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

**Затверджую**

Зав. кафедри інженерії  
програмного забезпечення  
та кібербезпеки

Криворучко О. В.

"29" грудня 2020 р.

## **Завдання на випускний кваліфікаційний проєкт студентів**

Марценюку Антону Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи: «Проектування та розробка соціальної мережі для бігунів на платформі iOS»

Затверджена наказом ректора від «28» грудня 2020 р. № 3923

2. Строк здачі студентом закінченої роботи 25 листопада 2021 р.

3. Цільова установка та вихідні дані до роботи:

**Мета роботи:** розробка мобільного додатка на платформі iOS, що дозволяє відстежувати показники здоров'я та продуктивності, такі як темп, відстань, пробіг, частота серцевих скорочень та калорії під час бігу та ділитися результатами з друзями

**Об'єкт дослідження:** додаток для iOS, архітектура системи

**Предмет дослідження:** розробка програмного продукту з відстежування показників бігу

4. Консультанти роботи із зазначенням розділів, які консультують:

| Розділ | Консультант<br>(прізвище, ініціали) | Підпис, дата   |                  |
|--------|-------------------------------------|----------------|------------------|
|        |                                     | Завдання видав | Завдання прийняв |
|        |                                     |                |                  |
|        |                                     |                |                  |

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ФУНКЦІОНАЛЬНІ ВИМОГИ

1.1 Runkeeper

1.2 Runtastic

1.3 Nike+ Run Club

1.4. Функціональні вимоги

1.5. Діаграми варіантів використання

1.6. Діаграми варіантів використання

1.7 Висновки до розділу 1

РОЗДІЛ 2. АРХІТЕКТУРА IOS ДОДАТКУ

2.1. Архітектура ОС iOS

2.1.1. Життєвий цикл iOS додатку

2.1.2. Потоки iOS додатка

2.1.3. Використовувані інструменти розробки

2.1.3.1. Середовище розробки

2.1.3.2. Мова програмування

2.2. Архітектура iOS додатків

2.2.1. View

2.2.2. ViewController

2.3. MVVM и Coordinators в iOS приложениях

2.3.1. MVVM

2.3.2. Coordinator

2.4 Висновки до розділу 2

## РОЗДІЛ 3. РОЗРОБКА ДОДАТКУ

### 3.1. Архітектура системи

### 3.2. Структура програми

#### 3.2.1. AuthFlowCoordinator

#### 3.2.2. IntroFlowCoordinator

#### 3.2.3. MainFlowCoordinator

### 3.3. Інтеграція соціальних мереж

### 3.4. Тестування розробленого додатка

### 3.5 Висновки до розділу 3

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

## ДОДАТКИ

## 6. Календарний план виконання роботи

| № пор. | Назва етапів випускної кваліфікаційної роботи                               | Строк виконання етапів роботи |            |
|--------|-----------------------------------------------------------------------------|-------------------------------|------------|
|        |                                                                             | за планом                     | фактично   |
| 1      | 2                                                                           | 3                             | 4          |
| 1.     | <i>Вибір теми випускного кваліфікаційного проєкту</i>                       | 21.09.2020                    | 21.09.2020 |
| 2.     | <i>Розробка та затвердження завдання на проєкту магістра</i>                | 29.12.2020                    | 29.12.2020 |
| 3.     | <i>Вступ та перелік літературних джерел</i>                                 | 27.02.2021                    | 27.02.2021 |
| 4.     | <i>Розробка технічного завдання</i>                                         | 20.03.2021                    | 20.03.2021 |
| 5.     | <i>Розділ 1. Огляд існуючих рішень та функціональні вимоги</i>              | 16.04.2021                    | 16.04.2021 |
| 6.     | <i>Розділ 2. Архітектура іос додатку</i>                                    | 24.05.2021                    | 24.05.2021 |
| 7.     | <i>Розділ 3 Розробка додатку</i>                                            | 21.06.2021                    | 21.06.2021 |
| 8.     | <i>Розробка програми та методики тестування</i>                             | 18.10.2021                    | 18.10.2021 |
| 9.     | <i>Написання наукової статті</i>                                            | 22.05.2021                    | 22.05.2021 |
| 10.    | <i>Керівництво користувача</i>                                              | 21.10.2021                    | 21.10.2021 |
| 11.    | <i>Висновки та пропозиції</i>                                               | 01.11.2021                    | 01.11.2021 |
| 12.    | <i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>  | 03.11.2021                    | 03.11.2021 |
| 13.    | <i>Підготовка автореферату та презентації доповіді</i>                      | 03.11.2021                    | 03.11.2021 |
| 14.    | <i>Попередній захист випускного кваліфікаційного проєкту</i>                | 22.11.2021<br>—<br>25.11.2021 | 22.11.2021 |
| 15.    | <i>Здача зброшурованої випускного кваліфікаційного проєкту</i>              | 25.11.2021                    | 25.11.2021 |
| 16.    | <i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>               | 26.11.2021                    | 26.11.2021 |
| 17.    | <i>Підготовка до публічного захисту випускного кваліфікаційного проєкту</i> |                               |            |

7. Дата видачі завдання «29» грудня 2020 р.

8. Науковий керівник випускного кваліфікаційного проєкту \_\_\_\_\_

Харченко О.А.

(прізвище, ініціали, підпис)

6. Гарант освітньої програми Токар В.В.

(прізвище, ініціали, підпис)

7. Завдання прийняв до виконання студент Марценюк А.С.

(прізвище, ініціали, підпис)

## This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист Харченко О.А.  
(ПІБ, підпис, дата)

Випускний кваліфікаційний проект студента Марценюка А.С.  
(прізвище, ініціали)

Гарант освітньої програми Токар В.В.  
(прізвище, ініціали, підпис)

Завідувач кафедри \_\_\_\_\_ Криворучко О.В.  
(підпис, прізвище, ініціали)

« \_\_\_\_\_ » 20 \_\_\_\_ г.

## АНОТАЦІЯ

Дипломна робота на тему «Проектування та розробка соціальної мережі для бігунів на платформі iOS» містить 41 сторінки, 28 рисунків та 1 додаток. Перелік посилань налічує 7 найменувань.

**Мета дослідження:** розробка мобільного додатка на платформі iOS, що дозволяє відстежувати показники здоров'я та продуктивності, такі як темп, відстань, пробіг, частота серцевих скорочень та калорії під час бігу та ділитися результатами з друзями.

**Об'єкт дослідження:** додаток для iOS, архітектура системи.

**Предметом дослідження:** розробка програмного продукту з відстежування показників бігу.

В першому розділі були проаналізовані існуючі рішення схожих додатків. Були виявлені недоліки та переваги кожного з них, які допомогли поєднати в собі функціональні вимоги додатку. Опрацьовані сценарії використання додатку. Перш за все, при складанні сценарію виявляється мета, якої будуть досягати користувачі на кожному з екранів додатка.

В другому розділі ми розглянули детально всі шари iOS архітектури, переглянули життєвий цикли мобільного додатку, як відбувається компілюється додаток, та обрали архітектурний патерн який будеме застосовувати для дизайну. Обрали мову програмування котра відповідає сучасним вимогам. Розглянули переваги середі розробки iOS додатків, яку підтримує Apple. Для оптимізація екранів застосовуємо Кординатор.

В третьому розділі було створено зв'язок з сервером котрий буде зберігати данні про користувача. Створили навігацію по проекту між контролерами. Інтегрували дані до соціальних мереж та протестували додаток на коректність різних інтерфейсів на системі iOS.

## ANNOTATION

The thesis on the topic "Development of a mobile application for the integration of social networks on the iOS platform" contains 41 pages, 31 drawings, and 1 appendix. The list of links includes 6 items.

**The purpose of the study** is to development of a mobile application on the ios platform, which allows you to integrate multiple social networks for viewing, creating, editing and publishing posts.

**Research subject:** application for iOS, system architecture.

**The study's subject** is software product development with integration of social networks.

The first section analyzed the existing solutions of similar applications. The disadvantages and advantages of each of them were identified, which helped to combine the functional requirements of the application. Scenarios for using the application have been processed. First of all, when compiling a script, the goal is revealed, which will be achieved by users on each of the screens of the application.

In the second section, we looked in detail at all the layers of the iOS architecture, reviewed the life cycles of the mobile application, how the application is compiled, and chose the architectural pattern that we will use for the design. They chose a programming language that meets modern requirements. Considered the benefits of the iOS application development environment supported by Apple. To optimize the screens we use the Coordinator.

In the third section there was a real connection with the server which will store data on the user. Created project navigation between controllers. We integrated data into social networks and tested the application for the correctness of various interfaces on the iOS system.

## ЗМІСТ

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>ВСТУП.....</b>                                                   | <b>4</b>  |
| <b>РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ФУНКЦІОНАЛЬНІ ВИМОГИ.....</b> | <b>6</b>  |
| 1.1 Runkeeper.....                                                  | 6         |
| 1.2 Runtastic.....                                                  | 7         |
| 1.3 Nike+ Run Club.....                                             | 8         |
| 1.4. Функціональні вимоги.....                                      | 9         |
| 1.5. Діаграми варіантів використання.....                           | 10        |
| 1.6. Діаграми варіантів використання.....                           | 11        |
| 1.7 Висновки до розділу 1.....                                      | 14        |
| <b>РОЗДІЛ 2. АРХІТЕКТУРА IOS ДОДАТКУ.....</b>                       | <b>15</b> |
| 2.1. Архітектура ОС iOS.....                                        | 15        |
| 2.1.1. Життєвий цикл iOS додатку.....                               | 18        |
| 2.1.2. Потoki iOS додатка.....                                      | 19        |
| 2.1.3. Використовувані інструменти розробки.....                    | 21        |
| 2.1.3.1. Середовище розробки.....                                   | 21        |
| 2.1.3.2. Мова програмування.....                                    | 22        |
| 2.2. Архітектура iOS додатків.....                                  | 22        |
| 2.2.1. View.....                                                    | 21        |
| 2.2.2. ViewController.....                                          | 25        |
| 2.3. MVVM и Coordinators в iOS приложениях.....                     | 25        |
| 2.3.1. MVVM.....                                                    | 25        |
| 2.3.2. Coordinator.....                                             | 26        |
| 2.4 Висновки до розділу 2.....                                      | 28        |

|              |                 |         |        |            |                                                                         |                                                      |       |         |
|--------------|-----------------|---------|--------|------------|-------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |                 |         |        |            | КНТЕУ 121 02з-10.МР                                                     |                                                      |       |         |
| Зм.          | Аркуш           | № докум | Підпис | Дата       | Проектування та розробка соціальної мережі для бігунів на платформі iOS | Стадія                                               | Аркуш | Аркушів |
| Зав. кафедри | Криворучко О.В. |         |        | 27.02.2021 |                                                                         | 3                                                    | 2     | 41      |
| Керівник     | Харченко О.А.   |         |        | 27.02.2021 |                                                                         |                                                      |       |         |
| Гарант       | Токар В.В.      |         |        | 27.02.2021 |                                                                         |                                                      |       |         |
| Розроб.      | Марценюк А.С.   |         |        | 27.02.2021 |                                                                         |                                                      |       |         |
|              |                 |         |        |            | Зміст                                                                   | Факультет інформаційних технологій, 2 курс, 23 група |       |         |

|                                           |           |
|-------------------------------------------|-----------|
| <b>РОЗДІЛ 3. РОЗРОБКА ДОДАТКУ.....</b>    | <b>29</b> |
| 3.1. Архітектура системи.....             | 29        |
| 3.2. Структура програми.....              | 30        |
| 3.2.1. AuthFlowCoordinator.....           | 32        |
| 3.2.2. IntroFlowCoordinator.....          | 34        |
| 3.2.3. MainFlowCoordinator.....           | 34        |
| 3.3. Інтеграція соціальних мереж.....     | 37        |
| 3.4. Тестування розробленого додатка..... | 38        |
| 3.5 Висновки до розділу 3.....            | 39        |
| <b>ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....</b>        | <b>40</b> |
| <b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>   | <b>41</b> |
| <b>ДОДАТКИ.....</b>                       | <b>42</b> |

|     |       |         |        |      |                            |     |
|-----|-------|---------|--------|------|----------------------------|-----|
|     |       |         |        |      | <i>КНТЕУ 121 023-10.МР</i> | Арк |
| Зм. | Аркуш | № докум | Підпис | Дата |                            | 3   |

## ВСТУП

У цьому сучасному світі, де «гарному зовнішньому вигляду» надається виняткове значення, кожен хоче виглядати чудово. Саме тому вони вирішують піти в спортзал або сісти на швидку дієту, приймаючи новорічні обіцянки. Проте нас не дивує той факт, що ці постанови діють не більше тижня. Тоді всі ми думаємо, чому не може бути чарівної палички, щоб схуднути/набрати вагу чи залишатися у формі? Що ж, у нас є мобільний додаток у вигляді чарівної палички, який допомагає нам з здоров'ям і фітнесом.

У цьому сучасному світі, де є програми для всього, наприклад ігор, медіа та розваг, освіти, банківської справи та фінансів, є також багато мобільних програм для здоров'я та фітнесу. Сьогодні сфера розробки мобільних додатків для здоров'я та фітнесу набуває все більшої популярності, оскільки кількість людей, які користуються такими додатками, зросла. Є багато причин, чому ви повинні використовувати мобільний додаток для здоров'я та фітнесу. Ось деякі з найважливіших переваг використання мобільного додатка для здоров'я та фітнесу.

