

**Uniwersytet Warszawski**  
Wydział Matematyki, Informatyki i Mechaniki

**Michał Mański**

Nr albumu: 209268

# **Archiwum dokumentów podpisanych elektronicznie**

**Praca magisterska  
na kierunku INFORMATYKA**

Praca wykonana pod kierunkiem  
**dr Janiny Mincer-Daszkiewicz**  
Instytut Informatyki

Czerwiec 2007

## **Oświadczenie kierującego pracą**

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

Data

Podpis kierującego pracą

## **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.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora pracy

## **Streszczenie**

W ramach pracy magisterskiej zaprojektowałem i zaimplementowałem elektroniczne archiwum na przechowywanie dokumentów podpisanych certyfikowanymi podpisami elektronicznymi, na potrzeby uczelni wyższej. Archiwum pozwala trzymać różnego rodzaju dokumenty, razem z metadanymi je opisującymi i umożliwia w prosty sposób rozszerzanie zestawu dopuszczalnych dokumentów. Przechowywane pliki mogą być zabezpieczone podpisem kwalifikowanym (na potrzeby kontaktów ze światem zewnętrznym), albo zwykłym podpisem elektronicznym (na wewnętrzne potrzeby uczelni). Archiwum dostępne jest poprzez WWW dla osób uprawnionych, a system uprawnień pozwala na przydzielanie indywidualnych praw do wykonywania różnych operacji dla poszczególnych osób. Archiwum bazuje na danych pobranych automatycznie z Uniwersyteckiego Systemu Obsługi Studiów i mogłoby w przyszłości zastąpić całkowicie papierowe archiwa uczelni.

## **Słowa kluczowe**

Archiwum Dokumentów, USOS, Java, MySQL, podpis elektroniczny, migrator, repozytorium, XML, X.509, PGP

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

11.3 Informatyka

## **Klasyfikacja tematyczna**

H. Information Systems  
H.3. Information Search and Retrieval  
H.3.4. Systems and Software  
H.3.5. Online Information Systems

## **Tytuł pracy w języku angielskim**

Archive of electronically signed documents



# Spis treści

<b>1. Wprowadzenie</b>	9
<b>2. Wstęp</b>	11
2.1. Definicje	11
2.2. Archiwa dokumentów	12
2.2.1. Archiwum tradycyjne vs archiwum elektroniczne	13
2.2.2. Bezpieczeństwo elektronicznych archiwów	15
2.3. Podpis elektroniczny	15
2.3.1. Cechy podpisu elektronicznego	15
2.3.2. Podpis elektroniczny — problemy	16
2.3.3. Formaty podpisu elektronicznego	17
2.4. Bezpieczne Archiwum Elektroniczne	18
2.4.1. Funkcjonalność BAE	19
2.4.2. Przesyłanie i przechowywanie dokumentów	20
2.5. Uniwex	20
2.5.1. Funkcjonalność systemu	20
2.5.2. Architektura systemu	22
2.6. Wnioski	22
<b>3. Analiza wymagań</b>	23
3.1. Wymagania funkcjonalne	23
3.1.1. Wymiana danych z USOS	23
3.1.2. Logowanie do systemu	23
3.1.3. Umieszczanie i przechowywanie dokumentów	23
3.1.4. Elastyczność systemu	23
3.1.5. Wyszukiwanie i pobieranie dokumentów	24
3.1.6. System uprawnień	24
3.1.7. Podpisy elektroniczne i certyfikaty	24
3.2. Wymagania нефункционалне	24
3.2.1. Pomocne informacje o stronach portalu	24
3.2.2. Obsługa dwóch języków	24
3.2.3. Interfejs użytkownika	25
<b>4. Projekt systemu</b>	27
4.1. Osobna aplikacja czy integracja z jakąś istniejącą?	27
4.2. Użytkownicy systemu	27
4.2.1. Zwykły użytkownik	27
4.2.2. Administrator jednostki organizacyjnej	28

4.2.3.	Administrator systemu . . . . .	28
4.3.	Dokumenty . . . . .	28
4.3.1.	Format pliku XML opisującego typ dokumentu . . . . .	29
4.3.2.	Przechowywanie dokumentów . . . . .	32
4.3.3.	Weryfikacja umieszczanych metadanych . . . . .	33
4.4.	Dekompozycja logiczna systemu . . . . .	33
4.4.1.	Warstwa danych . . . . .	33
4.4.2.	Schemat bazy danych . . . . .	33
4.4.3.	Tabele systemu . . . . .	33
4.5.	Struktura systemu . . . . .	40
4.6.	Warstwa prezentacji . . . . .	41
<b>5.</b>	<b>Implementacja . . . . .</b>	<b>43</b>
5.1.	Wybór technologii . . . . .	43
5.1.1.	Java . . . . .	43
5.1.2.	Apache Tomcat . . . . .	44
5.1.3.	MySQL . . . . .	44
5.1.4.	L <sup>A</sup> T <sub>E</sub> X . . . . .	44
5.2.	Narzędzia . . . . .	44
5.2.1.	Eclipse . . . . .	44
5.2.2.	Apache Ant . . . . .	44
5.2.3.	DBDesigner . . . . .	44
5.3.	Elementy systemu . . . . .	44
5.4.	Obsługa żądań WWW . . . . .	45
5.5.	Komunikacja z bazą danych . . . . .	45
5.6.	Logika systemu . . . . .	46
5.6.1.	Typy dokumentów . . . . .	46
5.6.2.	Uprawnienia . . . . .	46
5.6.3.	Podpisy elektroniczne . . . . .	46
5.7.	Prezentacja danych . . . . .	48
5.7.1.	Obsługa dwóch języków . . . . .	49
5.7.2.	AJAX . . . . .	49
<b>6.</b>	<b>Instalacja i uruchomienie systemu . . . . .</b>	<b>51</b>
6.1.	Wymagania <i>Archiwum</i> względem środowiska . . . . .	51
6.1.1.	Java . . . . .	51
6.1.2.	System operacyjny . . . . .	51
6.1.3.	Apache Tomcat . . . . .	51
6.1.4.	Apache Ant . . . . .	51
6.1.5.	MySQL . . . . .	51
6.2.	Kompilacja <i>Archiwum</i> . . . . .	52
6.3.	Instalacja <i>Archiwum</i> . . . . .	52
6.4.	Uruchomienie <i>Archiwum</i> . . . . .	53
<b>7.</b>	<b>Podręcznik użytkownika . . . . .</b>	<b>55</b>
7.1.	Interfejs zwykłego użytkownika . . . . .	55
7.1.1.	Rozpoczęcie pracy z systemem . . . . .	55
7.1.2.	Nowy dokument . . . . .	55
7.1.3.	Moje dokumenty . . . . .	59

7.1.4.	Wyszukiwarka . . . . .	59
7.1.5.	Przeglądanie archiwum . . . . .	60
7.1.6.	Podgląd dokumentu . . . . .	62
7.1.7.	Edycja dokumentu . . . . .	62
7.1.8.	Podgląd załącznika dokumentu . . . . .	64
7.1.9.	Podgląd klucza publicznego PGP . . . . .	64
7.1.10.	Podgląd certyfikatu X.509 . . . . .	64
7.1.11.	Blokada systemu . . . . .	68
7.2.	Interfejs administratora jednostki . . . . .	68
7.2.1.	Typy dokumentów . . . . .	68
7.2.2.	Certyfikaty . . . . .	70
7.2.3.	Użytkownicy . . . . .	70
7.3.	Interfejs administratora systemu . . . . .	73
7.3.1.	Role . . . . .	73
7.3.2.	Inne . . . . .	73
<b>8.</b>	<b>Podsumowanie . . . . .</b>	<b>77</b>
8.1.	Zrealizowane założenia . . . . .	77
8.2.	Możliwe rozszerzenia . . . . .	78
8.2.1.	Podpisy elektroniczne . . . . .	78
8.2.2.	Inne funkcje . . . . .	78
8.3.	Podziękowania . . . . .	78
<b>A.</b>	<b>Opis zawartości płyty CD . . . . .</b>	<b>79</b>
<b>B.</b>	<b>Skrypt tworzący widoki po stronie systemu USOS na potrzeby <i>Archiwum</i> . . . . .</b>	<b>81</b>
<b>C.</b>	<b>Konfiguracja migratora . . . . .</b>	<b>83</b>





# Spis rysunków

2.1. Znakowanie czasem — źródło: en.wikipedia.org . . . . .	17
2.2. Aplikacja kliencka BAE . . . . .	19
2.3. System Uniwex . . . . .	21
4.1. Schemat bazy danych cz. 1 . . . . .	34
4.2. Schemat bazy danych cz. 2 . . . . .	35
4.3. Struktura <i>Archiwum</i> . . . . .	41
7.1. Logowanie do <i>Archiwum</i> . . . . .	56
7.2. Strona powitalna <i>Archiwum</i> . . . . .	56
7.3. Dodawanie nowego dokumentu . . . . .	58
7.4. Wyszukiwarka cykli dydaktycznych . . . . .	59
7.5. Ekran <i>Moje dokumenty</i> . . . . .	60
7.6. Wyszukiwarka dokumentów . . . . .	61
7.7. Wyniki wyszukiwarki dokumentów . . . . .	61
7.8. Przeglądanie <i>Archiwum</i> . . . . .	62
7.9. Podgląd dokumentu . . . . .	63
7.10. Edycja dokumentu . . . . .	65
7.11. Podgląd załącznika . . . . .	66
7.12. Szczegóły podpisu załącznika . . . . .	66
7.13. Podgląd klucza publicznego PGP . . . . .	67
7.14. Podgląd certyfikatu X.509 . . . . .	67
7.15. Blokada systemu . . . . .	68
7.16. Zakładka <i>TYPY DOKUMENTÓW</i> . . . . .	69
7.17. Edycja typu dokumentu . . . . .	69
7.18. Zakładka <i>CERTYFIKATY</i> . . . . .	71
7.19. Zakładka <i>UŻYTKOWNICY</i> . . . . .	72
7.20. Zakładka <i>ROLE</i> . . . . .	74
7.21. Edycja roli . . . . .	74
7.22. Zakładka <i>INNE</i> . . . . .	75



# Rozdział 1

## Wprowadzenie

Obecnie prawie każda firma lub organizacja wykorzystuje każdego dnia różnego rodzaju dokumenty, zarówno do wewnętrznych celów jak i w kontaktach zewnętrznych. Bardzo często pojawia się wówczas potrzeba gromadzenia, klasyfikowania, udostępniania i przechowywania tych dokumentów. Tradycyjnym podejściem do tego problemu jest archiwum dokumentów trzymany w postaci papierowej, jednakże ma ono wiele wad i obecnie w epoce silnie postępującej informatyzacji jedynym słusznym rozwiązaniem wydaje się być archiwum dokumentów trzymany w postaci elektronicznej. Takie elektroniczne repozytorium otwiera wiele nowych możliwości i pozwala pod wieloma względami na usprawnienie pracy z dokumentami, jednakże podczas korzystania z dokumentów elektronicznych pojawiają się nowe problemy, które trzeba rozwiązać. Najważniejszą sprawą jest zapewnienie odpowiedniego bezpieczeństwa przechowywanym danym, aby niepowołane osoby nie miały dostępu do informacji, które nie są dla nich przeznaczone, a ponadto, by niemożliwe było sfałszowanie gromadzonych w ten sposób dokumentów. W tym celu konieczne jest zaimplementowanie w elektronicznym archiwum systemu uprawnień dostosowanego do konkretnych potrzeb oraz zabezpieczenie trzymany dokumentów podpisem elektronicznym.

Również na Uniwersytecie Warszawskim wiele dokumentów jest używanych na codzień i trzymany w postaci papierowej. Liczba takich dokumentów stale rośnie i problematyczne staje się ich przechowywanie, a w związku z tym dostęp do nich często jest mocno ograniczony. Zatem na UW także przydatne byłoby utworzenie elektronicznego repozytorium, które pozwalałoby na szybki dostęp poprzez internet do różnych dokumentów z dowolnego miejsca.

Moim zadaniem w ramach pracy magisterskiej było zaprojektowanie i zaimplementowanie prototypowego elektronicznego archiwum dokumentów, które może być załącznikiem do uruchomienia w przyszłości na całym UW elektronicznego archiwum, które pozwalałoby na szybki dostęp do wszystkich przechowywanych dokumentów — zarówno tych bieżących jak i archiwalnych, i w rezultacie przyczyniło się do całkowitego wyeliminowania papierowych archiwów uczelni.

W niniejszej pracy omówię na początku archiwa dokumentów, porównam tradycyjne podejście do archiwizacji z przechowywaniem dokumentów w postaci elektronicznej i przedstawię sposoby na zwiększenie bezpieczeństwa przechowywanych dokumentów elektronicznych. Podam także zastosowania takich archiwów, w szczególności omówię krótko bezpieczne archiwum dokumentów, które jest oferowane przez Unizeto, a także system Uniwex [Uniwex], który wykorzystuje podpisy elektroniczne do zarządzania kursami i egzaminami na uczelni wyższej. Opiszę także dokładnie podpisy elektroniczne, ich cechy i zastosowania oraz porównam między sobą różne standardy i formaty takich podpisów.

Następnie opiszę *Archiwum dokumentów podpisanych elektronicznie* (w skrócie *Archiwum*), które jest przeznaczone do przechowywania różnego rodzaju dokumentów używanych na Uniwersytecie Warszawskim. Pozwala ono na trzymanie dokumentów wraz z metadanymi je opisującymi oraz

zawiera elastyczny system ról i uprawnień umożliwiający przydzielanie indywidualnych praw do wykonywania różnych operacji dla poszczególnych osób w zależności od rodzaju danego dokumentu. Dane wykorzystywane do opisywania dokumentów oraz część definicji uprawnień pobierana jest automatycznie z systemu USOS za pomocą programu Migrator [Migrator]. *Archiwum* umożliwia umieszczanie zarówno dokumentów podpisanych za pomocą certyfikatu kwalifikowanego jak i dokumentów ze zwykłym podpisem elektronicznym (akceptowane są podpisy w standardzie PGP oraz X.509), przy czym możliwe jest wstawienie dokumentów podpisanych jedynie za pomocą wcześniej zarejestrowanych w nim certyfikatów. Najpierw zostanie przedstawiony projekt, a następnie implementacja zaprojektowanego systemu. Ponadto opisałem sposób instalacji systemu oraz zamieściłem podręcznik dla użytkowników. Ostatni rozdział zawiera wnioski, jakie pojawiły się po napisaniu programu.

Ponadto w skład pracy wchodzi materiały dostarczone na płycie CD, na której znajdują się:

- kod źródłowy *Archiwum*,
- plik konfiguracyjny Migratora,
- skrypt generujący odpowiednie widoki po stronie systemu USOS na potrzeby *Archiwum* wykorzystywane przez Migratora,
- niniejszy dokument w formacie PDF wraz ze źródłami w formacie  $\text{\LaTeX}$ .

# Rozdział 2

## Wstęp

### 2.1. Definicje

**Definicja 1** *Archiwum* — system do przechowywania dokumentów podpisanych elektronicznie na potrzeby Uniwersytetu Warszawskiego (projekt i implementacja tego systemu jest tematem niniejszej pracy magisterskiej) — pełna nazwa systemu to: Archiwum dokumentów podpisanych elektronicznie.

**Definicja 2** *Podpis elektroniczny* — dane w postaci elektronicznej, które wraz z innymi danymi, do których zostały dołączone lub z którymi są logicznie powiązane, służą do identyfikacji osoby składającej podpis elektroniczny. Najpopularniejsze standardy pozwalające na złożenie podpisu elektronicznego to X.509 oraz PGP.

**Definicja 3** *Certyfikat* — Certyfikat w rozumieniu Ustawy o podpisie elektronicznym [Ust01] jest elektronicznym zaświadczeniem, za pomocą którego dane służące do weryfikacji podpisu elektronicznego są przyporządkowane do osoby składającej podpis elektroniczny i które umożliwiają identyfikację tej osoby.

**Definicja 4** *Bezpieczny podpis elektroniczny* — jest to taki podpis elektroniczny, któremu ustawa [Ust01] nadaje status równoważny z podpisem własnoręcznym. Może być wykorzystywany do podpisywania wszelkiego rodzaju oświadczeń woli, np. PIT-ów, podań, ofert, umów itp. Bezpieczny podpis elektroniczny (zwany często podpisem kwalifikowanym) może zostać złożony tylko przy użyciu certyfikatu kwalifikowanego. Klucze i certyfikaty używane do składania bezpiecznego podpisu przechowywane są przeważnie na kartach kryptograficznych. Do takich podpisów wykorzystuje się standard X.509.

