

Czytnik kanałów RSS dla smartfonów z systemem operacyjnym iOS z synchronizacją danych w chmurze

Spis treści

1. Cel projektu	2
2. Opis dziedziny problemowej	2
2.1. Kanały internetowe	3
2.2. Różnorodność urządzeń mobilnych	4
2.3. Synchronizacja danych	4
3. Opis produktów rynkowych	5
3.1. Czytniki internetowe	9
4. Architektura systemu	9
5. Opis głównych procesów	12
6. Wymagania funkcjonalne i pozafunkcjonalne	13
6.1. Wymagania funkcjonalne	13
6.2. Wymagania pozafunkcjonalne	13
7. Analiza – diagram czynności dla dodania nowego kanału RSS	15
9. Interfejs użytkownika aplikacji mobilnej - spis treści	19
10. Projekt interfejsu użytkownika	19
10.1. Makieta	19
10.2. Wybrane ekrany aplikacji	20

1. Cel projektu

Głównym celem projektu jest zaprojektowanie i wykonanie aplikacji do czytania kanałów internetowych w formatach RSS i Atom. Jej podstawowym zadaniem będzie przechowywanie kanałów RSS/Atom dodanych przez użytkownika oraz wyświetlanie ich treści w przyjazny dla odbiorcy sposób. Aplikacja przeznaczona będzie dla użytkowników smartfonów z systemem operacyjnym iOS (iPhone) w wersji 8.0 i wyższych. Dodatkowo przygotowana zostanie wybranie rozwiązanie, dzięki któremu możliwe będzie przechowywanie danych z aplikacji mobilnej oraz ich synchronizacja na wielu urządzeniach. Kolejne punkty niniejszej pracy przedstawiają przegląd wybranych aplikacji służących do korzystania z kanałów RSS a także tych, które pozwalają zebrać w jednym miejscu informacje z wielu źródeł: kanałów RSS, portali społecznościach (Twitter, Facebook).

2. Opis dziedziny problemowej

Żeby efektywnie i bez dużego zaangażowania czasowego śledzić nieustannie treści w Internecie warto zapisać sobie w jednym miejscu wybrane przez nas strony internetowe, portale, artykuły ulubionych felietonistów czy blogi. Wiele stron udostępnia tzw. kanały RSS (really simple syndication), przez które przekazywane są wiadomości opierają się na języku XML. Dzięki nim możliwe jest oglądanie wszystkich nowych informacji na tych stronach w jednym miejscu, bez konieczności odwiedzania każdej z nich. Aby je odczytać wystarczy jedynie użyć specjalnego programu – czytnika RSS, który umożliwi nam odbieranie informacji z poszczególnych, wybranych przez nas kanałów.

Warto wybrać czytnik RSS, który przechowuje dane na serwerze a nie tylko na naszym komputerze. Mamy gwarancję, że (jak to czasem bywa w przypadku narzędzi do RSS instalowanych na dysku twardym komputera) nasze dane nie zostaną skasowane przy nieprawidłowym zamknięciu systemu. Możemy też eksportować albo importować pliki z definicjami adresów. Pozwala to na korzystanie z już zbudowanych baz kanałów RSS. W związku z tym, że coraz częściej korzystamy z urządzeń mobilnych z dostępem do Internetu,

będziemy mogli modyfikować listę kanałów na każdym urządzeniu, synchronizować ją z serwerem i już w zmienionej wersji oglądać na innym urządzeniu, na którym korzystamy z czytnika RSS powiązanego z tym samym kontem użytkownika.

2.1. Kanały internetowe

Wiele stron internetowych, których zawartość jest często aktualizowana, umożliwia użytkownikom śledzenie nowości za pomocą kanałów internetowych. Kanał internetowy (lub kanał informacyjny) to format danych (w języku XML lub JSON), który zawiera streszczenie wiadomości z powiązanej z nim strony www albo jej pełen tekst. Zwykle użytkownicy subskrybują wiele kanałów w celu „bicia na bieżąco” z informacjami pochodzącymi z różnych źródeł (serwerów). Subskrypcję umożliwiają czytniki kanałów (ang. *feed reader* lub *news aggregator*), programy dzięki którym użytkownik łatwo może przeglądać treści, bez konieczności otwierania przeglądarki internetowej.

Najpopularniejsze standardy kanałów internetowych to:

- RSS (ang. *Really Simple Syndication*)
- Atom (standard)

W sieci dominują kanały przygotowane w formacie RSS 2.0. Rozwiązanie to zawiera liczne **błędy i niejasności** i nie jest już rozwijane i poprawiane. W celu rozwiązania tego problemu i przejrzystej prezentacji danych stworzono standard Atom, który jeszcze nie jest powszechnie używany. Równolegle dostępne są **dwa różniące się standardy** (Tabela 1.). W związku z tym przy próbie zgromadzenia danych z wielu kanałów pojawią się zagadnienia takie jak:

- konieczność obsługi kanałów skonstruowanych w oparciu o specyfikację RSS 2.0 jak również Atom
- poprawne przeprowadzanie aktualizacji i agregacji wpisów (tylko standard Atom gwarantuje unikalne **globalne identyfikatory kanałów** i wpisów dzięki czemu znika ryzyko duplikowania pobieranych informacji)

- poprawne wyświetlanie treści kanału w standardzie Atom oraz w każdej z różniących się wersji RSS 2.0