### 1. Легко відстежуйте свій прогрес:

Звичайно, ви поставили перед собою цілі. Однак ви повинні стежити за повсякденними результатами, щоб перевірити, чи працює він для вас чи ні. Додаток допомагає відстежувати свій прогрес. Це дає вам зрозуміти, чи йдете ви в правильному напрямку чи ні. Якщо є прогрес, продовжуйте виконувати свої плани, інакше настав час їх змінити.

### 2. Отримайте безкоштовні ідеї для тренувань:

|              |       |                 |        |            |                                                                         |                                                      |       |         |
|--------------|-------|-----------------|--------|------------|-------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |       |                 |        |            | КНТЕУ 121 02з-10.МР                                                     |                                                      |       |         |
| Зм.          | Аркуш | № докум         | Підпис | Дата       | Проектування та розробка соціальної мережі для бігунів на платформі iOS | Стадія                                               | Аркуш | Аркушів |
| Зав. кафедри |       | Криворучко О.В. |        | 27.02.2021 |                                                                         | В                                                    | 4     | 41      |
| Керівник     |       | Харченко О.А.   |        | 27.02.2021 |                                                                         | Факультет інформаційних технологій, 2 курс, 23 група |       |         |
| Гарант       |       | Токар В.В.      |        | 27.02.2021 |                                                                         |                                                      |       |         |
| Розроб.      |       | Марценюк А.С.   |        | 27.02.2021 | Вступ                                                                   |                                                      |       |         |

У мобільному додатку також є блоги та статті, які дають вам деякі з найкращих і найефективніших ідей для тренувань, які допоможуть вам досягти поставлених цілей.

### 3. Поставте цілі, яких ви можете досягти:

Коли ви починаєте тренуватися, ви повинні поставити перед собою реалістичні цілі. Ви повинні встановити цілі щодо того, скільки ваги ви хочете схуднути за скільки часу. Отже, ви можете використовувати мобільні додатки для здоров'я та фітнесу, щоб ставити реалістичні цілі та досягати їх.

*Метою дослідження випускної кваліфікаційної роботи є розробка мобільного додатка на платформі iOS, що дозволяє відстежувати показники здоров'я та продуктивності, такі як темп, відстань, пробіг, частота серцевих скорочень та калорії під час бігу та ділитися результатами з друзями. Об'єктом дослідження є додаток для iOS, архітектура системи.*

#### *Завдання дослідження:*

- проаналізувати існуючі рішення;
- вивчити особливості мобільної платформи iOS;
- сформулювати список вимог щодо розроблюваного з додатком;
- ознайомитися з різними API соціальних мереж;
- спроектувати архітектуру програми;
- реалізувати програму;
- протестувати додаток.

*Метод дослідження:* теоретичне дослідження і практична реалізація.

Область застосування – організації, для поширення в AppStore.

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
|     |       |         |        |      |                     |      |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 5    |

## РОЗДІЛ 1

### ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ФУНКЦІОНАЛЬНІ ВИМОГИ

У даній предметній сфері існує багато готових рішень. Розглянемо кілька найпопулярніших на даний момент, що допомагають відстежувати показники бігу

#### 1.1.Runkeeper

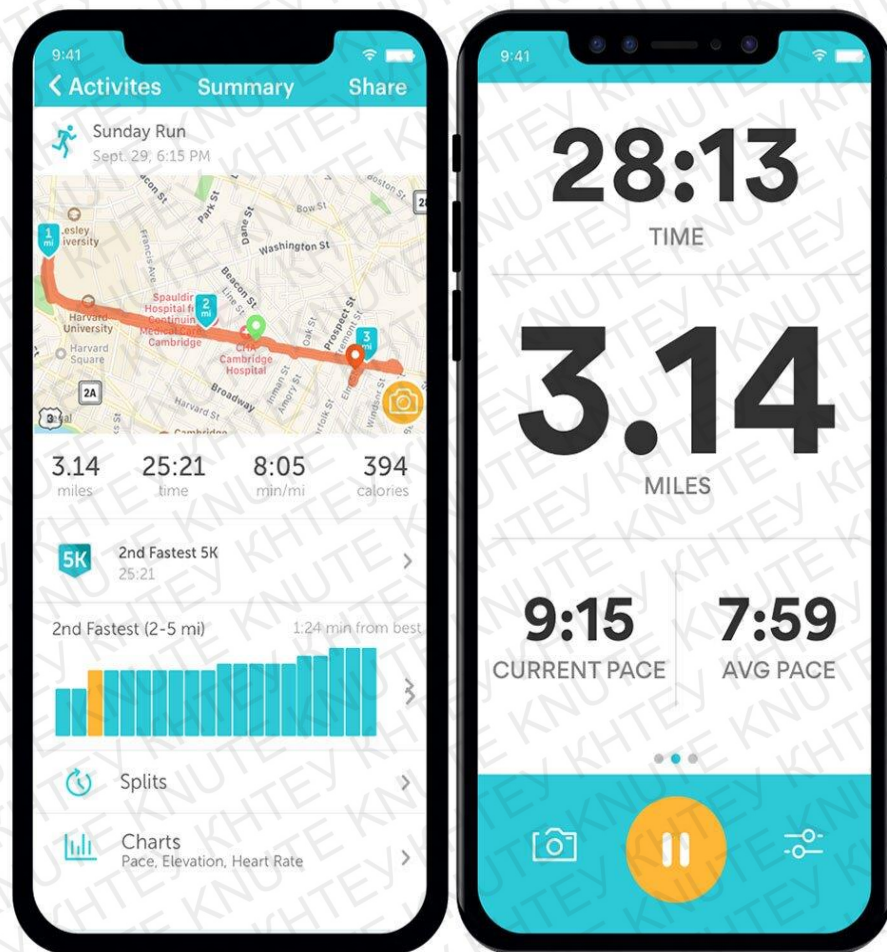


Рис. 1.1. Runkeeper додаток

|              |       |                 |        |            |                                                                         |                                                      |       |         |
|--------------|-------|-----------------|--------|------------|-------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |       |                 |        |            | КНТЕУ 121 02з-10.МР                                                     |                                                      |       |         |
| Зм.          | Аркуш | № докум         | Підпис | Дата       | Проектування та розробка соціальної мережі для бігунів на платформі iOS | Стадія                                               | Аркуш | Аркушів |
| Зав. кафедри |       | Криворучко О.В. |        | 16.04.2021 |                                                                         | P1                                                   | 6     | 41      |
| Керівник     |       | Харченко О.А.   |        | 16.04.2021 |                                                                         | Факультет інформаційних технологій, 2 курс, 23 група |       |         |
| Гарант       |       | Токар В.В.      |        | 16.04.2021 |                                                                         |                                                      |       |         |
| Розроб.      |       | Марценюк А.С.   |        | 16.04.2021 | Огляд існуючих рішень та функціональні вимоги                           |                                                      |       |         |

Переваги:

- сильна бігова спільнота та підтримка;
- підходить для всіх рівнів бігу;
- ставте цілі та відстежуйте прогрес;
- завдання в програмі та віртуальні бігові групи.

Недоліки:

- потрібно підписка, щоб отримати доступ до преміальних функцій;
- подальше використання GPS у фоновому режимі може зменшити час роботи акумулятора.

## 1.2. Runtastic

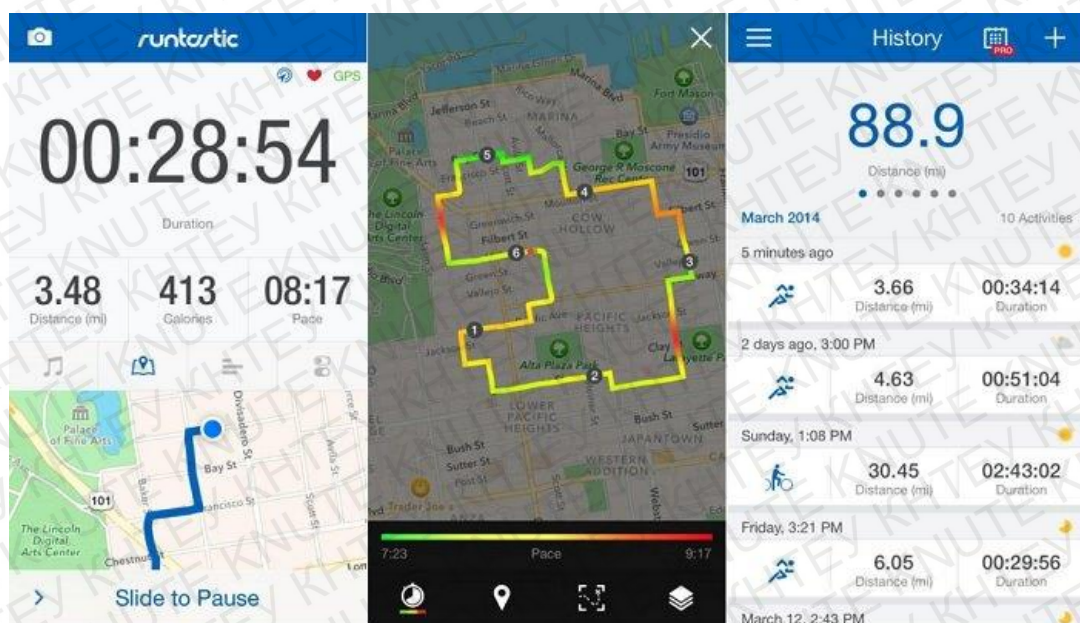


Рис. 1.2. Runtastic додаток

Переваги:

- одне місце для збереження всіх ваших бігових маршрутів і статистики;
- публікація статистики в Twitter або Facebook із програми.

|     |       |         |        |      |                     |           |
|-----|-------|---------|--------|------|---------------------|-----------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк.<br>7 |
| Зм. | Аркуш | № докум | Підпис | Дата |                     |           |

- відстежує та реєструє понад 120 заходів.

Недоліки:

- голосові сповіщення не такі гучні, як в інших додатках;
- для деяких функцій забігу і тренувань потрібна преміальна підписка.

### 1.3. Nike+ Run Club

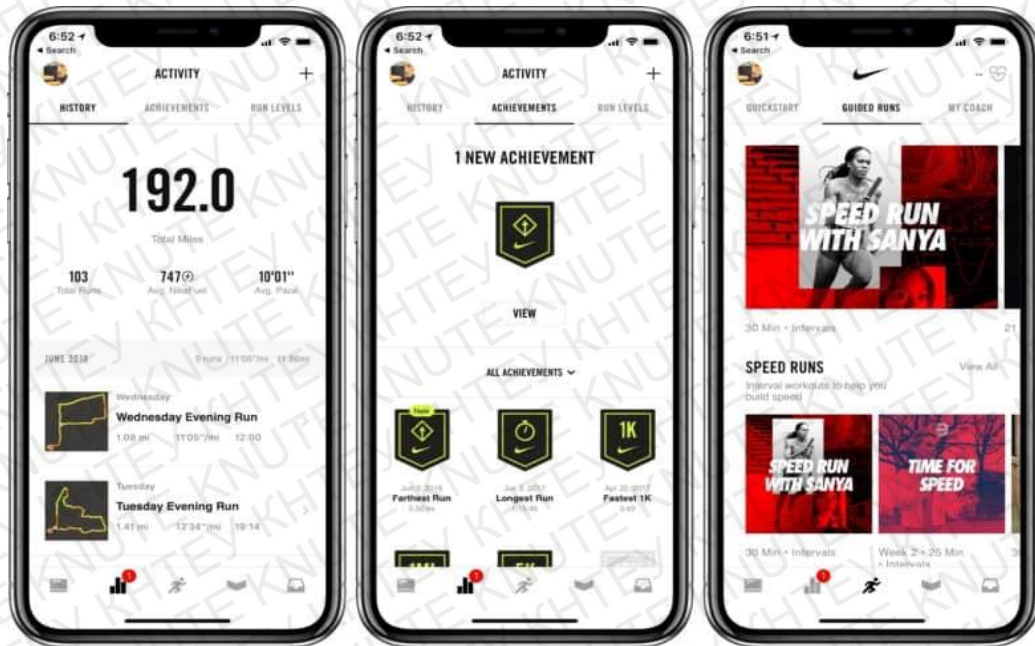


Рис. 1.3. Nike+ Run Club додаток

Переваги:

- швидкий огляд темпу, місцезнаходження, відстані, висоти, частоти серцевих скорочень і кілометрів;
- коучинг та індивідуальні плани;
- аудіо працює з тренерами Nike.

Недоліки:

- тільки для Apple Watch;
- користувачі повідомляють, що GPS може бути плямистим.

Таким чином, вже існуюче програмне забезпечення не дозволяє

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

8

повністю охопити весь список вимог, що свідчить про актуальність цієї роботи.

#### **1.4. Функціональні вимоги**

Система повинна являти собою мобільний додаток працює на операційній система iOS. Додаток має мати доступ в інтернет, а також підтримувати різні пристрої, такі як iPhone, iPad, iPod. Розглянемо основні компоненти проектування, описані в керівництві.