**Definicja 5** *Certyfikat kwalifikowany* — służy do weryfikacji bezpiecznego podpisu elektronicznego, oznacza to, że w stosunku do zwykłych certyfikatów musi on spełnić dodatkowe wymagania związane z bezpieczeństwem oraz weryfikacją tożsamości osoby posługującej się tym certyfikatem. Certyfikaty kwalifikowane mogą być wydawane tylko przez centra certyfikacji — w Polsce są 3 takie centra: Certum [CERTUM], Sigillum [SIGILLUM] i KIR (Krajowa Izba Rozliczeniowa) [KIR].

**Definicja 6** *OpenPGP [PGP]* — standard kryptograficzny pozwalający na szyfrowanie i podpisywanie wiadomości, opisuje system zdecentralizowany, w którym poziom autentyczności danego klucza jest determinowany przez sumę podpisów, złożonych przez różne osoby znające posiadacza klucza. Został oparty na programie PGP (Pretty Good Privacy) — jednym z najpopularniejszych narzędzi do szyfrowania poczty elektronicznej. W niniejszej pracy będę używał nazwy PGP do określenia podpisów w tym standardzie.

**Definicja 7** X.509 [X509] — standard opisujący system scentralizowany, w którym autentyczność klucza jest gwarantowana przez hierarchię centrów certyfikacji formalnie poświadczających związek klucza z tożsamością jego właściciela. Ze względów formalnych X.509 jest obecnie dominującym systemem, na którym opiera się aktualnie obowiązujące prawodawstwo o podpisie elektronicznym.

**Definicja 8** Klucz publiczny PGP — nazwa Certyfikatu w systemie PGP.

**Definicja 9** Certyfikat X.509 — nazwa Certyfikatu w systemie X.509.

**Definicja 10** Ścieżka certyfikacji — uporządkowany ciąg certyfikatów, prowadzący od certyfikatu punktu zaufania (centrum certyfikacji), aż do wybranego certyfikatu, utworzony w celu weryfikacji tego certyfikatu. Certyfikat niższego poziomu jest zawsze podpisany kluczem prywatnym związanym z certyfikatem z jednego poziomu wyżej.

**Definicja 11** PKCS#11 — standard opisujący abstrakcyjny interfejs programisty (niezależny od platformy) dla różnych żetonów (ang. token) kryptograficznych. Żeton to element (np. urządzenie elektroniczne), który przechowuje dane (głównie klucze i certyfikaty, ale nie tylko) i ma możliwość wykonywania operacji kryptograficznych.

**Definicja 12** Uniwersytecki System Obsługi Studiów [USOS] — system wspomagający wszelkie czynności związane z tokiem studiów. W bazie danych systemu USOS przechowywane są informacje dotyczące wszystkich studentów i pracowników uczelni, a także wszystkich przedmiotów, zajęć, jednostek organizacyjnych itd.

**Definicja 13** USOSweb [USOSweb] — aplikacja internetowa, pozwalająca ogółowi użytkowników Internetu na dostęp do wybranych danych z systemu USOS, udostępniająca m.in. rejestracje na przedmioty i do grup, wyniki ze sprawdzianów i egzaminów, informacje o programach studiów itp.

**Definicja 14** Migrator [Migrator] — system służący do synchronizacji baz danych systemów USOS z innymi aplikacjami UW, takimi jak USOSweb i Archiwum Prac Dyplomowych [APD]. Będzie służył również do przenoszenia danych między aplikacjami USOS i Archiwum — migracja będzie odbywała się tylko w jedną stronę: z systemu USOS do Archiwum.

**Definicja 15** Użytkownik — osoba korzystająca z Archiwum. Wszystkie informacje o użytkownikach pochodzą z systemu USOS.

**Definicja 16** JEE [JavaEE] — definiuje standard tworzenia aplikacji opartych na architekturze wielowarstwowej. JEE wykorzystuje język Java jako podstawę programowania logiki aplikacji oraz definiuje środowisko wykonania i model aplikacji.

## 2.2. Archiwa dokumentów

Obecnie prawie każda firma lub organizacja przetwarza wiele dokumentów różnych typów, które muszą być przechowywane i później udostępniane poszczególnym osobom. Standardowo realizowane jest to poprzez tworzenie archiwów dokumentów trzymanyh w postaci papierowej, obecnie jednak często nie sprawdza się takie rozwiązanie, gdyż jest ono bardzo niewydatne i często mocno ogranicza dostęp do szukanych informacji. Nowoczesnym podejściem jest elektroniczne archiwum dokumentów, które pozwala na szybki i bezpośredni dostęp zarówno do bieżących, jak i starych informacji. Jednocześnie poprawiona jest jakość obsługi klientów, która jest bardzo ważna w dzisiejszym biznesie.

Przechodząc z archiwum papierowego na elektroniczne trzeba pamiętać o jednej sprawie — konieczne jest przetworzenie wszystkich dotychczas używanych dokumentów na postać elektroniczną. Oczywiście nikt nie będzie takich dokumentów przepisywał ręcznie do komputera i tworzył plików w formacie *Word* czy *PDF*. Jedynym sensownym rozwiązaniem jest tutaj zeskanowanie takich dokumentów i trzymanie w systemie plików np. w formacie *TIFF* lub *JPG*.

Podstawową funkcjonalność elektronicznego archiwum stanowi: umieszczanie dokumentów w archiwum, pobieranie wcześniej wstawionych dokumentów oraz wyszukiwanie ich wg różnych kryteriów. Często wraz z nimi trzymane są też metadane je opisujące, co pozwala na ich łatwiejsze wyszukiwanie. Przydatne jest też klasyfikowanie ich na różne rodzaje, wówczas różnym dokumentom można przypisywać inne metadane — w zależności od ich rodzaju.

### 2.2.1. Archiwum tradycyjne vs archiwum elektroniczne

W celu przedstawienia wszystkich zalet elektronicznych systemów zarządzania dokumentami w załączonej w tym punkcie tabeli porównam je z systemami tradycyjnymi. Pozwala to łatwo zorientować się zarówno w zaletach, jak i wynikających z nich korzyściach finansowych.

<b>Kryterium</b>	<b>System tradycyjny</b>	<b>System elektroniczny</b>
<b>Czas dostępu</b>	<b>Minuty</b> W zależności od sposobu organizacji archiwum papierowego, czas dostępu mierzony jest w minutach, a — w niektórych przypadkach — w godzinach lub dniach. Przez czas dostępu rozumieć należy czas podejścia do miejsca przechowywania, odszukania odpowiedniego dokumentu oraz powrotu na stanowisko robocze.	<b>Sekundy</b> Większość systemów elektronicznych pozwala na udostępnienie poszukiwanego dokumentu w ciągu kilku sekund od przesłania zapytania.
<b>Sposób dostępu</b>	<b>Fizyczny</b> Dostęp wyłącznie fizyczny do oryginału papierowego lub ewentualnie do jego kopii. W wielu przypadkach uniemożliwia to jednoczesną pracę kilku osób nad jednym dokumentem.	<b>Sieciowy</b> Dokument elektroniczny jest dostępny w sieci komputerowej. Wszyscy uprawnieni użytkownicy mają natychmiastowy dostęp do tego samego dokumentu.
<b>Kolejność składowania</b>	<b>Narzucona</b> Dokumenty papierowe mogą być składowane wyłącznie według jednego kryterium (np. numer, data, nazwa kontrahenta itp.).	<b>Dowolna</b> W archiwum elektronicznym nie ma takiego ograniczenia.

<b>Sposób wyszukiwania</b>	<b>Utrudniony</b> Przeszukanie zbioru dokumentów po kryterium niezgodnym z przyjętą kolejnością składowania może być bardzo czasochłonne.	<b>Wielokryterialny</b> Wyszukiwanie może się odbywać według wielu jednoczesnych kryteriów, niezależnie od kolejności składowania. Rezultaty mogą być automatycznie sortowane, a w niektórych przypadkach możliwe jest odszukanie dokumentu według dowolnego słowa w opisie lub treści.
<b>Objętość archiwum</b>	<b>Duża</b> Archiwum tradycyjne zajmuje znaczną powierzchnię. Dodatkowo konieczność zapewnienia łatwego dostępu do dokumentów wymaga inwestycji w wyposażenie archiwum (regały przesuwne, segregatory, szafy obrotowe itp.). Często także wymagane jest powielanie dokumentu i przechowywanie kopii tych samych dokumentów (np. w przypadku dużych biur pozwala to na skrócenie czasu dostępu do dokumentu), co dodatkowo zwiększa objętość archiwum.	<b>Znikoma</b> Elektroniczne dokumenty przechowywane są na dyskach i praktycznie nie zajmują powierzchni biurowej. Źródłowe dokumenty papierowe — jeśli nie mogą być zniszczone — mogą być przechowywane w tańszych lokalach lub archiwach prowadzonych usługowo przez firmy zewnętrzne.
<b>Ryzyko zniszczenia</b>	<b>Znaczne</b> Dokumenty mogą ulec zniszczeniu na skutek zdarzeń losowych (pożar, powódź) lub celowych działań pracowników. Wykonanie kopii zapasowej i jej zabezpieczenie jest zazwyczaj bardzo kosztowne.	<b>Znikome</b> Także istnieje ryzyko zniszczenia na skutek zdarzeń losowych, ale ze względu na łatwość wykonania kopii zapasowej to ryzyko prościej ograniczyć.
<b>Ryzyko zgubienia</b>	<b>Duże</b> Przetrzymanie lub nieprawidłowe odłożenie dokumentu utrudnia lub uniemożliwia późniejsze jego odszukanie.	<b>Znikome</b> Praktycznie brak możliwości zgubienia dokumentu.
<b>Jakość zapisu</b>	<b>Zmienna</b> Na skutek intensywnej eksploatacji dokument może ulec zużyciu lub wyblaknąć.	<b>Stać</b> Cyfrowa jakość kopii jest niezależna od częstotliwości dostępu.
<b>Administracja</b>	<b>Utrudniona</b> Przesyłanie oraz zapewnienie odpowiednich praw dostępu do dokumentu wymaga rozbudowanego systemu rozdzielnictwa i ewidencji.	<b>Łatwa</b> Prawa dostępu przydzielane przez administratora. Brak konieczności kopiowania i przesyłania dokumentów.



Widać, że przedstawione porównanie ma swoje implikacje w warstwie ekonomicznej. Dzięki szybkości dostępu obniżone zostają koszty osobowe związane z czasem wyszukiwania. Dodatkowo obniżeniu ulegają koszty kopiowania, przechowywania i administrowania dokumentami.

### **2.2.2. Bezpieczeństwo elektronicznych archiwów**

Bardzo istotną kwestią archiwum dokumentów jest bezpieczeństwo przechowywanych danych. Elektroniczne archiwum powinno być bardzo bezpieczne, tzn. skonstruowane tak, żeby niepowołany użytkownik nie miał dostępu do informacji, które nie są dla niego przeznaczone. W tym celu w archiwum powinny być zdefiniowane różne role lub grupy uprawnień określające, kto może umieszczać i czytać dokumenty danego typu.

Trzeba też pamiętać o tym, że zwykle dokumenty elektroniczne nie mają takiej mocy prawnej jak dokumenty przechowywane w tradycyjnym archiwum. Dokumenty papierowe są z reguły podpisane przez pewną osobę (np. autora lub właściciela dokumentu) i po kilku latach możemy zweryfikować, z kim są one związane lub do kogo należą. W momencie, gdy włożymy do systemu taki zeskanowany dokument lub wygenerowany plik PDF, nie może on za jakiś czas posłużyć za dowód, że dotyczy pewnej osoby. Aby to osiągnąć musimy trzymać w archiwum dokumenty podpisane elektronicznie. Wówczas w każdej chwili będziemy mogli zweryfikować podpis złożony pod dokumentem i dowiedzieć się przez kogo został podpisany, czyli poznać osobę która zaświadcza o jego autentyczności. Natomiast w przypadku, gdy podpis nie zostanie poprawnie zweryfikowany, będziemy wiedzieli, że ten dokument prawdopodobnie został sfałszowany i nie powinniśmy ufać jego treści.

Rozważając dokument podpisany elektronicznie można mówić o różnych typach podpisów. Gdy używane są podpisy kwalifikowane, to taki dokument z archiwum będzie miał taką samą moc prawną jak podpisany ręcznie dokument papierowy. Jednakże nie zawsze opłaca się używać certyfikatów kwalifikowanych (choćby ze względu na koszt takiego certyfikatu). Na wewnętrzne potrzeby często wystarcza używanie zwykłych podpisów w systemie *X.509* lub podpisów *PGP*.

Innym aspektem zwiększającym bezpieczeństwo archiwum jest przechowywanie dokumentów na dysku w postaci zaszyfrowanej. Wówczas nawet jak ktoś niepowołany dostanie się do takiego dokumentu, będzie miał problem z odczytaniem go, gdyż nie będzie w stanie go odszyfrować. Ponadto elektroniczne archiwum powinno znajdować się w bezpiecznym budynku wyposażonym w różnej klasy zabezpieczenia, które dodatkowo chroniłyby dostęp do przechowywanych danych.

Bezpieczne systemy zarządzania dokumentami posiadające wspomniane cechy, mają wiele zastosowań i mogą przynieść spore korzyści różnym firmom i organizacjom, a także osobom prywatnym. W przypadku potrzeby złożenia podania lub oświadczenia czy podpisania umowy, taki system umożliwia wykonanie tego z dowolnego miejsca, bez konieczności umawiania się na konkretne spotkanie lub czekania w urzędach w długich kolejkach, a jedynie poprzez wymianę podpisanych dokumentów w systemie — pozwala to wielu osobom zaoszczędzić mnóstwo czasu w stosunku do sytuacji, w której musieliby tą czynność wykonywać korzystając z dokumentów papierowych.

## **2.3. Podpis elektroniczny**

### **2.3.1. Cechy podpisu elektronicznego**

Podpisu elektronicznego służy zapewnieniu następujących cech:

- autentyczności, czyli pewności co do autorstwa dokumentu,
- niezaprzeczalności informacji, czyli autor nie może wyprzeć się utworzenia wiadomości, gdyż podpis cyfrowy stanowi dowód jego utworzenia,

- integralności, czyli pewności, że wiadomość nie została zmodyfikowana po złożeniu podpisu przez autora.

Podpisy elektroniczne w wielu zastosowaniach mogą ułatwić życie. W porównaniu z podpisanymi odręcznymi trudniej jest je podrobić i pozwalają na łatwiejszą weryfikację zmian w podpisywanych dokumentach. Najpopularniejsze standardy pozwalające na złożenie podpisu elektronicznego to *X.509* oraz *PGP*.

### 2.3.2. Podpis elektroniczny — problemy

Podpisy elektroniczne mają wiele zalet, jednakże dokument w formie elektronicznej i podpisany elektronicznie stwarza szereg nowych wyzwań i problemów, które trzeba rozwiązać.

