Użycie języka Swift i elementów paradygmatu reaktywnego na przykładzie aplikacji mobilnej do polecania filmów i seriali

Bartosz Woliński

18 października 2016

Spis treści

Rozdział 1

WSTĘP

Rozdział 2

CEL I ZAKRES PRACY

Rozdział 3

ŹRÓDŁA I DEFINICJE

3.1 Historia systemu iOS

3.2 Historia języka Swift

3.3 Paragydmat funkcyjny

3.4 Paragydmat reaktywny

3.4.1 Idea programowania reaktywnego

3.4.2 Podejście reaktywne w praktyce

3.4.3 Operatory reaktywne

3.4.4 Współbieżność

3.4.5 Wady i zalety podejścia reaktywnego

3.4.6 Implementacja na platformie iOS

Rozdział 4

PRACA WŁASNA

4.1 Czynności przygotowawcze

Stworzenie aplikacji mobilnej było możliwe dzięki wykonaniu uprzednio czynności przygotowawczych. Do czynności tych należało: wybór środowiska programistycznego, instalacja bibliotek i frameworków, wstępna konfiguracja projektu oraz stworzenie repozytorium.

4.1.1 Instalacja narzędzi

Program XCode jest jednym z niewielu dostępnych środowisk programistycznych na platformę iOS. Jest on darmowy i dostarczany wraz systemem MacOS. XCode posiada wbudowany kompilator LLVM oraz szereg przydatnych narzędzi tj. interface builder - graficzny edytor do tworzenia elementów interfejsu, dynamiczną kontrolę składni czy menedżer systemu kontroli wersji. Ze względu na powyższe właściwości zdecydowałem się użyć właśnie tego środowiska programistycznego. Ponadto użyłem także narzędzia do rozwiązywania zależności - CocoaPods. Jest ono bardzo przydatne szczególnie przy instalacji bibliotek i frameworków do projektu.

4.1.2 Wstępna konfiguracja projektu

Podczas konfigurowania projektu w środowisku XCode istotne jest abyśmy stworzyli go z użyciem CoreData. Jest to baza danych środowiska iOS i może stanowić integralną część aplikacji.

4.1.3 Stworzenie repozytorium

Dostępnych jest wiele systemów kontroli wersji, ale najpopularniejszym z nich jest GIT. Zdecydowałem się na jego wykorzystanie, ponieważ jest dosyć prosty w użyciu a większość serwerów GITa jest darmowa. Użycie systemu typu GIT pozwoli mi na kontrolę postępów mojej pracy jak i na dokumentowanie jej.

Oprócz tego użycia GITa powoduje, że błąd popełniony przeze mnie na dalszym etapie mogę odwrócić przywracając poprzedni stan projektu.

4.2 Budowa aplikacji właściwej

4.2.1 Wymagania funkcjonalne

4.2.2 Wymagania niefunkcjonalne

4.2.3 Diagram przypadków użycia aplikacji

4.2.4 Zależności w bazie danych

4.2.5 Diagram klas

4.2.6 Opis wybranych klas

4.2.7 Opis użytych bibliotek i frameworków

4.2.8 Implementacja aplikacji mobilnej - zastosowana architektura

4.2.9 Prezentacja aplikacji

Rozdział 5

PODSUMOWANIE