Система повинна мати можливість:

- 1) Реєструвати користувача.
- 2) Відновлювати пароль.
- 3) Авторизувати користувача.
- 4) Додати аккаунт соціальної мережі.
- 5) Видаляти акаунт соціальної мережі.
- 6) Додавання особистої інформації.
- 7) Налаштування.
- 8) Постановка цілей.
- 9) Попередньо встановлені процедури вправ.
- 10) Гейміфікація.
- 11) Пряма трансляція.
- 12) Інтеграція з допоміжними пристроями.
- 13) Відстеження активності.
- 14) Геолокація.

#### **1.5. Діаграми варіантів використання**

На рисунках 1.7- 1.9 показані діаграми варіантів використання додатку.

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
|     |       |         |        |      |                     | 9    |
| Зм. | Аркуш | № докум | Підпис | Дата |                     |      |

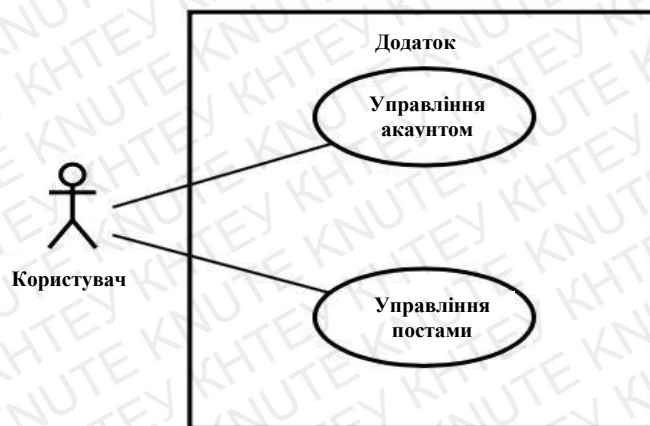


Рис. 1.7. Загальна діаграма варіантів використання.

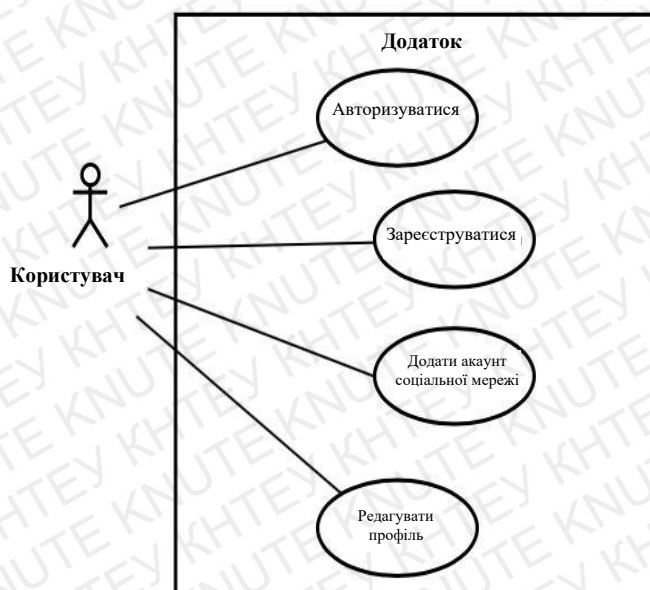


Рис. 1.8 .Діаграма варіантів використання «Управління акаунтом».

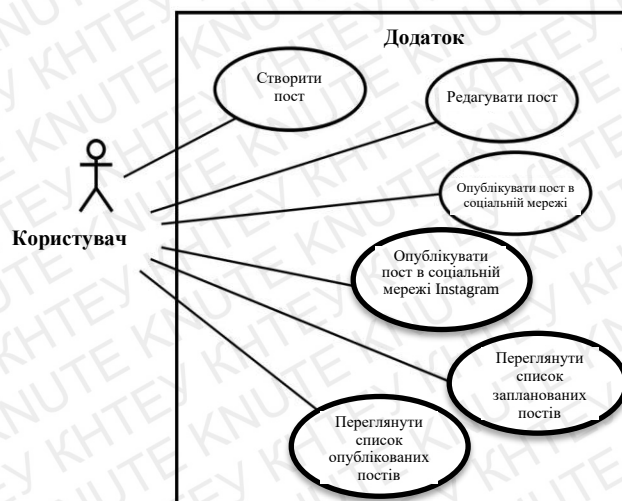


Рис. 1.9. Діаграма варіантів використання «Управління постами»

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

10

## **1.6. Сценарії варіантів використання**

*Сценарій варіанта використання «Авторизуватись»:*

1. Користувач вводить логін і пароль в текстові поля на екрані авторизація.
2. Система перевіряє дані на валідність.
3. Користувач натискає на кнопку «Авторизуватись».
4. Система відправляє на сервер введені дані.
5. Система отримує відповідь від сервера і показує помилку якщо користувач ввів некоректні дані або направляє його на головний екран.

*Сценарій варіанта використання «Зареєструватися»:*

1. Користувач вводить username, пошту, пароль, перевірки пароль в текстові поля на екрані реєстрація.
2. Система перевіряє, що паролі збігаються.
3. Користувач натискає на кнопку «Зареєструватися».
4. Система відправляє на сервер введені дані.
5. Система отримує відповідь, після чого відкриває головний екран додатка якщо авторизація пройшла успішно, в іншому випадку показує помилку.

*Сценарій варіанта використання «Редагувати профіль»:*

1. Користувач натискає на кнопку «Налаштування» в лівому меню головного екрану.
2. Система відкриває екран настройки.
3. Користувач може змінити пошту і картинку, потім натиснути на кнопку "Зберегти"
4. Система відправляє данні на сервер та оновлює дані користувача.

*Сценарій варіанти використання «Почати забіг»:*

1. Користувач натискає на кнопку "Вперед" на головному екрані.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

**КНТЕУ 121 023-10.МР**

Арк.

11

2. Система відкриває екран початку забігу.
3. Користувач бачить локація в котрій розпочався забіг, відстань та швидкість.
4. Користувач натискає на кнопку «Закінчити забіг».
5. Система відкриває екран зі списком показників забігу та виділяє шлях пробігу на карті.
6. Користувач має змогу поділитися результатами в соціальних мережах.
7. Система повертає користувача на головний екран, а також відображає останній забіг.

*Сценарій варіанта використання «Редагувати забіг»:*

1. Користувач натискає кнопку "Редагувати" на конкретному забігу.
2. Система відкриває екран для редагування результатів.
3. Користувач змінює заголовок, фото або відео, обирає акаунти соціальних мереж і виставляє дату і натискає кнопку «Готово».
4. Система відправляє нові данні на сервер та повертає користувача на головний екран.

*Сценарій варіанта використання «Опублікувати пост в соціальну мережу»:*

1. Користувач натискає кнопку «Опублікувати».
2. Система відкриває екран з детальною інформацією про інформацію забіга.
3. Користувач переглядає вміст поста і натискає кнопку "Опублікувати".
4. Система публікує пост в соціальну мережу, повертає користувача на головний екран.

*Сценарій варіанта використання «Поставлення цілей»:*

1. Користувач натискає на кнопку «Цілі» в лівому меню головного

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
|     |       |         |        |      |                     | 12   |
| Зм. | Аркуш | № докум | Підпис | Дата |                     |      |

екрану.

2. Система відкриває екран з поставленими цілями та її результатами.

3. Користувач натискає на кнопку «Додати».

4. Система відкриває новий екран на котрому потрібно заповнити інформацію по цілі: термін дії, вправи та бажаний результат.

5. Користувач натискає кнопку «Зберегти» та система відкриває екран цілей.

6. Система відкриває новий екран з початком нового челенджа.

*Сценарій варіанта використання «Подивитися перелік запланованих забігів»:*

1. Користувач натискає на кнопку «Заплановані» в лівому меню головного екрану.

2. Система відкриває екран зі списком запланованих забігів.

*Сценарій варіанта використання «Подивитися результати»:*

1. Користувач натискає на кнопку «Результати» в лівому меню головного екрану.

2. Система відкриває екран зі списком результатів забігів.

### **1.7. Висновки до розділу 1**

В першому розділі були проаналізовані існуючі рішення схожих додатків. Були виявлені недоліки та переваги кожного з них, які допомогли поєднати в собі функціональні вимоги додатку. Опрацьовані сценарії використання додатку. Перш за все, при складанні сценарію виявляється мета, якої будуть досягати користувачі на кожному з екранів додатка. Фітнес-додатки повинні дозволяти користувачам встановлювати цілі для тренувань, відстеження активності. Ця функція є специфічною для програм для тренувань. Основна проблема більшості програм для тренувань полягає в тому, що

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 13   |

процедури неможливо налаштувати. Після отримання результату користувач може перейти на відповідну сторінку. Існує багато показників, які потрібно відстежувати, але програми повинні переконатися, що вони виділяють лише кілька найбільш важливих для їх програми, щоб уникнути перевантаження інформацією. Графічний огляд показників також був би гарним варіантом, щоб користувач отримував підсумок своєї забігу під рукою.

Користувач може переходити в профіль іншого користувача та відстежувати його сторінку, може написати кому-небудь повідомлення.

|     |       |         |        |      |                     |     |
|-----|-------|---------|--------|------|---------------------|-----|
|     |       |         |        |      | КНТЕУ 121 023-10.MP | Арк |
|     |       |         |        |      |                     |     |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 14  |

## РОЗДІЛ 2

### АРХІТЕКТУРА IOS ДОДАТКУ

#### 2.1. Архітектура ОС iOS

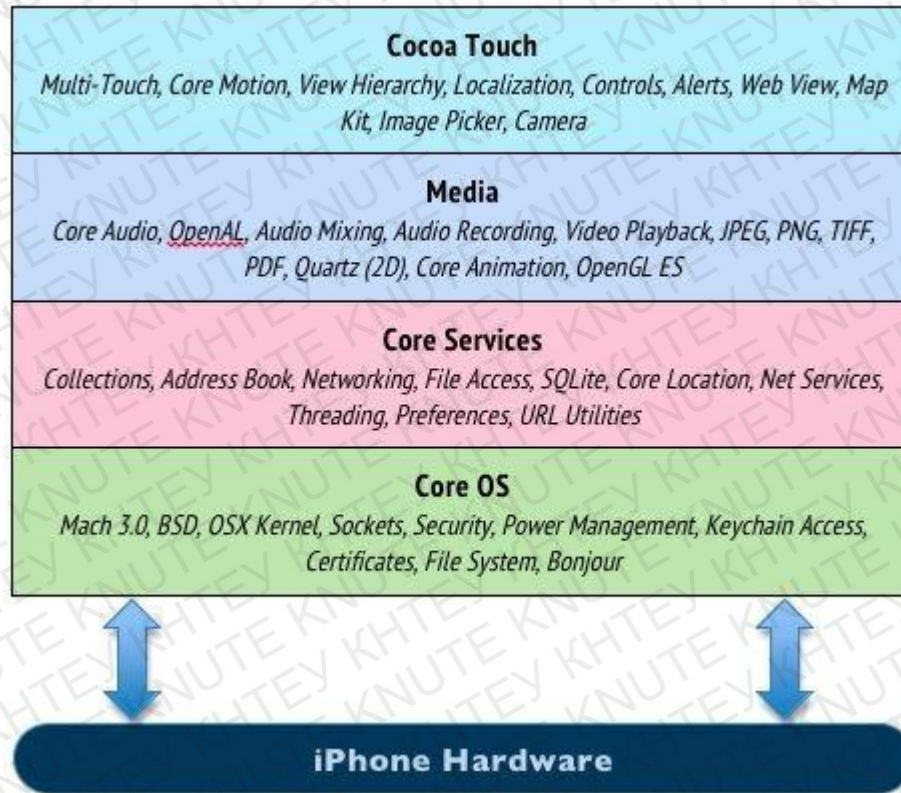


Рис. 2.1. Шари ОС iOS

Архітектуру iOS можна розділити на 4 шари (Рис. 2.1):

- 1) Рівень ядра (Core OS) - працює з файловою системою, контролює дійсність різних сертифікатів, що належать додаткам [2]. Також відповідає за безпеку всієї системи в цілому. Містить низький рівень доступу до елементів пристрою.
- 2) Рівень сервісів ядра (Core Service) - надає доступ до таких сервісів

|              |                 |         |        |            |                                                                         |                                                      |       |         |
|--------------|-----------------|---------|--------|------------|-------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |                 |         |        |            | КНТЕУ 121 02з-10.МР                                                     |                                                      |       |         |
|              |                 |         |        |            |                                                                         |                                                      |       |         |
| Зм.          | Аркуш           | № докум | Підпис | Дата       |                                                                         |                                                      |       |         |
| Зав. кафедри | Криворучко О.В. |         |        | 24.05.2021 | Проектування та розробка соціальної мережі для бігунів на платформі iOS | Стадія                                               | Аркуш | Аркушів |
| Керівник     | Харченко О.А.   |         |        | 24.05.2021 |                                                                         | P2                                                   | 15    | 41      |
| Гарант       | Токар В.В.      |         |        | 24.05.2021 |                                                                         | Факультет інформаційних технологій, 2 курс, 23 група |       |         |
| Розроб.      | Марценюк А.С.   |         |        | 24.05.2021 |                                                                         |                                                      |       |         |
|              |                 |         |        |            | Архітектура IOS додатку                                                 |                                                      |       |         |

як:

- Вбудована база даних (SQLite), що дозволяє створювати модель даних, а також здійснювати SQL запити.
- Файловий диспетчер (File Accesses), сервіс має високий рівень інтерфейсу для доступу до різних файлів належить додатком.
- Локація (Core Location) – сервіс, який дозволяє відстежити своє поточне місцезнаходження, а також відстежувати його тривалі переміщення.
- Інтернет сервіс (Net Service) – сервіс, який дозволяє отримувати і передавати дані через мережу інтернет.

