

# bardzofajnyspiewnik.pl — Plan rozbudowy projektu

Jakub Onufry Wojtaszczyk

Piotr Tabor

9 marca 2007

**Wersja: 0.2**

## Historia

Data	Wersja	Autor	Zmiany
2007-03-08	0.1	Piotr Tabor	Pierwsza wersja dokumentu.
2007-03-08	0.2	Jakub Wojtaszczyk	Informacje o web-servisie.

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Cel dokumentu . . . . .	2
1.2	O projekcie . . . . .	2
<b>2</b>	<b>Sytuacja obecna</b>	<b>3</b>
2.1	Funkcjonalność osiągnięta . . . . .	3
2.2	Jakie problemy projekt rozwiązuje? Jakie daje zyski? . . . . .	3
2.3	Zależności i elementy składowe produktu . . . . .	4
2.4	Technologie o które obecnie oparty jest system . . . . .	4
<b>3</b>	<b>Funkcjonalność zaplanowana do wykonania w okresie marzec 2007 - czerwiec 2007</b>	<b>5</b>
3.1	Umożliwienie przygotowywania piosenki do wydruku . . . . .	5
3.1.1	Klient web-service'u . . . . .	5
3.1.2	Serwer web-service'u . . . . .	5
3.2	Interfejs wyszukiwania piosenek - wyszukiwanie pełnotekstowe . . . . .	5
3.3	Rozbudowa logiki biznesowej . . . . .	5
3.3.1	Szybkość dostępu do SVN'a . . . . .	5
3.3.2	Odciążenie bazy danych . . . . .	5
3.4	Budowanie projektu . . . . .	5
<b>4</b>	<b>Przypadki użycia dla nowej funkcjonalności</b>	<b>6</b>
4.1	Przeszukanie dostępnych piosenek . . . . .	6
4.2	Otrzymanie pliku w formacie PDF z wybraną piosenką . . . . .	8
4.3	Wyszukanie śpiewnika . . . . .	9
4.4	Wydrukowanie śpiewnika . . . . .	10
<b>5</b>	<b>Podział pracy</b>	<b>11</b>
5.1	Jakub Onufry Wojtaszczyk . . . . .	11
5.2	Piotr Tabor . . . . .	11

# 1 Wstęp

## 1.1 Cel dokumentu

Celem tego dokumentu jest opisanie funkcjonalności o jaką zostanie rozbudowany projektu bardzofajnyśpiewnik.pl - utworzony przez Annę Jasińską, Marcina Pilipczuka, Michała Korcha i Piotra Tabora w czasie zajęć z ZPP w roku 2006/2007.

Dokument ten w pierwszej sekcji streszcza wizję projektu w jego obecnym kształcie, a w następnych sekcja opisuje funkcjonalność o którą projekt zostanie rozszerzony.

Szczegółowa dokumentacja projektu znajduje się pod adresem: <http://duch.mimuw.edu.pl/malcinżpp>

## 1.2 O projekcie

Dokument ten dotyczy projektu do prowadzenia, edycji, zarządzania śpiewnikami w postaci elektronicznej. Działający projekt ma umożliwić prosty i estetyczny wydruk piosenek oraz zapewnić wygodne i bogate w możliwości narzędzie do ich edycji. W architekturze projektu zostaną uwzględnione możliwości rozszerzenia systemu do zarządzania całymimi śpiewnikami, które mogą być dostępne dla poszczególnych osób.

## 2 Sytuacja obecna

### 2.1 Funkcjonalność osiągnięta

Celem projektu jest stworzenie narzędzia umożliwiającego zebranie w spójnym systemie informatycznym kolejnej części dorobku ludzkiego.