#### Konserwacja podpisu

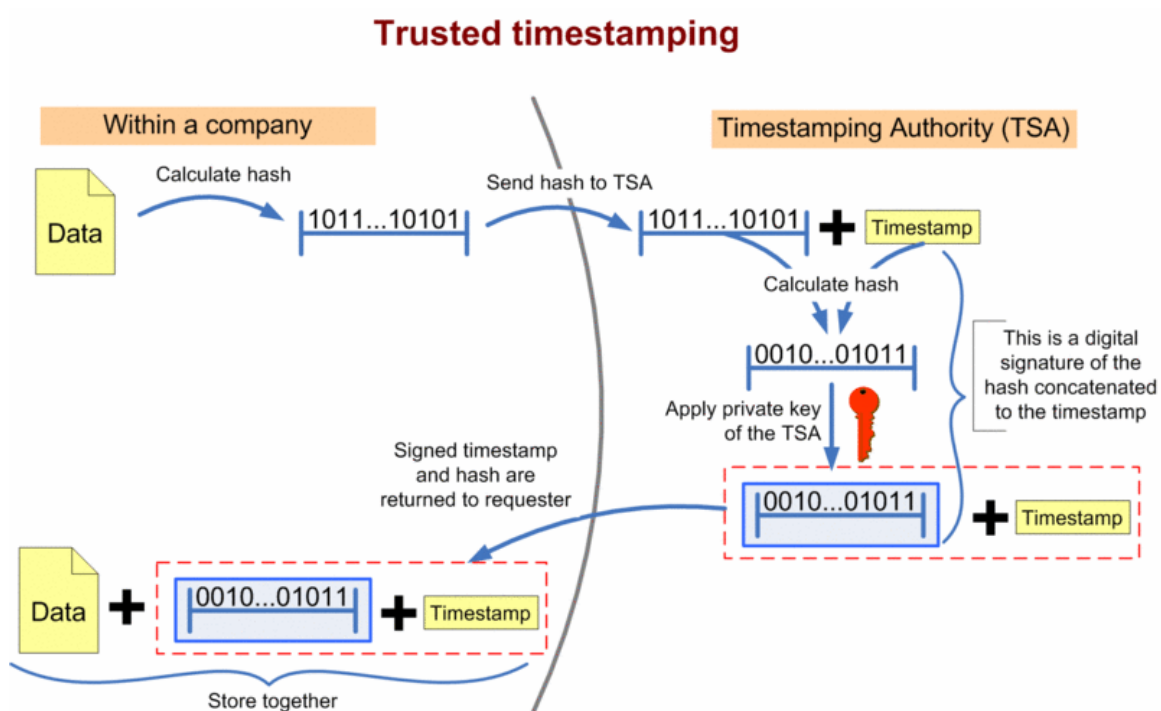
Jednym z nich jest kwestia konserwacji podpisu elektronicznego. Jeśli dziś podpiszemy dokument używając funkcji skrótu SHA-1, to czy będzie on równie bezpieczny za 20 lat?

Odpowiedź brzmi — nie. Niezaprzeczalność autorstwa i autentyczność podpisanego dokumentu opierają się na specyficznej cesze funkcji skrótu (znanej pod angielską nazwą *preimage resistance*), która powoduje, że praktycznie niemożliwe jest wygenerowanie dokumentu dającego określony skrót, którym dysponujemy. Innymi słowy, mając dokument z dołączonym podpisem nie możemy dzisiaj wygenerować dokumentu o innej treści, do którego ten podpis będzie pasować. Jednak za 20 lat obecne funkcje kryptograficzne mogą stracić swoją moc. Technika idzie do przodu, ludzie produkują coraz szybsze maszyny i wymyślają nowe algorytmy, zatem wówczas złamanie np. podpisu, do którego użyto funkcji SHA-1 może być trywialne. Możemy więc wyobrazić sobie sytuację, w której wyciągamy z archiwum dokument podpisany przez kogoś podpisem kwalifikowanym, oddzielamy od niego podpis, a następnie do tego podpisu dorabiamy fałszywy dokument o niekorzystnej dla tej osoby treści.

W przypadku dokumentu papierowego możemy łatwo wykryć takie fałszerstwo analizując papier, na którym spisano umowę, pieczętki, tusz z długopisów i wreszcie charakter pisma. Tymczasem udowodnienie wspomnianego fałszerstwa dokumentu elektronicznego może być bardzo trudne. Konieczna staje się zatem konserwacja dokumentów podpisanych elektronicznie, która polega na cyklicznym, co kilka lat, znakowaniu czasem przy użyciu coraz to nowszych algorytmów kryptograficznych. Według polskiego prawa znakowanie musi być wykonywane co 10 lat, aby dokument elektroniczny można było uznawać dalej za wiarygodny.

Znakowanie czasem zapewnia, że dokument elektroniczny istniał w danym momencie w danej formie, zaś łańcuszek wzajemnie pokrywających się podpisów-znaczników czasowych pozwala zachować bezpieczeństwo dokumentu. Sposób generowania znacznika czasu jest przedstawiony na rys. 2.1. Polega ono na utworzeniu skrótu z dokumentu (np. za pomocą funkcji SHA-1) i wysłaniu go do zaufanego centrum wydające certyfikowane znaczniki. Tam do tego skrótu zostaje dołączony znacznik czasu (czyli bieżący czas) i z całości ponownie liczony jest skrót, który jest podpisywany za pomocą klucza prywatnego tego centrum. Na koniec ten podpis razem ze znacznikiem czasu (użyтым do wyliczenia tego podpisanego skrótu) jest odsyłany z powrotem i od tej pory razem z dokumentem będzie przechowywany w archiwum odebrany od centrum znacznik czasu wraz z podpisem.

W ten sposób za 20 lat będzie możliwe uznanie dokumentu za autentyczny, jeśli od momentu jego podpisania będzie można zweryfikować szereg kolejnych znaczników czasowych złożonych coraz to doskonalszymi funkcjami skrótu — a na końcu właściwy podpis elektroniczny, złożony przed 20-tu laty.



Rysunek 2.1: Znakowanie czasem — źródło: en.wikipedia.org

### Dostępność opisu formatu danych

Innym problemem może być dostępność opisu formatu danych. Przykładowo już dzisiaj problemem może być odczytanie dokumentów w formacie popularnych w latach 90-tych edytorów tekstu QR Tekst czy TAG oraz rozpakowanie archiwów stworzonych co bardziej egzotycznymi archiwizatorami ze środowiska DOS. Tak samo za kilka lat część obecnych edytorów może być już nieużywana i nawet jeśli pobierzemy wówczas z archiwum pewien dokument (i zostanie on poprawnie zweryfikowany), to możemy mieć problem z odczytaniem treści tego dokumentu, gdyż może być on zapisany w nieużywanym już formacie.

### 2.3.3. Formaty podpisu elektronicznego

Istnieją różne formaty podpisów X.509. Najbardziej znane i używane w Polsce to:

- *PKCS#7* — zdefiniowany w RFC 2315 [RFC2315].
- *CMS* (Cryptographic Message Syntax) — powstał na bazie *PKCS#7* i trochę rozszerza jego funkcjonalność, zdefiniowany w RFC 3852 [RFC3852].
- *XaDES* (XML Advanced Electronic Signatures) [XaDES] — jest formatem dedykowanym dla podpisu elektronicznego w dokumentach XML. Rozszerza format *XML Signature* np. o funkcje dla podpisu kwalifikowanego, na co *XML Signature* nie pozwalał. *XaDES* definiuje 6 różnych form podpisów w zależności od poziomu bezpieczeństwa, którego oczekujemy od wygenerowanego podpisu.

W Polsce do zapisywania podpisu kwalifikowanego są dopuszczone wszystkie 3 podane formaty. Niestety są one wzajemnie niezgodne. Każdy z nich ma inną strukturę i jest inaczej kodowany (*CMS* i *PKCS#7* są plikami binarnymi, a *XaDES* jako XML jest plikiem tekstowym). Formaty *XaDES* i *CMS*

są mniej więcej równoważne treścią, *PKCS#7* jest uboższy (na przykład nie przechowuje informacji o znacznikach czasowych).

Można stąd wywnioskować, że istnienie różnych formatów może komplikować używanie podpisów elektronicznych. Sytuację dodatkowo utrudnia to, że każde z polskich centrów certyfikacji zdecydowało się wykorzystywać inny format! Certum (Unizeto) udostępnia aplikację do podpisu, która tworzy podpis w formacie *CMS*. KIR (Krajowa Izba Rozliczeniowa) korzysta z *PKCS#7*, a Sigillum używa jeszcze innego formatu *SDOC*, który sami zaprojektowali. Co prawda korzysta on z *PKCS#7*, ale i tak jest niezgodny z wszystkimi innymi formatami. Natomiast Signet (zakończył działalność w czerwcu zeszłego roku) korzystał z formatu *XaDES*. Na szczęście obecnie sytuacja się poprawia, bo centra certyfikacją wspólnie dążą w kierunku ujednoczenia do formatu *XaDES*, który staje się coraz bardziej popularny.

Normalną rzeczą może być jednak sytuacja, w której ktoś prześle nam dokument elektroniczny z podpisem w jednym formacie, a my nie będziemy w stanie tego dokumentu zweryfikować i uznamy go za błędny, gdyż nasza aplikacja będzie tylko potrafiła weryfikować podpisy w innym formacie i nie rozpozna poprawnie podpisu pod dokumentem. Zatem jeśli byśmy chcieli móc weryfikować podpisy od dowolnej osoby, niezależnie w jakim formacie wygenerowała podpis, to byśmy musieli posiadać aplikację, która sprawdza wszystkie możliwe formaty, co dodatkowo komplikuje np. tworzenie bezpiecznych archiwów dokumentów, które powinny być uniwersalne, niezależne od formatu podpisanego dokumentu.

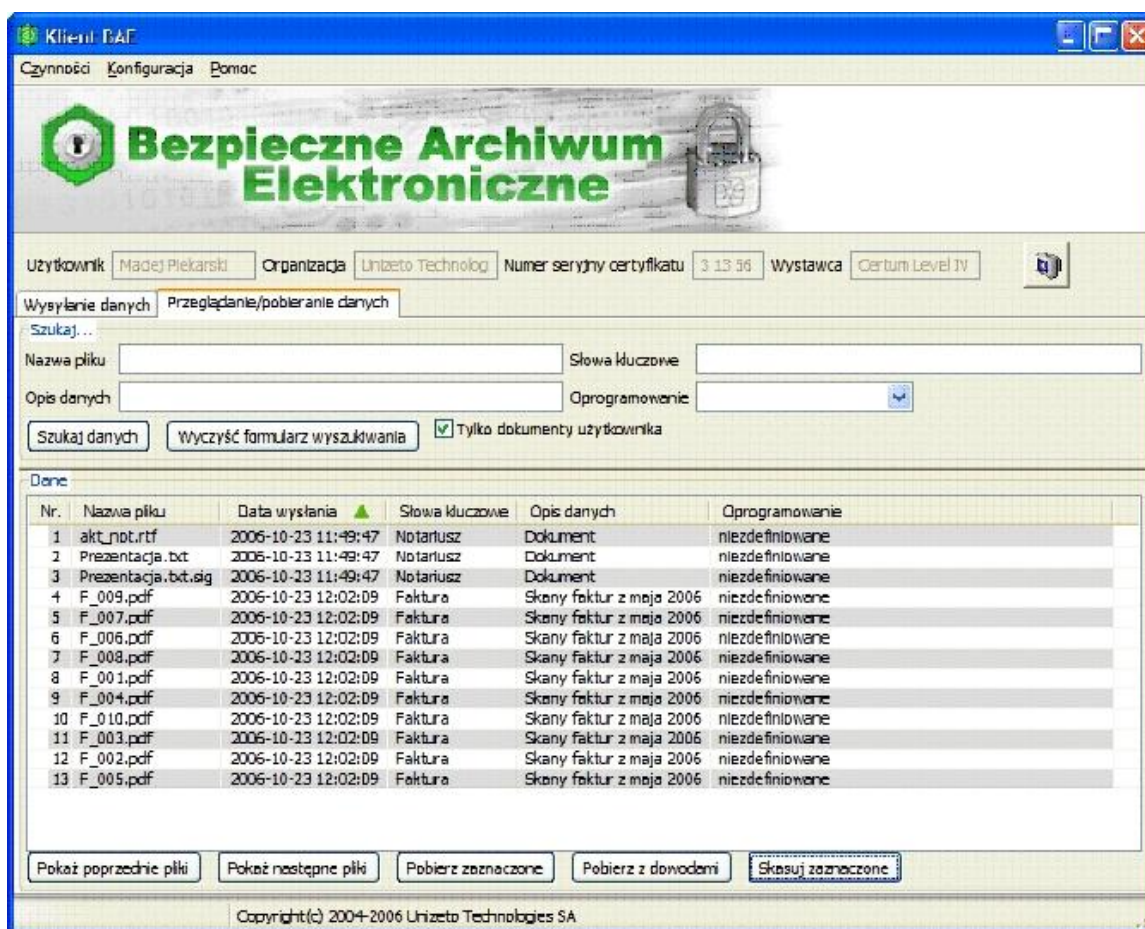
## 2.4. Bezpieczne Archiwum Elektroniczne

Bezpieczne Archiwum Elektroniczne (BAE) jest systemem oferowanym przez Unizeto Technologies SA [Unizeto], który umożliwia archiwizację dokumentów elektronicznych. System przeznaczony jest do przechowywania dowolnych dokumentów, a w szczególności dokumentów podpisanych elektronicznie. Umożliwia konserwację podpisu elektronicznego, prowadzącą w efekcie do utrzymania wartości dowodowej przechowywanych dokumentów, opatrzonych podpisem elektronicznym.

Bezpieczne Archiwum Elektroniczne zapewnia:

- nadawanie cech niezaprzeczalności przechowywanym danym,
- integralność archiwizowanych danych w dowolnie długim okresie czasu,
- dostępność przechowywanych danych 24 godziny dziennie przez 7 dni w tygodniu,
- gromadzenie danych dowodowych w odniesieniu do archiwizowanych dokumentów, archiwizowanie, oprócz podpisanych elektronicznie dokumentów, także certyfikatów, list odwołań CRL, znaczników czasu,
- stosowanie mechanizmów kryptograficznych, zapewniających bezpieczeństwo archiwizowanych danych.

Zasadniczą funkcjonalnością Bezpiecznego Archiwum Elektronicznego jest konserwacja wartości dowodowej przechowywanych dokumentów. Zgodnie z zapisami ustawy [Ust01], BAE realizuje konserwację wartości dowodowej przez znakowanie czasem. W tym celu wykorzystywane są kwalifikowane znaczniki czasu, wydawane przez CERTUM — Powszechne Centrum Certyfikacji. Ponieważ BAE przyjmuje do archiwizacji dowolne pliki, którymi w szczególności mogą być dokumenty podpisane elektronicznie z wykorzystaniem różnych standardów podpisu i różnych certyfikatów (zwykłych i kwalifikowanych), każdy dokument przyjmowany do archiwizacji jest znakowany czasem.



Rysunek 2.2: Aplikacja kliencka BAE

### 2.4.1. Funkcjonalność BAE

Bezpieczne Archiwum Elektroniczne działa w technologii grubego klienta. Dostęp do archiwum (składanie i pobieranie dokumentów, przeszukiwanie archiwum) realizowany jest za pomocą aplikacji klienckiej, natomiast archiwizacja oraz konserwacja wartości dowodowej przechowywanych dokumentów realizowana jest przez część serwerową systemu, znajdującą się w Unizeto.

W celu korzystania z usługi BAE konieczne jest posługiwanie się aplikacją kliencką. Korzystanie z aplikacji jest możliwe po uwierzytelnieniu się certyfikatem niekwalifikowanym, zarejestrowanym na serwerze BAE, który zlokalizowany jest w Unizeto Technologies SA. Certyfikaty wykorzystywane do uwierzytelniania użytkowników mogą być przechowywane na kartach kryptograficznych lub zapisane lokalnie w pliku na stacji roboczej użytkownika. Stosowanie certyfikatów umieszczonych na kartach zapewnia mobilność umożliwiając uwierzytelnianie użytkownika na dowolnej stacji roboczej, na której zainstalowana jest aplikacja kliencka. Funkcjonalność aplikacji podzielona jest na dwie grupy, co znajduje odzwierciedlenie w postaci zakładek (por. rys. 2.2), z których jedna nazywa się *Wysyłanie danych* i przeznaczona jest do składania dokumentów elektronicznych do archiwum, a druga opisana jest jako *Przeglądanie/pobieranie danych* przeznaczona jest do przeszukiwania BAE i pobierania z niego dokumentów.

Dokumenty przesyłane od użytkownika do archiwum za pomocą aplikacji klienckiej są przed wysyłką opisywane przez aplikację metadanymi wprowadzonymi przez użytkownika i wraz z nimi przesyłane do archiwum. Metadane nie są opatrywane podpisem elektronicznym ani nie są częścią

archiwizowanego dokumentu. Są natomiast powiązane z danym dokumentem, ponieważ wykorzystywane są do jego identyfikacji i wyszukiwania w archiwum. Metadane są wprowadzane za pomocą aplikacji klienckiej i dowiązywane przez tę aplikację do dokumentów w momencie ich wysyłania do BAE. Ilość i rodzaj metadanych jest definiowany przed uruchomieniem usługi i później nie może już ulec zmianie. Wszystkie dokumenty muszą być opisywane tymi samymi metadanymi — nie ma możliwości kategoryzacji dokumentów na różne rodzaje.