3) Рівень медіа (Media) – містить інструменти дозволяють обробляти більшість форматів мультимедійних файлів, наприклад:

- Аудіо (Core Audio) – дозволяє записувати, прослуховувати, а також і редагувати різні аудіо доріжки. Можливість дістати з відео, певну доріжку.
- Відео (Video playback) – сервіс, який дозволяє відтворювати, перемотувати, зупиняти відео. Редагування відео на увазі зміну розширення, довжини або якості.
- Анімація (Core Animation) – бібліотека дозволяє створювати анімації. При створенні анімації може використовуватися як показ картинок послідовно, так і зміна різних параметрів елементів інтерфейсу, таких як висота, ширина, розташування, видимість і т.д.

4) Рівень ядра (Core OS) – працює з файловою системою, контролює дійсність різних сертифікатів, що належать додаткам [2]. Також відповідає за безпеку всієї системи в цілому. Містить низький рівень доступу до елементів пристрою.

5) Рівень сервісів ядра (Core Service) – надає доступ до таких сервісів як:

- Вбудована база даних (SQLite), що дозволяє створювати модель даних, а також здійснювати SQL запити.

|     |       |         |        |      |                     |            |
|-----|-------|---------|--------|------|---------------------|------------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк.<br>16 |
| Зм. | Аркуш | № докум | Підпис | Дата |                     |            |

- Файловий диспетчер (File Accesses), сервіс має високий рівень інтерфейсу для доступу до різних файлів належить додатком.
- Локація (Core Location) – сервіс, який дозволяє відстежити своє поточне місцезнаходження, а також відстежувати його тривалі переміщення.
- Інтернет сервіс (Net Service) – сервіс, який дозволяє отримувати і передавати дані через мережу інтернет.

6) Рівень медіа (Media) – містить інструменти дозволяють обробляти більшість форматів мультимедійних файлів, наприклад:

- Аудіо (Core Audio) – дозволяє записувати, прослуховувати, а також і редагувати різні аудіо доріжки. Можливість дістати з відео, певну доріжку.
- Відео (Video playback) – сервіс, який дозволяє відтворювати, перемотувати, зупиняти відео. Редагування відео на увазі зміну розширення, довжини або якості.
- Анімація (Core Animation) – бібліотека дозволяє створювати анімації. При створенні анімації може використовуватися як показ картинок послідовно, так і зміна різних параметрів елементів інтерфейсу, таких як висота, ширина, розташування, видимість і т.д.

7) Рівень інтерфейсу (Cocoa Touch) – має безліч елементів для створення мобільних інтерфейсів, а а також надає іншим верствам інформацію вхідну від користувача. У нього входять такі елементи як:

- Оброблювач множинних натискань (Multi-Touch) – дозволяє обробляти кілька натискань на екран.
- Оброблювач жестів (Core Motion) – дозволяє обробити різні жести виробляються на екрані пристрою, такі як зрушення вправо, вниз і т.д.
- Галерея (ImagePickers) – надає доступ до фотографій і відео лежать на пристрої. За допомогою нього додатки можуть отримати доступ і завантажити до себе файли з галереї.
- Камера (Camera) – використовується для знімком і подальшим

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 17   |

завантаженням даних в додаток.

### 2.1.1. Життєвий цикл iOS додатку

Додатки мають кілька станів, щоб дізнатися як вони змінюються, розглянемо життєвий цикл (Рис. 2.2).

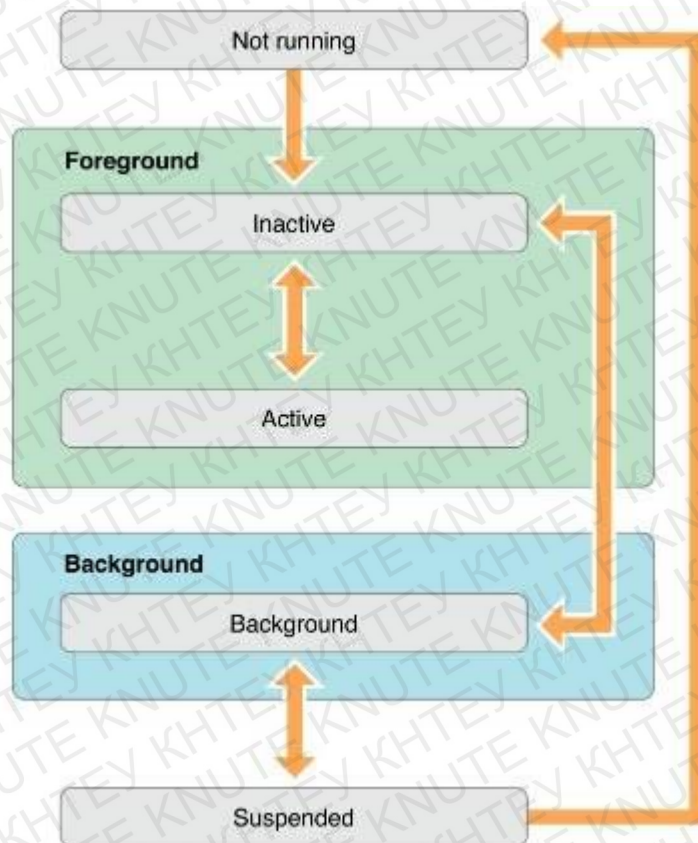


Рис. 2.2. Стани iOS додатку

Стану залежать один від одного і змінюються строго своєю ієрархією. Кожна програма починає свою роботу після натискання іконки програми [3]. Розглянемо стану докладніше:

- 1) Not running – програма не запущено або система видалила його з пам'яті.
- 2) Inactive – додаток запущено на передньому плані, але все ще не

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

18

приймає події. Зазвичай додаток залишається в цьому стані тільки не надовго, оскільки переходить в інший стан.

3) Active – звичайний режим при якому додаток працює на передньому плані і приймає події.

4) Background – додаток знаходиться в фоновому режимі і виконує якийсь код. Більшість додатків вводять цей стан не надовго, після чого йде припинення програми. Однак якщо потрібно більше часу для роботи у фоновому режимі, додаток може запитати його у системи і залишитися в цьому стані на деякий час.

5) Suspended – додаток знаходиться в фоновому режимі і не виконує код. Система автоматично переводить додатки в цей стан і не повідомляє їх про це. Додаток призупинено, але залишається в пам'яті. При виникненні ситуації, коли закінчується пам'ять на пристрої, система може видалити призупинені додатки без повідомлення, щоб звільнити місце для наступного додатку.

Додатки повинні бути готові до припинення роботи в будь-який час і не повинні чекати, щоб зберегти призначені для користувача дані і виконати інші важливі завдання.

Користувач також як і система здатний видалити програму з пам'яті використовуючи багатозадачність інтерфейс, в цьому випадку програма не отримає ніяких повідомлень про це.

### **2.1.2. Потіки iOS додатка**

Додаток створює головний потік вашого додатку і при необхідності можна створювати додаткові потоки для виконання інших завдань [4]. Для iOS додатків найкращим методом є використання Grand Central Dispatch (GCD), операційних об'єктів і інших асинхронних інтерфейсів програмування, а не створення і управління потоками самостійно. Такі технології, як GCD, дозволяють визначати роботу, яку ви хочете зробити, і порядок, в якому ви

|     |       |         |        |      |                     |     |
|-----|-------|---------|--------|------|---------------------|-----|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк |
|     |       |         |        |      |                     | 19  |
| Зм. | Аркуш | № докум | Підпис | Дата |                     |     |

хочете це зробити, а також дозволяти системі вирішувати, як краще за все виконати цю роботу на доступних CPU (Рис. 2.3).

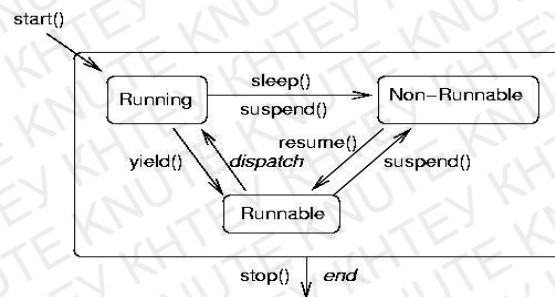


Рис. 2.3. Робота потоків в iOS додатку

Надання системі управління потоками спрощує код, який ви повинні писати, спрощує забезпечення правильності цього коду і забезпечує більш високу загальну продуктивність.

Розглянемо основні підходи роботи з потоками:

1. Робота з уявленнями, основний анімацією і багатьма іншими класами UIKit зазвичай повинна відбуватися в головному потоці програми. Існують деякі винятки з цього правила – наприклад, маніпуляції на основі зображень часто можуть виконуватися у фоновому потоці, але якщо є якісь труднощі з виконанням, тоді слід запускати цю задачу в головному потоці.
2. Довгі завдання, завжди повинні виконуватися у фоновому потоці.
3. Будь-які завдання, пов'язані з мережевим доступом, доступом до файлів або великими обсягами обробки даних, повинні виконуватися асинхронно з використанням об'єктів GCD або операцій.

Під час запуску ваше застосування повинне використовувати доступне час для установки свого призначеного для користувача інтерфейсу якомога швидше. У головному потоці повинні виконуватися тільки ті завдання, які сприяють налаштування призначеного для користувача інтерфейсу. Всі інші завдання повинні виконуватися асинхронно, результати яких мають бути показані користувачеві, як тільки вони будуть готові.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

20

### **2.1.3. Використовувані інструменти розробки**

#### **2.1.3.1. Середовище розробки**

Xcode – інтегроване середовище розробки під macOS і iOS, розроблена компанією Apple, що дозволяє створювати додатки для iPhone, iPad, Apple Watch, Mac і Apple TV. Першу версію анонсували в 2001 році. Програмне забезпечення є безкоштовним. Xcode містить в собі велику частину документації розробників від Apple. На момент написання використовувався Xcode 11.

Основні переваги Xcode:

- 1) Текстовий редактор (Source Editor) – інструмент дозволяє розширення коду, згортання коду, здійснює підсвічування синтаксису, відображає попередження, помилки та іншу контекстно-залежну інформацію, вбудовану в ваш код.
- 2) Каталог картинок (Asset Catalog) – інструмент дозволяє управляти зображеннями вашого додатка, групуючи дозволу одного і того ж ресурсу.
- 3) Редактор версії (Version Editor) – інструмент дозволяє відображати поточну тимчасову шкалу коммітів, допомагає визначити причину збою і дозволяє повертати файли до минулих коммітів для порівняння з вихідним файлом.
- 4) Симулятор (Simulator) – за допомогою iOS SDK XCode створювати, встановлювати, запускати і налагоджувати написані програми в симуляторі на базі Mac, що оптимізує робочий процес розробки.
- 5) Вбудований інтерфейс билдер (Interface Builder) – інструмент дозволяє створювати і тестувати інтерфейс без написання рядка коду.

#### **2.1.3.2. Мова програмування**

Swift - відкритий компільований мову програмування використовується в першу чергу для створення програмного забезпечення під iOS і Mac. Мова

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
|     |       |         |        |      |                     |      |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 21   |

сумісний з основною кодовою базою, написаної на Objective-C.

Переваги мови:

- 1) Більш легкий для читання і стійкий до помилок програміста мова, ніж передував йому Objective-C.
- 2) Більш безпечний. Swift містить безліч інструментів для роботи з опціональними типами, що дає програмісту впевненість в написаному їм коді.
- 3) Підтримує динамічні бібліотеки.
- 4) Swift набагато швидше, ніж його попередник Objective-C.
- 5) Спрощує успадкування класів.

## 2.2. Архітектура iOS додатків

Для початку потрібно познайомитися з двома основними компонентами iOS додатки це View і ViewController (Рис. 2.4).



Рис. 2.4. Взаємозв'язок головних компонентів

### 2.2.1. View

У додатках на платформі iOS за користувацький інтерфейс відповідає один або кілька компонентів уявлень (view). Загальний інтерфейс складається з безлічі компонентів (subviews) успадковані від загального класу View (Рис. 2.5).

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

22

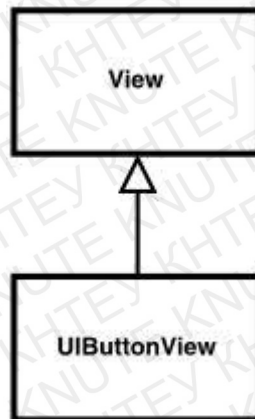


Рис. 2.5. Успадкування UIButtonView від View

До таких доступним елементів інтерфейсу відносяться: текстові поля, кнопки, таблиці, контейнери для картинок та інше.