Tabela 1. Różnice między RSS 2.0 a Atom 1.0

RSS 2.0	Atom 1.0
Specyfikacja przewiduje tylko tekst, a w praktyce używa się HTML "ukrytego" za pomocą encji. Nie ma możliwości użycia XHTML, ani innych typów danych. Przez tę niejasność i ograniczenia RSS jest nazywany formatem stratnym ^[1] .	Atom dla każdego elementu pozwala jednoznacznie określić typ danych i obsługuje tekst, HTML , XHTML , inne warianty XML oraz załączanie plików.
RSS ma tylko jedno ogólne pole na opis elementu, przez co autorzy czasem budują osobne kanały zawierające tylko streszczenia lub tylko pełne treści wpisów	Atom unika tego problemu pozwalając wstawić streszczenia i rozwinięcia do tego samego kanału
RSS nie wymaga podania identyfikatorów i dat do wpisów, przez co agregacja i aktualizacja kanałów RSS jest dla czytelników skomplikowana	Atom wymaga użycia globalnych unikatowych identyfikatorów dla kanału i wpisów, co czyni aktualizację prostymi i gwarantuje, że wpisy nie będą przypadkiem duplikowane
RSS 2.0 jest specyfikacją należącą do Uniwersytetu Harvarda i nie będzie rozwijany	Atom jest otwartym standardem IETF
RSS występuje w 9 odmianach ^[2] , które nie są zupełnie zgodne	Atom ma jedną oficjalną wersję 1.0. W sieci można spotkać kanały używające "roboczej" 0.3.
RSS nie ma własnej przestrzeni nazw XML, przez co nie może być osadzony wewnątrz innych dokumentów XML. Ma własny element <language>, nie obsługuje linków relatywnych.	Atom poprawnie i w pełni wykorzystuje XML, przez co może być łatwo mieszany z innymi językami XML, korzysta z xml:lang i xml:base.
RSS jest starszym i przez to popularniejszym od Atom formatem. Stał się <i>de facto</i> standardem i synonimem dla kanałów informacyjnych	Atom jest młodym standardem, jego specyfikacja osiągnęła wersję 1.0 dopiero pod koniec 2005 roku.

2.2. Różnorodność urządzeń mobilnych

Na rynku dominują urządzenia mobilne z systemami operacyjnymi: Android, iOS oraz Windows Phone, przy czym ostatni chociaż obecny na rynku ma znacznie mniejszy w nim udział. Istnieją również inne młode platformy takie jak Firefox OS czy Ubuntu Mobile, natomiast mają pomijalny udział w rynku w chwili pisania tego tekstu. Wyzwaniem dla developerów jest takie przygotowanie aplikacji aby działała poprawnie na grupie urządzeń z danym OS w każdej jego wersji oraz na wielu urządzeniach o różnej rozdzielczości ekranów.

2.3. Synchronizacja danych

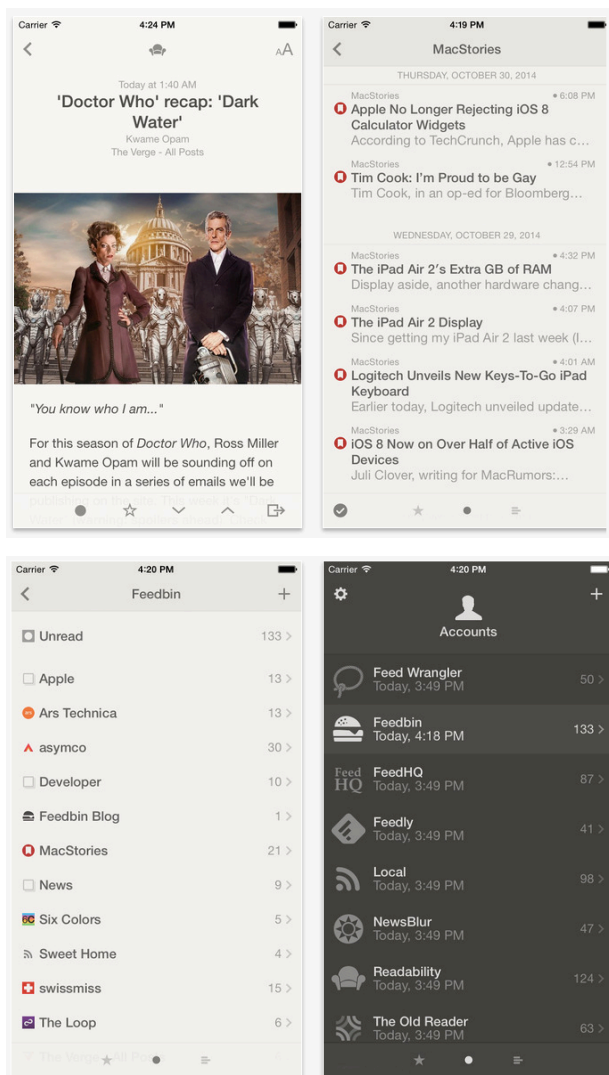
Użytkownicy oczekują możliwości korzystania z aplikacji na wielu urządzeniach (np. na smartfonie i na tablecie) po zalogowaniu się do niej na swoje konto. Jest to możliwe dzięki zapisywaniu danych na serwerze (przechowywaniu ich w chmurze) i synchronizacji z urządzeniem.

3. Opis produktów rynkowych

Na rynku istnieją darmowe oraz komercyjne czytniki RSS w formie aplikacji na urządzenia smartphona oraz tablet. Różnią się m.in. sposobem prezentowania listy zasubskrybowanych kanałów oraz prezentacją samych wpisów. Posiadają integrację z różnymi serwisami pozwalającymi na agregację danych o kanałach (e.g. Feedly, The Old Reader). Poniżej znajdują się przykłady aplikacji dostępnych dla iOS.