Podobnie jak system wikipedia odniósł sukces w zbieraniu wiedzy encyklopedycznej, tak oparty o podobne mechanizmy system bardzofajnyspiewnik.pl ma umożliwić zgromadzenie w jednym miejscu, ponad podziałami językowymi, dorobku muzycznego. Otwarty kod, bezpłatne możliwości korzystania ze zgromadzonej wiedzy, oparcie o otwarte standardy i umożliwienie wszystkim maksymalnie łatwego uzupełniania zasobów, które stały się przyczyną sukcesów wikipedi - na tym chcemy oprzeć nasz projekt.

Jednak system bardzofajnyspiewnik.pl w swojej długofalowej wizji rozwoju zamierza udostępnić nieco więcej. Chcemy, by każdy mógł założyć sobie konto, a w jego obrębie zarządzać interesującymi go piosenkami, w szczególności zbierać utwory w śpiewniki i udostępniać je przez internet lub eksportować do formatu PDF i drukować.

Tym samym pozwoli to zmniejszyć wiedzę i pracę potrzebną, by mieć na swojej prywatnej stronie, stronie grupy turystycznej, drużyny harcerskiej, kapeli czy grupy religijnej śpiewnik z utworami ważnymi dla tej grupy. A przygotowanie tego śpiewnika poprzez profesjonalny skład tekstu dokona się za pomocą jednego kliknięcia. Zmniejszy to liczbę kiepskich śpiewników przygotowywanych i drukowanych przy pomocy oprogramowania nie najlepszej jakości i nie przystosowanego do tego celu, np. Microsoft Word.

Wykorzystanie tego systemu na innych stronach będzie wymagało linkowania do systemu bardzofajnyspiewnik.pl, co w długoterminowym rozrachunku spowoduje wysoki ranking strony w wyszukiwarkach klasy Google i potencjalnie umożliwi utrzymanie systemu z wpływów pochodzących z reklam.

### 2.2 Jakie problemy projekt rozwiązuje? Jakie daje zyski?

- Repozytorium piosenek dostępne przez internet umożliwi wykorzystywanie ich przez wiele narzędzi. Piosenki w formacie XML mogą być przetwarzane w wygodny sposób, przez bardzo wiele już istniejących narzędzi.
- Zastosowanie Javy i modelu cienkiego klienta służy uniezależnieniu systemu od platformy, a tym samym czyni je użytecznym dla szerszej grupy użytkowników.
- Oddzielenie warstwy prezentacji od warstwy danych. Ten sam zbiór piosenek użytkownik chce często w jednorodny sposób drukować, bądź prezentować na komputerze. Zastosowanie standardu XML do przechowywania piosenki zdecydowanie to ułatwia.
- Wielojęzykowość – poprzez zastosowanie standardu kodowania UTF-8 piosenki mogą zawierać znaki niemal wszystkich języków (nawet w obrębie tej samej piosenki).
- Historia – system będzie prowadził historię wszystkich modyfikacji piosenki (w formie drzewa).
- Drukowanie – system będzie przeprowadzał porządną skład piosenki do druku i udostępniał go w formacie PDF. Piosenka będzie mogła zostać złożona do druku na kartkach A4 lub A5.
- Edycja piosenki – dostępna natychmiast ze strony WWW (nie wymagająca ściągnięcia piosenki ani żadnego oprogramowania), przy pomocy wygodnego edytora.
- Prowadzenie śpiewników prywatnych (przypisanych do konta (użytkownika, organizacji)).
- Hierarchia organizacji i użytkowników (konto może być związane z innym kontem i tym samym udostępniać mu szersze uprawnienia). Np. rozważmy konto Jasia, konto zastępu i konto drużyny. Jaś jest w hierarchii przypisany do zastępu, a zastęp do drużyny. Piosenki chronione zastępu są dostępne dla Jasia, ale nie są dostępne dla ogółu.
- Sterowanie udostępnianiem piosenki ze swojego konta. Piosenka może być: prywatna, chroniona (dostępna dla niższych w hierarchii) oraz publiczna. Rozwiąże to problem utworów, które bądź nie nadają się do publicznej prezentacji, bądź są „jedyną słuszną wersją” używaną tylko w danym środowisku.
- Sterowanie udostępnianiem śpiewników ze swojego konta. Śpiewnik może być: prywatny, chroniony (dostępny dla niższych w hierarchii) oraz publiczny.
- Interfejs do prowadzenia dyskusji edytorów nad konkretną piosenką.