Компоненти здатні групувати між собою елементи:

- 1) TableView – групує елементи TableViewCell у вигляді горизонтального списку, де в одному рядку буде знаходитися тільки один елемент.
- 2) CollectionView – групує елементи CollectionViewCell, як в горизонтальний, так і вертикальний список, в одному рядку може бути кілька елементів.
- 3) Horizontal Stack View і Vertical Stack View – стек елементів, що дозволяє з легкістю налаштувати загальні розміри, відступи елементів.
- 4) ScrollView – контейнер, може містити в собі будь-які елементи, а також розширюється в залежності від кількості елементів в ньому.

Існує кілька способів створення view:

- 1) Створювати і налаштовувати view програмно.
- 2) Використовувати інтерфейс білдер (Рис. 2.6).

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

23

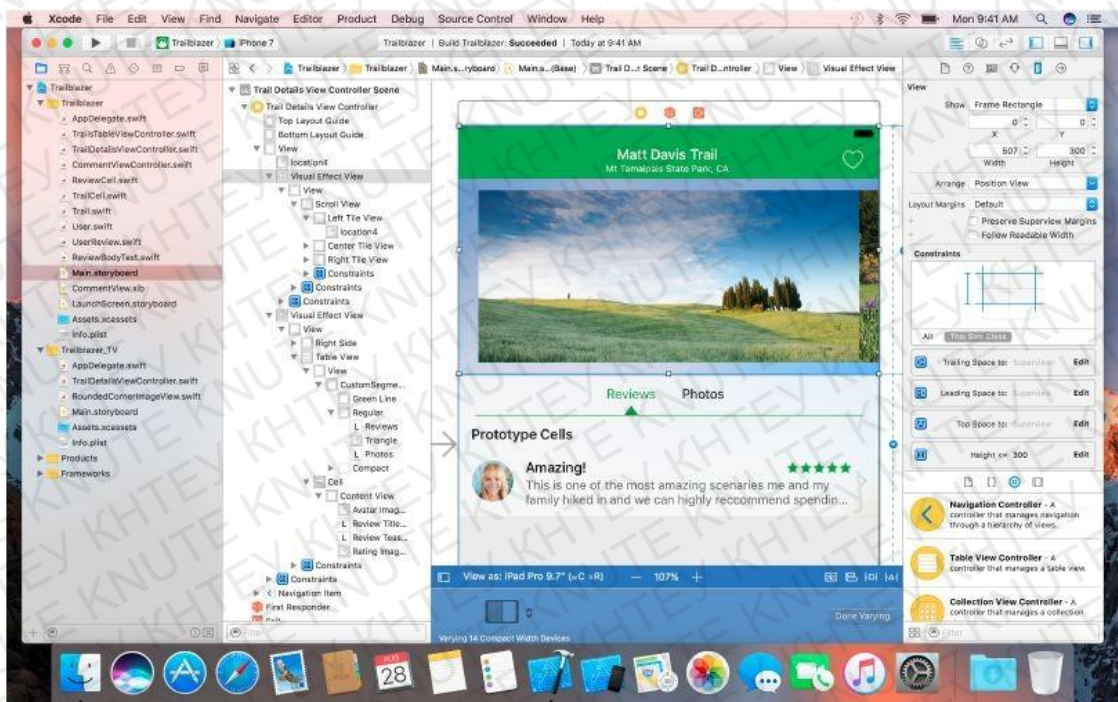


Рис. 2.6. Інтерфейс білдер

При створення своєї view надалі з її допомогою можна будувати інтерфейси різних екранів.

Основні вимоги до дизайну інтерфейсу мобільних додатків:

1) Мінімізація кількості елементів на екрані. Не слід перевантажувати невеликий простір дисплея мобільного пристрою великою кількістю об'єктів. Необхідно постаратися вмістити максимум функціоналу в лаконічний і доброзичливий інтерфейс.

2) Управління сучасними смартфонами з сенсорними екранами здійснюється за допомогою пальця. З огляду на те, що площа дотику пальця набагато більше розмірів покажчика комп'ютерної миші, інтерфейс не повинен містити дрібних елементів. Якщо ж наприклад кнопка має маленькі розміри, що ускладнює потрапляння по ній пальцем, то слід збільшити простір навколо кнопки, яке також обробляє дотику.

3) Розмір тексту в полях додатки повинен бути досить великий, щоб була можливість з легкістю його читати на віддаленому відстань від пристрою.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

24

### 2.2.2. ViewController

ViewControllers – компоненти iOS додатку, що відповідають за управління набором view і займаються навігацією за додатком [5]. Зазвичай компоненти створюються ізольовано один від одного, мають різні уявлення [6]. Так наприклад один контролер може показувати список елементів, а в той час інший контролер буде відображати обраний елемент з цього списку (Рис. 2.7).

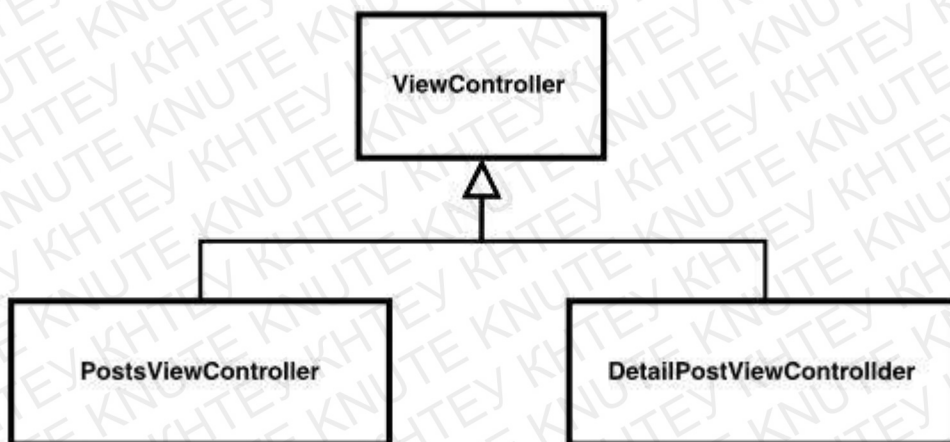


Рис. 2.7. Успадкування контролерів

ViewController виконує такі завдання як:

- 1) Оновлює зміст view, звичайно при оновленні будь-яких даних, оновлюється і уявлення.
- 2) Реагує на взаємодію користувача з інтерфейсом.
- 3) Змінює розміри і перемальовує уявлення в залежності від отриманих подій: переверот екрану, додавання нового елемента та інші.

## 2.3. MVVM u Coordinators в iOS приложениях

### 2.3.1. MVVM

MVVM - шаблон застосовують для проектування архітектури додатку, орієнтований на сучасні платформи розробки [7]. Реалізація патерну для iOS архітектури містить 4 елементи: ViewController, View, Model, ViewModel

(Рис. 2.8).

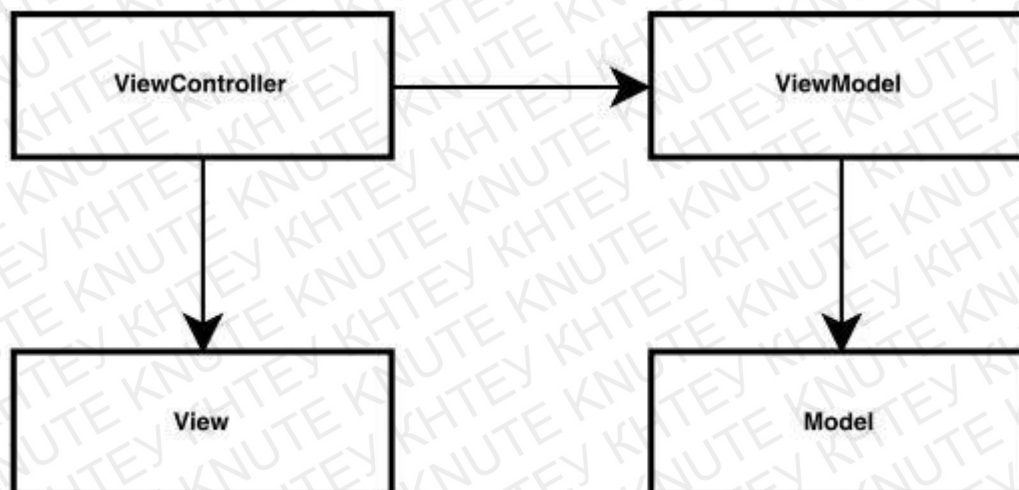


Рис. 2.8. Модель MVVP в iOS додатку

MVVM застосовується для поділу моделі і її уявлення, що необхідно для зміни їх окремо один від одного. Наприклад, розробник працює з логікою даних, в той час дизайнер відповідно розробляє призначений для користувача інтерфейс. MVVM зручно використовувати замість класичного MVC і йому подібних в тих випадках, коли в платформі, на якій ведеться розробка, присутня явна зв'язаність між призначеним для користувача інтерфейсом і бізнес логікою.

### 2.3.2. Coordinator

Coordinator (Координатор) - патерн дозволяє інкапсулювати всю навігацію по додатком [8]. Зазвичай відповідальний за перехід на наступний контролер був поточний контролер. Об'єкт координатор дозволяє взяти всі ці обов'язки на себе. Він візьме на себе управління переходами між контролерами, що сприяє збільшенню слабких залежностей і правильного перевикористання коду. Координатор може управляти також координаторами, доданими до головного батьківського координатору. Розглянемо основний інтерфейс координатора (Рис. 2.9)

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
|     |       |         |        |      |                     |      |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 26   |



Рис. 2.9. Інтерфейс координатора

Зазвичай додаток стартує з головного координатора, який в залежності від стану покаже необхідний додатком координатор. Об'єднавши координатор з патерном MVVM ми отримаємо наступну структуру (Рис. 2.10)

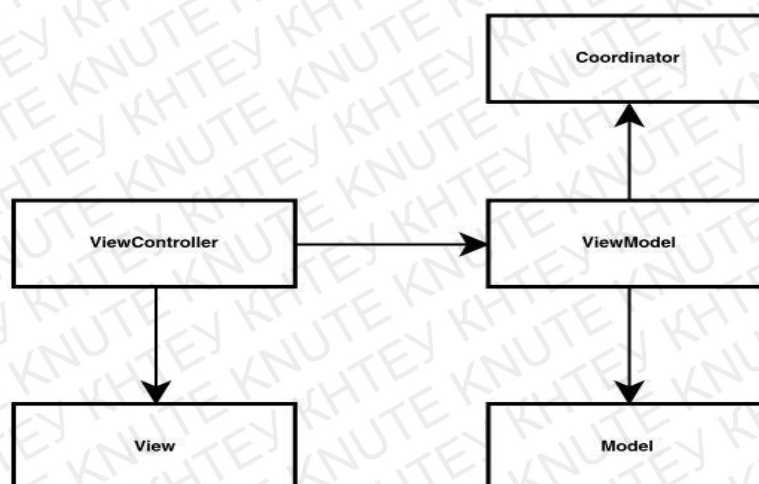


Рис. 2.10. Модель MVVM и Coordinator

Розглянемо обов'язки кожного елемента архітектури докладніше:

- 1) View (Представлення) - відображає призначений для користувача інтерфейс.
- 2) Model (Модель) – містить в собі призначені для користувача дані, так ж виробляє з ними маніпуляції і повідомляє про всі зміни ViewModel.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк

27

3) ViewModel (Модель подання) – отримує дані від моделі і перетворює їх в потрібний вид. Отримує повідомлення від контролера і передає їх координатору, для коректної навігації в додатку.

4) ViewController (Контролер) – запитує дані у ViewModel і використовуючи їх формує уявлення потрібним чином. Обробляє і передає події вчинені користувачем ViewModel.

5) Coordinator (Координатор) – обробляє повідомлення отримані від ViewModel, тим самим робить навігацію по додатком.

Позбавлення контролеов від обязанностей роботи з моделью даних і навігації за додатком, робить їх набагато читабельнее і дозволяє використовувати їх в інших проектах. Избавление контролеов от обязанности работы с моделью данных и навигации по приложению, делает их намного читабельнее и позволяет использовать их в других проектах.

#### **2.4. Висновки до розділу 2**

В другому розділі ми розглянули детально всі шари iOS архітектури, переглянули життєвий цикли мобільного додатку, як відбувається компіляція додатку, та обрали архітектурний патерн який будемо застосовувати для дизайну. Обрали мову програмування котра відповідає сучасним вимогам. Розглянули переваги середі розробки iOS додатків, яку підтримує Apple. Для оптимізація екранів застосовуємо Кординатор.

