

Instytut Informatyki Uniwersytetu Wrocławskiego
Studencka Pracownia Inżynierii Oprogramowania, G4

Michał Kopacz, Mateusz Nahalewicz, Karol Bajko

Dokumentacja projektu
WiDz – Wirtualny Dziennik
Architektura oprogramowania

Wersja 1.0

Wrocław 2009

Tablica 1. Historia zmian w dokumencie

Lp.	Data	Nr wersji	Autor	Zmiana
1.	2009-12-08	1.0	Karol Bajko	Utworzenie dokumentu

Spis treści

1. Wstęp	3
2. Elementy architektury WiDz	3
2.1. Serwer	3
2.2. Baza danych	3
2.3. Pliki źródłowe	3
2.4. Interfejs użytkownika	4
3. Perspektywy architektoniczne	4
3.1. Perspektywa logiczna	4
3.2. Perspektywa implementacyjna	4
3.3. Perspektywa wdrożeniowa	4
3.4. Perspektywa procesowa	4
3.5. Perspektywa przypadków użycia	4
4. Słownik	4
Literatura	5

1. Wstęp

Niniejszy dokument ma na celu przedstawienie czytelnikowi wizji architektury aplikacji WiDz. Prezentuje on jej kluczowe składniki, a także skupia się na analizie architektury z kilku perspektyw – logicznej, implementacyjnej, wdrożeniowej, procesowej i przypadków użycia. Wszystko po to by ułatwić potencjalnemu klientowi zrozumienie kwestii architektonicznych WiDz.

Opisana poniżej architektura może ulec zmianie w fazie implementacji, lecz ogólny schemat aplikacji nie powinien zostać naruszony.

2. Elementy architektury WiDz

2.1. Serwer

Serwer jest podstawowym składnikiem architektury, gdyż na nim zostanie uruchomiona aplikacja WiDz. Posiada on zainstalowany system operacyjny, oparty o jądro Linux, a w nim dostępny interpreter języka Ruby, menadżer pakietów RubyGems do zarządzania bibliotekami, koniecznymi do uruchomienia aplikacji, oprogramowanie serwerowe HTTP Apache z modułem Passenger – niezbędny do współpracy z aplikacją stworzoną w Ruby on Rails, biblioteka kryptograficzna OpenSSL oraz SZBD MySQL.

Serwer jest odpowiedzialny za umożliwienie ciągłego dostępu do aplikacji każdemu użytkownikowi, bez względu na porę dnia. Odpowiada za okresowe tworzenie kopii bezpieczeństwa danych zawartych w bazie danych.

2.2. Baza danych

Baza danych jest kluczowym elementem aplikacji, ponieważ przechowuje wszystkie dane związane z oferowanymi usługami aplikacji WiDz. Zawiera dane nauczycieli, opiekunów i uczniów placówki szkolnej, wraz z informacjami o postępach w nauce, frekwencji. Przechowuje dane o dokonanych opłatach, obecnym harmonogramie zajęć w placówce oraz archiwizuje wszelkie wiadomości przesyłane pomiędzy użytkownikami WiDz.

Baza danych MySQL jest umieszczona na serwerze. Korzystać z niej może jedynie aplikacja WiDz, która będzie miała bezpośredni dostęp do danych w niej przechowywanych. Komunikacja pomiędzy aplikacją a SZBD MySQL opiera się na języku zapytań SQL.

2.3. Pliki źródłowe

Kod źródłowy WiDz został napisany w języku Ruby przy wykorzystaniu ramy projektowej Ruby on Rails. Język ten okazał się najlepszy do stworzenia aplikacji, ponieważ oferuje szybkie, łatwe tworzenie rozbudowanych projektów, umożliwiając przy tym większe skupienie na samych funkcjonalnościach, które ma oferować, sprowadzając do minimum konieczność niskopoziomowego kodowania, gdyż Ruby on Rails wprowadza mnóstwo gotowych,

domyślnych konwencji i bibliotek do wykorzystania¹. Sama aplikacja została stworzona przy wykorzystaniu wzorca projektowego MVC, które wydzieliło aplikację na trzy części:

- warstwa modelu odpowiedzialna za komunikację z bazą danych i przetwarzanie danych,
- warstwa widoku odpowiedzialna za prezentowanie wyników działania aplikacji na ekranie monitora,
- warstwa kontrolera odpowiedzialna za przechwytywanie żądań użytkownika i na ich podstawie generowanie odpowiedzi, wykorzystując dwie powyższe warstwy.

Jak widać, pliki źródłowe pełnią krytyczną rolę w aplikacji. Są one najważniejszą częścią całego projektu, ponieważ to one są odpowiedzialne za sprawną współpracę bazy danych i użytkowników korzystającym z WiDz.

2.4. Interfejs użytkownika

Interfejs użytkownika to element bardzo istotny w całym projekcie, którego sukces zależy w dużym stopniu od intuicyjności, przejrzystości i łatwości w obsłudze. Języki, które wykorzystano w celu zaprojektowania przyjaznego interfejsu to HTML, CSS, JavaScript.

3. Perspektywy architektoniczne

3.1. Perspektywa logiczna

3.2. Perspektywa implementacyjna

3.3. Perspektywa wdrożeniowa

3.4. Perspektywa procesowa

3.5. Perspektywa przypadków użycia

4. Słownik

- CSS
- HTML
- JavaScript
- MVC

Słownik terminów użytych w dokumencie znajduje się w [1].

¹ zasada "Konwencja ponad Konfigurację" (ang. *Convention over configuration*)

Literatura

- [1] Michał Kopacz, Mateusz Nahalewicz, Karol Bajko, *Dokumentacja projektu WiDz – Wirtualny Dziennik. Słownik*. Wrocław, SPIO IIUWr 2009.