

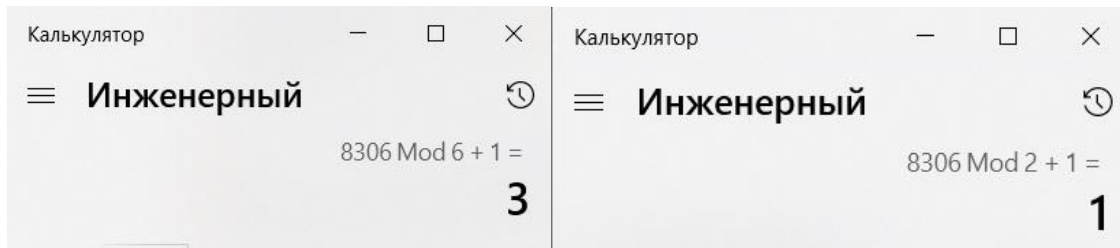
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота № 6
з дисципліни
“Програмування мобільних систем”

Виконав:
студент групи ІВ-83
ЗК ІВ-8306
Грисюк Михайло

Київ 2021

Варіант



Варіант 3
REQUEST – “small+animals” COUNT – 18

Варіант 1	
Предметна область – фільми URL-адреса для отримання даних – http://www.omdbapi.com/?apikey=API_KEY&s=REQUEST&page=1 , де API_KEY – 7e9fe69e, REQUEST – запит із поля для пошуку. Зверніть увагу на те, що використання ключа API_KEY обмежене до 1000 запитів на день, якщо буде вичерпано кількість запитів на день для наведеного ключа, то можна отримати власний ключ, зареєструвавшись за посиланням .	Предметна область – фільми URL-адреса для отримання даних – http://www.omdbapi.com/?apikey=API_KEY&i=IDENTIFIER , де API_KEY – 7e9fe69e, IDENTIFIER – ідентифікатор відповідного фільму (поле imdbID із сутності фільму). Зверніть увагу на те, що використання ключа API_KEY обмежене до 1000 запитів на день, якщо буде вичерпано кількість запитів на день для наведеного ключа, то можна отримати власний ключ, зареєструвавшись за посиланням .

Лістинг коду

NetworkManager.swift

```
//  
// NetworkManager.swift  
// Lab1-1  
//  
// Created by misha_hrysiuk on 07.05.2021.  
//  
import Foundation  
import UIKit  
  
protocol NetworkManagerDelegate {  
    func getMovies(with name: String, completion: @escaping (Data?, Error?)  
-> Void)  
    func getMovieDetails(with identifier: String, completion: @escaping  
(Data?, Error?) -> Void)  
    func getPhotos(with request: String, count: Int, completion: @escaping  
(Data?, Error?) -> Void)
```

```

        func getImage(with coverURL: String, completion: @escaping (UIImage?,
Error?) -> Void)
        func cancel()
    }

final class NetworkManager: NSObject, NetworkManagerDelegate,
NSURLSessionDelegate {
    static let sharad = NetworkManager()

    lazy var urlsession: URLSession = {
        let config = URLSessionConfiguration.default
        return URLSession(configuration: config)
    }()

    var datatask: URLSessionTask?
}

extension NetworkManager {
    func getMovies(with name: String, completion: @escaping (Data?, Error?)
-> Void) {
        let urlString =
"http://www.omdbapi.com/?apikey=7e9fe69e&s=\(name)&page=1"

        guard let components = URLComponents(string: urlString), let url =
components.url else {return}
        DispatchQueue.global(qos: .background).async {[weak self] in
            self?.datatask = self?.urlsession.dataTask(with:
URLRequest(url: url)) { (data, response, error) in
                guard let httpResponse = response as? HTTPURLResponse,
                    let data = data,
                    httpResponse.statusCode == 200 else {return}
                DispatchQueue.main.async {
                    completion(data, nil)
                }
            }
            self?.datatask?.resume()
        }

    }

    func getMovieDetails(with identifier: String, completion: @escaping
(Data?, Error?) -> Void) {
        let urlString =
"http://www.omdbapi.com/?apikey=7e9fe69e&i=\(identifier)"
        guard let components = URLComponents(string: urlString), let url =
components.url else {return}
        DispatchQueue.global(qos: .background).async { [weak self] in
            self?.datatask = self?.urlsession.dataTask(with:
URLRequest(url: url)) { (data, response, error) in
                guard let httpResponse = response as? HTTPURLResponse,
                    let data = data,
                    httpResponse.statusCode == 200 else {return}
                print(httpResponse.statusCode)
                DispatchQueue.main.async {
                    completion(data, nil)
                }
            }
            self?.datatask?.resume()
        }
    }
}