- Opcjonalnie: włączenie do opisów piosenek tabulatur lub nut.

## 2.3 Zależności i elementy składowe produktu

Na całość systemu `bardzofajny.piosennik.pl` składają się następujące elementy:

**Schemat bazy danych** – przechowuje informacje o użytkownikach systemu i ich profilach, a także elementy zawarte w dyskusjach nad piosenkami. Dopuszczalna jest także sytuacja, gdy baza danych będzie zawierała indeks piosenek znajdujących się w repozytorium na potrzeby przyśpieszenia przeszukowania. W tej sytuacji zostaną dostarczone mechanizmy przebudowy indeksu.

**Repozytorium piosenek** – w postaci repozytorium Subversion o odpowiedniej strukturze. Przechowuje wszystkie wersje piosenek (zarówno historię, jak i poszczególne gałęzie - także prywatne).

**Moduł przygotowywania wydruków** – składający się z systemu LaTeX wraz z odpowiednimi pakietami naszego autorstwa (umożliwiający skład piosenek i opcjonalnie śpiewników). Moduł ten będzie prawdopodobnie postawiony na oddzielnej maszynie, a wysyłanie plików w formacie TEX i odbiór wyników (PDF) będzie realizowany poprzez protokół HTTP (technologia web-service).

**Aplet edytora** – wygodny aplet umożliwiający edycję piosenek. W szczególności:

- wprowadzenie tekstu piosenek,
- wyróżnianie zwrotek,
- jednorazowe wprowadzanie powtarzających się fragmentów,
- rozmieszczanie akordów nad tekstem,
- edycję metadanych (wspomaganie poprzez np. udostępnianie list wcześniej wprowadzonych twórców),
- wygodne wpisywanie piosenek dostępnych gdzie indziej (wspomaganie procesu Copy-Paste),
- wielojęzyczność (UTF-8),
- dostęp bezpośrednio do źródła XML.

Aplet będzie pilnował zachowania odpowiedniego formatu piosenek. Opcjonalnie aplet będzie umożliwiał pracę offline.

**Strony JSP** – zarówno publiczne, jak i prywatne strony umożliwiające przeszukiwanie piosenek i pracę ze swoim profilem.

Opcjonalnie udostępnią też możliwość dyskusowania nad piosenką.

## Prezentacja produktu od strony fizycznej

System będzie miał możliwość działania na trzech rozłącznych maszynach (oczywiście wymienione komputery mogą być nawet jednym, jeżeli znajdzie się taki, spełniający wszystkie wymogi), a są to:

- komputer z Tomcat,
- komputer z LaTeX i Apache,
- komputer z PostgreSQL i Subversion.

System zostanie przygotowany w ten sposób, bo bardzo trudno znaleźć istniejącą maszynę spełniającą wszystkie trzy wymagania.

## 2.4 Technologie o które obecnie oparty jest system

- JSP, Struts (1.2.9)
- Ant - jako system budowania projektu
- SVN, (JavaSVN) - jako repozytorium piosenek
- PostgreSQL
- Hibernate 3.2
- Saxon - jako silnik transformacji XML (XSLT)

## **3 Funkcjonalność zaplanowana do wykonania w okresie marzec 2007 - czerwiec 2007**

### **3.1 Umożliwienie przygotowywania piosenki do wydruku**

Na chwilę obecną tej funkcjonalności praktycznie nie ma. Zatem zostanie on skonstruowany od zera. Jako, że kompilacja TeXa do formatu PDF jest dość zasobożerna, nie będzie się ona odbywać na serwerze, na którym stoi cały system, ale będzie wykonywana na innych komputerach (potencjalnie wielu), które będą udostępniały web-service'y przyjmujące TeXa i udostępniające w zamian PDFa.