## РОЗДІЛ 3

### РОЗРОБКА ДОДАТКУ

#### 3.1. Архітектура системи

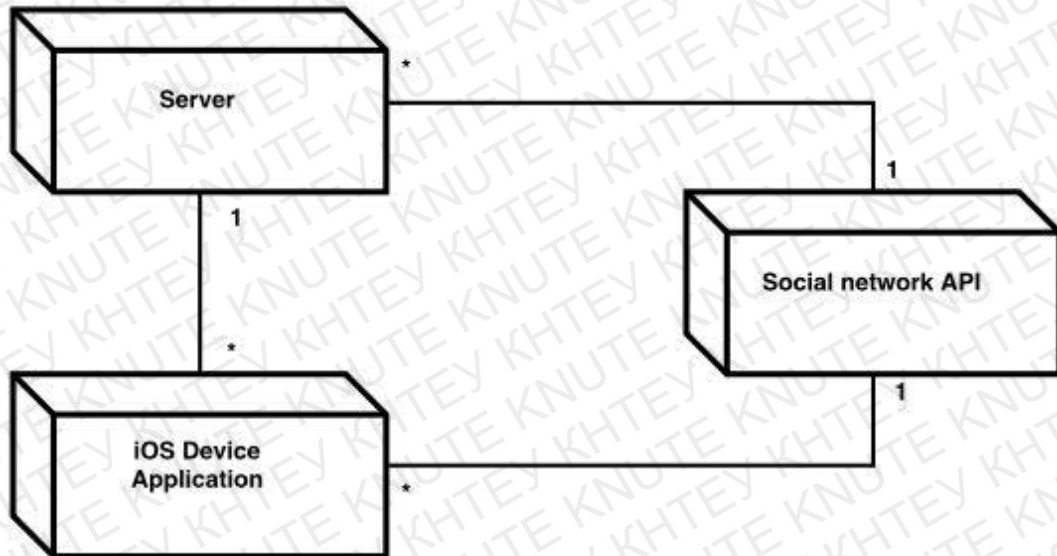


Рис. 3.1. Діаграма розгортання системи

Розглянемо систему розгортання (Рис. 3.1). Система складається з 3 частин:

- 1) Додатки для платформи iOS.
- 2) WebService
- 3) Група серверів соціальних мереж, що надають API.

У даній дипломній роботі сервер не розглядається, тому що він реалізований іншим розробником.

Додаток взаємодіє з сервером по https протоколу. Додаток відправляє на сервер post, get, upload запити. Сервер і додаток обмінюються інформацією у вигляді текстового формату json.

Переваги використання json формату:

|              |                 |         |        |            |                                                                                                       |                                                      |       |         |
|--------------|-----------------|---------|--------|------------|-------------------------------------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |                 |         |        |            | <i>КНТЕУ 121 02з-10.МР</i>                                                                            |                                                      |       |         |
| Зм.          | Аркуш           | № докум | Підпис | Дата       | Проектування та розробка<br>соціальної мережі для бігунів на<br>платформі iOS<br><br>Розробка додатку | Стадія                                               | Аркуш | Аркушів |
| Зав. кафедри | Криворучко О.В. |         |        | 21.06.2021 |                                                                                                       | РЗ                                                   | 29    | 41      |
| Керівник     | Харченко О.А.   |         |        | 21.06.2021 |                                                                                                       | Факультет інформаційних технологій, 2 курс, 2м група |       |         |
| Гарант       | Токар В.В.      |         |        | 21.06.2021 |                                                                                                       |                                                      |       |         |
| Розроб.      | Марценюк А.С.   |         |        | 21.06.2021 |                                                                                                       |                                                      |       |         |

- 1) Легко читається людьми (Рис. 3.2)
- 2) Більш відповідний для серіалізації складних структур, ніж XML.
- 3) В серед iOS розробки існує безліч бібліотек, спрощують серіалізацію.

```
{
  "firstName": "Иван",
  "lastName": "Иванов",
  "address": {
    "streetAddress": "Московское ш., 101, кв.101",
    "city": "Ленинград",
    "postalCode": 101101
  },
  "phoneNumbers": [
    "812 123-1234",
    "916 123-4567"
  ]
}
```

Рис. 3.2. Приклад json відповіді

Додаток такоємодіє з серверами соціальних мереж. Деякі соціальні мережі такі як Facebook, Google+ надають SDK вбудовуються в проект. SDK спрямовані на інтеграцію додатки з соціальною мережею, що істотно спрощує розробку. Але у більшості соціальних мереж немає свого SDK, тоді потрібно використовувати їх API і створювати свої веб запити.

### 3.2. Структура програми

Додаток починає працювати з виклику applications в AppDelegate. В цьому методі ми створимо AppCoordinator і виконаємо метод didFinishLaunchingWithOptions (Рис. 3.3).

```
func application(_ application: UIApplication,
                 didFinishLaunchingWithOptions launchOptions:
                 [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    let keyWindow = UIWindow(frame: UIScreen.main.bounds)
    window = keyWindow
    appCoordinator = AppCoordinator(window: keyWindow)
    appCoordinator?.start()
    return true
}
```

Рис. 3.3. Запуск головного координатора

|     |       |         |        |      |                     |      |
|-----|-------|---------|--------|------|---------------------|------|
|     |       |         |        |      | КНТЕУ 121 023-10.МР | Арк. |
| Зм. | Аркуш | № докум | Підпис | Дата |                     | 30   |

AppCoordinator в свою чергу містить і управляє 3 координаторами (Рис. 3.4).

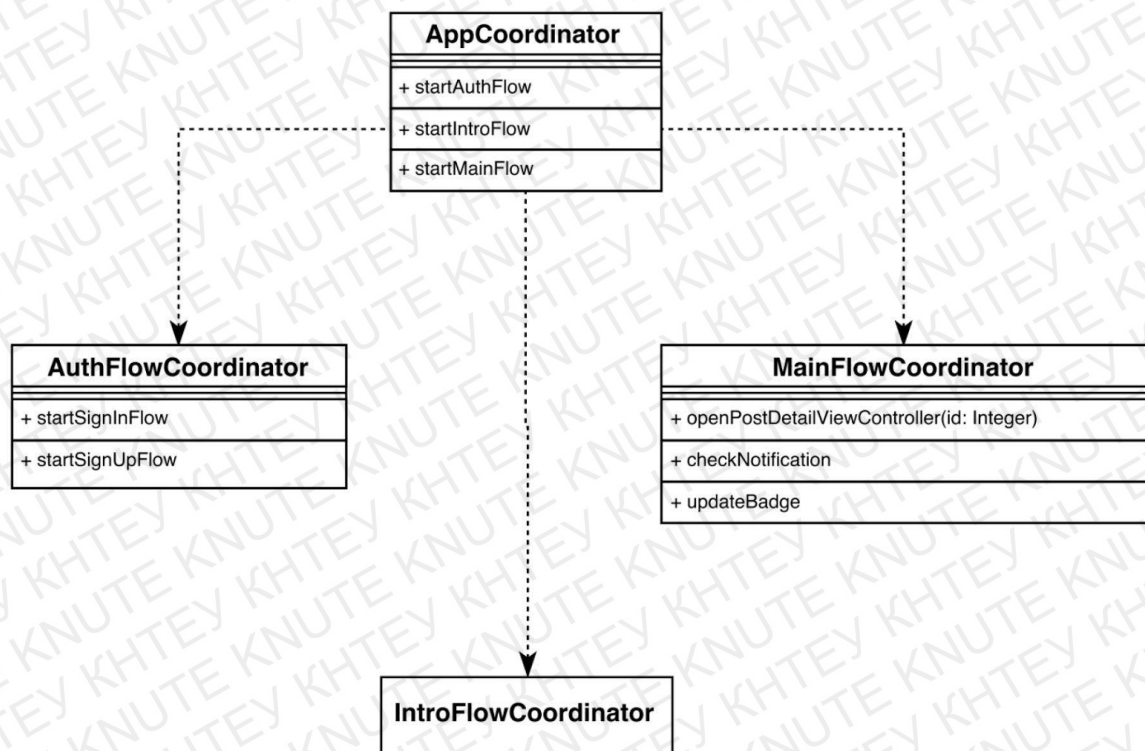


Рис. 3.4. Діаграма класів, показує залежності між координаторами

Залежно від стану головний координатор вирішує, який показати.

Якщо користувач перший раз запустив додаток, то AppCoordinator виконає метод `startIntroFlow`. Це в свою чергу запустить роботу IntroFlowCoordinator, який відповідає за показ навчальних слайдів.

Якщо користувач не авторизований в системі, то AppCoordinator виконає метод `startAuthFlow`, тим самим запустить роботу AuthFlowCoordinator. Цей координатор відповідає за авторизацію і реєстрацію користувача в додатку.

Якщо користувач вже авторизований в додатку, то AppCoordinator запустить MainFlowCoordinator, який відповідає за основний функціонал програми.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

31

### 3.2.1. AuthFlowCoordinator

AuthFlowCoordinator – координатор програми відповідає за навігацію при авторизації і реєстрації в додатку. Розглянемо його структуру докладніше (Рис. 3.5).

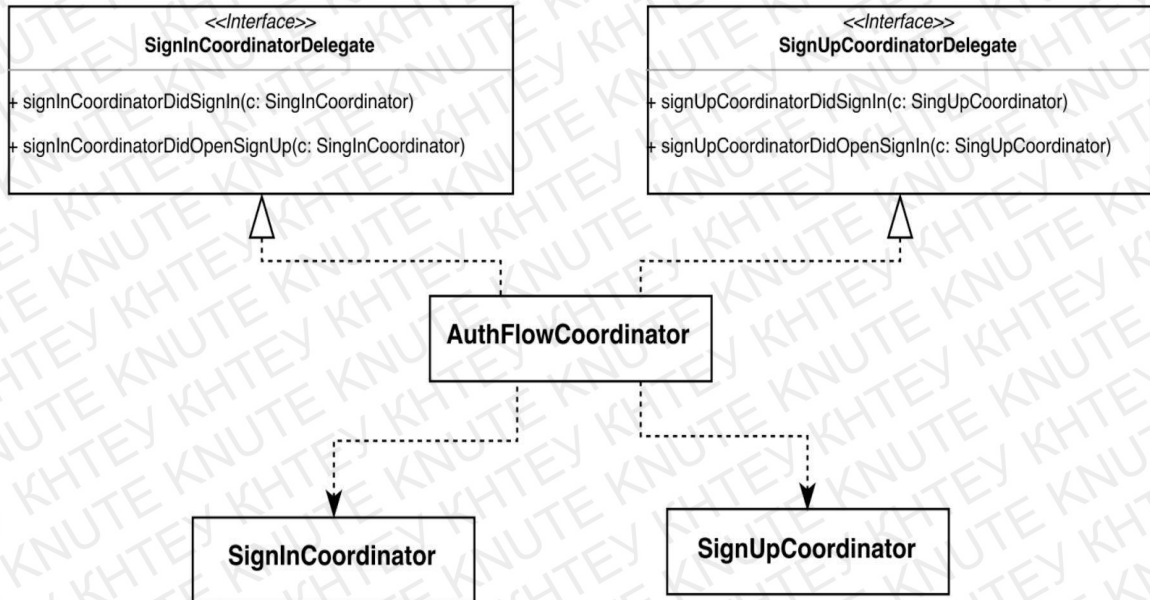


Рис. 3.5. Діаграма класів, показує залежності AuthFlowCoordinator

Як ми можемо помітити, AuthFlowCoordinator містить і управляє 2 координаторами, а також реалізує два інтерфейсу. Головний координатор при необхідності заміщає один координатор іншим.

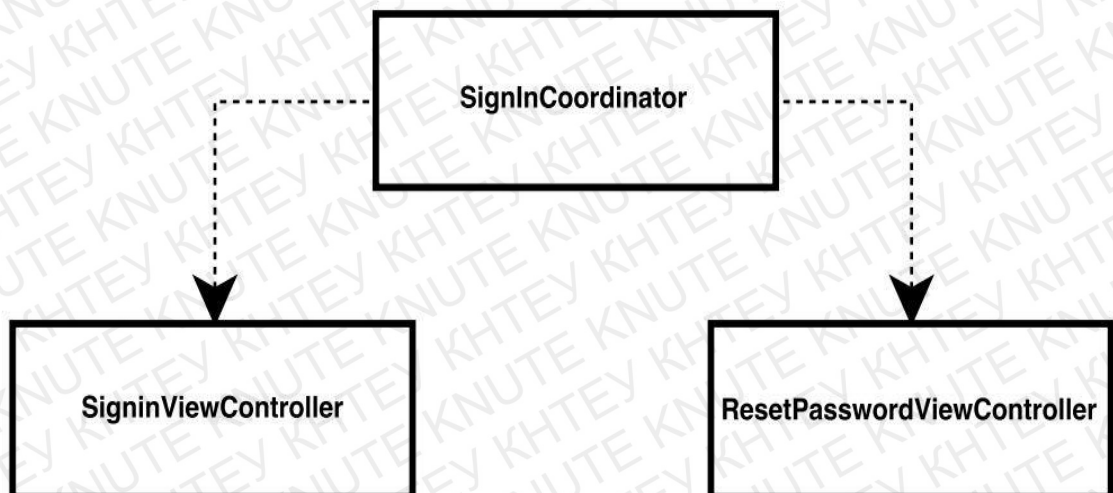


Рис. 3.6. Діаграма класів, показує залежності SignInCoordinator

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

32

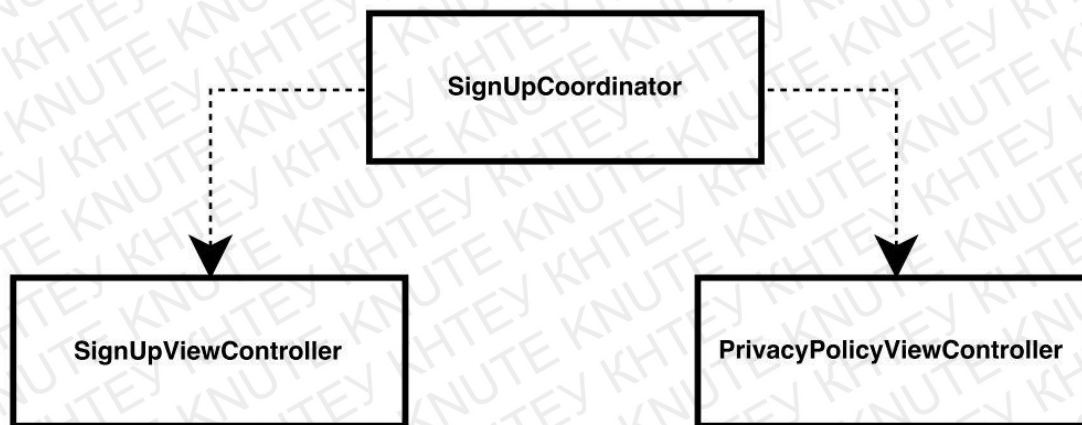


Рис. 3.7. Діаграма класів, показує залежності SignUpCoordinator з контролерами

ViewModel кожного контролера має посилання на свій координатор. Розглянемо кожен контролер докладніше:

- 1) SignInViewController: необхідний для реалізації самої авторизації.
- 2) ResetPasswordViewController: необхідний для відновлення пароля.
- 3) SignUpViewController: необхідний для самої реєстрації в додатку.
- 4) PrivacyPolicyViewController: необхідний для перегляду політики конфіденційності додатку.

