

# POLITECHNIKA WROCŁAWSKA

## WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: INFORMATYKA

SPECJALNOŚĆ: INS

## PROJEKT INŻYNIERSKI

Budowa graficznego interfejsu  
użytkownika aplikacji sieciowej z  
wykorzystaniem JavaScript

Development of a web application's  
GUI with the use of JavaScript

AUTOR:

Mateusz Adamiak

PROWADZĄCY PRACĘ:

dr inż. Tomasz Kubik

OCENA PRACY:

---

WROCŁAW, 2012

# Spis treści

<b>1. Wstęp . . . . .</b>	<b>4</b>
1.1. Cel . . . . .	5
1.2. Zakres . . . . .	5
<b>2. Wymagania projektowe . . . . .</b>	<b>6</b>
2.1. Wymagania funkcjonalne . . . . .	6
2.2. Wymagania нефункционалне . . . . .	7
2.3. Cykl życia projektu . . . . .	7
<b>3. Analiza przypadków użycia? . . . . .</b>	<b>10</b>
<b>4. Opis implementacji . . . . .</b>	<b>11</b>
4.1. Pierwsza iteracja . . . . .	11
4.2. Druga iteracja . . . . .	12
<b>5. Testowanie . . . . .</b>	<b>14</b>
<b>6. Podsumowanie . . . . .</b>	<b>15</b>
<b>Literatura . . . . .</b>	<b>16</b>
<b>A. Instrukcja wdrożenia . . . . .</b>	<b>17</b>
<b>B. Instrukcja użytkownika . . . . .</b>	<b>18</b>

# Skróty

**XSD** (ang. *XML Schema Definition*)

**XML** (ang. *eXtensible Markup Language*)

**J2S** (ang. *Java2Script*)

# Rozdział 1

## Wstęp

W dzisiejszych czasach Internet to główne źródło wiedzy i informacji. Dziennie przeglądamy dziesiątki stron WWW, pobieramy wiele plików, kontaktujemy się z innymi użytkownikami, używamy bankowości internetowej. Nie sposób wymienić dzisiaj wszystkich funkcji, jakie oferuje nam sieć.

Jednym z podstawowych elementów działania sieci są serwisy internetowe. To dzięki nim możemy korzystać z dobrodziejstw, jakie daje nam Internet. Przeglądanie towarów w sklepie internetowym, skrzynka pocztowa, czytanie najnowszych wiadomości ze świata, oglądanie filmów online – takie aktywności wymagają stron internetowych lub aplikacji sieciowych.

Stajemy tu przed problemem komunikacji użytkownika z serwisem lub aplikacją sieciową. Człowiek, używając interfejsu użytkownika, ma możliwość interakcji z komputerem i korzystania z funkcji programu czy też strony internetowej.

Historia interfejsu użytkownika jest tak długa, jak długa jest historia komputera. Od czasu powstania maszyny liczącej, projektant stawał przed zadaniem skonstruowania odpowiedniego zestawu przyrządów, które ułatwią korzystającemu z maszyny interakcję z urządzeniem.

Rozróżniamy trzy rodzaje interfejsu użytkownika: tekstowy, graficzny i interfejs strony internetowej. Najuboższy jest tryb tekstowy, oferujący nam urządzenia wejścia-wyjścia w postaci klawiatury i wyświetlacza lub drukarki znakowej.

Jednak jako ludzie, jesteśmy nauczeni wskazywać, jest to dla nas naturalne, stąd drugi typ interfejsu, jakim jest interfejs graficzny. Aby umożliwić łatwiejszą interakcję z komputerem, wymyślono urządzenie, które popularnie nazywamy myszką. Pozwala nam ona wskazywać na elementy wyświetlacza i wchodzić w bardziej naturalną dla nas interakcję z komputerem. Postęp techniki w tym względzie prowadzi nas w stronę ekranów dotykowych, co ułatwia nam komunikację z maszynami. Niepotrzebne są wtedy żadne urządzenia zewnętrzne, takie jak myszka.