#### **2.4.2. Przesyłanie i przechowywanie dokumentów**

Po uwierzytelnieniu się w systemie, zestawiane jest bezpieczne połączenie pomiędzy aplikacją kliencką a częścią serwerową BAE. Od tego momentu cała komunikacja pomiędzy aplikacją a serwerami systemu odbywa się z wykorzystaniem tego połączenia. Dotyczy to w szczególności składania i pobierania dokumentów oraz towarzyszących im metadanych.

Ponadto dokumenty przesyłane do archiwum są przed wysyłką szyfrowane przez aplikację kliencką za pomocą klucza publicznego archiwum i przesyłane w postaci zaszyfrowanej do części serwerowej. Zatem w momencie przesyłania dokumenty są zabezpieczone w dwojaki sposób — poprzez zaszyfrowanie kluczem prywatnym archiwum oraz przez szyfrowane połączenie pomiędzy aplikacją kliencką a serwerem BAE. Dokumenty złożone w BAE przez cały okres ich archiwizacji są składowane w postaci zaszyfrowanej. Dołączone do nich metadane nie są szyfrowane, lecz przechowywane w postaci jawnej, dzięki czemu możliwe jest przeszukiwanie zawartości archiwum i wyszukiwanie dokumentów odpowiadających tym metadanym bez konieczności deszyfrowania samych dokumentów.

W momencie pobierania dokumentu jest on deszyfrowany kluczem prywatnym archiwum i szyfrowany kluczem publicznym użytkownika, który pobiera dokument. W tej postaci poprzez zaszyfrowane połączenie jest on przesyłany do aplikacji klienckiej użytkownika, która po otrzymaniu zaszyfrowanego dokumentu automatycznie deszyfruje go z użyciem klucza prywatnego użytkownika i umożliwia zapisanie na dysku w postaci jawnej.

### **2.5. Uniwex**

Uniwex jest systemem należącym do włoskiej firmy Unimatica, który wspomaga wykonywanie różnych czynności związanych z egzaminami i protokołami egzaminacyjnymi, czyli podobnie jak system USOSweb, ale ma węższą funkcjonalność. Natomiast bardziej dba o bezpieczeństwo przechowywanych danych, szczególnie poprzez powszechne wykorzystywanie podpisów elektronicznych. Uniwex pozwala m.in. na rejestrację na przedmioty oraz obsługę egzaminów, czyli wyznaczanie terminów egzaminów przez wykładowców i zapisywanie się przez studentów na te egzaminy. Ponadto umożliwia wpisywanie i zapamiętywanie wyników egzaminów, które później są udostępniane uprawnionym osobom.

Obecnie system ten jest wykorzystywany na 3 uczelniach włoskich, w szczególności z powodzeniem jest stosowany na wszystkich wydziałach Uniwersytetu Bolońskiego (rys 2.3) i pozwolił na znaczną eliminację obiegu dokumentów papierowych. Ponadto przyśpieszył wykonywanie wielu czynności na uczelni oraz przyniósł spore korzyści finansowe. Szacuje się, że od 2005 roku pozwolił zaoszczędzić Uniwersytetowi Bolońskiemu 900 tysięcy euro rocznie.

#### **2.5.1. Funkcjonalność systemu**

Większość wykonywanych operacji wymaga zalogowania się przy pomocy ważnego certyfikatu wydanego przez kwalifikowane centrum certyfikacji, który jest wcześniej rejestrowany w systemie.



Risunek 2.3: System Uniwex

Użytkownik loguje się poprzez podanie kodu PIN do karty kryptograficznej, na której jest przechowywany klucz prywatny związany z certyfikatem dla niego zarejestrowanym.

W zależności od pełnionej funkcji po zalogowaniu użytkownik ma prawo wykonywać różne operacje. Nauczyciel akademicki przedmiotów może m.in. definiować szczegóły przeprowadzania egzaminów na kursach, które prowadzi oraz wprowadzać wyniki po skończonym egzaminie dla poszczególnych studentów. Dla każdego egzaminu wykładowca może podać miejsce i terminy, w których będzie się odbywał, sposób jego przeprowadzenia (np. liczbę pytań na teście) itp. Możliwe jest także zweryfikowanie czy nie koliduje on z egzaminami dla innych przedmiotów. Studenci po zalogowaniu mogą zapisywać się na egzaminy z zajęć, na które uczęszczają oraz później sprawdzać otrzymane na nich oceny. Wprowadzone sprawozdanie z egzaminu dla danego studenta jest podpisywane przez wykładowcę, które — aby było uznane za wiarygodne — musi później być jeszcze podpisane co najmniej przez jednego z pozostałych członków komisji egzaminacyjnej. W każdej chwili każdy nauczyciel może przejrzeć listę sprawozdań, które nie są podpisane przez 2 osoby i pod którymi ma prawo złożyć własny podpis. Dane na temat egzaminów mogą być eksportowane lub importowane z plików XML i plików w formacie Excel.

Ponadto wykładowcy przed rozpoczęciem danego przedmiotu, który mają prowadzić, opisują go wskazując daty i tematy poszczególnych wykładów. Te informacje są później przez nich podpisywane i wysyłane do dziekana do weryfikacji, a na koniec system generuje wykaz wszystkich zajęć dla poszczególnych wykładowców na dany rok akademicki.

Uniwex pozwala także na zarządzanie egzaminami dyplomowymi i sporządzanie dokumentacji poegzaminacyjnej z podpisami osób egzaminujących i przechowuje je do wglądu dla uprawnionych osób.

## 2.5.2. Architektura systemu

Uniwex jest udostępniany w modelu ASP (Application Service Provider), który polega na wynajmowaniu programu komputerowego poprzez internet. System nie jest każdorazowo instalowany poszczególnym u poszczególnych klientów, ale działa na serwerze dostawcy, często bardzo oddalonym od siedziby klienta. Inny jest także sposób rozliczania się pomiędzy dostawcą programu a jego użytkownikiem. W modelu ASP rezygnuje się z wnoszenia opłaty za licencję za pewien okres z góry, na rzecz wnoszenia opłaty proporcjonalnej do stopnia wykorzystania. Ze względu na charakter tych rozliczeń czasami ASP określa się terminem *Oprogramowania na żądanie* (ang. *On-demand software*).

Uniwex jest aplikacją napisaną w technologii JEE (która uruchomiona jest na serwerze Apache Tomcat [Tomcat]), czyli w takiej samej jak *Archiwum*. Dodatkowo wykorzystana została biblioteka Struts [Struts], która udostępnia podobną funkcjonalność jak biblioteka Webwork [WW], której użyłem w mojej implementacji (por. p. 5.1).

System Uniwex zapewnia wysoki poziom bezpieczeństwa poprzez:

- utrzymywanie bezpiecznego centrum przechowującego dane, które posiada różne mechanizmy chroniące dostęp przed niepowołanymi osobami, takie jak stała obserwacja budynku czy elektroniczne karty wstępu zabezpieczone kodem,
- okresowe wykonywanie kopii zapasowej dla trzymany danych, aby np. w przypadku pożaru lub innej sytuacji losowej utracone dane mogły możliwie szybko zostać przywrócone,
- zapory ogniowe (ang. *firewalls*) — chroniące przed nieautoryzowanym dostępem poprzez sieć.

W celu osiągnięcia wysokiej dostępności i wydajności systemu Uniwex stosuje:

- równoważenie obciążenia systemu poprzez efektywne rozdzielanie zadań na poszczególne serwery w sieci tak, aby pojedynczy serwer nie był nigdy przeciążony dużo mocniej od pozostałych,
- dynamiczną naprawę awarii — w przypadku, gdy pewien komponent sprzętowy ulegnie uszkodzeniu, system automatycznie podłącza się do alternatywnego dysku, serwera czy fragmentu sieci.

## 2.6. Wnioski

Uniwex poprzez wykorzystywanie podpisów elektronicznych oraz stosowanie wielu innych mechanizmów zabezpieczeń przechowuje dane w bardzo bezpieczny sposób, jednakże zbiór dokumentów, które w ten sposób pozwala przechowywać jest mocno ograniczony. System ten ma spełniać trochę inne zadania niż *Archiwum* i oprócz archiwizowania danych, także pozwala zarządzać nimi w różnych procesach, w związku z czym jest on bardziej rozbudowany od *Archiwum*. Istotne jednak jest to, że jest mało elastyczny, gdyż trudno jest go rozszerzyć o możliwość przechowywania nowych dokumentów. Ponadto często się zdarza, że na różnych uczelniach wykorzystywane są trochę inne dokumenty albo część procesów przebiega w inny sposób, zatem dostosowanie systemu Uniwex na potrzeby innej uczelni może wymagać dokonania zmian w kodzie programu. Natomiast głównym celem *Archiwum* jest umożliwienie dynamicznego rozszerzania dopuszczalnego zbioru dokumentów, co pozwala prościej dostosować go na potrzeby innego środowiska.



## Rozdział 3

# Analiza wymagań

### 3.1. Wymagania funkcjonalne

#### 3.1.1. Wymiana danych z USOS

Źródłem potrzebnych danych o jednostkach organizacyjnych, przedmiotach, kierunkach studiów itp. ma być system USOS, zatem do *Archiwum* należy dołączyć niezbędną konfigurację Migratora, za pomocą którego dane będą synchronizowane. Migracja będzie odbywała się tylko w jedną stronę: z systemu USOS do *Archiwum*, gdyż dane z systemu USOS po stronie *Archiwum* nie będą modyfikowane, a umieszczane dokumenty nie będą przechowywane w bazie USOS. Powinno być możliwe zablokowanie systemu na czas migracji danych.

#### 3.1.2. Logowanie do systemu

Logowanie do systemu będzie odbywało się poprzez CAS (Central Authentication Service [CAS]). Zalogować się będą mogły tylko te osoby, które mają aktywne konto w systemie USOS (migrowane będą tylko te konta, dla których atrybut *INST\_WWW\_KOD* ma wartość *CUS\_WWW*). Zatem logowanie będzie odbywało się identycznie jak w istniejącym już Archiwum Prac Dyplomowych [APD].

#### 3.1.3. Umieszczanie i przechowywanie dokumentów

Każdy użytkownik będzie mógł umieszczać w systemie dokumenty w ramach swoich uprawnień. Razem z dokumentem będą trzymane metadane (zbiór atrybutów) go opisujące, które będzie uzupełniał użytkownik przed wstawieniem dokumentu. Przechowywane dokumenty będą podzielone na różne typy — każdy typ będzie mógł posiadać inne metadane lub inaczej się zachowywać, prawa do wykonania danej operacji będą zależne od typu dokumentu. Typ dokumentu będzie opisywany przez jeden plik XML. Umieszczane dokumenty będą mogły być podpisane podpisem w standardzie PGP lub X.509 (zarówno podpisem zwykłym, jak i kwalifikowanym), ponadto będzie też możliwość dodania dokumentu nie podpisanego żadnym podpisem. W obrębie danego typu dokumentów będzie określone w jakiej postaci dokumenty tego typu powinny być wkładane do *Archiwum*. Jeden dokument w systemie będzie mógł zawierać więcej niż jeden fizyczny plik.

#### 3.1.4. Elastyczność systemu

*Archiwum* powinno być na tyle ogólne i elastyczne, aby umożliwiała przechowywanie dowolnych dokumentów i opisywanie ich dowolnymi atrybutami (gdzie każdy atrybut może być dowolnego typu), a w dodatku być łatwo rozszerzalne o nowe funkcjonalności. W każdym archiwum najważniejsze są dokumenty i ich właściwa kategoryzacja według poszczególnych kryteriów, dlatego też bardzo

dużą zaletą takiego archiwum będzie możliwość rozszerzenia zestawu dopuszczalnych dokumentów bez potrzeby ingerowania w kod aplikacji, nawet bez potrzeby restartowania jej. Zatem powinna być możliwość zarejestrowania w systemie poprzez interfejs administratora nowego typu dokumentu, co jednocześnie umożliwi uprawnionym użytkownikom dodawanie dokumentów tego nowego typu.

### **3.1.5. Wyszukiwanie i pobieranie dokumentów**

Każdy użytkownik będzie miał możliwość wyszukiwania dokumentów wg różnych kryteriów. Ponadto w *Archiwum* będzie tworzona wirtualna struktura katalogów (czyli w rzeczywistości na dysku te katalogi nie będą istniały), w których będą umieszczane dokumenty. Użytkownik będzie miał możliwość poruszania się po takiej strukturze i przeglądania uporządkowanych w ten sposób dokumentów (katalogami będą np. nazwy typów dokumentów, nazwy jednostek organizacyjnych i lata akademickie, a umiejscowienie danego dokumentu będzie zależało od wartości jego metadanych). W wynikach wyszukiwania i katalogach będą pojawiały się tylko te dane, do których użytkownik posiada prawo odczytu. Dla każdego wyszukanego dokumentu użytkownik będzie mógł obejrzeć opisujące go metadane, pobrać pliki z nim związane (w przypadku posiadania prawa do pobrania ich) oraz edytować te metadane i pliki (w przypadku posiadania odpowiednich praw do edycji).

### **3.1.6. System uprawnień**

Elastyczny system ról i uprawnień powinien umożliwiać w prosty sposób przypisanie lub zmianę różnych uprawnień poszczególnym osobom. Główne znaczenie będą miały uprawnienia definiowane przez konkretne typy dokumentów, jednakże część uprawnień będzie definiowana globalnie dla systemu poprzez przypisanie poszczególnych praw poszczególnym rolom. Częściowo przypisanie osób do ról będzie wynikało bezpośrednio z danych zmigrowanych z systemu USOS o studentach czy pracownikach uczelni. Natomiast przynależność do pewnych ról będzie określana po stronie *Archiwum* niezależnie od systemu USOS. Będzie możliwość prostego dodawania nowych ról z interfejsu administratora.

### **3.1.7. Podpisy elektroniczne i certyfikaty**

Uprawnione osoby powinny mieć możliwość rejestrowania dla poszczególnych użytkowników nowych certyfikatów X.509 i kluczy publicznych PGP. Każdy użytkownik będzie mógł posiadać więcej niż jeden certyfikat danego typu, a dokumenty wstawiane przez niego, które będą podpisane elektronicznie będą weryfikowane tylko za pomocą zarejestrowanych na niego certyfikatów. Dokument błędnie zweryfikowany nie będzie mógł zostać umieszczony w *Archiwum*.

## **3.2. Wymagania niefunkcjonalne**

### **3.2.1. Pomocne informacje o stronach portalu**

*Archiwum* powinno udostępniać wskazówki dotyczące obsługi systemu. Pomoc powinna być dostępna z większości stron i opisywać ich przeznaczenie oraz przypadki użycia.

### **3.2.2. Obsługa dwóch języków**

Ze względu na znaczną liczbę osób z zagranicy studiujących na polskich uczelniach, którym jest wygodniej posługiwać się językiem angielskim niż polskim, *Archiwum* powinno być dostępne w obydwu wymienionych wersjach językowych, szczególnie, że istniejące już aplikacje na UW, takie jak USOSweb i Archiwum Prac Dyplomowych, dostępne są w tych dwóch językach.

### **3.2.3. Interfejs użytkownika**

Obsługa systemu powinna być intuicyjna oraz na tyle łatwa, aby system pomocy serwisu był wykorzystywany stosunkowo rzadko. Wygląd stron *Archiwum* ma być podobny do wyglądu systemu USOSweb.



## Rozdział 4

# Projekt systemu

*Archiwum* zostało zaprojektowane jako aplikacja internetowa. Wybór ten wydaje się być jedynym słusznym ze względu na liczbę potencjalnych użytkowników oraz powszechność takiego rozwiązania. Daje to możliwość korzystania z gotowych rozwiązań, jakimi są istniejące protokoły przesyłania danych oraz technologie WWW. Dzięki takiemu rozwiązaniu nie ma potrzeby instalowania dodatkowego oprogramowania po stronie użytkownika.