Після успішної авторизації або реєстрації, відповідний координатор повідомляє про це AuthFlowCoordinator, який в свою чергу повідомляє це головному координатору, які вже вирішувє показувати чи головний функціонал додатка.

### 3.2.2. IntroFlowCoordinator

IntroFlowCoordinator – відповідає за навігацію між екранами в додатку, що допомагають новому користувачі з легкістю розібратися з основним функціоналом.

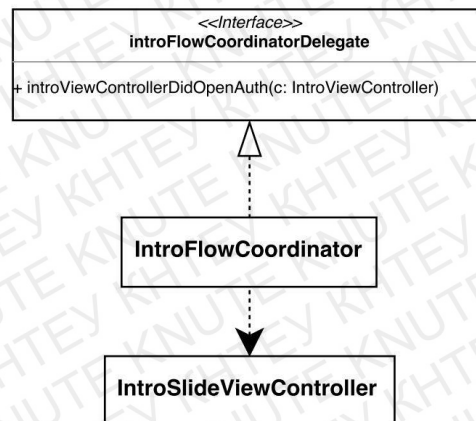


Рис. 3.8. Діаграма класів, показує залежності SignUpCoordinator

Координатор керує IntroSlideViewController і реалізує інтерфейс IntroFlowCoordinatorDelegate. IntroSlideViewController: перемикає view, реагуючи на жести користувача (Рис. 3.8). Можна переміщатися вліво і вправо. На останній view здійснюється перехід на координатор AuthFlowCoordinator.

### 3.2.3. MainFlowCoordinator

MainFlowCoordinator - відповідає за навігацію між основними екранами приложенія. Координатор приймає кілька інтерфейсів і має метод openPostDetailViewController, який дозволяє переглядати пост прийшов в push повідомлення (Рис. 3.9). Незалежно на якому екрані ми будемо знаходитися, контролер PostDetailViewController буде відображатися вище всіх в ієрархії контролерів.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

34

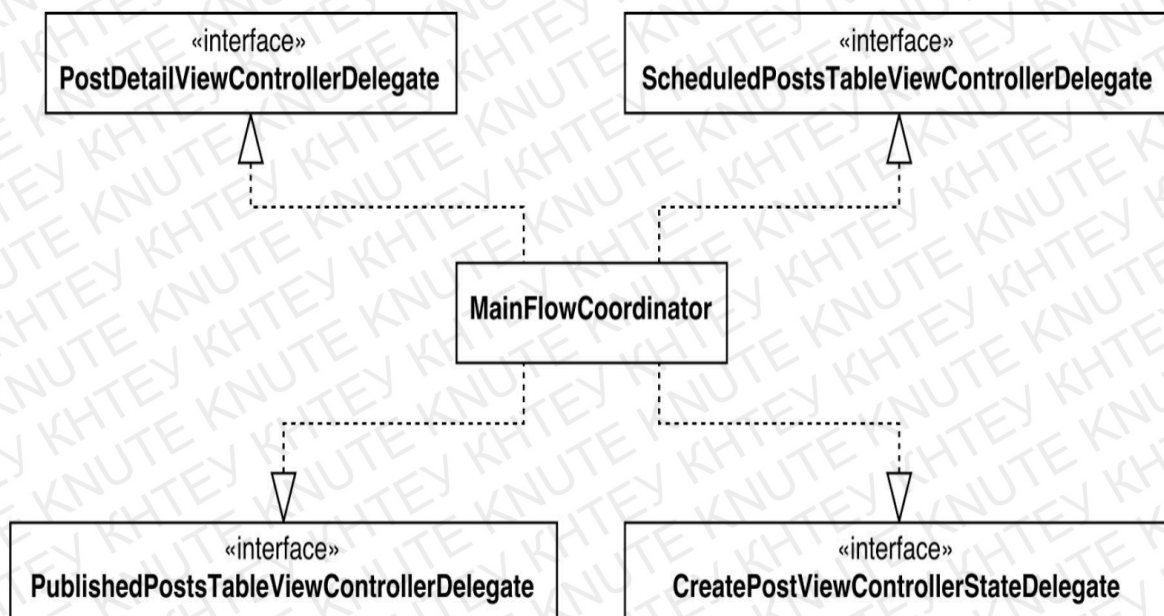


Рис. 3.9. Діаграма класів, показуюча залежності  
MainFlowCoordinator

MainFlowCoordinator взаємодіє з декількома контролерами, які обмінюються повідомленнями один з одним через делегати. Користувач управляє навігацією через висувний меню зліва, за це відповідає SlideMenuViewController (Рис. 3.10).

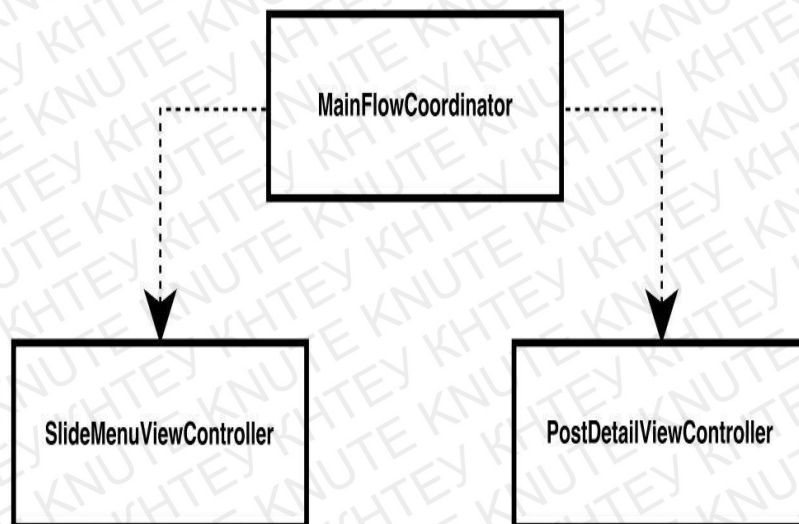


Рис. 3.10. Діаграма класів, показує залежності SignUpCoordinator з  
котроллерами

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

35

SlideMenuViewController містить і керує безліччю окремих контролерів (Рис. 3.11).

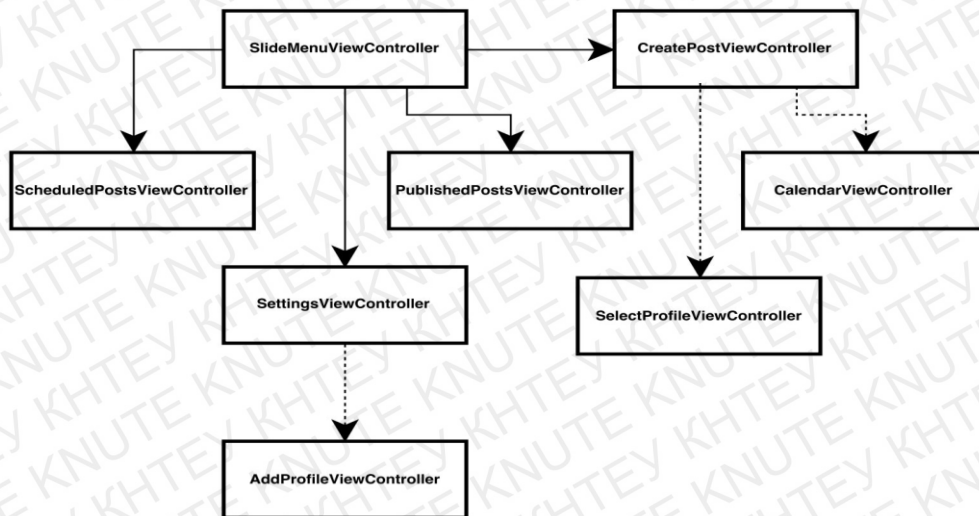


Рис. 3.11. Діаграма класів, що показує залежно SlideMenuViewController

Розглянемо кожен контролер докладніше:

- **ScheduledPostsViewController:** необхідний для відображення списку запланованих постів, відображається у вигляді таблиці, де кожна клітинка є постом.
- **PublishedPostsViewController:** необхідний для відображення списку опублікованих постів.
- **CreatePostViewController:** контролер дозволяє створювати нові пости.
- **SelectProfileViewController:** дозволяє вибрати один або кілька акаунтів соціальних мереж, що потрібно при створенні поста.
- **CalendarViewController:** контролер який дозволяє вибрати день і виставити час для посту, під час його створення.
- **SettingsViewController:** дозволяє змінювати профіль користувача аккаунта.
- **AddProfileViewController:** дозволяє додати новий обліковий запис для соціальних мереж.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

36

### 3.3. Інтеграція соціальних мереж

Соціальні мережі має різні технології авторизації і публікації постів. Потрібно навчитися взаємодіяти з різними API і SDK. Для інтеграції соціальних мереж, наведемо функціонал до загального класу SocialNetwork. Для кожної соціальної мережі створимо успадкований клас від SocialNetwork і реалізуємо кожен метод в залежності від технології (Рис. 3.12). Зберігати всі об'єкти будемо в упорядкованому масиві. Як тільки потрібно доступ до певної соціальної мережі, з масиву дістається заздалегідь сконфігурованих об'єкт.

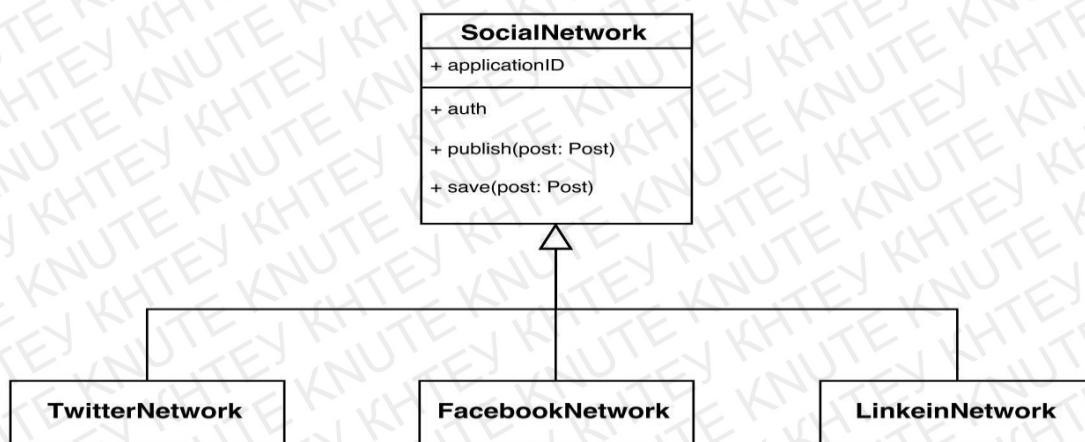


Рис. 3.12. Діаграма класів соціальних мереж

У соціальній мережі Instagram немає можливості публікації постів через API. Єдиний спосіб, це відправляти зміст посту заздалегідь встановленої програми Instagram і використовувати його публікувати.

|     |       |         |        |      |
|-----|-------|---------|--------|------|
|     |       |         |        |      |
| Зм. | Аркуш | № докум | Підпис | Дата |

КНТЕУ 121 023-10.МР

Арк.

37

### 3.4. Тестування розробленого додатка

Тестування програми було вироблено на пристроях з різними характеристиками (Таблиця 1).

Таблиця 1

#### Тестові пристрої

| Пристрій      | Дозвіл      | Діагональ | Версія iOS |
|---------------|-------------|-----------|------------|
| iPhone 6s     | 1334 × 750  | 4.7"      | 13.4       |
| iPhone 8      | 1334 × 750  | 4.7"      | 12.2       |
| iPhone 8 Plus | 1920 × 1080 | 5.5"      | 12.1       |
| iPhone X      | 2436×1125   | 5.8"      | 12.5       |
| iPhone XR     | 1792 × 828  | 6.1"      | 13.1       |
| iPhone XS Max | 2688 ×1242  | 6.5"      | 13.5       |
| iPad Pro      | 2732 x 2080 | 12.9"     | 10         |

При тестуванні приділяється увага характеристики: діагональ екрана, дозвіл екрана, версія ОС iOS.