```

```

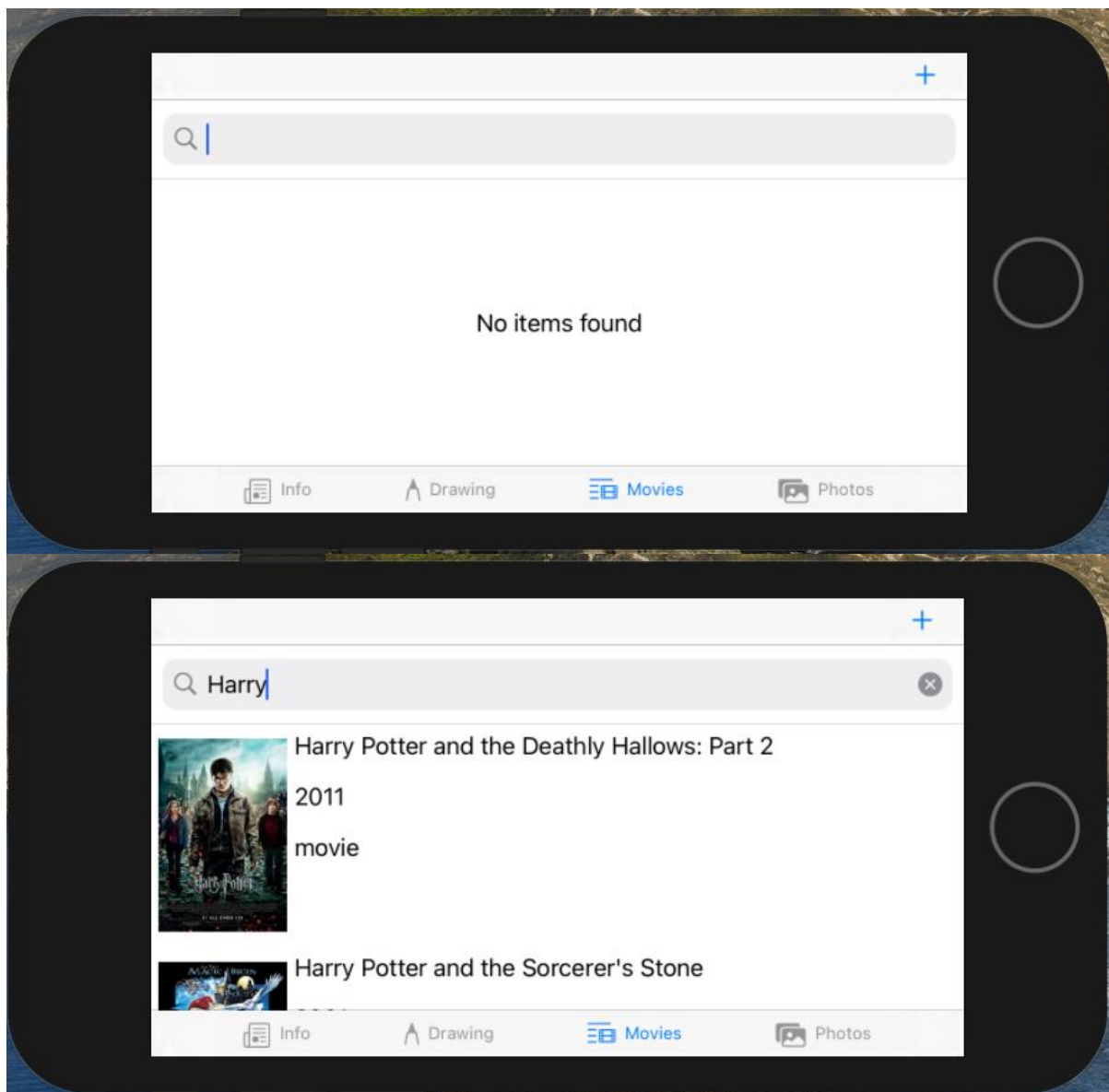
    func getPhotos(with req: String, count: Int, completion: @escaping
(Data?, Error?) -> Void) {
        let urlString = "https://pixabay.com/api/?key=19193969-
87191e5db266905fe8936d565&q=\(req)&image_type=photo&per_page=\(count)"
        guard let components = URLComponents(string: urlString), let url =
components.url else {return}
        DispatchQueue.global(qos: .background).async { [weak self] in
            self?.datatask = self?.urlsession.dataTask(with:
URLRequest(url: url)) { (data, response, error) in
                guard let httpResponse = response as? HTTPURLResponse,
                    let data = data,
                    httpResponse.statusCode == 200 else {return}
                DispatchQueue.main.async {
                    completion(data, nil)
                }
            }
            self?.datatask?.resume()
        }
    }

    func cancel() {
        datatask?.cancel()
    }

    func getImage(with imageURL: String, completion: @escaping (UIImage?,
Error?) -> Void) {
        guard let url = URL(string: imageURL) else {return}
        DispatchQueue.global(qos: .background).async { [weak self] in
            self?.datatask = self?.urlsession.downloadTask(with: url) {
(tempURL, response, error) in
                if let tempURL = tempURL,
                    let data = try? Data(contentsOf: tempURL),
                    let image = UIImage(data: data) {
                    DispatchQueue.main.async {
                        completion(image, nil)
                    }
                } else {
                    completion(nil, error)
                }
            }
            self?.datatask?.resume()
        }
    }
}