### 4.1. Osobna aplikacja czy integracja z jakąś istniejącą?

Na Uniwersytecie Warszawskim jest kilka aplikacji korzystających z systemu USOS, takich jak USOSweb czy Archiwum Prac Dyplomowych. Dostyc istotną decyzją było rozstrzygnięcie, czy *Archiwum* powinno być zintegrowane, z którymś z tych serwisów, czy działać jako osobna i niezależna aplikacja. Lepszym rozwiązaniem wydaje się stworzenie osobnego serwisu, gdyż *Archiwum* co prawda ma wykorzystywać dane z systemu USOS, ale tylko te najważniejsze, więc migracja danych nie będzie aż tak czasochłonna. Nie ma konieczności przechowywania wstawianych dokumentów w systemie USOS, a dane z systemu USOS nie będą modyfikowane po stronie *Archiwum*, więc migracja będzie wykonywana tylko w jedną stronę. Zważywszy zatem na to, że dodatkowa funkcjonalność mogłaby mocno obciążać już którąś z istniejących aplikacji (w przypadku gdyby *Archiwum* zostało z nią zintegrowane) oraz w związku z tym, że jest to projekt testowy, zdecydowałem się na stworzenie osobnej aplikacji.

### 4.2. Użytkownicy systemu

#### 4.2.1. Zwykły użytkownik

Użytkownikami systemu będą wszystkie osoby, które mają aktywne konta w systemie USOS. Każdy z nich będzie miał prawo się zalogować oraz umieszczać, przeglądać, modyfikować i pobierać dokumenty według swoich uprawnień, wynikających z pełnienia odpowiednich ról i zarejestrowanych typów dokumentów. Rola jest przydzielana konkretnej osobie w ramach danej jednostki organizacyjnej, zatem jeśli np. ktoś jest studentem na dwóch wydziałach, to będzie występował w dwóch rolach. Podstawowe role w systemie to: *student*, *pracownik*, *pracownik dydaktyczny* (nauczyciel akademicki), *pracownik dziekanatu*, *pracownik kwestury*, *pracownik BSS*, *dziekan* oraz *administrator*.

### 4.2.2. Administrator jednostki organizacyjnej

Dodatkowe uprawnienia będą posiadali użytkownicy, którzy będą przydzieleni w ramach danej jednostki do roli *administrator*. W ramach takiej jednostki będą mogli umieszczać, przeglądać, modyfikować i pobierać dowolne dokumenty bez żadnych ograniczeń. Ponadto taki użytkownik będzie miał prawo do korzystania także z innych opisanych w tym punkcie funkcjonalności.

#### Zarządzanie certyfikatami

Administrator będzie odpowiedzialny za rejestrowanie kluczy publicznych PGP oraz certyfikatów X.509 dla poszczególnych użytkowników. Będzie odbywało się to poprzez załadowanie odpowiedniego klucza czy pliku z certyfikatem z panelu administratora. Dla danej osoby będzie można zarejestrować dowolną liczbę kluczy publicznych PGP i certyfikatów X.509.

#### Przydzielanie użytkowników do ról

Z panelu administratora będzie możliwe przydzielanie poszczególnych osób do ról, przy czym tylko w obrębie tych jednostek, w których posiada się uprawnienia administratora. Ponadto użytkowników można przypisywać tylko do tych ról, które nie są migrowane z bazy USOS, jak np. *pracownik dziekanatu* czy *pracownik kwestury*. Modyfikacja przynależności do takich ról jak *student* i *pracownik* nie będzie możliwa.

#### Zarządzanie typami dokumentów

Z panelu administratora będzie można także zarejestrować nowy typ dokumentu poprzez załadowanie pliku XML, w którym jest on opisany. Ta operacja nie będzie wymagała wykonywania żadnych dodatkowych zmian w bazie danych — odpowiednia tabela przechowująca dokumenty tego typu zostanie automatycznie utworzona w bazie danych. Możliwa będzie także modyfikacja istniejącego typu dokumentu poprzez załadowanie zaktualizowanego pliku XML. Przy tej operacji, jeśli zaktualizowany typ dokumentu wymaga zmian w strukturze tabeli w bazie, przechowującej dokumenty tego typu, konieczne będzie ręczne wykonanie w bazie skryptu dokonującego odpowiednich zmian.

### 4.2.3. Administrator systemu

Istnieje także specjalny użytkownik — główny administrator systemu. Posiada on wszystkie te prawa, które ma użytkownik z przypisaną rolą *administrator*, przy czym może je wykonywać w obrębie wszystkich jednostek organizacyjnych. Dodatkowo administrator systemu może poprzez interfejs z panelu administratora zmieniać globalne uprawnienia przypisane poszczególnym rolom oraz tworzyć nowe role w systemie. Ponadto może zablokować system (czyli uniemożliwić użytkownikom wykonania żadnych funkcji) np. na czas migracji danych.

## 4.3. Dokumenty

Najważniejszymi obiektami w systemie będą Dokument i Typ dokumentu. Dokument to zbiór plików wraz z metadanymi go opisującymi oraz informacje o typie tego dokumentu, dacie dodania go do *Archiwum*, osobie. To typ dokumentu będzie określał właściwości danych dokumentów i po części definiował funkcjonalność *Archiwum*. Jeden typ dokumentu będzie opisywany poprzez jeden plik w formacie XML. W takim pliku zdefiniowane są:

- nazwa typu dokumentu,

- kategoria dokumentów, do której ten typ należy,
- atrybuty dokumentów specyficzne dla tego typu,
- uprawnienia do tworzenia dokumentów tego typu,
- osoby, którym ten dokument jest dedykowany, tzn. osoby, z którymi dokument jest w pewien sposób związany (np. dla typu *Praca dyplomowa* takimi osobami będą autorzy pracy oraz opiekunowie pracy),
- uprawnienia do odczytu i edycji dokumentów tego typu,
- typy podpisów, którymi mogą być podpisane pliki wchodzące w skład dokumentów tego typu (nie zawsze będzie potrzebna, aby umieszczane dokumenty były podpisane podpisem kwalifikowany, ale również złym rozwiązaniem mogłoby być pozwolenie na umieszczenie wszystkich dokumentów w dowolnej postaci — umożliwienie określenia dopuszczalnych podpisów dla danego typu dokumentu rozwiązuje ten problem),
- sposób generowania opisu dokumentu — zamiast ręcznego określania opisu dokumentu, możliwe będzie automatycznie wygenerowanie takiego opisu na podstawie wartości atrybutów tego dokumentu według wcześniej ustalonego wzoru.

#### 4.3.1. Format pliku XML opisującego typ dokumentu

Elementem głównym tego pliku XML jest `<typ>`. Wewnątrz niego powinny znajdować się kolejno następujące elementy:

- `<nazwa_pl>` — zawartością tego elementu jest tekst oznaczający nazwę typu dokumentu w języku polskim,
- `<nazwa_en>` — zawartością jest tekst oznaczający nazwę typu dokumentu w języku angielskim,
- `<nazwa_kod>` — zawartością jest tekst określający kod (nazwę kodową) typu dokumentu, który będzie używany w *Archiwum* do identyfikacji typu dokumentu — musi być unikatowy w obrębie kodów wszystkich typów,
- `<kategoria>` — zawiera tekst oznaczający kod kategorii dokumentów,
- `<atrybuty>` — zawiera listę elementów `<atrybut>` opisujących poszczególne atrybuty; każdy `<atrybut>` powinien zawierać następujące elementy:
  - `<nazwa_kod>` — kod danego atrybutu, który będzie używany w *Archiwum* do identyfikacji tego atrybutu, musi być unikatowy w obrębie wszystkich atrybutów jednego typu,
  - `<nazwa_pl>` — nazwa atrybutu w języku polskim,
  - `<nazwa_en>` — nazwa atrybutu w języku angielskim,
  - `<typ>` — określa typ atrybutu, musi posiadać atrybuty *rodzaj* (*prosty/złożony*) i *klasa*
    - \* dla rodzaju *prosty* wartością *klasa* może być: *Integer*, *Long*, *String*, *Enumeration*, *Boolean*, *Date*,
    - \* dla rodzaju *złożony* wartością *klasa* jest pełna nazwa klasy z *Archiwum* (np. *pl.archiwum.core.base.Person* oznaczająca osobę lub *pl.archiwum.core.university.StudyProgram* oznaczająca program studiów),

Typ *Enumeration* oznacza wybór ze zbioru jednej wartości wyliczeniowej.

Pozostałe nieobowiązkowe atrybuty elementu `<typ>` to:

- \* *wielokrotny* — gdy wartością jest *tak*, możliwe będzie nadanie kilku wartości dla tego atrybutu (np. dla typu dokumentu *Praca dyplomowa* możliwe jest wybranie więcej niż jednego autora pracy lub więcej niż jednego opiekuna pracy),
- \* *wymagany* — gdy wartością jest *tak*, ten atrybut będzie obowiązkowy, czyli niemożliwe będzie utworzenie dokumentu bez podania wartości tego atrybutu,
- \* *dlugosc* — maksymalna długość wartości atrybutu (obowiązkowe dla pola *String*, dla innych nie ma znaczenia).

Zawartość elementu `<typ>` może być pusta albo może zawierać element `<zakres>` określający szczegółowo dostępne wartości dla tego atrybutu dokumentu. Obecnie `<zakres>` można zdefiniować tylko dla *Enumeration* oraz dla typu złożonego *pl.archiwum.core.base.Person*, przy czym tylko dla *Enumeration* jego zdefiniowanie jest obowiązkowe.

Dla *Enumeration* wewnątrz tego elementu powinna znajdować się lista elementów `<wartosc>`, które określają poszczególne wartości wyliczeniowe dostępne w tym atrybucie. Element `<wartosc>` powinien mieć pustą zawartość i zawierać następujące atrybuty:

- \* *id* — unikatowa liczba w obrębie wszystkich wartości wyliczeniowych tego atrybutu,
- \* *nazwa\_kod* — kod wartości wyliczeniowej unikatowy w obrębie wszystkich wartości wyliczeniowych tego atrybutu,
- \* *nazwa\_pl* — nazwa w języku polskim,
- \* *nazwa\_en* — nazwa w języku angielskim.

Dla *pl.archiwum.core.base.Person* wewnątrz elementu `<zakres>` powinien znajdować się element `<typ>`, którego zawartością może być *student* albo *pracownik* albo *pracownik\_dydaktyczny*, powodujący ograniczenie zbioru wartości tego atrybutu tylko do studentów, pracowników lub pracowników dydaktycznych.

Przykładowa definicja atrybutu wygląda następująco:

```
<atrybut>
  <nazwa_kod>koordynatorzy</nazwa_kod>
  <nazwa_pl>Koordynatorzy</nazwa_pl>
  <nazwa_en>Instructors</nazwa_en>
  <typ rodzaj="zlozony" klasa="pl.archiwum.core.base.Person"
  wielokrotny="tak" wymagany="tak">
    <zakres>
      <typ>pracownik_dydaktyczny</typ>
    </zakres>
  </typ>
</atrybut>
```

- W kilku kolejnych punktach opisane są elementy określające uprawnienia. Użyte są tam elementy definiujące różne grupy osób posiadających dane uprawnienia do dokumentów. Są to elementy jednego z 4 typów:
  - `<wszyscy/>` — oznacza, że wszyscy użytkownicy systemu będą posiadali dane uprawnienie do dokumentów tego typu we wszystkich jednostkach organizacyjnych,
  - `<rola>${kod_rola}</rola>` — oznacza, że użytkownicy należący do roli o kodzie `${kod_rola}` w jednostce *X* będą posiadali dane uprawnienie do dokumentów w jednostce *X*,



- <atrybut\_ref>\${kod\_atrybutu}</atrybut\_ref> — oznacza, że dane uprawnienie będą posiadały osoby wpisane jako wartość atrybutu o kodzie \${kod\_atrybutu},
  - <tworca/> — oznacza, że uprawnienie do danego dokumentu tego typu posiada osoba, która wstawiła ten dokument.
- <tworcy> — zawiera listę złożoną z elementów <wszyscy> lub <rola> definiujących kolejne grupy osób posiadających prawo do tworzenia dokumentów tego typu,
  - <odbiorcy> — zawiera listę złożoną z elementów <wszyscy>, <rola>, <atrybut\_ref> lub <tworca>, definiującą, komu będą dedykowane dokumenty tego typu.

Przykładowa definicja odbiorców wygląda następująco:

```
<odbiorcy>
  <atrybut_ref>studenci</atrybut_ref>
  <rola>pracownik_dziekanatu</rola>
  <rola>dziekan</rola>
</odbiorcy>
```

- <uprawnienia> — określa sposób przydzielania wszystkich pozostałych uprawnień dotyczących dokumentów, zawiera elementy oznaczające kolejne uprawnienia (każdy z tych elementów zawiera listę złożoną z elementów <wszyscy>, <rola>, <atrybut\_ref> lub <tworca>, definiującą kto będzie posiadał dane uprawnienie). Możliwe uprawnienia to:
  - <czytanie> — prawo do przeglądania/czytania dokumentu,
  - <zalaczniki\_czytanie> — prawo do pobierania plików związanych z dokumentem,
  - <edycja> — prawo do modyfikacji metadanych (atrybutów) dokumentu,
  - <zalaczniki\_edycja> — prawo do dodawania/modyfikacji/usuwania plików związanych z dokumentem.

Przykładowa definicja uprawnień wygląda następująco:

```
<uprawnienia>
  <czytanie>
    <wszyscy/>
  </czytanie>
  <zalaczniki_czytanie>
    <tworca/>
    <atrybut_ref>studenci</atrybut_ref>
    <rola>pracownik_dziekanatu</rola>
  </zalaczniki_czytanie>
  <edycja>
    <tworca/>
    <rola>pracownik_dziekanatu</rola>
  </edycja>
  <zalaczniki_edycja>
    <rola>pracownik_dziekanatu</rola>
  </zalaczniki_edycja>
</uprawnienia>
```

- `<podpis>` — zawiera listę elementów, z których każdy definiuje jeden dozwolony typ podpisu dla plików w dokumentach tego typu. Ma on postać `<dozwolony>${typ}</dozwolony>`, gdzie `${typ}` to *brak*, *PGP* lub *X.509*,
- `<inne>` — definiuje dodatkowe właściwości tego typu dokumentu. Obecnie możliwe jest tylko umieszczenie tu wzoru do generowania opisu dokumentu. Wzór opisu jest zapisywany jako zwykły tekst z umieszczonymi wewnątrz wstawkami oznaczającymi odwołanie do nazwy typu dokumentu (`${DOCTYPE}`) lub wartości atrybutu w tym typie dokumentu (`${kod_atrybutu}`).

Przykładowa definicja elementu `<inne>` wygląda następująco:

```
<inne>
    <auto_opis wzor="${DOCTYPE} (${przedmiot}, ${nr_protokolu})"/>
</inne>
```

### 4.3.2. Przechowywanie dokumentów

Projektując każde archiwum dokumentów trzeba podjąć decyzję w jaki sposób będą przechowywane umieszczane tam dokumenty (pliki). W zasadzie do wyboru są dwie możliwości: trzymanie plików na dysku w systemie plików lub trzymanie ich razem z pozostałymi danymi używanymi w archiwum w bazie danych jako pole typu *BLOB*. W przypadku mojego *Archiwum* zdecydowałem się na to pierwsze rozwiązanie, gdyż obecnie w kilku aplikacjach na UW jest ono stosowane (np. w systemie *APD*) i bardzo dobrze się sprawdza.

Drugą ważną sprawą, którą należało rozstrzygnąć jest sposób trzymania w bazie metadanych (atrybutów) opisujących poszczególne dokumenty. W związku z tym, że dostępnych ma być wiele różnych typów dokumentów (które można dynamicznie dodawać podczas działania aplikacji) niemożliwe staje się wykorzystanie standardowego rozwiązania, czyli użycia jednej tabeli, w której każda kolumna przechowywałaby wartości jednego atrybutu. Można ten problem rozwiązać na dwa sensowne sposoby.

Pierwszym rozwiązaniem jest utworzenie dla specyficznych atrybutów każdego typu oddzielnej tabeli, w której te dane byłyby przechowywane (w jednym wierszu byłyby zapisane wszystkie metadane dotyczące jednego dokumentu). Drugim z nich jest przechowywanie specyficznych metadanych dla wszystkich typów w jednej tabeli — w jednym wierszu trzymana byłaby jedna wartość jednego atrybutu z jednego dokumentu. W obu tych przypadkach wspólne atrybuty wszystkich dokumentów można trzymać w standardowy sposób w osobnej tabeli.

