

Uniwersytet Jagielloński w Krakowie  
Wydział Fizyki, Astronomii i Informatyki Stosowanej



**WITOLD KAROL BOBROWSKI**

NUMER ALBUMU: 1115454

## **CZYTNIK KSIĄŻEK ELEKTRONICZNYCH NA IOS**

PRACA LICENCJACKA NA KIERUNKU INFORMATYKA

PRACA WYKONANA POD KIERUNKIEM:  
dra Karola Przystalskiego  
Zakład Technologii Informatycznych

Kraków 2017

## **Oświadczenie autora pracy**

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Kraków, dnia

Podpis autora pracy

## **Oświadczenie kierującego pracą**

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Kraków, dnia

Podpis kierującego pracą

*podziękowania*

*i więcej podziękowań*

## **Streszczenie**

streszczenie

## **Abstract**

abstract

## Spis treści

<b>1. Wstęp</b>	6
<b>2. Wykorzystane technologie</b>	7
2.1. Xcode i Developer Tools	7
2.2. Swift	9
2.3. iOS SDK	9
2.4. Zewnętrzne biblioteki	9
<b>3. Charakterystyka EPUB</b>	10
3.1. Omówienie	10
3.2. Specyfikacja	10
<b>4. Framework EPUBKit</b>	11
4.1. Tworzenie frameworku na iOS	11
4.2. Model	11
4.3. Parser	11
4.4. Widok	11
4.5. Dystrybucja	11
<b>5. Aplikacja demonstracyjna</b>	12
5.1. Tworzenie aplikacji na iOS	12
5.2. Wykorzystanie frameworku w aplikacji	12
5.3. Publikacja	12
<b>6. Podsumowanie</b>	13

# 1. Wstęp

Mobilny system od Apple, iOS niedługo będzie obchodził 10 lat od wprowadzenia go na rynek. W 2014 roku Apple ogłosiło, że jest ponad miliard aktywnych urządzeń z tym właśnie systemem a dziś jest ich z pewnością znacznie więcej. Każdy kolejny model telefonu komórkowego marki Apple, prezentowany z roczną częstotliwością cieszy się coraz większym powodzeniem. Oprócz nowych urządzeń dostajemy w pakiecie nową odsłonę iOS która jest nie tylko udoskonaleniem poprzedniej wersji, ale również jej pełnoprawnym następcą wprowadzając nowy zbiór elementów zarówno funkcjonalnościowych jak i wizualnych. Tak dynamicznie rozwijający się system jest bardzo atrakcyjny dla użytkownika, który dzięki darmowym aktualizacjom dla starszych urządzeń wciąż może cieszyć się najnowszym oprogramowaniem.

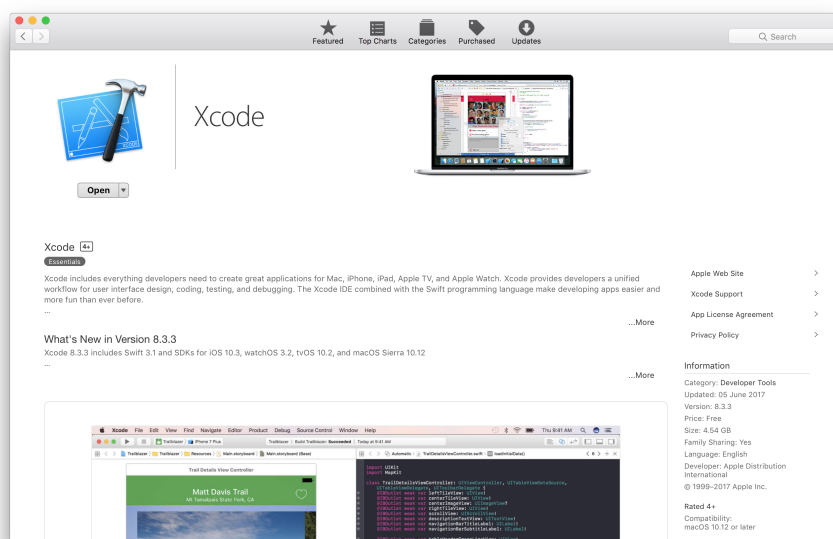
Dla programisty, iOS jest również atrakcyjnym miejscem. Język Swift nad którym prace zostały rozpoczęte w 2010 roku przez Chrisa Lattnera oraz wielu innych programistów z Apple, a w 2014 roku miał swój debiut, dzisiaj jest już głównym językiem programowania mobilnych aplikacji na platformę iOS oraz aplikacji desktopowych na MacOS i wyparł dotychczas używany w tych celach język Objective-C, który swoją historię ma również ściśle związaną z Apple. Korzenie fundamentalnych frameworków z iOS (CocoaTouch) sięgają lat 80 poprzedniego stulecia, a przez ten czas były bardzo silnie rozwijane i wykorzystywane w desktopowym systemie od Apple, MacOS. Nowoczesny język oraz potężne SDK (Software Development Kit) stanowią dziś podstawę pracy z aplikacjami na te platformy. Z roku na rok, wraz z nową wersją systemu, Apple wypuszcza uaktualniony do istniejących API (Application Programming Interface) oraz dostarcza nowe biblioteki zapewniające dostęp do najnowszych elementów systemu.