Ostatnim rodzajem interfejsu jest strona internetowa. To rozwiązanie powstało na potrzeby rozwoju sieci. Przeglądarka internetowa wyświetla nam pożądaną stronę, po której następnie nawigujemy i wchodzimy w interakcję z systemem, który jest po stronie serwera.

Przyszłość interfejsu użytkownika czeka jeszcze długa droga, by komunikacja z maszyną odbywała się w środowisku naturalniejszym dla człowieka. Prowadzone są badania nad rzeczywistością wirtualną, czy interfejsami mózg-maszyna, które mogą ułatwić użytkownikowi współpracę z komputerem.

Ten projekt inżynierski skupia się właśnie na implementacji interfejsu użytkownika. Jednak interfejs musi mieć jakąś funkcjonalność, którą reprezentuje, stąd projekt zakłada również implementację logiki biznesowej lub podłączenie się pod już zaimplementowaną funkcjonalność.

## **1.1. Cel**

Celem niniejszej pracy jest implementacja interfejsu graficznego dla użytkownika aplikacji sieciowej, realizującej funkcje wybranego systemu.

## **1.2. Zakres**

Zakres powstałego projektu obejmuje:

1. Wykonanie logiki biznesowej w języku Java,
2. Implementacja interfejsu graficznego dla użytkownika aplikacji, używając biblioteki graficznej SWT,
3. Wygenerowanie strony HTML – z zawartym skryptem Javascript – przy użyciu wtyczki do Eclipse’a – Java2Script,
4. Przetestowanie działania serwisu,
5. Napisanie dokumentacji projektowej wraz z instrukcją wdrożenia.

# Rozdział 2

## Wymagania projektowe

### 2.1. Wymagania funkcjonalne

Zrealizowany projekt został wykonany w oparciu o serwis EDAP na potrzeby pracy inżynierskiej na kierunku Informatyka na Politechnice Wrocławskiej. Projekt nosi nazwę AtEditor. Aplikacja stanowi narzędzie, mające na celu usprawnienie pisania aktów prawnych zgodnie ze schematem narzuconym przez ministerstwo.

Serwis EDAP – elektroniczna forma aktów prawnych, został przygotowany przez MSWiA w związku z wejściem w życie rozporządzenia w sprawie wymaga technicznych dokumentów elektronicznych zawierających akty normatywne i inne akty prawne, elektronicznej formy dzienników urzędowych oraz środków komunikacji elektronicznej i informatycznych nośników danych (Dz. U. 2008 Nr 75, poz 451 z późn. zm.).

Aplikacja służy do tworzenia oraz edycji aktów prawnych, wspomaga także zarządzanie oraz przechowywanie stworzonych dokumentów.

Akty prawne są generowane w postaci pliku z rozszerzeniem .xml oraz .pdf w oparciu o elektroniczny schemat aktów prawnych, przygotowany przez MSWiA w postaci pliku schema.xsd. W tym projekcie generowanie pliku zostało zastąpione wyświetleniem zawartości takiego pliku w aplikacji.

Aplikacja ma prezentować użytkownikowi przyjazny interfejs, który ułatwi mu korzystanie z takiego narzędzia, jak generator aktów prawnych. Istniejący serwis EDAP bywa pod tym względem mało intuicyjny. Korzystanie z JavaScript powinno umożliwić korzystanie ze stworzonej aplikacji za pomocą interfejsu webowego.

Poniżej przedstawiony został diagram przypadków użycia serwisu EDAP. Wyróżnione zostały funkcje systemu, które zostały zaimplementowane w ActEditor. • TODO: tu jakieś diagramy przypadków użycia EDAPa.

## 2.2. Wymagania niefunkcjonalne

Poniżej zostały przedstawione języki programowania, języki znaczników, narzędzia wspomagające oraz technologie użyte w projekcie.

### **Java**

Główny język programowania, użyty w projekcie. Oprogramowana jest w nim cała logika biznesowa, opierająca się na wczytaniu danych z pliku schema.xsd, wyświetleniu i obsłudze danych w oknie aplikacji, generowaniu rezultatu, jakim jest plik XML.