#### **3.1.1 Klient web-service'u**

Klient web-service'u będzie częścią głównego modułu systemu bardzofajnyspiewnik.pl . Zostanie dodany on do już stworzonego interfejsu, zostanie też napisany moduł umożliwiający przekazywanie żądań do różnych dostępnych web-service'ów.

#### **3.1.2 Serwer web-service'u**

Sam web-service będzie mógł być napisany w różnych technologiach, w ramach projektu zostanie zrealizowana przynajmniej jedna wersja w PHP lub w Javie. Będzie udostępniał informacje o obciążeniu oraz o wykonanych już zadaniach oraz wystawiał do ściągnięcia już przygotowane PDFy.

### **3.2 Interfejs wyszukiwania piosenek - wyszukiwanie pełnotekstowe**

Obecnie system udostępnia dość niewygodny interfejs wyszukiwania piosenek. Zostanie on przebudowany - w szczególności zmniejszy się liczba wymaganych pól przez użytkownika, a pola edycyjne zostaną zastąpione listami wyboru.

Jednak kluczową modyfikacją będzie wprowadzenie mechanizmu szybkiego wyszukiwania ogólnego. Mechanizm będzie polegał na wprowadzeniu indeksu pełnotekstowego - indeksującego wszystkie piosenki (jako treść i metadane).

Użytkownik będzie mógł wpisać dowolny zbiór skojarzeń z poszukiwaną piosenką - i w ten sposób otrzymać interesującą go pozycję. Piosenki zostaną zaindeksowane za pomocą silnika Apache Lucene. System automatycznie będzie utrzymywał aktualny indeks.

### **3.3 Rozbudowa logiki biznesowej**

#### **3.3.1 Szybkość dostępu do SVN'a**

Poważnym mankamentem systemu w jego obecnym kształcie jest wydajność. Problem ten objawia się w szczególności przy odwołaniach do repozytorium piosenek (SVN). Operacja zbudowania drzewa zależności poszczególnych wersji piosenki potrafi zająć nawet kilkanaście sekund przy jednym działającym użytkowniku - co jest sytuacją nie dopuszczalną w systemie tej klasy.

By rozwiązać ten problem, zaimplementujemy następujące rozwiązania:

- Wprowadzenie puli połączeń do obsługi repozytorium SVN
- Wprowadzenie ceche'a zapamiętującego dane otrzymane od SVN'a - by zmniejszyć liczbę odwołań do svn'a

#### **3.3.2 Odciążenie bazy danych**

Poprawek wymaga także szybkość dostępu do bazy danych. W wielu miejscach kodu występuje problem n+1 (albo i jeszcze więcej) zapytań. Zostanie dokonana analiza dostępu do bazy danych i zostaną wprowadzone konieczne poprawki.

### **3.4 Budowanie projektu**

Projekt zostanie zreorganizowany w ten sposób, by był budowany przy pomocy narzędzia „Maven 2”.