Ta praca dokumentuje framework "EPUBKit" którego zadaniem jest pełna obsługa (parsowanie oraz wyświetlanie) książek elektronicznych w formacie EPUB (Electronic Publication) a następnie wykorzystanie go w aplikacji. Rozpoczynając od dokładnego opisu środowiska, wykorzystanych narzędzi, scharakteryzowano format EPUB i jego spycyfikację techniczną, opisano proces tworzenia frameworku, jego strukturę oraz możliwości dystrybucji biblioteki jako moduł gotowy do wykorzystania przez developerów. Następnie w celu demonstracji funkcjonalności frameworku opisaną proces tworzenia aplikacji z jego wykorzystaniem. Celem pracy jest stworzenie prostego i lekkiego narzędzia w języku Swift, które uprości pracę innym programistom tworzącym aplikacje na iOS rozwiązując problem jakim jest obsługa formatu EPUB. Na dzień dzisiejszy natywne biblioteki iOS nie zapewniają programistom takiego narzędzia, a publicznie istnieje niewiele rozwiązań, które cieszą się mniejszą lub większą popularnością.

## 2. Wykorzystane technologie

### 2.1. Xcode i Developer Tools

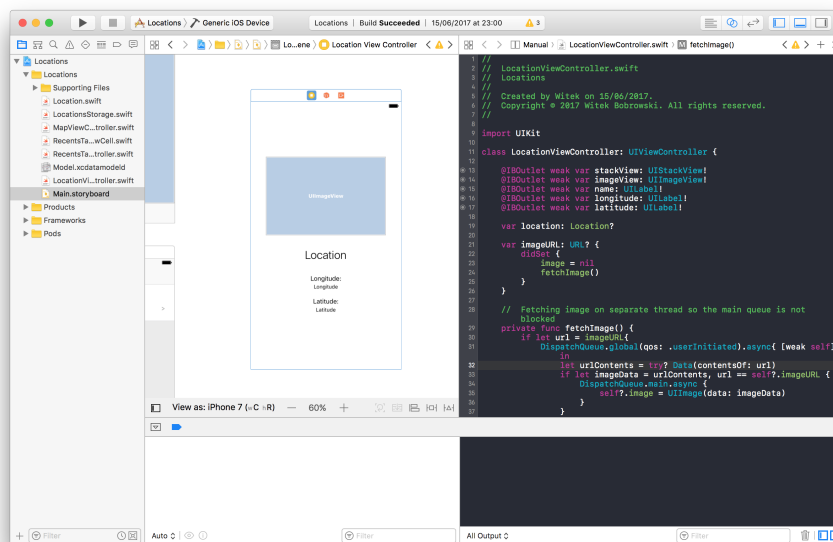
Xcode jest IDE (Integrated development environment) stworzonym przez Apple i dostępnym za darmo do pobrania z App Store, sklepu z aplikacjami do którego dostęp mają wyłącznie użytkownicy komputerów z systemem MacOS. Jest wyposażony w pakiet wszystkich narzędzi (Developer Tools) potrzebnych dla developerów aby tworzyć aplikacje na iOS. Główną aplikacją pakietu jest Xcode IDE który wraz z wspomagającymi aplikacjami dostępnymi w pakiecie takimi jak Simulator czy Instruments czyni pracę przy tworzeniu aplikacji płynną i efektywną. W tym rozdziale przedstawię właśnie te narzędzia ze względu na ich rolę w procesie tworzenia aplikacji.