```

Скріншот роботи програми



< Back



Info

Drawing

Movies

Photos

< Back



Title: Harry Potter and the Deathly Hallows: Part 2

Year: 2011

Genre: Adventure, Drama, Fantasy, Mystery

Director: David Yates

Actors: Ralph Fiennes, Michael Gambon, Alan Rickman, Daniel Radcliffe

Country: UK

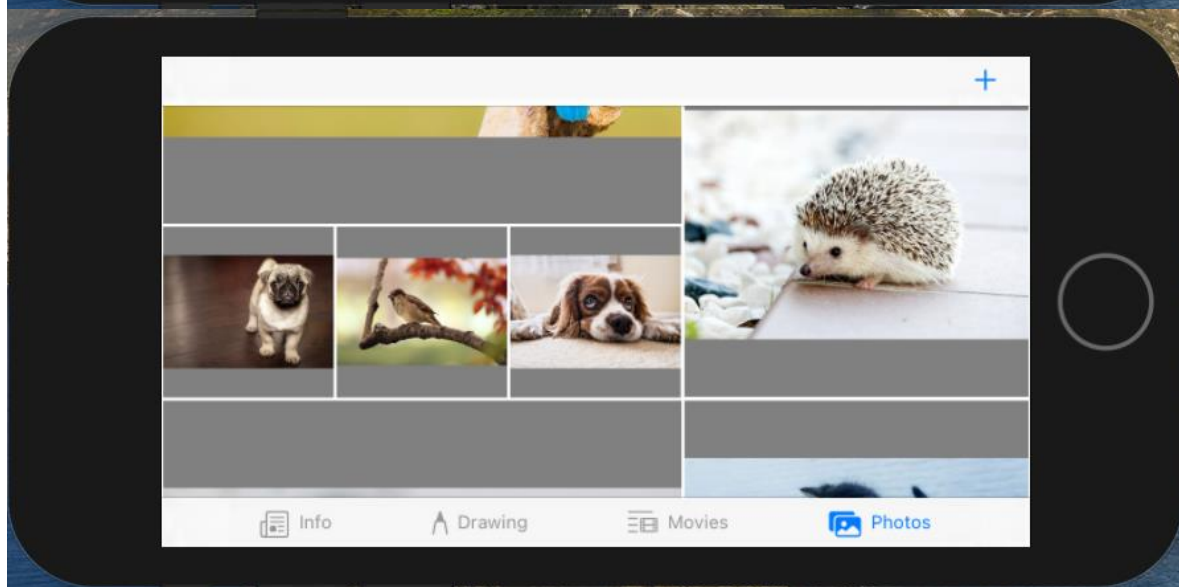
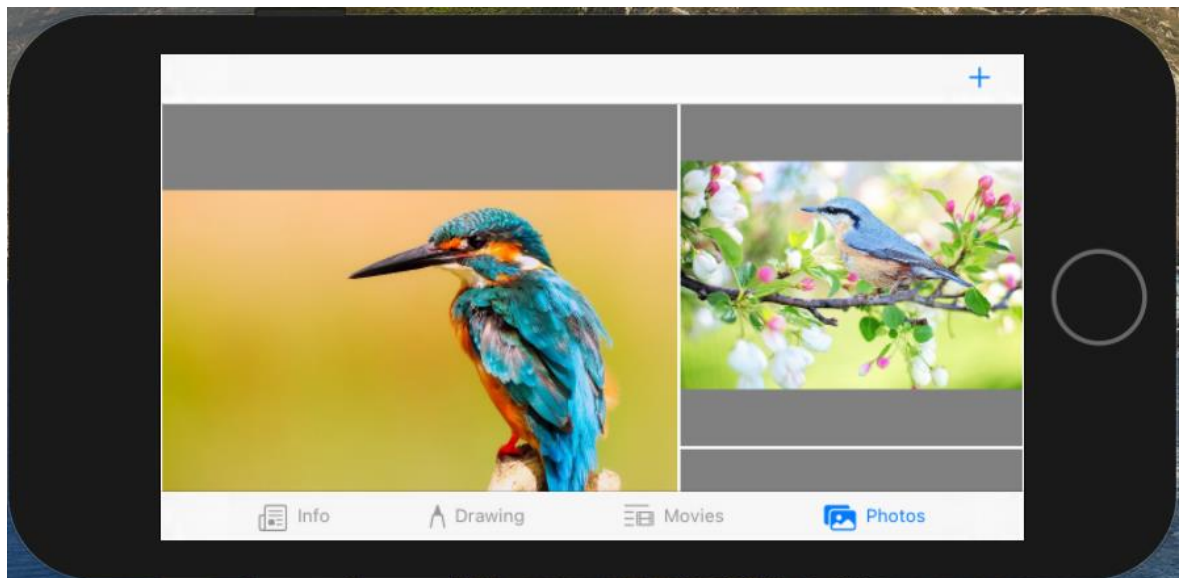
Language: English