### **JavaScript**

### **HTML**

### **XML**

### **XSD**

### **Eclipse IDE**

### **Java2Script**

### **Github, EGit**

...//TODO: uzupełnić

## 2.3. Cykl życia projektu

Cykl życia projektu reprezentuje model przyrostowy (iteracyjny). W stosunku do modelu kaskadowego daje o wiele większą elastyczność, dzięki powrotom do wstępnych faz projektu i pozwala reagować na opóźnienia, czyli przyspieszać prace nad innymi częściami projektu. Daje też elastyczność pod względem definiowania wymagań – możemy początkowo zdefiniować wstępne wymagania, podczas gdy końcowa specyfikacja może powstać dopiero na etapie testowania.

## 1. Definiowanie projektu

### a) Etap inicjowania projektu

Inicjowanie projektu wymaga odpowiedzi na pytania: jaka jest tematyka przedsięwzięcia? Jakie są główne problemy zagadnienia? Jakie problemy ma rozwiązać projekt?

### b) Etap definiowania celu

Definiowanie zagadnienia polega na wyspecyfikowaniu kluczowego problemu, który formułujemy jako cel główny. Identyfikuje się tutaj również cele cząstkowe, które muszą zostać zrealizowane, by osiągnąć cel końcowy, a także prognozuje się potencjalne przeszkody, które mogą wpłynąć na sukces projektu.

### c) Etap weryfikacji

W tym etapie ocenia się możliwość wykonania postanowionych wymagań oraz weryfikuje szanse na osiągnięcie sukcesu projektu.

## 2. Planowanie

Podsumowanie tego, co musi być wykonane w ramach projektu, aby osiągnąć cel. Tworzy się tu podział na podzadania, które są łatwe do zorganizowania i zarządzania.

## 3. Wykonanie planu projektu

### a) Organizacja kamieni milowych

Ustalenie kolejności wykonywanych podzadań wraz z rozmieszczeniem ich w czasie. Specyfikowana jest tutaj kolejność wykonywania zadań oraz ich przewidywany czas wykonania.

### b) Wprowadzenie planu w życie

Wykonanie wyspecyfikowanych zadań i dążenie do ich realizacji przed upływem ich szacowanego czasu trwania.

### c) Przetestowanie aplikacji

Testowanie implementowanych części systemu. Najczęściej jest to etap iteracyjny, który jest wykonywany po zaimplementowaniu konkretnej funkcji w systemie.

## 4. Zamykanie projektu

### a) Ocena projektu



Następuje tu ocena odpowiedniości, efektywności, skuteczności projektu. Jest to przegląd osiągnięć realizowanego projektu w stosunku do początkowych oczekiwań. Odpowiada na pytania: czy rezultaty projektu były zgodne z oczekiwaniami? Czy prace zostały wykonane zgodnie z planem?

#### b) Wnioski

Wyciągnięcie wniosków z realizacji projektu. Opis doświadczeń zdobytych w trakcie wykonywania zadania, które mają służyć poprawie przyszłych projektów i programów.

## **Rozdział 3**

### **Analiza przypadków użycia?**

# Rozdział 4

## Opis implementacji

### 4.1. Pierwsza iteracja

#### 1. Definiowanie projektu

##### a) Etap inicjowania projektu

Pierwsza wersja aplikacji miała dotyczyć implementacji graficznego interfejsu użytkownika aplikacji sieciowej, jaką był serwis semWeb, którego tematyką było wykorzystanie metadanych i sieci semantycznych do wyszukiwania, integracji i weryfikacji urzędowych danych posiadających aspekt czasowo-przestrzenny.

##### b) Etap definiowania celu

Celem projektu miało być stworzenie aplikacji sieciowej, napisanej w języku Java, stworzenie interfejsu graficznego przy użyciu biblioteki graficznej SWT oraz wygenerowanie strony HTML za pomocą wtyczki do Eclipse'a JavaScript. Interfejs miał łączyć się z funkcjonalnością bazowego systemu semWeb, który dostarcza endpoint, umożliwiając zarządza-nie dziedzinową bazą wiedzy. Serwis bazowy jest oparty na fasadowym wzorcu projektowym, co było jednym z powodów, dla którego został on wybrany jako potencjalna baza, ze względu na łatwość integracji modułu interfejsu z resztą aplikacji.