Rysunek 2.1: Xcode w App Store

**Xcode IDE** Xcode jako nowoczesne, produktywnie środowisko jest miejscem w którym programista aplikacji na iOS spędza znaczną większość swojego czasu. Całość prac wykonywanych przy produkcji aplikacji może zostać wykonana właśnie tutaj. Najbardziej podstawowy element jakim jest edytor tekstu dobrze współgra z takimi narzędziami jak Interface Buildier, który pozwala w prosty sposób zaprojektować stronę wizualną aplikacji przy użyciu Storyboardów a następnie stworzyć referencję w kodzie do wybranych przez nas elementów przez proste przeciągnięcie myszką. Storyboardy są opcjonalnym acz-

kolwiek bardzo pożytecznym narzędziem szczególnie dla programistów stawiających swoje pierwsze kroki na tej platformie. Zapewniają one wizualne wyobrażenie interfejsu aplikacji nad którą wykonywana jest praca, a projektowanie dowolnego widoku który będzie wyglądał dobrze na każdym urządzeniu w dowolnej orientacji, jest relatywnie proste po zapoznaniu się z kilkoma elementarnymi zasadami.



Rysunek 2.2: Xcode pokazujący "Assistant editor"

Ponieważ Storyboardy tworzy się w jednym pliku o formacie .storyboard, często w profesjonalnej produkcji rezygnuje się z nich ze względu na konflikty w systemach kontrolii wersji. Konflikty te powstają w wyniku pracy wielu programistów, a ponieważ plik .storyboard jest w rzeczywistości plikiem XML, który został wygenerowany automatycznie, rozwiązywanie konfliktów bywa kłopotliwe, a przy dużych projektach problematyczne. Dlatego rezygnuje się z nich na rzecz tworzenia widoków tylko przy użyciu kodu, oraz niezależnych plików XIB. Pliki te pozwalają na ustawienie elementów w stylu znanym ze Storyboardów lecz w przeciwieństwie do nich reprezentują pojedynczy widok, dzięki czemu problem z konfliktami zostaje uniknięty a jednocześnie tworzenie bardziej skomplikowanych widoków pozostaje znacznie ułatwione.

Xcode zapewnia wsparcie dla systemu kontroli wersji git. Przy tworzeniu nowego projektu, gdy jest o to poproszony, inicjalizuje nowe repozytorium. Dodatkowo w nawigаторze projektu, w którym widać strukturę projektu, Xcode oznaczy literą "M" pliki które git oznacza jako pliki w których dokonano zmian (modified) a literą "A" (add) pliki które zostały dodane od czasu poprzedniego zachowania zmian. W najnowszej wersji 9.0, Xcode zyskał nową funkcjonalność - Source Control Navigator, który pozwala na eksplorowanie poszczególnych gałęzi repozytorium i podglądu dowolnego momentu w historii repozytorium.

## Simulator

## Instruments



## **2.2. Swift**

## **2.3. iOS SDK**

## **2.4. Zewnętrzne biblioteki**

### **3. Charakterystyka EPUB**

#### **3.1. Omówienie**

#### **3.2. Specyfikacja**

## **4. Framework EPUBKit**

### **4.1. Tworzenie frameworku na iOS**

### **4.2. Model**

### **4.3. Parser**

### **4.4. Widok**

### **4.5. Dystrybucja**

## **5. Aplikacja demonstracyjna**

### **5.1. Tworzenie aplikacji na iOS**

### **5.2. Wykorzystanie frameworku w aplikacji**

### **5.3. Publikacja**

## **6. Podsumowanie**

## **Bibliografia**