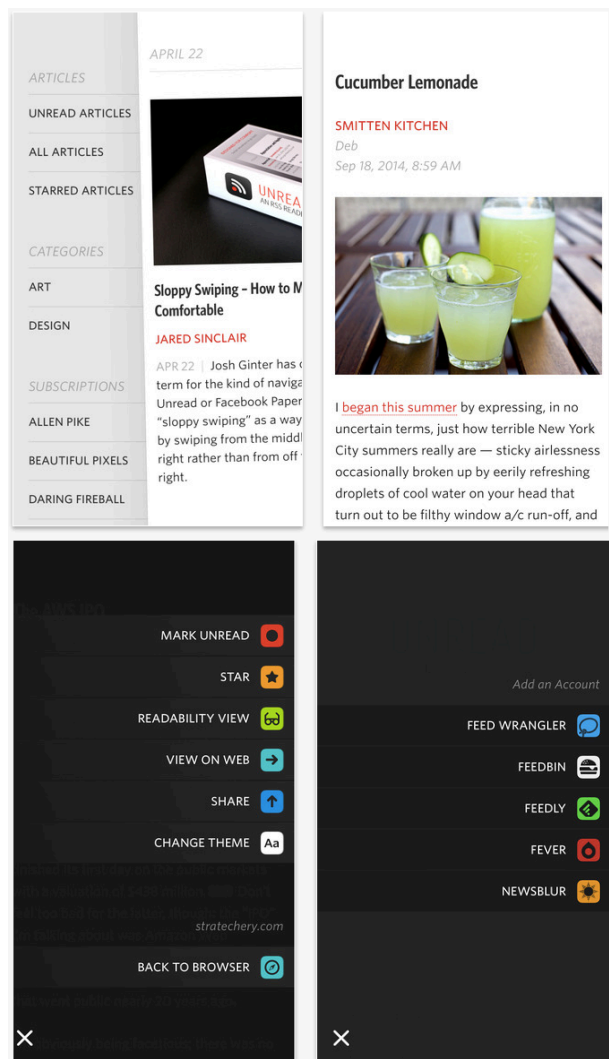
Reader 2

Jedną z lepiej rozpoznawalnych aplikacji jest Reader 2. Rozpoczęcie korzystania z aplikacji jest bardzo proste – po otwarciu aplikacji pojawia się interfejs „accounts” - przez kliknięcie „plusika” na górze można się zalogować swoje konto w chmurze. Można też bezpośrednio w aplikacji dodać swoje feedy. Ustawienia takie jak tryb synchronizacji (w tle, przy uruchomieniu, manualnie) czy okres trzymania przeczytanych artykułów można skonfigurować. Przy zalogowaniu do chmury, kanały zostaną posortowane dokładnie tak samo w tagi/foldery, jak to ma miejsce w przeglądarkowej wersji naszego serwisu. Jest jednak możliwość zmieniania folderu/taga, czy usunięcia subskrypcji danego kanału. Reader 2 wspiera protokół feed://. Oznacza to, że przeglądając daną stronę w przeglądarce Safari, wystarczy kliknąć w link RSS, a zostanie on automatycznie otwarty w Reader 2 z pytaniem, czy chcemy dodać go zasubskrybować. Użytkownik może czytać to, co nas w danej chwili go interesuje – ma możliwość czytania wszystkich newsów, ze wszystkich kanałów, wszystkich z konkretnego folderu/tagu, lub wszystkich z konkretnego feeda. Samo nawigowanie po aplikacji jest bardzo naturalne i intuicyjne. Dostępne są 2 paski nawigacyjne – 1 na górze i drugi na dole, oraz gesty standardowe dla platformy iOS (pull to refresh, swipe etc.).



Unread

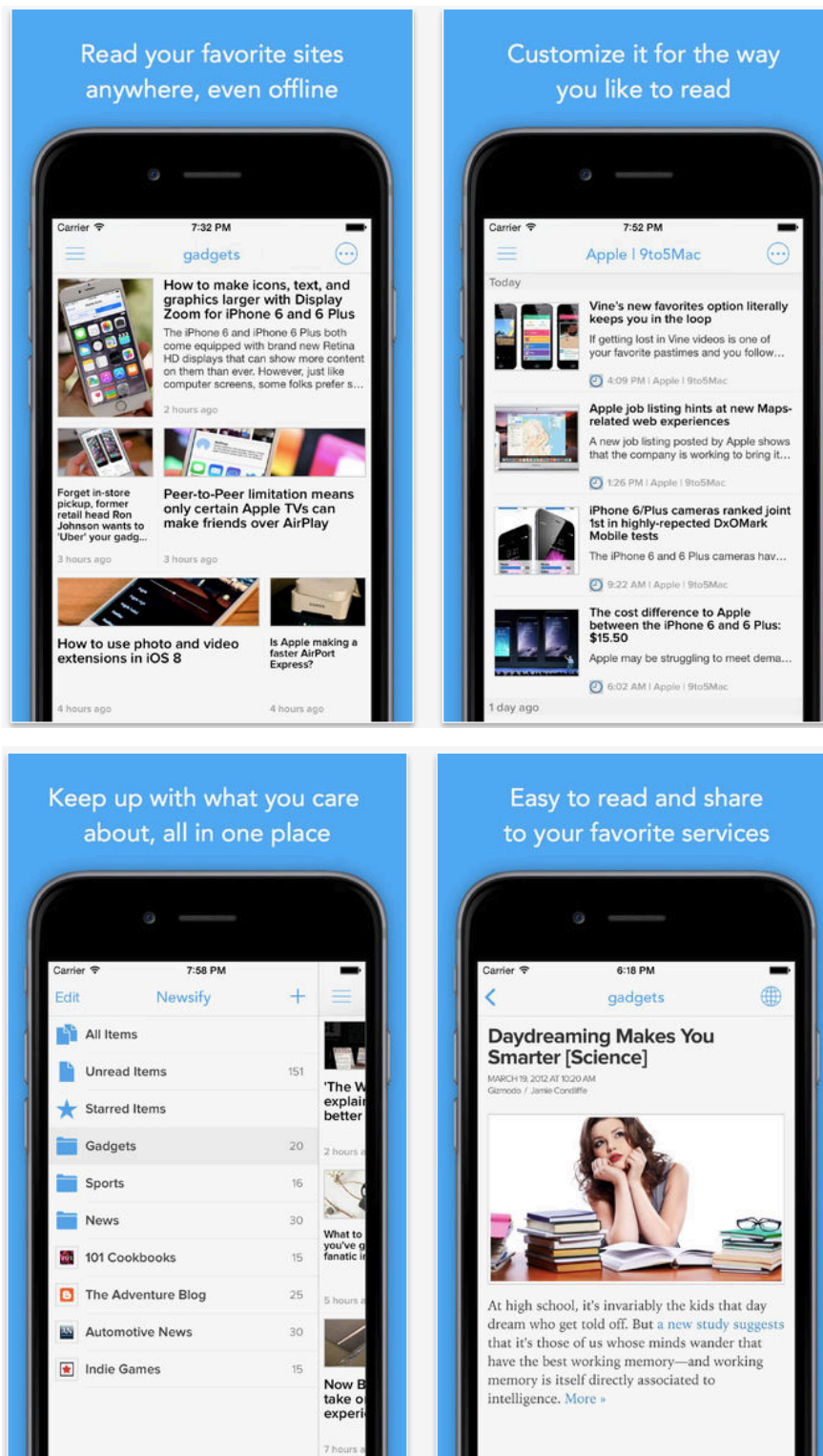
Unread autorstwa Jareda Sinclaira, to tworzona przez 8 miesięcy aplikacja o minimalistycznej szacie graficznej. Interfejs Unread został przygotowany w duchu minimalizmu. Dla wielu użytkowników przytłoczonych przeładowanym interfejsem np. Reader 2. Wadą aplikacji Unread może być dla niektórych użytkowników to, że nie pozwala na dodawanie nowych subskrypcji, trzeba to robić ręcznie bezpośrednio w jednej ze wspieranych usług (np. desktopowej lub przeglądarkowej wersji aplikacji Feedly, której zawartość użytkownik odczytuje następnie w czytniku Unread na urządzeniu mobilnym).