Potencjalnymi zagrożeniami, wynikającymi z takiego rozwiązania, były problemy z importem plików o rozszerzeniu .jar do projektu. Powodem tego jest obecna implementacja wtyczki JavaScript, która opiera się na wczytywaniu skompilowanych klas środowiska Java, ale tylko tych, które umieszczone są bezpośrednio w projekcie, lub należą do konkretnych, wbudowanych bibliotek, jak java.lang (//TODO: odnośnik do strony J2S).

##### c) Etap weryfikacji

Z uwagi na to, że nie dało się dołączyć plików .jar do projektu w taki sposób, by kompilator JavaScript mógł prawidłowo załadować klasy oraz skonwertować je do plików z rozszerzeniem .js, projekt nie przeszedł pozytywnie przez fazę weryfikacji i został odrzucony.

Brak możliwości dołączania zewnętrznych archiwów .jar oraz wbudowanych bibliotek, takich jak java.sql, uniemożliwiło połączenie się z bazą danych. Konsekwencją tego był brak możliwości zbudowania aplikacji, która spełniałaby minimalne wymagania zawarte w etapie definiowania projektu. W związku z tym kolejnym krokiem w cyklu życia projektu było przejście do kolejnej iteracji i zdefiniowanie nowego problemu.

## 4.2. Druga iteracja

### 1. Definiowanie projektu

#### a) Etap inicjowania projektu

Druga wersja aplikacji miała dotyczyć implementacji graficznego interfejsu użytkownika aplikacji sieciowej, jaką był serwis EDAP, którego tematyką było tworzenie dokumentów prawnych, ich edycja oraz zarządzanie już istniejącymi dokumentami. Projekt ułatwia skonstruowanie wybranego dokumentu prawnego zgodnie ze schematem narzuconym przez MSWiA. Dokument generowany jest w postaci pliku o formacie XML, a wszelkie obostrzenia i zasady jego utworzenia zawiera plik schema.xsd. Używając serwisu EDAP, użytkownik może także wczytać już istniejący plik z rozszerzeniem .xml i zmodyfikować go.

#### b) Etap definiowania celu

Celem projektu było stworzenie aplikacji sieciowej, zawierającej główną funkcję serwisu EDAP, czyli konstruowanie dokumentu prawnego w postaci pliku XML na podstawie schematu zawartego w pliku schema.xsd.

Aby osiągnąć cel główny, należało wykonać 3 podstawowe kroki: odpowiednio wczytać zawartość pliku ze schematem XML, następnie stworzyć interfejs użytkownika, który wyświetlałby dostępne pozycje do wypełnienia przez użytkownika, z których potem generowany byłby plik XML.

Głównymi potencjalnymi problemami były operacje na plikach, takie jak wczytywanie i generowanie, które nie są wspierane przez JavaScript. Oprócz tego problemem mogło się okazać odpowiednie wczytanie zawartości pliku schema.xsd, gdyż nie można było skorzystać z zewnętrznych bibliotek, które by to ułatwiały, z uwagi na brak możliwości dołączenia takich bibliotek w taki sposób, by JavaScript mógł je załadować.

c) Etap weryfikacji

Projekt przeszedł pozytywnie etap weryfikacji z uwagi na możliwość stworzenia aplikacji, której celem byłoby skonstruowanie dokumentu prawnego w postaci pliku XML.

Projekt zakłada jedynie generowanie takiego pliku, nie jego modyfikację. Kolejnym założeniem jest brak generowania samego pliku XML, a jedynie wyświetlenie jego zawartości w aplikacji. Oba założenia wynikają z braku możliwości operacji na plikach, które nie są wspierane przez wtyczkę JavaScript.

2. Planowanie

3. Wykonanie planu projektu

a) Organizacja kamieni milowych

b) Wprowadzenie planu w życie

c) Przetestowanie aplikacji

4. Zamykanie projektu

a) Ocena projektu

b) Wnioski

## **Rozdział 5**

### **Testowanie**

## **Rozdział 6**

### **Podsumowanie**

# Literatura



## **Dodatek A**

### **Instrukcja wdrożenia**

## **Dodatek B**

### **Instrukcja użytkownika**