## Dodatek: Zestawienie technologii, które zostaną wprowadzone do projektu

- System stanie się klientem web-service.
- Maven 2
- Apache Lucene

## 4 Przypadki użycia dla nowej funkcjonalności

### Spis treści

#### 4.1 Przeszukanie dostępnych piosenek

<b>Krótki opis</b>	Dowolny użytkownik ma dostęp do repozytorium wszystkich piosenek. Może wyszukiwać w nim to, czego potrzebuje.
<b>Cel</b>	Szybkie znajdowanie potrzebnych piosenek użytkownika.
<b>Wydajność</b>	Wyszukanie piosenek powinno trwać do kilku sekund.
<b>Aktorzy</b>	Dowolny użytkownik
<b>Warunki wstępne</b>	brak
<b>Przebieg zdarzenia</b>	<ol style="list-style-type: none"><li>1. Aktor wchodzi w zakładkę „wyszukaj piosenki”.</li><li>2. Aktor uzupełnia formularz HTML parametrów wyszukiwania:<ul style="list-style-type: none"><li>• autor tekstu i muzyki,</li><li>• tytuł lub tytuły,</li><li>• słowa kluczowe zwrotek i refrenu,</li><li>• początek pierwszej zwrotki,</li><li>• początek refrenu,</li><li>• data ostatniej modyfikacji,</li><li>• kategoria piosenki,</li><li>• album,</li><li>• rok powstania,</li><li>• tłumacz,</li><li>• ilość informacji muzycznych dostępnych dla tej piosenki.</li></ul></li><li>3. Aktor submituje formularz.</li><li>4. Aktor dostaje listę piosenek spełniających kryterium.</li></ol>
<b>Efekt</b>	Aktor dostaje listę piosenek spełniających kryterium.
<b>Kategoria</b>	core dla wyszukiwania po autorze, tytule i kategorii; reszta wyszukiwania - supporting

**Ryzyko**

średnie

**Uwagi**

W wersji podstawowej systemu wyszukiwanie jest po:

- autor tekstu i muzyki,
- tytuł lub tytuły,
- kategoria piosenki.

## 4.2 Otrzymanie pliku w formacie PDF z wybraną piosenką

<b>Krótki opis</b>	Użytkownik chce mieć możliwość wydruku aktualnie oglądanej piosenki. W tym celu może kliknąć „wydrukuj piosenkę” — otrzyma wtedy plik PDF z piosenką.
<b>Cel</b>	Możliwość druku pojedynczych piosenek.
<b>Wydajność</b>	Wygenerowanie PDFa z jedną piosenką powinno się mieścić w 30s limicie na request w protokole HTTP.
<b>Aktorzy</b>	Dowolny użytkownik
<b>Warunki wstępne</b>	Aktor wyświetla jakąś piosenkę.
<b>Przebieg zdarzenia</b>	<ol style="list-style-type: none"><li>1. Aktor klika na link „wydrukuj piosenkę”.</li><li>2. Aktor dostaje w zwrocie PDFa z piosenką.</li></ol>
<b>Efekt</b>	Wygenerowanie i przesłanie PDFa z piosenką.
<b>Kategoria</b>	core
<b>Ryzyko</b>	wysokie
<b>Uwagi</b>	Piosenka w PDFie jest zgodna z aktualną transpozycją akordów.



### 4.3 Wyszukanie śpiewnika

<b>Krótki opis</b>	Dowolny gość projektu może przeglądać publiczne śpiewniki. Do tego służy wyszukiwarka — po właścicielu i po nazwie Zalogowany użytkownik zarejestrowany może wyświetlić też „moje śpiewniki” — jest to de facto wyszukanie swoich śpiewników.
<b>Cel</b>	Przeszukiwanie śpiewników, znalezienie określonych śpiewników
<b>Wydajność</b>	System powinien spokojnie utrzymywać po kilka-kilkanaście śpiewników każdego użytkownika, czyli w sumie kilkadziesiąt śpiewników, wykonując na nich w niezauważalnym czasie operacje. Wyszukiwanie może jednak potrwać do kilku sekund, w zależności od złożoności zapytania.
<b>Aktorzy</b>	Gość, ew. użytkownik zarejestrowany.
<b>Warunki wstępne</b>	brak, ew. użytkownik zarejestrowany jest zalogowany
<b>Przebieg zdarzenia</b>	<p><b>Główny przebieg</b></p> <ol style="list-style-type: none"><li>1. Aktor wchodzi w zakładkę „wyszukaj śpiewnik”.</li><li>2. Aktor wypełnia formularz zapytania:<ul style="list-style-type: none"><li>• login właściciela,</li><li>• imię i nazwisko właściciela,</li><li>• słowo kluczowe w tytule.</li></ul></li><li>3. Aktor submituje formularz.</li><li>4. Aktor dostaje listę śpiewników spełniających zależności.</li></ol> <p><b>Przebieg alternatywny: użytkownik zarejestrowany klika</b></p> <ol style="list-style-type: none"><li>1. Aktor wchodzi w zakładkę „moje śpiewniki”, zostaje wysłane zapytanie „wyszukaj śpiewniki”:<ul style="list-style-type: none"><li>• login właściciela = zalogowanemu,</li><li>• imię i nazwisko właściciela — puste,</li><li>• słowo kluczowe w tytule — puste.</li></ul></li><li>2. Aktor dostaje listę śpiewników spełniających zależności, czyli jego własne śpiewniki.</li></ol>
<b>Efekt</b>	Aktor dostaje listę śpiewników spełniających zależności
<b>Kategoria</b>	supporting
<b>Ryzyko</b>	średnie
<b>Możliwe ulepszenia i rozwinięcia</b>	<ul style="list-style-type: none"><li>• Wyszukiwanie tylko wśród kont-rodziców.</li><li>• Podział wyników wyszukiwania na strony, po 50 wyników.</li></ul>
<b>Uwagi</b>	-