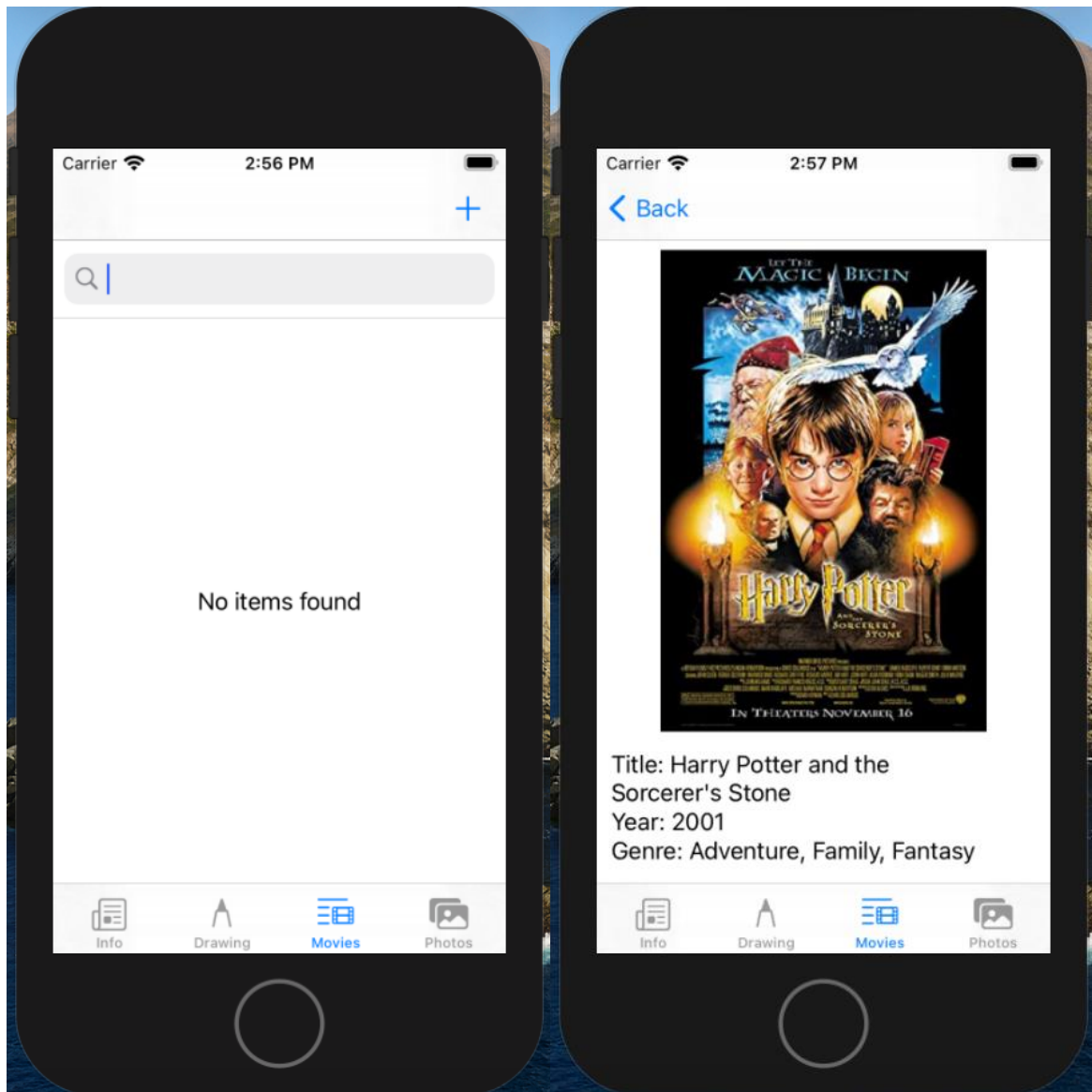
Info

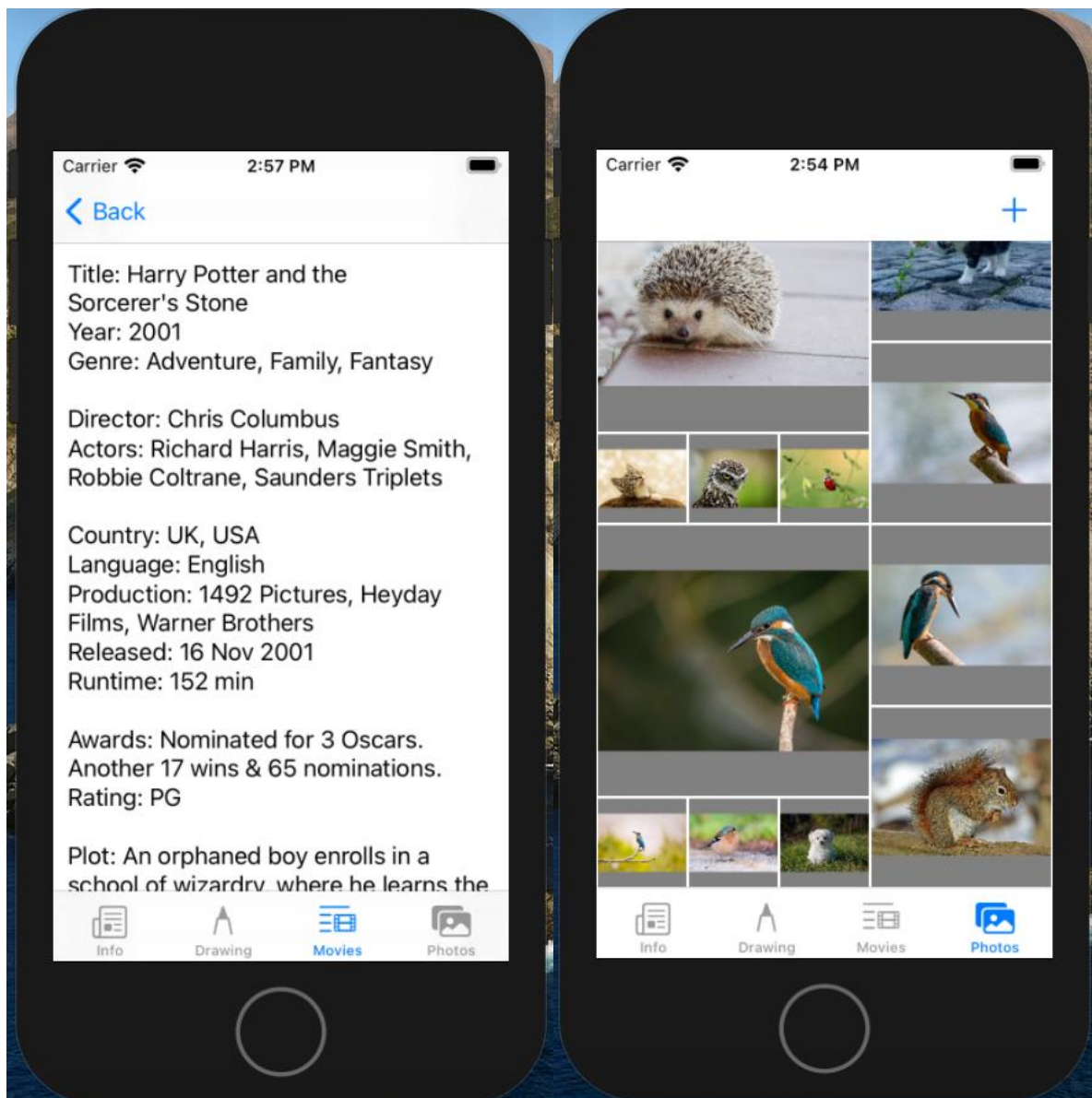
Drawing

Movies

Photos







Висновок

В цій лабораторній роботі була дороблена програма з лабораторної роботи №6. Було оновлено вкладки нову вкладку "Photos" та "Movies", в яких дані вже будуть братись з інтернету за варіантом. Було оновлено код з попередніх робіт та створені нові для правильного виконання завдання. Як видно на скріншотах програма працює коректно при вертикальній і горизонтальній положеннях.