Newsify

Newsify umożliwia stworzenie strony z wiadomościami pochodzącymi z konkretnego kanału lub z całego folderu dodanego przez użytkownika. W kolumnie po lewej znajduje się lista subskrypcji, z poziomu której można przeglądać wszystkie treści lub sortować je pod kątem tych jeszcze nieprzeczytanych. Pozostała część ekranu jest przeznaczona do wyświetlania treści wiadomości i przewijania ich listy. Szybkie przeskakiwanie pomiędzy kolejnymi artykułami polega na przesuwaniu palcem po ekranie w górę i w dół. W przypadku przejścia do konkretnego postu, jego widok wewnątrz aplikacji ogranicza się do wyświetlenia jednego obrazka i pierwszych kolumn tekstu – i to wystarczy do wstępnego zorientowania się w jego treści. Do pełnej treści postu przechodzimy poprzez kliknięcie w ikonkę globusa, która znajduje się w prawym górnym rogu. Z poziomu Newsify wybrane wiadomości można też dodawać do

ulubionych (oznaczać gwiazdką) lub wysłać je do przeczytania na później (np. do aplikacji takich jak Pocket czy Readability).



3.1. Czytniki internetowe

Na rynku istnieją czytniki w całości działające w przeglądarce internetowej, zapewniające większość funkcji aplikacji ale wymagające stałego połączenia z internetem.

Aż do roku 2013 najlepiej rozwiniętym internetowym agregatorem kanałów RSS był Google Reader. W oparciu o to rozwiązanie powstało wiele aplikacji desktopowych jak i mobilnych, które korzystały z jego funkcji. Wraz z jego zamknięciem pojawiła się luka na rynku, która szybko została zastąpiona nowymi rozwiązaniami.

Jednym z takich rozwiązań jest The Old Reader. Jego twórcy poza stworzeniem własnej aplikacji przygotowali webservice z dobrze opisanym API, do którego dostęp jest bezpłatny.. Webservice umożliwia integrację The Old Reader w aplikacjach tworzonych przez niezależnych dostawców, przykłady takich aplikacji to: Reeder, Feeder dla iOS, Greader dla urządzeń z Androidem czy Old Reader dla WindowsPhone. Elementy jakie obsługuje The Old Reader to:

- Foldery: tworzenie, usuwanie, zmiana nazwy, przechowywanie kolejności ustalonej przez użytkownika.
- Sybskrybcje: dodawanie, usuwanie, odświeżanie subskrypcji, zliczanie nieprzeczytanych elementów
- Elementy/Posty (Items): identyfikacja postów po id, zwracanie zawartości postów, zaznaczanie wszystkich postów jako przeczytane, odświeżanie.

The Old Reader ogranicza ilość darmowych subskrypcji dla jednego użytkownika do 100. Pomimo bezpłatnego dostępu do API występują więc opłaty w tej usłudze.

4. Architektura systemu

System będzie składał się z dwóch części: aplikacja iOS oraz część serwerowa do synchronizacji danych użytkownika. Głównym elementem projektu jest aplikacja iOS, natomiast część serwerowa zapewnia jedynie dodatkową funkcjonalność.

Rozważane były dwie opcje dla części serwerowej: tworzenie własnego web service do synchronizacji danych (wymaga to stworzenia własnej części serwerowej) oraz wykorzystanie istniejącego, publicznie dostępnego web service. Poniżej znajduje się porównanie zalet i wad obydwu rozwiązań:

Utworzenie własnego Web Service	Wykorzystanie istniejącego Web Service
<ul style="list-style-type: none"> + Nieograniczone możliwości dostosowywania do potrzeb aplikacji – Konieczność utrzymania serwera – Konieczność obsłużenia zwiększonego ruchu sieciowego – Konieczność obsługi zagadnień związanych z bezpieczeństwem 	<ul style="list-style-type: none"> – Brak możliwości dostosowania + Brak konieczności utrzymania serwera + Brak konieczności zajmowania się problemem obciążenia serwera i obsługi zwiększonego ruchu sieciowego + Brak konieczności zajmowania się

Po rozważeniu wad i zalet zdecydowano o wykorzystaniu istniejącego rozwiązania web service. Podczas analizy rozważano rozwiązanie The Old Reader i zostanie ono wykorzystane do zapewnienia synchronizacji danych użytkownika. Schemat architektury rozwiązania został zaprezentowany na poniższym diagramie.