Під час процесу тестування були виявлені і своєчасно усунені такі помилки:

1. Некоректне відображення призначеного для користувача інтерфейсу: через великий діагоналі на пристроях iPad деякі елементи інтерфейсу розташовувалися в неправильному місці, а також розтягувалися.
2. Відмінності в версіях операційної системи вплинуло на push

повідомлення, в 9 і 10 версіях iOS SDK надає різну обробку приходять повідомлень, що вимагало реалізації двох варіантів.

3. Некоректно відображення анімацій переходу між екранами.
4. Помилка при конвертації відео і фото.

### **3.5. Висновки до розділу 3**

В третьому розділі було створено зв'язок з сервером котрий буде зберігати данні про користувача. Створили навігацію по проекту між контролерами. Інтегрували дані до соціальних мереж та протестували додаток на коректність різних інтерфейсів на системі iOS.

|     |       |         |        |      |                            |      |
|-----|-------|---------|--------|------|----------------------------|------|
|     |       |         |        |      | <b>КНТЕУ 121 023-10.МР</b> | Арк. |
|     |       |         |        |      |                            |      |
| Зм. | Аркуш | № докум | Підпис | Дата |                            | 39   |

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У виконаній роботі:

- 1) Проаналізовано існуючі рішення.
- 2) Вивчено особливості мобільної платформи iOS.
- 3) Сформовано вимоги до розробляється з додатком.
- 4) Вивчено взаємодію з різними API соціальних мереж.
- 5) Спроектowana архітектура додатку.
- 6) Реалізовано додаток.
- 7) Протестовано додаток.

В роботі розроблено iOS-додаток що, дозволяє відстежувати дані забігу, ділитися, редагувати і втсановлювати цілі, тобто поставлена мета роботи була виконана. Крім того, в рамках даної роботи були проаналізовані технології взаємодії з соціальними мережами. В даний момент в додаток інтегровано в 5 соціальних мереж. Однак реалізований простий приклад додавання нових мереж, що дозволить в майбутньому розширити функціональність.

Результати роботи – вивчене середовище розробки Xcode для пристроїв від компанії Apple; досліджена архітектура ОС iOS; проаналізовані API соціальних мереж; спроектована структура мобільного додатка; розроблено мобільний додаток, доступне через AppStore.

|              |       |                 |        |            |                                                                         |                                                      |       |         |
|--------------|-------|-----------------|--------|------------|-------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |       |                 |        |            | <i>КНТЕУ 121 023-10.МР</i>                                              |                                                      |       |         |
| Зм.          | Аркуш | № докум         | Підпис | Дата       | Проектування та розробка соціальної мережі для бігунів на платформі iOS | Стадія                                               | Аркуш | Аркушів |
| Зав. кафедри |       | Криворучко О.В. |        | 01.11.2021 |                                                                         | ВП                                                   | 40    | 41      |
| Керівник     |       | Харченко О.А.   |        | 01.11.2021 |                                                                         | Факультет інформаційних технологій, 2 курс, 23 група |       |         |
| Гарант       |       | Токар В.В.      |        | 01.11.2021 |                                                                         |                                                      |       |         |
| Розроб.      |       | Марценюк А.С.   |        | 01.11.2021 | Висновки та пропозиції                                                  |                                                      |       |         |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фаулер М., Архітектура корпоративних програмних додатків. // М.Фаулер. – М. : Видавничий будинок "Вільяме", 2006. – 544 с.
2. iOS Developer Library URL:  
<http://developer.apple.com/library/ios/navigation/> (Дата звернення: 10.11.2019).
3. Life Cycle [Електронний ресурс] // Apple Inc. – Електрон. дан. – [Б.м.], 2017 – URL:  
<https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle.html>
4. Thread [Електронний ресурс // Google Inc. – Електрон. дан.– [Б.м.], 2020 – URL: <https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/Multithreading/>(Дата звернення: 24.03.2020).
5. ViewController [Електронний ресурс] // Apple Inc. – Електрон. дан. – [Б.м.], 2019 – URL:  
<https://developer.apple.com/library/content/featuredarticles/ViewControll er/>  
 (Дата звернення: 10.11.2019).
6. Introduction to MVVM [Електронний ресурс] – Електрон. дан. – Режим доступу: <https://www.objc.io/issues/13-architecture/mvvm/>, вільний.
7. An iOS Coordinator Pattern [Електронний ресурс] – Електрон. дан. – Режим доступу: <https://will.townsend.io/2016/an-ios-coordinator-pattern>, вільний.

|              |                 |         |        |            |                                                                         |                                                      |       |         |
|--------------|-----------------|---------|--------|------------|-------------------------------------------------------------------------|------------------------------------------------------|-------|---------|
|              |                 |         |        |            | <i>КНТЕУ 121 02з-10.МР</i>                                              |                                                      |       |         |
| Зм.          | Аркуш           | № докум | Підпис | Дата       | Проектування та розробка соціальної мережі для бігунів на платформі iOS | Стадія                                               | Аркуш | Аркушів |
| Зав. кафедри | Криворучко О.В. |         |        | 27.02.2021 |                                                                         | СД                                                   | 41    | 41      |
| Керівник     | Харченко О.А.   |         |        | 27.02.2021 |                                                                         | Факультет інформаційних технологій, 2 курс, 23 група |       |         |
| Гарант       | Токар В.В.      |         |        | 27.02.2021 |                                                                         |                                                      |       |         |
| Розроб.      | Марценюк А.С.   |         |        | 27.02.2021 |                                                                         |                                                      |       |         |
|              |                 |         |        |            | Список використаних джерел                                              |                                                      |       |         |

## ДОДАТКИ

### Додаток А

```
import
UIKit

enum AMLoginSignupViewMode {
    case login
    case signup
}

class ViewController: UIViewController {

    let animationDuration = 0.25
    var mode:AMLoginSignupViewMode = .signup

    //MARK: - background image constraints
    @IBOutlet weak var backgroundImageLeftConstraint: NSLayoutConstraint!
    @IBOutlet weak var backgroundImageBottomConstraint: NSLayoutConstraint!

    //MARK: - login views and constrains
    @IBOutlet weak var loginView: UIView!
    @IBOutlet weak var loginContentView: UIView!
    @IBOutlet weak var loginButton: UIButton!
    @IBOutlet weak var loginButtonVerticalCenterConstraint: NSLayoutConstraint!
    @IBOutlet weak var loginButtonTopConstraint: NSLayoutConstraint!
    @IBOutlet weak var loginWidthConstraint: NSLayoutConstraint!

    //MARK: - signup views and constrains
    @IBOutlet weak var signupView: UIView!
    @IBOutlet weak var signupContentView: UIView!
    @IBOutlet weak var signupButton: UIButton!
    @IBOutlet weak var signupButtonVerticalCenterConstraint: NSLayoutConstraint!
    @IBOutlet weak var signupButtonTopConstraint: NSLayoutConstraint!

    //MARK: - logo and constrains
    @IBOutlet weak var logoView: UIView!
    @IBOutlet weak var logoTopConstraint: NSLayoutConstraint!
    @IBOutlet weak var logoHeightConstraint: NSLayoutConstraint!
    @IBOutlet weak var logoBottomConstraint: NSLayoutConstraint!
    @IBOutlet weak var logoBottomInSingupConstraint: NSLayoutConstraint!
    @IBOutlet weak var logoCenterConstraint: NSLayoutConstraint!
```

```

@IBOutlet weak var forgotPassTopConstraint: NSLayoutConstraint!
@IBOutlet weak var socialsView: UIView!

//MARK: - input views
@IBOutlet weak var loginEmailInputView: AMInputView!
@IBOutlet weak var loginPasswordInputView: AMInputView!
@IBOutlet weak var signupEmailInputView: AMInputView!
@IBOutlet weak var signupPasswordInputView: AMInputView!
@IBOutlet weak var signupPasswordConfirmInputView: AMInputView!

//MARK: - controller
override func viewDidLoad() {
    super.viewDidLoad()

    // set view to login mode
    toggleViewMode(animated: false)

    //add keyboard notification
    NotificationCenter.default.addObserver(self, selector:
#selector(keyboardFrameChange(notification:)), name: .UIKeyboardWillChangeFrame,
object: nil)
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
}

//MARK: - button actions
@IBAction func loginButtonTouchUpInside(_ sender: AnyObject) {

    if mode == .signup {
        toggleViewMode(animated: true)
    }else{

        //TODO: login by this data
        NSLog("Email:\(loginEmailInputView.textFieldView.text)
Password:\(loginPasswordInputView.textFieldView.text)")
    }
}

@IBAction func signupButtonTouchUpInside(_ sender: AnyObject) {

```

```

        if mode == .login {
            toggleViewMode(animated: true)
        } else {
            //TODO: signup by this data
            NSLog(@"Email:\(signupEmailInputView.textFieldView.text)
Password:\(signupPasswordInputView.textFieldView.text),
PasswordConfirm:\(signupPasswordConfirmInputView.textFieldView.text)")
        }
    }

    //MARK: - toggle view
    func toggleViewMode(animated: Bool) {
        // toggle mode
        mode = mode == .login ? .signup : .login

        // set constraints changes
        backImageLeftConstraint.constant = mode == .login ? 0 : -
self.view.frame.size.width

        loginWidthConstraint.isActive = mode == .signup ? true : false
        logoCenterConstraint.constant = (mode == .login ? -1 : 1) *
(loginWidthConstraint.multiplier * self.view.frame.size.width) / 2
        loginButtonVerticalCenterConstraint.priority = mode == .login ? 300 : 900
        signupButtonVerticalCenterConstraint.priority = mode == .signup ? 300 : 900

        //animate
        self.view.endEditing(true)

        UIView.animate(withDuration:animated ? animationDuration:0) {

            //animate constraints
            self.view.layoutIfNeeded()

            //hide or show views
            self.loginContentView.alpha = self.mode == .login ? 1 : 0
            self.signupContentView.alpha = self.mode == .signup ? 1 : 0

            // rotate and scale login button
            let scaleLogin:CGFloat = self.mode == .login ? 1 : 0.4

```

```

        let rotateAngleLogin:CGFloat = self.mode == .login ? 0:CGFloat(-M_PI_2)

        var transformLogin = CGAffineTransform(scaleX: scaleLogin, y: scaleLogin)
        transformLogin = transformLogin.rotated(by: rotateAngleLogin)
        self.loginButton.transform = transformLogin

        // rotate and scale signup button
        let scaleSignup:CGFloat = self.mode == .signup ? 1:0.4
        let rotateAngleSignup:CGFloat = self.mode == .signup ? 0:CGFloat(-M_PI_2)

        var transformSignup = CGAffineTransform(scaleX: scaleSignup, y:
scaleSignup)
        transformSignup = transformSignup.rotated(by: rotateAngleSignup)
        self.signupButton.transform = transformSignup
    }

}

//MARK: - keyboard
func keyboardFrameChange(notification:NSNotification){

    let userInfo = notification.userInfo as! [String:AnyObject]

    // get top of keyboard in view
    let topOfKeyboard = (userInfo[UIKeyboardFrameEndUserInfoKey] as!
NSNumber).cgRectValue .origin.y

    // get animation curve for animate view like keyboard animation
    var animationDuration:TimeInterval = 0.25
    var animationCurve:UIViewAnimationCurve = .easeOut
    if let animDuration = userInfo[UIKeyboardAnimationDurationUserInfoKey] as?
NSNumber {
        animationDuration = animDuration.doubleValue
    }

    if let animCurve = userInfo[UIKeyboardAnimationCurveUserInfoKey] as? NSNumber
{
        animationCurve = UIViewAnimationCurve.init(rawValue: animCurve.intValue)!
    }

    // check keyboard is showing
    let keyboardShow = topOfKeyboard != self.view.frame.size.height

```

```

//hide logo in little devices
let hideLogo = self.view.frame.size.height < 667

// set constraints
backImageBottomConstraint.constant = self.view.frame.size.height -
topOfKeyboard

logoTopConstraint.constant = keyboardShow ? (hideLogo ? 0:20):50
logoHeightConstraint.constant = keyboardShow ? (hideLogo ? 0:40):60
logoBottomConstraint.constant = keyboardShow ? 20:32
logoBottomInSingupConstraint.constant = keyboardShow ? 20:32

forgotPassTopConstraint.constant = keyboardShow ? 30:45

loginButtonTopConstraint.constant = keyboardShow ? 25:30
signupButtonTopConstraint.constant = keyboardShow ? 23:35

loginButton.alpha = keyboardShow ? 1:0.7
signupButton.alpha = keyboardShow ? 1:0.7

// animate constraints changes
UIView.beginAnimations(nil, context: nil)
UIView.setAnimationDuration(animationDuration)
UIView.setAnimationCurve(animationCurve)

self.view.layoutIfNeeded()

UIView.commitAnimations()
}

//MARK: - hide status bar in swift3
override var prefersStatusBarHidden: Bool {
    return true
}
}

```