Każde z tych dwóch rozwiązań ma wady i zalety. Dla tego pierwszego rozwiązania przy dodawaniu nowego typu trzeba tworzyć zawsze nową tabelę — co akurat dość łatwo daje się zautomatyzować (czyli zaimplementować tak, aby w momencie dodawania nowego typu, automatycznie został wygenerowany i uruchomiony skrypt na podstawie pliku XML, który utworzy w bazie nową tabelę), ale już kłopotliwe może być automatyczne wygenerowanie takiego skryptu, w momencie, gdy trzeba zaktualizować istniejący typ dokumentu. To rozwiązanie pozwala jednak na większą przejrzystość w bazie danych od sytuacji, w której jest jedna tabela trzymająca w jednym wierszu jedną wartość. Zwłaszcza, że ta tabela może urosnąć do bardzo wielkich rozmiarów i jej przeszukiwanie może być dość kosztowne (szczególnie, że metadane są kluczowymi danymi w *Archiwum* i bardzo często będą odczytywane i zapisywane).

Zdecydowałem się na to pierwsze rozwiązanie, czyli trzymanie metadanych dla poszczególnych typów dokumentów w osobnych tabelach. Można je lekko zmodyfikować poprzez trzymanie wspólnych metadanych dla wszystkich dokumentów także w tych tabelach (zamiast tworzenie jednej oddzielnej na wspólne metadane), dzięki temu atrybuty dotyczące danego dokumentu byłyby przechowywane razem w jednym miejscu i przy odczytywaniu czy zapisie nie trzeba byłoby sięgać za każdym

razem do dwóch tabel. Podstawową wadą takiego rozwiązania jest jednak utrudnione przeszukiwanie dokumentów po wspólnych atrybutach, gdyż konieczne byłoby sięganie do wielu tabel jednocześnie i używanie w zapytaniu do bazy kosztownej operacji *UNION* — kłopotliwe szczególnie byłoby sortowanie wyników takiego wyszukiwania. W związku z tym zdecydowałem się na trzymanie oddzielnej tabeli trzymającej wspólne metadane.

### 4.3.3. Weryfikacja umieszczanych metadanych

Kolejna sprawa, która wymagała głębokiego przemyślenia to weryfikacja poprawności i spójności umieszczanych przez użytkownika metadanych opisujących dokumenty, czyli określenie jak mocno chcemy te dane weryfikować. Czy na przykład pozwolić na dodanie protokołu egzaminacyjnego, dla którego w metadanych zostanie wpisany przedmiot X, a jako koordynator podany pracownik Y, a w rzeczywistości koordynatorem tego przedmiotu jest ktoś inny? Albo czy można dodać dokument dla Wydziału MIM, gdzie w polu student podany zostanie student Wydziału Chemii? Tego typu różnych przypadków czy koniecznych warunków do sprawdzania może być bardzo dużo, dlatego też zdecydowałem się do sensownego minimum ograniczyć taką weryfikację, gdyż takie sprawdzania byłyby dość skomplikowane, a w dodatku trzeba byłoby migrować o wiele więcej danych z systemu USOS, aby móc te dane weryfikować. Szczególnie, że i tak do końca nigdy poprawności byśmy nie zweryfikowali, bo zawsze ktoś może dodać dokument, który kompletnie nie jest związany z przypisanymi metadanymi, a nie jesteśmy przecież w stanie zweryfikować zawartości wstawianych plików (które mogą być w dowolnym formacie).

W związku z tym do *Archiwum* można np. dodać protokół z egzaminu dla przedmiotu i wykładowcy, który go nie prowadził, ale już nie można dodać dokumentu dla jednostki X i studenta Y, który studiuje na innym wydziale niż X.

## 4.4. Dekompozycja logiczna systemu

### 4.4.1. Warstwa danych

Wszystkie dane *Archiwum* można podzielić na dwie kategorie:

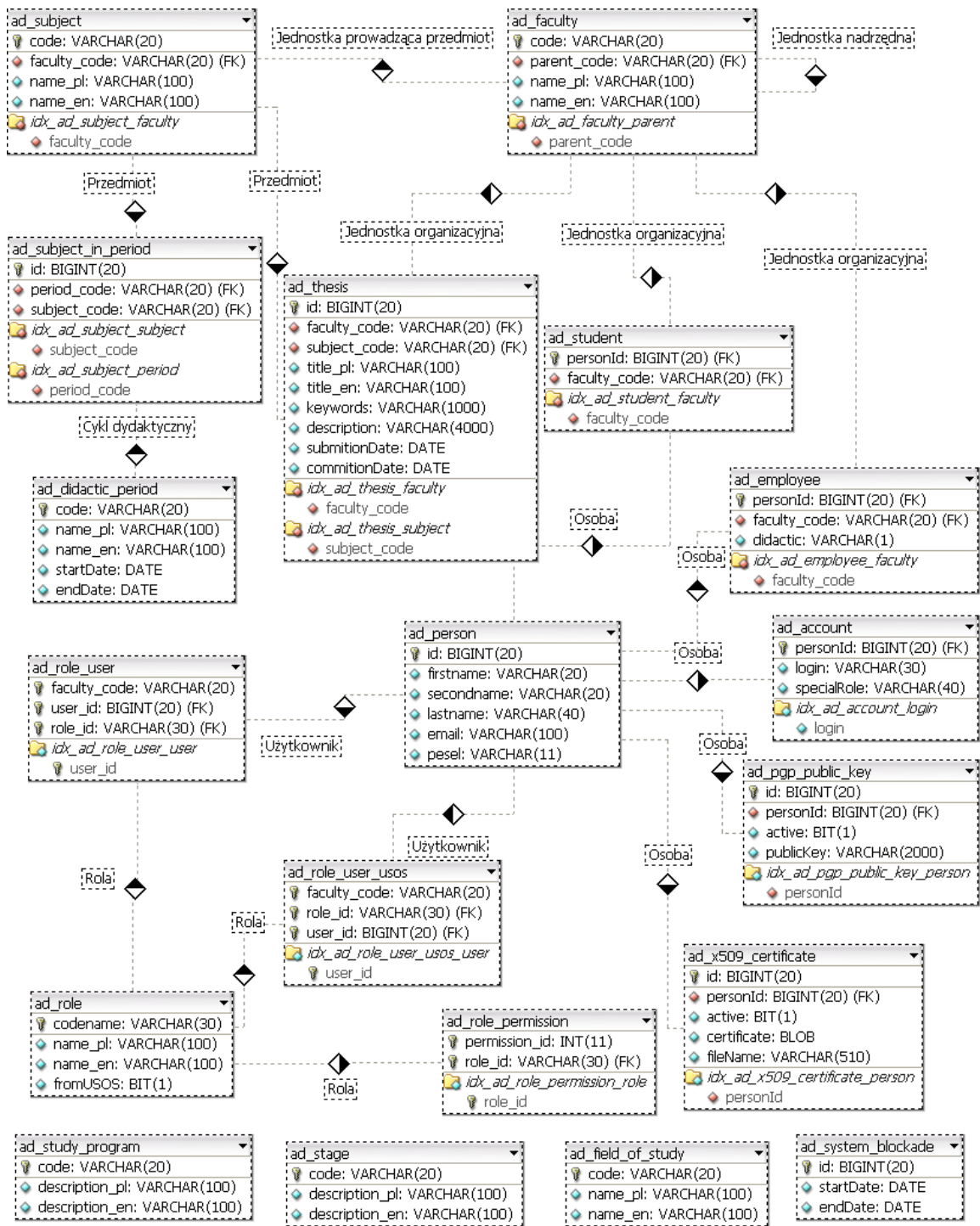
- przechowywane w bazie danych, czyli dane zmigrowane z systemu USOS (użytkownicy, jednostki, przedmioty, programy studiów, itd.) oraz informacje o typach dokumentów, dokumentach, uprawnieniach i certyfikatach użytkowników,
- zapisywane na dysku, czyli pliki (załączniki do dokumentów) umieszczane w *Archiwum* przez osoby tworzące dokumenty.

### 4.4.2. Schemat bazy danych

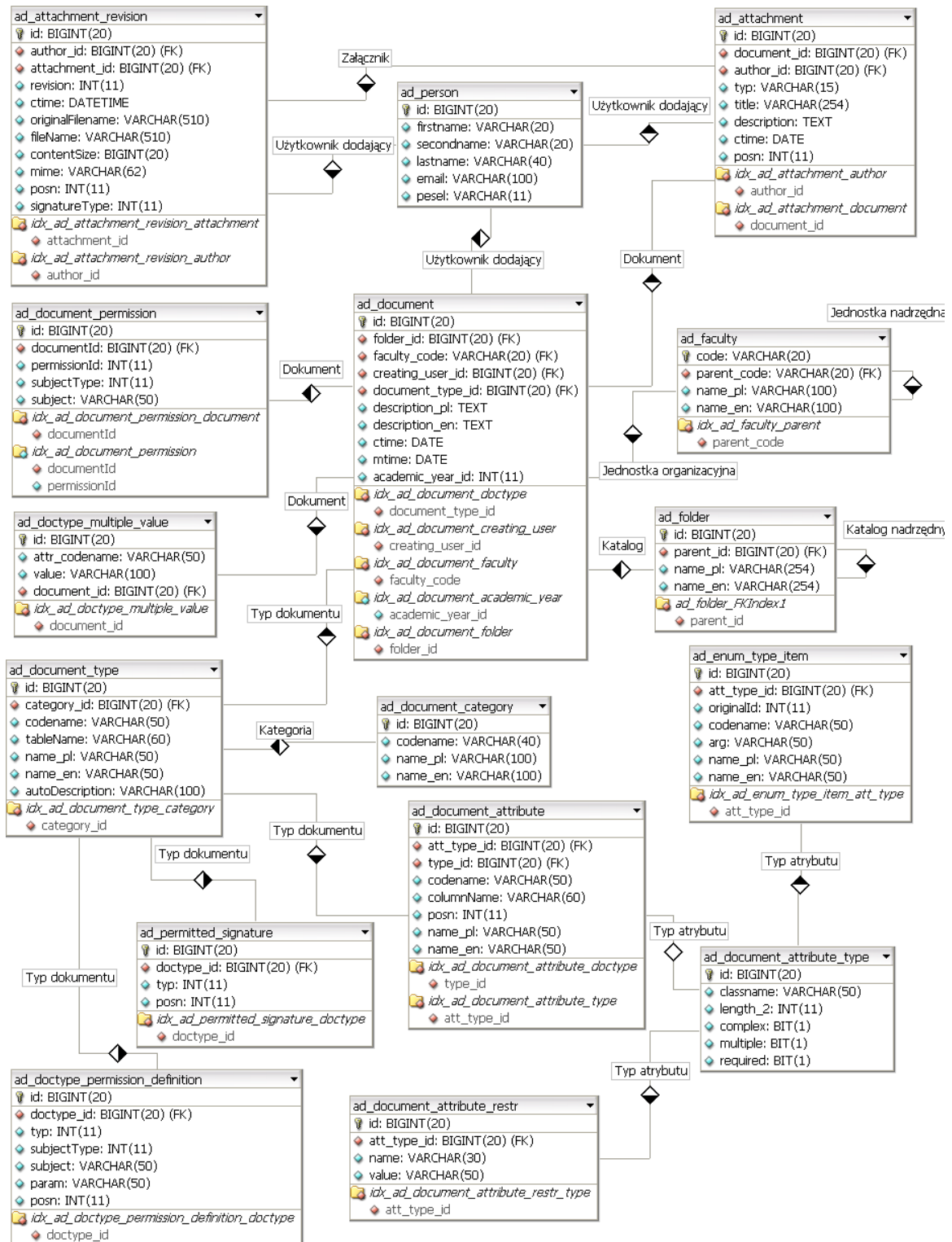
Schemat bazy danych przedstawiono na rysunkach 4.1 i 4.2.

### 4.4.3. Tabele systemu

W tym punkcie zostaną omówione wszystkie tabele *Archiwum* przedstawione na rysunku. W nawiasach obok tabel, które mają swoje odpowiedniki po stronie USOS, podane są nazwy tych oryginalnych tabel.



Rysunek 4.1: Schemat bazy danych cz. 1



Rysunek 4.2: Schemat bazy danych cz. 2

### **Tabela *ad\_person* (DZ\_OSOBY)**

Przechowuje dane dotyczące wszystkich osób w systemie. Każda osoba ma następujące atrybuty: PESEL, imiona, nazwisko oraz email.

### **Tabela *ad\_account* (DZ\_KONTA)**

Przechowuje informacje o kontach użytkowników w systemie. Zawiera wskaźnik na osobę *ad\_person* oraz identyfikator tej osoby w systemie. Na potrzeby *Archiwum* dodano pole *specialRole*, które służy do przydzielania użytkownikowi specjalnych ról — obecnie tylko można wpisać tam słowo *admin*, które oznacza rolę głównego administratora systemu.

### **Tabela *ad\_faculty* (DZ\_JEDNOSTKI\_ORGANIZACYJNE)**

Zawiera informacje o strukturze jednostek organizacyjnych. Każda jednostka może mieć nazwę w języku polskim oraz angielskim, identyfikatorem jest kod jednostki.

### **Tabela *ad\_field\_of\_study* (DZ\_KIERUNKI\_STUDIOW)**

Zawiera informacje o kierunkach studiów (nazwę kierunku w języku polskim oraz angielskim).

### **Tabela *ad\_study\_program* (DZ\_PROGRAMY)**

Opisuje programy studiów w języku polskim oraz angielskim.

### **Tabela *ad\_stage* (DZ\_ETAPY)**

Opisuje etapy studiów w języku polskim oraz angielskim.

### **Tabela *ad\_didactic\_period* (DZ\_CYKLE\_DYDAKTYCZNE)**

Opisuje cykle dydaktyczne (kod, nazwa w języku polskim i angielskim, data początku i data końcowa cyklu).

### **Tabela *ad\_subject* (DZ\_PRZEDMIOTY)**

Każdy wpis w tabeli opisuje konkretny przedmiot. Określa jego przynależność do jednostki organizacyjnej, ponadto zawiera kod przedmiotu oraz nazwę w wersji polskiej i angielskiej.

### **Tabela *ad\_subject\_in\_period* (DZ\_PRZEDMIOTY\_CYKLI)**

Zawiera informacje o przedmiotach w danych cyklach dydaktycznych. Kluczem tej tabeli nie jest para (kod przedmiotu, kod cyklu), jak to miało miejsce w tabeli *DZ\_PRZEDMIOTY\_CYKLI*, ale sztuczny identyfikator *id*. Zostało to tak zaprojektowane głównie ze względu na to, aby wszystkie wartości atrybutów opisujących poszczególne dokumenty były „wartościami pojedynczymi”, co pozwala na prostsze odwoływanie się do tych danych w mocno uogólnionym *Archiwum* (zwłaszcza, że jest to jedyna tabela zmigrowana z systemu USOS, która zawiera klucz podwójny). Pole *id* jest typu *auto\_increment*, w związku z czym identyfikatory są przydzielane kolejnym przedmiotom automatycznie podczas migracji. Trzeba jednak pamiętać, że w przypadku wyczyszczenia takiej tabeli i wykonaniu ponownej migracji dany przedmiot może mieć przydzielone inny *id* niż przy pierwszej migracji. W związku z tym powinno unikać się wykonywania takiej operacji, gdyż może spowodować ona rozspójnienie danych w *Archiwum*.

### **Tabela *ad\_thesis* (DZ\_PRACE\_CERT)**

Opisuje prace dyplomowe, zawiera następujące dane: kod przedmiotu związanego z pracą, jednostka organizacyjną (w której powstała praca), temat pracy (w wersji polskiej oraz angielskiej), data złożenia pracy, data zatwierdzenia tematu, słowa kluczowe, streszczenie.

### **Tabela *ad\_student* (AD\_STUDENCI)**

Zawiera informacje o studentach poszczególnych jednostek organizacyjnych (zawiera tylko identyfikator osoby oraz kod jednostki). Odpowiednikiem w bazie USOS jest widok *AD\_STUDENCI* — nie istnieje pojedyncza tabela przechowujące te dane.