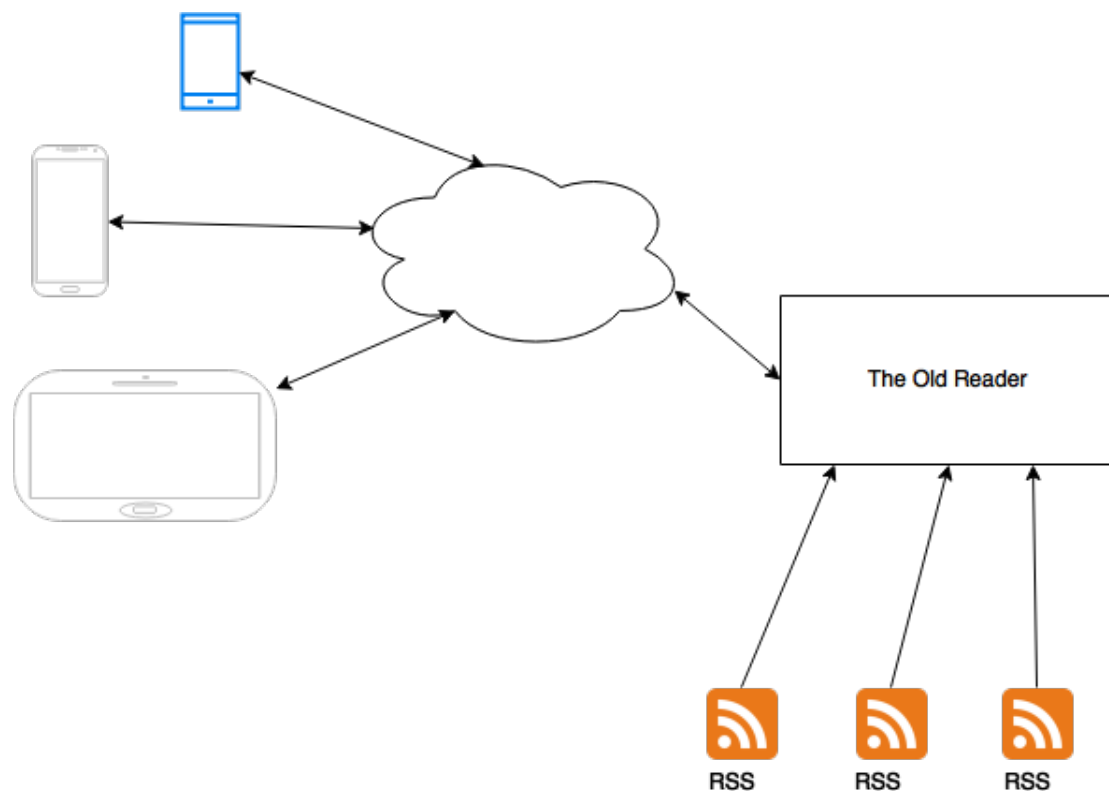


Diagram 1. Architektura systemu

Aplikacja mobilna może być zainstalowana na wielu urządzeniach (iPhone, iPad, iPod). Na każdym z nich użytkownik zalogowany jest na swoje konto i może subskrybować nowe treści. W czasie synchronizacji kanały RSS za pośrednictwem The Old Reader zapisywane są na serwerze.

5. Opis głównych procesów

Pobieranie i aktualizacja danych w aplikacji mobilnej

Użytkownik w czasie korzystania z aplikacji dodaje kolejne adresy URL kanałów RSS. Podstawowa funkcją aplikacji jest pobranie listy postów dla każdego kanału a następnie jej wyświetlenie w przyjazny dla użytkownika sposób. Zarówno lista kanałów RSS jak i lista postów jest przechowywana w bazie danych aplikacji (SQLite) dzięki czemu dane dostępne są po wyłączeniu ponownym włączeniu aplikacji. Przechowywane są także informacje o statusie postów (czy zostały przeczytane, czy zostały dodane przez użytkownika do kategorii „Ulubione”). Dla zapisanych w aplikacji kanałów RSS przeprowadzana jest okresowa aktualizacja – pobranie listy aktualnych postów z sieci. Aktualizacja może być wymuszona (przez użytkownika) lub wykonywana automatycznie w określonych odstępach czasu.

Synchronizacja danych

Przeprowadzana jest okresowa synchronizacja danych z web service. Zapisywane są dane: lista kanałów RSS, lista postów (wraz z ich statusami: przeczytany/nie przeczytany, dodany do „Ulubionych”), preferencje użytkownika (np. dotyczące częstości synchronizacji, częstości odświeżania, sposobu sortowania postów).

6. Wymagania funkcjonalne i pozafunkcjonalne

6.1. Wymagania funkcjonalne

Lista kanałów

- dodawanie adresu URL kanału RSS
- usuwanie adresu URL kanału RSS
- walidacja adresu i zawartości URL
- wyświetlanie listy adresów kanałów RSS