## 4.4 Wydrukowanie śpiewnika

<b>Krótki opis</b>	Gotowe śpiewniki nieraz, by zabrać na jakiś wyjazd lub po prostu pograć, trzeba wydrukować. Nasz system udostępnia opcję generowania plików PDF ze śpiewnikiem.
<b>Cel</b>	Wygenerowanie pliku PDF ze śpiewnikiem w celu późniejszego druku.
<b>Wydajność</b>	Generacja pliku PDF może zająć kilka minut.
<b>Aktorzy</b>	Użytkownik zarejestrowany.
<b>Warunki wstępne</b>	Aktor jest zalogowany. Posiada śpiewnik do wydrukowania. Kliknął na „moje śpiewniki” (wyszukał swoje śpiewniki). <ol style="list-style-type: none"><li>1. Aktor klika na link przy śpiewniku „wygeneruj PDF”.</li><li>2. Żądanie generacji zostaje wysłane do modułu LaTeXującego. W tym momencie link „wygeneruj PDF” staje się nieaktywny — plik jest w trakcie generacji.</li></ol>
<b>Przebieg zdarzenia</b>	<ol style="list-style-type: none"><li>3. Po zakończeniu generacji link obok „wygeneruj PDF” pojawia się „pobierz PDF”. Aktor klika na link.</li><li>4. PDF zostaje pobrany.</li><li>5. Po godzinie od generacji plik PDF zostaje usunięty z serwera i jest niedostępny.</li></ol>
<b>Efekt</b>	Zostaje wygenerowany i pobrany przez autora plik PDF ze śpiewnikiem.
<b>Kategoria</b>	supporting
<b>Ryzyko</b>	wysokie
<b>Uwagi</b>	<ul style="list-style-type: none"><li>• Generować swojego PDF może tylko właściciel śpiewnika. Jednak gotowego PDFa może on np. powieścić na swojej stronie internetowej.</li><li>• Dwuetapowa generacja - w pierw żądanie generacji, potem pobranie, jest konieczne ze względu na możliwy długi czas generacji PDF.</li><li>• Kolejna generacja PDF-a przykrywa poprzednią.</li></ul>

## 5 Podział pracy

### 5.1 Jakub Onufry Wojtaszczyk

- Dodanie interfejsu drukowania pojedynczej piosenki oraz śpiewnika
- Dodanie zarządzania zadaniami konwersji piosenek do formatu PDF
- Napisanie klienta web-service'u
- Napisanie serwera web-service'u

### 5.2 Piotr Tabor

- Przeniesienie projektu na Maven 2
- Wprowadzenie indeksu pełnotekstowego (Apache Lucene) i reorganizacja interfejsu „Wyszukiwania”
- Przyśpieszenie kodu dostępu do repozytorium SVN (pula połączeń - cache) i bazy danych (analiza zapytań i mapowania RO)