### **Tabela *ad\_employee* (AD\_PRACOWNICY)**

Zawiera informacje o pracownikach poszczególnych jednostek organizacyjnych (zawiera identyfikator osoby, kod jednostki oraz informację czy dany pracownik jest nauczycielem akademickim). Odpowiednikiem w bazie USOS jest widok *AD\_PRACOWNICY* — nie istnieje pojedyncza tabela przechowujące te dane. Obecnie widok ten jest pewnym uproszczeniem — kody jednostek pobierane są z tabeli *DZ\_OSOBY* (a powinny być z *DZ\_PRAC\_ZATR*), w związku z tym jeśli np. dana osoba pracuje na dwóch różnych wydziałach, to w *Archiwum* będzie pamiętane, że jest pracownikiem tylko na tym wydziale z tych dwóch, który jest zapisany w tabeli *DZ\_OSOBY*.

### **Tabela *ad\_role***

Spis wszystkich dostępnych ról w systemie — zawiera kod roli, nazwę polską, nazwę angielską, flagę informującą czy przynależność do tej roli wynika z systemu USOS, czy jest niezależnie określana po stronie *Archiwum*.

### **Tabela *ad\_role\_permission***

Określa globalne uprawnienia w systemie dla danej roli.

### **Tabela *ad\_role\_user***

Określa przynależność osób do poszczególnych ról (tylko tych nadawanych po stronie *Archiwum*, niezależnych od systemu USOS) w poszczególnych jednostkach organizacyjnych. Odpowiednikiem w bazie USOS jest widok *AD\_ROLE* — nie istnieje pojedyncza tabela przechowujące te dane.

### **Tabela *ad\_role\_user\_usos* (AD\_ROLE)**

Określa przynależność osób do poszczególnych ról (tylko tych migrowanych z systemu USOS) w poszczególnych jednostkach organizacyjnych. W przypadku wykonania w przyszłości intuicyjnego złączenia tej tabeli z *ad\_role\_user* w jedną tabelę, konieczne będzie napisanie widoku na potrzeby Migratora dla tej złączonej tabeli, który odpowiadałby obecnym danym z *ad\_role\_user\_usos* (ze względu na ograniczenie Migratora, który nie pozwala wybrać do migracji po stronie bazy MySQL podzbioru danej tabeli — migracja obejmuje zawsze całą wskazaną tabelę lub widok).

### **Tabela *ad\_document\_category***

Spis kategorii dokumentów — każda kategoria może posiadać podkategorie.

### **Tabela *ad\_document\_type***

Tabela opisująca zarejestrowane w *Archiwum* typy dokumentów, zawiera następujące dane:

- *codename* — kod typu dokumentu,
- *tableName* — nazwę tabeli, która przechowuje wartości atrybutów specyficznych dla tego typu dokumentu,
- *category\_id* — identyfikator kategorii dokumentu, do której przynależy ten typ,
- *name\_pl* i *name\_en* — nazwa typu w języku polskim i angielskim,
- *autoDescription* — wzór generowania opisu dla dokumentów tego typu (opis generowany jest na podstawie wartości atrybutów danego dokumentu).

### **Tabela *ad\_document\_attribute***

Tabela opisująca specyficzny atrybut dokumentu związany z danym typem dokumentu. Zawiera m.in. następujące dane:

- *codename* — kod atrybutu,
- *columnName* — nazwę kolumny przechowującej wartości tego atrybutu (w tabeli trzymającej wartości atrybutów specyficznych dla typu dokumentu związanego z tym atrybutem),
- *att\_type\_id* — wskaźnik na typ atrybutu (tabelę *ad\_document\_attribute\_type*),
- *type\_id* — wskaźnik na typ dokumentu,
- *name\_pl* i *name\_en* — nazwa atrybutu w języku polskim i angielskim.

### **Tabela *ad\_document\_attribute\_type***

Opisuje typy poszczególnych atrybutów dokumentu. Oprócz klucza, którym jest sztuczny identyfikator, zawiera następujące dane:

- *complex* — określa czy jest to typ złożony (pewna klasa zaimplementowana w *Archiwum*) czy prosty (liczba, tekst, data, itp.),
- *classname* — nazwa typu, dla typów prostych mogą to być: *Integer*, *Long*, *String*, *Date*, *Boolean*, *Enumeration*; dla typów złożonych: pełna nazwa klasy z *Archiwum* (np. *pl.archiwum.core.base.Person*),
- *length* — maksymalna długość wartości atrybutu (tylko dla pola *String*),
- *required* — określa czy atrybut o tym typie jest obowiązkowy — jeśli jest, to niemożliwe będzie utworzenie danego dokumentu bez podania wartości tego atrybutu,
- *multiple* — określa czy atrybut o tym typie jest wielokrotny, czyli czy możliwe jest nadanie kilku wartości dla tego atrybutu (np. dla typu dokumentu *Praca dyplomowa* możliwe jest wybranie więcej niż jednego autora pracy lub więcej niż jednego opiekuna pracy).



### **Tabela *ad\_document\_attribute\_restr***

Opisuje ograniczenia na dany typ atrybutu — używane tylko dla typów złożonych (np. dla typu *pl.archiwum.core.base.Person* można ustalić ograniczenie, aby wybór nie był ze wszystkich osób, ale tylko spośród studentów lub tylko spośród pracowników).

### **Tabela *ad\_enum\_type\_item***

Opisuje wartości wyliczeniowe dla typów atrybutów, które są typu *Enumeration*.

### **Tabela *ad\_doctype\_permission\_definition***

Określa definicje uprawnień dla typów dokumentów tzn. w jaki sposób będą nadawane dokumentom uprawnienia do odczytu, edycji itp. oraz kto może tworzyć dokumenty tego typu.

### **Tabela *ad\_permitted\_signature***

Zawiera informacje o dozwolonych typach podpisów dla dokumentów dodawanych w ramach poszczególnych typów dokumentów (dopuszczalne typy podpisów to: PGP, X.509 i *brak podpisu*).

### **Tabela *ad\_doctype\_multiple\_value***

Tabela przechowująca wartości atrybutów wielowartościowych w dokumentach. Zawiera identyfikator dokumentu, kod atrybutu i wartość atrybutu (kluczem jest sztuczny identyfikator).

### **Tabela *ad\_attachment***

Przechowuje informacje o załącznikach poszczególnych dokumentów. Załączniki mogą być dwóch rodzajów — dokument posiada jeden główny plik oraz dowolną liczbę dodatkowych załączników.

### **Tabela *ad\_attachment\_revision***

Przechowuje informacje o wersjach poszczególnych załączników. Zawiera m.in. takie dane jak: numer wersji załącznika, wskaźnik na załącznik (*ad\_attachment*), nazwę pliku na dysku na serwerze, gdzie zapisano plik odpowiadający tej wersji załącznika, typ podpisu, którym został podpisany ten plik.

### **Tabela *ad\_folder***

Opisuje wirtualne katalogi w *Archiwum*, w których znajdują się dokumenty.

### **Tabela *ad\_document***

Zawiera informacje o umieszczonych w *Archiwum* dokumentach. Zawiera m.in. wskaźnik na jednostkę organizacyjną, której ten dokument dotyczy, opis dokumentu w języku polskim i angielskim, datę umieszczenia i wskaźnik na osobę, która umieściła ten dokument w *Archiwum*, wskaźnik na katalog, w którym znajduje się ten dokument.

### **Tabela *ad\_document\_permission***

Opisuje uprawnienia do poszczególnych dokumentów. Zawiera następujące dane:

- *documentId* — identyfikator dokumentu, którego dotyczy dane uprawnienie,
- *permissionId* — identyfikator uprawnienia (może to być prawo do odczytu dokumentu, edycji dokumentu, pobierania załączników lub dodawania/modyfikacji załączników,
- *subjectType* — rodzaj podmiotu, któremu przydziela się uprawnienie (obecnie dostępne są 3 rodzaje: osoba, rola i *wszyscy*),
- *subject* — określa podmiot, któremu przydziela się uprawnienie — w zależności od *subjectType* będzie to identyfikator osoby, kod roli albo *null* (dla rodzaju *wszyscy*).

### **Tabela *ad\_pgp\_public\_key***

Przechowuje klucze publiczne PGP zarejestrowane dla poszczególnych użytkowników.

### **Tabela *ad\_x509\_certificate***

Przechowuje certyfikaty X.509 zarejestrowane dla poszczególnych użytkowników.

### **Tabela *ad\_system\_blockade***

Przechowuje informacje o blokadzie systemu (zawiera datę początku i końca blokady).

### **Tabele dla danych typów dokumentów**

Oprócz wymienionych tabel są też tabele przechowujące wartości atrybutów specyficznych dla poszczególnych typów dokumentów. Są one tworzone w momencie rejestrowania typu dokumentu. Nazwy tych tabel są tworzone według wzoru: *ad\_doc\_{kod\_typu}*, gdzie *{kod\_typu}* oznacza kod danego typu dokumentu. Tabele te zawierają kolumnę *document\_id* (identyfikator dokumentu, którego dotyczą wartości w danym wierszu) oraz kolumny odpowiadające wszystkim specyficznym atrybutom określonym dla tego typu dokumentu (o nazwach identycznych z kodami poszczególnych atrybutów).

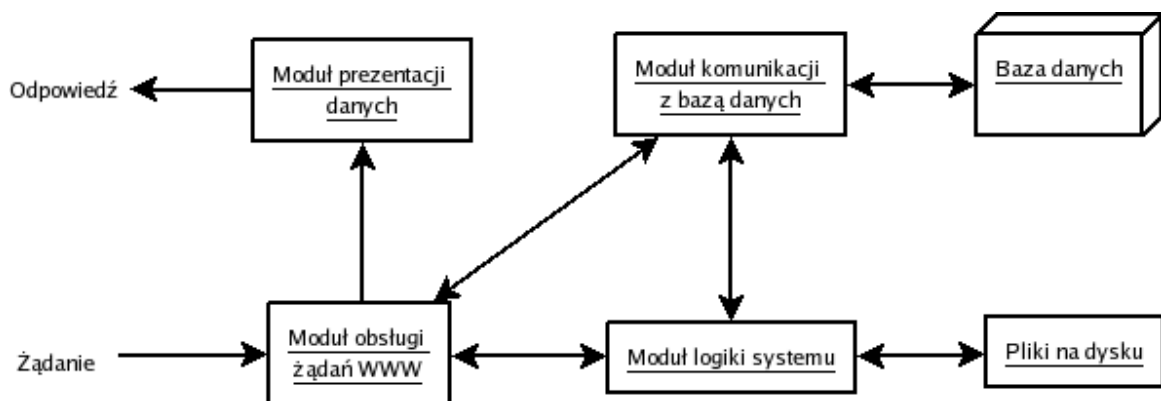
### **Tabele na potrzeby Migratora**

Ponadto istnieją dwie tabele *M\_CHANGE\_LOG* i *M\_CHANGE\_LOG\_VALUES*, które są wymagane do działania Migratora. Nie są one w żaden sposób wykorzystywane przez *Archiwum*.

## **4.5. Struktura systemu**

W tym punkcie omówiony jest podział *Archiwum* na logiczne części. Realizację każdego żądania przez system (komunikację między poszczególnymi modułami systemu) ilustruje schemat na rysunku 4.3. Strzałki określają przepływ informacji.

Moduł obsługi żądań WWW odpowiada za realizację wszystkich żądań od użytkowników. W zależności od otrzymanego zapytania, inicjowana jest odpowiednia akcja, która na podstawie parametrów przesłanych przez użytkownika powoduje wykonanie pewnych operacji, komunikując się z modułem logiki systemu lub w pewnych sytuacjach od razu z modułem komunikacji z bazą danych



Rysunek 4.3: Struktura *Archiwum*

(gdy konieczne jest tylko pobranie gotowych danych z bazy nie wymagających żadnego przetwarzania). Otrzymane wyniki przekazuje do modułu prezentacji danych, który wykorzystując je odpowiednio przygotowuje odpowiedź w HTML, która zostanie wysłana użytkownikowi.

Moduł logiki systemu odpowiada za przetwarzania danych i wykonywanie różnych operacji w systemie:

- autoryzację użytkownika,
- zarządzanie dokumentami, czyli dodawanie, modyfikacja i wyszukiwanie dokumentów,
- zarządzanie typami dokumentów, w szczególności parsowanie odebranego od użytkownika pliku XML, zawierającego opis typu dokumentu i tworzenie z odczytanych danych struktury obiektowej, która przez warstwę dostępu do bazy danych jest odpowiednio mapowana i zapisywana w bazie,
- dodawanie i usuwanie plików/załączników,
- przeszukiwanie różnych słowników z danymi (osób, przedmiotów, cykli dydaktycznych, kierunków studiów itd.),
- bezpieczeństwo systemu: sprawdzanie wszystkich praw dostępu do dokumentów i do wykonywania poszczególnych operacji, modyfikacja uprawnień i ról,
- zarządzanie certyfikatami i podpisami, w szczególności weryfikacja podpisanych plików za pomocą certyfikatów wcześniej zarejestrowanych w *Archiwum*.

Moduł komunikacji z bazą danych odpowiada za pobieranie z bazy i zapisywanie do bazy wszystkich danych, jak również za przeszukiwanie bazy poprzez wykonywanie odpowiednich zapytań. Odczytywane z bazy dane są mapowane na obiekty, której później są wykorzystywane przez inne warstwy systemu.

## 4.6. Warstwa prezentacji

W *Archiwum* istnieje jeden wspólny interfejs WWW obsługujący żądania od wszystkich użytkowników, przy czym w zależności od uprawnień część funkcjonalności może być niewidoczna dla pewnych użytkowników. Administrator systemu ma dostęp do pełnej funkcjonalności, zwykły użytkownik nie widzi panelu administratora, natomiast administrator jednostki organizacyjnej ma dostęp do

panelu administratora, ale tylko częściowy. W przypadku próby wyświetlenia strony, do której użytkownik nie ma wystarczających uprawnień, zostanie wyświetlony stosowny komunikat. Ponadto w zależności od uprawnień poszczególni użytkownicy mogą mieć możliwość wykonywania innych operacji na poszczególnych stronach, np. każdy użytkownik może przejść do ekranu dodawania nowego dokumentu, ale w zależności od uprawnień różni użytkownicy będą mieli możliwość umieszczania innych dokumentów.

Wygląd interfejsu jest wzorowany na istniejących już aplikacjach uniwersyteckich, takich jak USOSweb i APD.

## Rozdział 5

# Implementacja

W niniejszym rozdziale został przedstawiony sposób realizacji projektu.

### 5.1. Wybór technologii

W kolejnych punktach opiszę technologie, które wybrałem dla mojego systemu i postaram się uzasadnić taki właśnie wybór.

#### 5.1.1. Java

*Archiwum* zostało napisane z użyciem języka Java (jako aplikacja *JEE*). Z jednej strony decyzja ta może wydawać się niezrozumiała, zwłaszcza, że język skryptowy PHP sprawdził się i jest wykorzystywany w większości aplikacji uniwersyteckich. Jednakże język Java również jest bardzo znany i od wielu lat bardzo popularny i rozwijany, istnieje wiele różnych, dobrze udokumentowanych bibliotek dla języka Java wspomagających i przyspieszających pisanie aplikacji bazodanowych w technologii klient-serwer. Przykładem aplikacji *JEE* o podobnej funkcjonalności do *Archiwum*, która jest z powodzeniem wykorzystywana na innych uniwersytetach, jest system Uniwex (por. p. 2.5). Ponadto miałem większe doświadczenie w tworzeniu projektów w tej technologii, a w związku z tym, że jest to projekt testowy, uznałem, że lepszym rozwiązaniem będzie wykorzystanie Javy. Pozwoliło mi to na napisanie systemu z większą funkcjonalnością, który z pewnością przyczyni się do uruchomienia w przyszłości na całym Uniwersytecie w pełni działającego bezpiecznego archiwum elektronicznego.

Wykorzystałem następujące biblioteki Javowe (wszystkie one są dostępne bezpłatnie):

- Hibernate [Hibernate] — biblioteka odpowiadająca za warstwę dostępu do danych (ang. persistence layer). Zapewnia przede wszystkim translację (ang. O/R mapping) danych pomiędzy relacyjną bazą danych a obiektami Javy. Do opisu struktury danych wykorzystuje język XML. Dodatkowo Hibernate zwiększa wydajność operacji na bazie danych dzięki buforowaniu i minimalizacji liczby przesyłanych zapytań.
- Webwork [WW] — szkielet, struktura (ang. framework) wspomagająca tworzenie aplikacji webowych na bazie technologii JEE, pozwala zminimalizować kod i skupić się na logice biznesowej, udostępnia bibliotekę różnych komponentów i znaczników nadających się do wielokrotnego użycia na stronach JSP (Java Server Pages), zawiera m.in. wsparcie dla internacjonalizacji oraz do pisania stron w technologii AJAX.
- BouncyCastle [BC] — służy do zarządzania kluczami, certyfikatami i podpisami elektronicznymi (jest implementacją interfejsu do architektury kryptograficznej Java Cryptography Architecture [JCA]). Posiada wsparcie dla wielu różnych formatów podpisów, w szczególności

pozwala na podpisywanie i weryfikację podpisów w standardzie PGP oraz weryfikację podpisów w formatach używanych w podpisach kwalifikowanych takich jak *CMS* i *PKCS#7*.