Lista postów

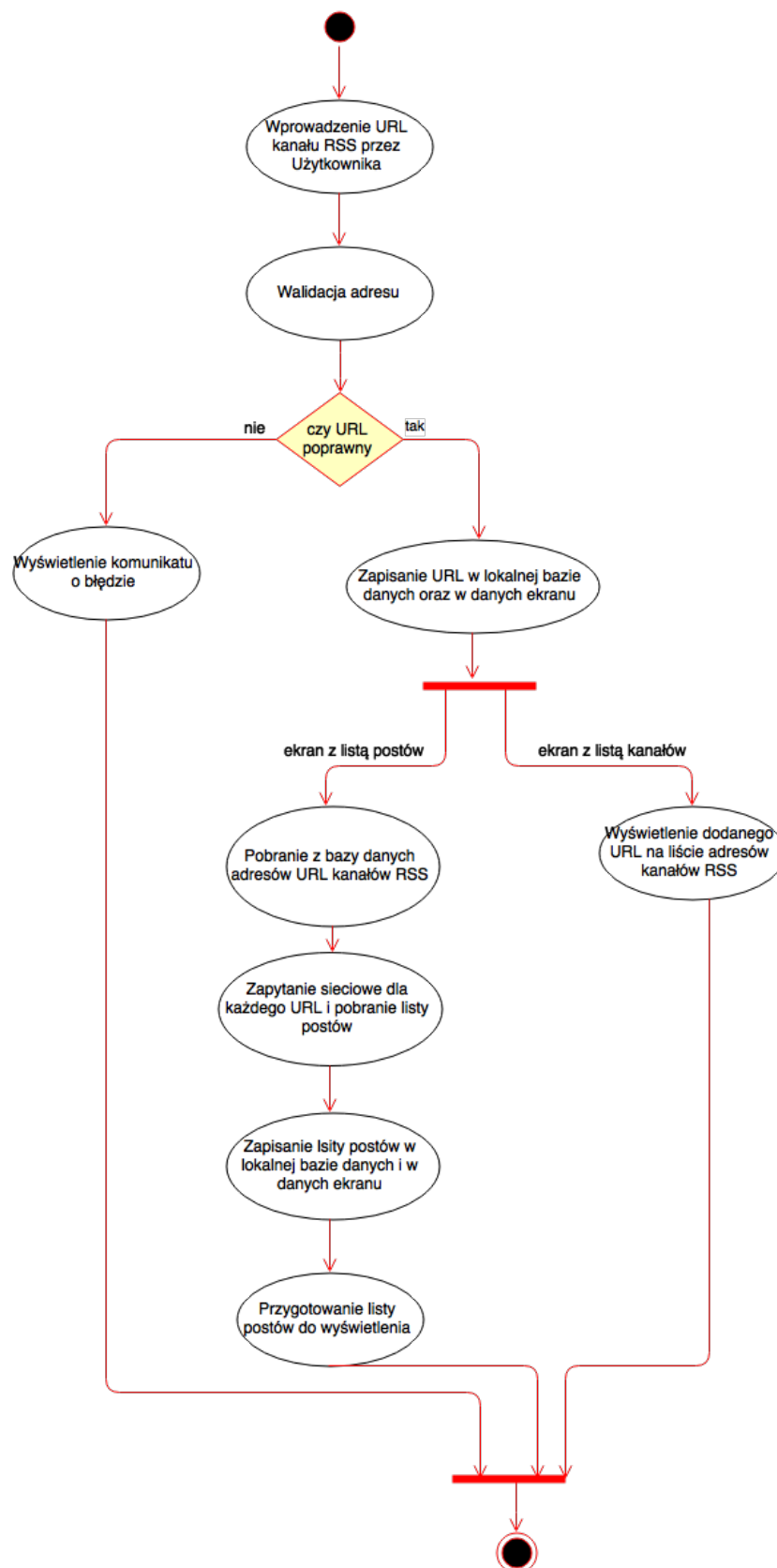
- wyświetlanie listy postów
 - filtrowanie listy (po dacie, przeczytane, nie przeczytane)
 - dodawanie postów do kategorii do „Ulubione”
 - udostępnianie posta (e-mail, SMS, serwisy społecznościowe)
 - wyświetlanie treści pojedynczego posta
-
- Logowanie/rejestracja
 - Synchronizacja danych z backendem
 - Wyświetlanie informacji w sekcji pomocy, zawierających zrozumiałe dla użytkownika podpowiedzi dotyczące działania aplikacji.

6.2. Wymagania pozafunkcjonalne

- Zgodność - Aplikacja mobilna powinna działać na smartfonach z systemem operacyjnym iOS w wersji 8.0 i wyższej.
- Uwzględnienie wytycznych dotyczących interfejsu użytkownika opisanych na [developers.apple.com](https://developer.apple.com).
- Zapewnienie działania w trybie offline - brak połączenia z Internetem nie powinien uniemożliwiać korzystania z aplikacji.
- Dostępność i niezawodność synchronizacji z backendem – 24/7 z maksymalnym czasem niedostępności 4h/30 dni roboczych
- Komunikaty dotyczące błędów powinny być zgodne z wytycznymi Apple dostępnymi na stronie developer.apple.com

- Bezpieczeństwo – dane użytkowników (login, hasło) powinny być zaszyfrowane w taki sposób, żeby uniemożliwić ich odczyt przez osoby trzecie zarówno w aplikacji jak i w czasie przesyłania ich przez sieć.
- Skalowalność – aplikacja powinna zostać napisana w taki sposób, żeby możliwe było jej późniejsze przystosowanie do innych urządzeń niż smartfony typu iPhone (iPad, iPod).
- Przenośność – aplikacja serwerowa nie może być uzależniona od jednego dostawcy systemu operacyjnego lub serwera aplikacyjnego.

7. Analiza – diagram czynności dla dodania nowego kanału RSS



Text

Diagram 2. Diagram czynności dla dodawania nowego adresu URL kanału RSS

Diagram 2 przedstawia czynności jakie są wykonywane przez użytkownika i aplikację w celu prawidłowego zapisania nowego adresu URL w pamięci lokalnej bazy danych aplikacji (dane dostępne po wyłączeniu i włączeniu aplikacji) oraz w źródle danych ekranów wyświetlających listę adresów URL i listę postów.

8. Lista testów

Możliwy sposób zaprezentowania wymagań w postaci przypadków użycia (use cases). Przypadku użycia stanowią podstawę do testowania głównych funkcji aplikacji mobilnej.

UC1: Dodanie URL kanału RSS do listy

Główny scenariusz:

1. Użytkownik wpisuje adres URL kanału RSS
2. Użytkownik zatwierdza wpisany URL
3. Aplikacja sprawdza poprawność URL i danych
4. Aplikacja dodaje URL do listy
5. Aplikacja wyświetla listę URLi

Rozszerzenia:

- 3.A. Aplikacja nie potwierdził poprawności URL
 - 3.A.1. Aplikacja prosi o ponowne wprowadzenie URL (powrót do 1)

UC2: Usuwanie URL kanału RSS z listy

Główny scenariusz:

1. Użytkownik wybiera z listy URL kanału, który chce usunąć
2. Użytkownik potwierdza decyzję o usunięciu URL kanału RSS
3. Aplikacja usuwa kanał z listy
4. Aplikacja wyświetla odświeżoną listę URL kanałów RSS

Rozszerzenia:

- 2.A. Użytkownik anuluje chęć usunięcia URL kanału RSS (przejdź do 4)

UC3: Wyświetlenie szczegółów posta

Główny scenariusz:

1. Użytkownik wyświetla ekran główny z listą postów

<p>2. Użytkownik wybiera post z listy</p> <p>3. Aplikacja wyświetla ekran szczegółów posta</p> <p>Rozszerzenia:</p> <p>3.A. Aplikacja wyświetla popup informujący o problemie z pobraniem danych (brak internetu, niepoprawny URL)</p> <p>3.A.1. Aplikacja pozwala użytkownikowi powtórnie wywołać zapytanie sieciowe (sukces - przejdź do 3 porażka - 3.A)</p> <p>3.B. Aplikacja wyświetla popup informujący o problemie z pobraniem danych (brak Internetu, niepoprawny URL)</p> <p>3.A.1. Aplikacja pozwala użytkownikowi zamknąć popup (przejdź do 1)</p>
<p>UC4: Dodanie/usunięcie posta do „Ulubionych”</p> <p>Główny scenariusz:</p> <ol style="list-style-type: none"> 1. Użytkownik wyświetla ekran główny z listą postów 2. Użytkownik wybiera post z listy 3. Aplikacja wyświetla ekran ze szczegółami postu 4. Użytkownik wybiera ikonę dodającą post do „Ulubionych” (ikona jest nieaktywna) 5. Aplikacji zmienia ikonę na aktywną w: widoku szczegółów postu, na liście postów 6. Aplikacja dodaje post do ekranu „Ulubione” <p>Rozszerzenia:</p> <p>4.A. Użytkownik wybiera ikonę dodającą post do „Ulubionych” (ikona jest aktywna)</p> <p>4.A.1. Aplikacji zmienia ikonę na nieaktywną w: widoku szczegółów postu, na liście postów</p> <p>4.A.2. Aplikacja usuwa post z ekranu „Ulubione”</p>
<p>UC5: Oznaczanie postów jako przeczytane</p> <p>Główny scenariusz:</p> <ol style="list-style-type: none"> 1. Użytkownik wyświetla ekran główny z listą postów 2. Użytkownik wybiera post z listy 3. Aplikacja wyświetla ekran szczegółów posta 4. Użytkownik nawiguje wstecz

5. Aplikacja oznacza wyświetlony przed chwilą posta jako przeczytany (posty przeczytane i nieprzeczytane na liście postów różnią się od siebie)

UC6: Zapisanie danych do pamięci aplikacji

Główny scenariusz:

1. Użytkownik wykonuje UC1
2. Użytkownik wyłącza aplikację
3. Użytkownik włącza ponownie aplikację
4. Dane w aplikacji mają taki sam stan jak w momencie jej zamykania

Rozszerzenia:

- 1.A. Użytkownik wykonuje UC2 (przejdź do 2)
- 1.B. Użytkownik wykonuje UC2 (przejdź do 2)
- 1.C. Użytkownik wykonuje UC4 (przejdź do 2)
- 1.D. Użytkownik wykonuje UC5 (przejdź do 2)

UC7: Pobranie danych z sieci

Główny scenariusz:

1. Użytkownik wyświetla ekran główny aplikacji
2. Użytkownik wykonuje gest „pul to refresh”
3. Aplikacja pobiera aktualne posty z sieci i wyświetla je na liście

Rozszerzenia:

- 3.A. Aplikacja wyświetla informację o braku możliwości pobrania danych
 - 3.A.1 Użytkownik potwierdza przeczytanie informacji (np. guzik OK na popupie)
 - 3.A.2. Aplikacja wyświetla listę postów z pamięci stałej

9. Interfejs użytkownika aplikacji mobilnej - spis treści

Ekrany aplikacji

- Splash screen z logiem aplikacji
- Ekran logowania
- Ekran rejestracji
- Ekran główny z listą postów
- Ekran dodawania kanałów RSS
- Ekran z listą ulubionych postów
- Ekran z widokiem szczegółów postu
- Ekran ustawień

Popupy

- Popup – brak obserwowanych kanałów RSS
- Popup – problem w czasie pobierania danych z sieci
- Popup – brak Internetu
- Popup – informacja o próbie powtórnej dodania tego samego kanału RSS do obserwowanych
- Popup – udostępnianie postu przez e-mail, SMS, wybrane portale społecznościowe

10. Projekt interfejsu użytkownika

10.1. Makieta

Przygotowana została makieta przedstawiająca ekrany aplikacji mobilnej. Dzięki wykorzystaniu narzędzia NinjaMock makieta jest „klikalna” i obrazuje przepływ nawigacji w aplikacji. Makieta dostępna pod linkiem:

<http://ninjamock.com/s/ukdnua>

Aplikacja posiada nawigację poziomą: użytkownik przechodzi do wybranego ekranu wybierając ikonę z menu dolnego (tab bar).

10.2. Wybrane ekrany aplikacji

Ekran Główny

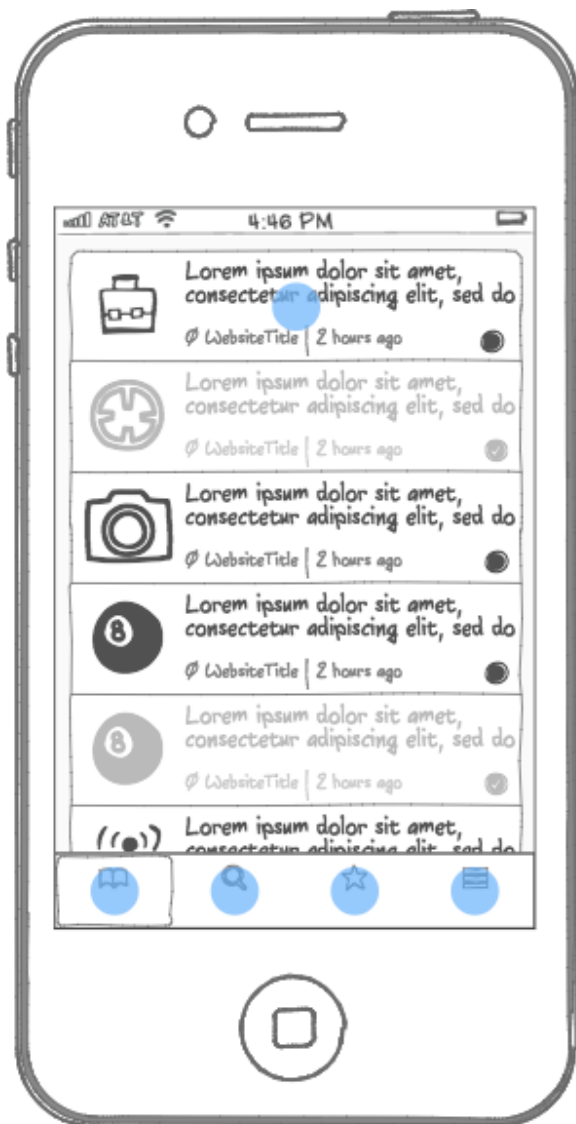
Ekran pojawia się po wybraniu pierwszej ikony z tab bara (ikona książki).

Ekran zawiera listę postów wszystkich kanałów RSS zapisanych w aplikacji.

Widok pojedynczego posta zawiera:

- miniaturkę zdjęcia
- tytuł posta
- datę publikacji
- skrócony opis zawartości
- ikonę pokazującą, czy post został dodany do „Ulubionych” (na makiecie: szare/czarne kółko w prawy dolnym rogu posta).

Wybranie posta z listy powoduje wyświetlenie ekranu z widokiem szczegółów.



Ekran – widok szczegółów

Ekran zawiera treść posta osadzoną w kontrolce WebView. Górna belka nawigacyjna zawiera guziki pozwalające na:

- powrót do poprzedniego ekranu
- dodanie posta do kategorii „Ulubione” (na makiecie: ikona gwiazdki)
- udostępnienie posta (e-mail, SMS, wybrane media społecznościowe)

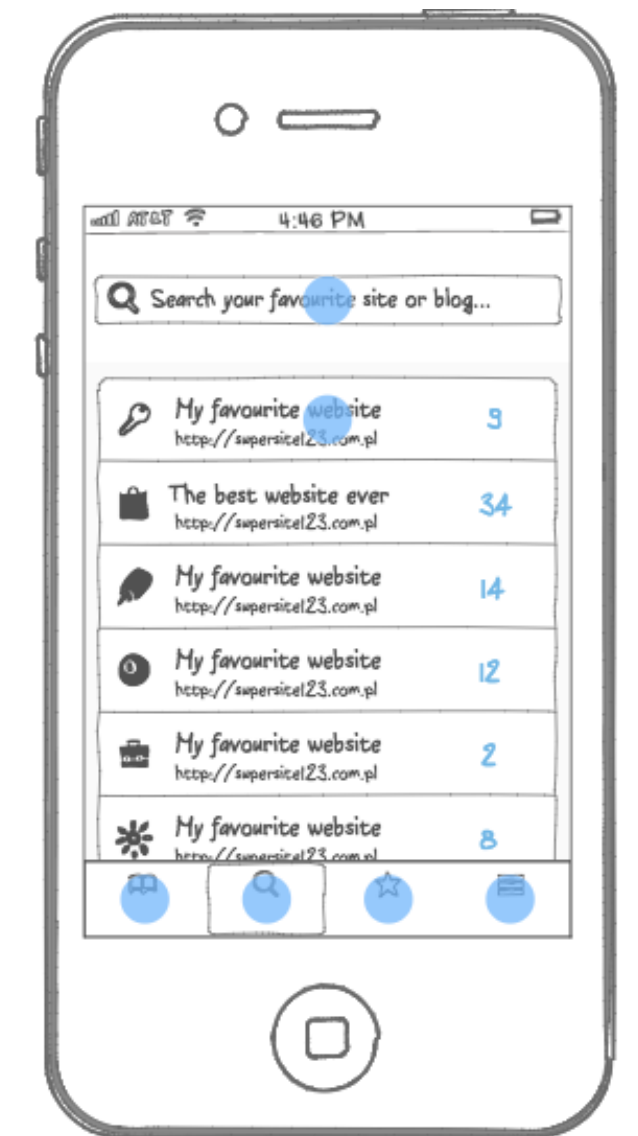


Ekran – dodawanie kanałów RSS

Ekran pojawia się po wybraniu drugiej ikony z tab bara (ikona lupy).

Użytkownik ma możliwość dodania kanału RSS poprzez wpisanie go w pole tekstowe „Search” w górnej części ekranu.

Ekran wyświetla listę kanałów. Przy każdym kanale widoczna jest liczba postów.



Ekran – ulubione posty

Ekran pojawia się po wybraniu trzeciej ikony z tab bara (ikona gwiazdki).

Wyświetlana jest lista postów, które zostały oznaczone jako „Ulubione”.

Wybranie postu z listy powoduje wyświetlenie Ekranu szczegółów (opis w punkcie 10.2.2).