### 5.1.2. Apache Tomcat

Serwerem, na którym będzie działać *Archiwum* jest kontener aplikacji webowych Apache Tomcat [Tomcat], który jest rozwijany w ramach projektu Apache [Apache]. Umożliwia on uruchamianie aplikacji internetowych w technologiach serwletów i stron JSP.

### 5.1.3. MySQL

Jako serwer bazy danych wybrałem serwer MySQL [MySQL], który również jest darmowy i sprawdził się w wielu aplikacjach na naszym Uniwersytecie. Ponadto Migrator został napisany z myślą o migracji między bazami Oracle i MySQL, w przypadku wyboru innej bazy, konieczne mogłoby być odpowiednie przerobienie i dostosowanie Migratora.

### 5.1.4. L<sup>A</sup>T<sub>E</sub>X

Wszystkie dokumenty, które powstały w wyniku mojej pracy magisterskiej, zostały napisane przy użyciu języka LaTeX [LaTeX2e] i skompilowane do formatu *PDF* za pomocą programu *pdflatex*.

## 5.2. Narzędzia

### 5.2.1. Eclipse

Jako edytora dla języka Java i innych tworzonych plików (m.in. JSP, XML, CSS) użyłem zaawansowanego edytora Eclipse [Eclipse]. Jest to produkt darmowy, posiadający wsparcie dla wielu języków programowania, w szczególności do pisania aplikacji JEE. Oferuje on podświetlanie składni języka, dopełnianie nazw, odnośniki do deklaracji i definicji funkcji oraz wiele innych, przydatnych dla programisty opcji.

### 5.2.2. Apache Ant

Apache Ant [Ant] to narzędzie służące do zautomatyzowania procesu budowy oprogramowania. Jest podobne do znanego z Linuxa programu Make, ale napisane jest w języku Java i jest przede wszystkim wykorzystane z programami napisanymi w tym języku. Do opisu procesu budowy i jego zależności używane są pliki w formacie XML.

### 5.2.3. DBDesigner

Do tworzenia projektu bazy danych systemu użyłem, dostępnego na licencji GPL, programu DBDesigner [DBDes]. Pozwala on na graficzne modelowanie związków encji oraz generowanie skryptów tworzących zaprojektowaną bazę danych. Posiada wsparcie dla takich serwerów baz danych jak MySQL, PostgreS, Oracle i wielu innych.

## 5.3. Elementy systemu

W skład *Archiwum* wchodzi:

- klasy w języku Java podzielone w następujący sposób:

- dostęp do danych (pakiet *pl.archiwum.core*) — zawiera wszystkie klasy związane z dostępem do bazy danych, w tym klasy używane przez Hibernate do mapowania zawartości tabel bazy danych na obiekty (wraz z opisem mapowania w odpowiednich plikach XML),
  - logika systemu (pakiet *pl.archiwum.logic*) — odpowiada za sprawdzanie uprawnień, wykonywanie wszystkich operacji na dokumentach, typach dokumentów, załącznikach, certyfikatach itp.
  - obsługa żądań WWW (pakiet *pl.archiwum.web* i *pl.archiwum.webwork*) — *pl.archiwum.web* odpowiada za obsługę żądań użytkowników — znajdują się tam wszystkie *akcje* Webworka, natomiast *pl.archiwum.webwork* dostarcza kilka dodatkowych znaczników używanych na stronach JSP oraz implementuje bazową klasę dla wszystkich *akcji* z pakietu *pl.archiwum.web*,
- prezentacja danych — strony JSP generujące strony HTML-owe w odpowiedzi na żądania użytkownika oraz statyczne zasoby — arkusze stylów definiujące wygląd stron WWW oraz obrazki wyświetlane na stronach WWW,
  - skrypty do zakładania tabel po stronie *Archiwum*,
  - plik konfiguracyjny aplikacji Migrator oraz skrypt tworzący widoki w bazie USOS, które są wykorzystywane w tym pliku,
  - pliki XML opisujące poszczególne typy dokumentów.

## 5.4. Obsługa żądań WWW

*Archiwum* zostało napisane przy użyciu *Webworka* i cała obsługa żądań WWW jest na nim oparta. Celem jest zaimplementowanie różnych *akcji*, które są później odpowiednio wywoływane przez filtr *Webworka*, który przechwytuje wszystkie żądania HTTP. *Akcja* to zbiór operacji, które powinny być wykonane na dane żądanie użytkownika. Wszystkie dostępne *akcje* są definiowane w pliku *xwork.xml* — każda *akcja* jest odwzorowywana na pewną klasę, która musi posiadać metodę *execute()*. Wywołanie danej *akcji* polega właśnie na wykonaniu tej metody *execute()*, która przekazuje wynik. Przed wywołaniem tej metody *Webwork* zadba o to, aby parametry żądania HTTP odwzorować na odpowiednie atrybuty *akcji* za pomocą odpowiednich akcesorów. W zależności od wyniku może nastąpić przekierowanie do innego zasobu/*akcji* albo wywołanie odpowiedniej strony JSP, która wygeneruje odpowiedź na żądanie HTTP. Sposób zachowania się w zależności od wyniku *akcji* też jest definiowany w *xwork.xml*.

Wszystkie *akcje* w *Archiwum* są zaprojektowane w następujący sposób: każda *akcja* składa się ze zbioru *zdarzeń*. Jedno *zdarzenie* to sekwencja pewnych operacji realizujących dane zadanie, często polegających na wywoływaniu odpowiednich funkcji udostępnianych przez logikę aplikacji (czyli pakiet *pl.archiwum.logic*), które przekazują pewne wyniki. Wyniki te *akcja* zapisuje na swoich atrybutach, które później poprzez akcesory są odczytywane przez strony JSP i używane do wygenerowania odpowiedzi dla użytkownika.

## 5.5. Komunikacja z bazą danych

Komunikacja z bazą danych jest realizowana przy pomocy biblioteki Hibernate. Większość tabel z bazy danych jest odwzorowywana na obiekty Javy i wczytywanie/zapisywanie danych do bazy odbywa się poprzez odczytywanie/zapisywanie atrybutów odpowiednich obiektów. Tylko mała część

danych jest odczytywana i zapisywana do bazy ręcznie (tzn. poprzez ręczne wskazanie biblioteki Hibernate, jakie zapytanie na bazie ma zostać wykonane) — są to głównie dane dokumentów dotyczące specyficznych atrybutów związanych z typem dokumentu. Wynika to z faktu, iż typy dokumentów są dodawane dynamicznie, w związku z czym nie mogłem utworzyć odpowiednich klas i statycznych odwzorowań tych danych na bazę. Każdej klasie w Javie, która jest odwzorowywana na bazę danych, jest przyporządkowany plik XML określający odwzorowanie, o takiej samej nazwie jak klasa, ale innym rozszerzeniu (dla pliku Xxx.java jest to Xxx.hbm.xml).

## 5.6. Logika systemu

Cała logika *Archiwum*, czyli odczytywanie danych z bazy, ich przetwarzanie i zapisywanie danych do bazy, znajduje się w pakiecie *pl.archiwum.logic*. Zaimplementowana jest ona poprzez zbiór statycznych klas (menedżerów) udostępniających odpowiednie funkcje przetwarzające dane. Każda klasa odpowiada za inną część systemu: jedna pozwala zarządzać dokumentami, inna odpowiada za uprawnienia w systemie, jeszcze inna zarządza certyfikatami i podpisami elektronicznymi itd.

### 5.6.1. Typy dokumentów

Dynamiczne dodanie nowego typu do *Archiwum* polega na załadowaniu poprzez interfejs administratora pliku XML opisującego ten typ. W systemie każdy typ dokumentu jest reprezentowany przez strukturę odpowiednio ze sobą powiązanych obiektów, która przez Hibernate odwzorowana jest na poszczególne tabele bazy danych. W związku z tym dodanie nowego typu polega na sparsowaniu pliku XML i utworzenie odpowiednich obiektów. Tabela w bazie danych trzymająca wartości atrybutów specyficznych dla dokumentów tego typu zostaje wówczas automatycznie utworzona.

Możliwe jest także zaktualizowanie istniejącego już w *Archiwum* typu dokumentu. Także odbywa się to poprzez załadowanie pliku XML i utworzeniu odpowiednich obiektów, które są zapisywane do bazy zamiast poprzedniej struktury opisującej ten typ. Jeśli dana aktualizacja wymaga modyfikacji tabeli z wartościami specyficznych atrybutów, to musi zostać wykonana ręcznie w bazie danych. Należy zauważyć, że automatyczne wygenerowanie takiego skryptu jest w pewnych przypadkach dość trudne i mogłoby się czasem zdarzyć, że wykonana automatyczna modyfikacja nie będzie identyczna z zamierzeniami twórcy zaktualizowanego pliku XML.

Obecnie zaimplementowane typy dokumentów (czyli typy, dla których zostały utworzone pliki XML) to *Decyzja administracyjna w sprawie studenta*, *Dyplom*, *Karta egzaminacyjna*, *Karta przebiegu studiów*, *Księga Albumów*, *Księga Dyplomów*, *Lista stypendialna*, *Podanie do dziekana*, *Praca Dyplomowa*, *Protokół egzaminacyjny* oraz *Rozliczenie pensum*.

### 5.6.2. Uprawnienia

System ról i uprawnień jest jednym z ważniejszych i często wykorzystywanych elementów systemu, gdyż przed wykonaniem różnych operacji trzeba sprawdzić czy dany użytkownik posiada odpowiednie uprawnienie. Dlatego, też ze względów wydajnościowych buforowane są informacje o rolach użytkowników i globalnych uprawnieniach. W momencie modyfikacji tych ról i uprawnień, odpowiednie dane w buforach są uaktualniane.

### 5.6.3. Podpisy elektroniczne

Podpisy elektroniczne wykorzystywane są poprzez przechowywanie plików podpisanych w różnych formatach. Samo podpisywanie plików nie jest wykonywane przez *Archiwum*. Obecnie do *Archiwum* wkładane mogą być zarówno pliki podpisane, jak i pliki niezawierające podpisu, akceptowane są



zarówno podpisy w standardzie PGP jak i X.509 (zwykłe i kwalifikowane), przy czym obecnie *Archiwum* rozpoznaje podpisy X.509 tylko w formacie *CMS*, czyli w takim jak podpisy generowane przez Unizeto. W przyszłości dobrze byłoby rozszerzyć tę funkcjonalność do akceptowania wszystkich formatów podpisów X.509.

W momencie dodawania pliku następuje proces jego weryfikacji za pomocą certyfikatów zarejestrowanych dla osoby umieszczającej ten dokument. W przypadku udanej weryfikacji, plik jest przyjmowany do *Archiwum*, wpp. wyświetlany jest komunikat o niemożności poprawnego zweryfikowania pliku i niemożliwe będzie jego dodanie. Oprócz wyboru pliku, interfejs umożliwia użytkownikowi wskazanie typu podpisu, którym umieszczany plik został podpisany — jednym z typów jest *brak podpisu* — w związku z czym niepodpisany plik nie będzie weryfikowany, tylko od razu umieszczany w *Archiwum*.

Nie jest obecnie możliwe wstawienie przez osobę X pliku podpisanego przez osobę Y. W niektórych sytuacjach być może przydatne byłoby umożliwienie wykonania takiej operacji, więc w przyszłości można o tym pomyśleć. Podobnie jeśli umieszczony zostanie plik podpisany przez więcej niż jedną osobę, to *Archiwum* będzie weryfikowało i wyświetlało informacje tylko o jednym podpisie (złożonym przez osobę, która dodała ten plik).

Obecnie podczas weryfikacji sprawdzana jest tylko poprawność podpisu, nie jest weryfikowana jego ważność. Podobnie podczas rejestrowania nowego certyfikatu w systemie tylko jest sprawdzana jego poprawność (nie jest weryfikowana jego ważność). Zatem mamy pewność, że zawartość pliku, który został podpisany, nie uległa zmianie od momentu podpisania, ale nie wiemy, czy ten plik został podpisany certyfikatem, który w momencie składania podpisu był ważny.

Aby osiągnąć ten cel, należałoby w momencie rejestracji certyfikatu (lub żeby było bezpiecznie także podczas weryfikacji każdego podpisu weryfikowanego tym certyfikatem) sprawdzać jego *ścieżkę certyfikacji*. Ponadto podczas weryfikacji podpisu trzeba by sprawdzać, czy certyfikat, którego użyto do złożenia podpisu, był ważny w momencie składania (z reguły certyfikat jest wystawiany tylko na okres 2 lat) i czy nie znajduje się na liście *CRL* (*Certificate Revocation List*), czyli liście unieważnionych certyfikatów (listy te są przechowywane, co jakiś czas aktualizowane i udostępniane poprzez internet przez poszczególne centra certyfikacji).

W chwili obecnej dla podpisów w standardzie X.509 nie jest rozpoznawane, czy jest to podpis zwykły czy kwalifikowany. Aby to osiągnąć należałoby sprawdzać politykę podpisu, która jest przypisana danemu certyfikatowi X.509 (gdyż właśnie polityka podpisu decyduje m.in. o tym, czy dany certyfikat ma właściwości certyfikatu kwalifikowanego). W związku z tym konieczne byłoby pamiętanie w bazie wszystkich polityk podpisu, które powodują, że certyfikat jest kwalifikowany (czyli także podpisy generowane przy jego użyciu są kwalifikowane) i wówczas jeśli podpis zostałby zweryfikowany certyfikatem X.509, który uznajemy za kwalifikowany, to podpis ten byłby oznaczany w *Archiwum* jako kwalifikowany, natomiast, gdyby okazało się, że jest on weryfikowany za pomocą certyfikatu (którego polityki podpisu nie ma oznaczonej jako tej, która służy do generowania podpisów kwalifikowanych), to oznaczany byłby w archiwum jako podpis zwykły. Wówczas możliwe byłoby rozszerzenie definiowania dopuszczalnych podpisów w typach dokumentów na podpisy zwykłe i kwalifikowane (obecnie jest tylko rozróżnienie na podpisy PGP i X.509).

Obecnie wyłącznie administratorzy mają prawo rejestrować nowe certyfikaty dla użytkowników, w przyszłości być może przydatne byłoby umożliwienie rejestracji własnych certyfikatów przez wszystkich użytkowników. Taki certyfikat mógłby być z początku nieaktywny, a dopiero administrator uaktywniałby go (po sprawdzeniu czy rzeczywiście dodany certyfikat należy do tej osoby). Dla każdego umieszczonego pliku użytkownicy (którzy mają do niego dostęp) mogą dowiedzieć się jakim podpisem i przez kogo został on podpisany. Również jest możliwe pobranie certyfikatu (lub obejrzenia w *Archiwum* podstawowych cech certyfikatu), który został użyty do złożenia podpisu.

Nie ma obecnie możliwości usunięcia z *Archiwum* zarejestrowanego certyfikatu. Z jednej strony jest to rozsądne, bo przecież już część umieszczonych plików mogło zostać podpisanych przy użyciu

danego certyfikatu, więc nie powinno być możliwości usunięcia takiego certyfikatu. Z drugiej jednak strony można ten problem obejść poprzez pamiętanie z każdym podpisanym plikiem dowiązania do certyfikatu zarejestrowanego w systemie, który został użyty do podpisania tego pliku (obecnie nie jest pamiętane to dowiązanie — podczas każdej weryfikacji pliku następuje próba zweryfikowania wszystkimi certyfikatami zarejestrowanymi dla osoby umieszczającej ten plik) i wówczas możliwe byłoby usuwanie certyfikatów, do których nie są dowiązane żadne podpisane pliki.

Na pewno w *Archiwum* przydatna byłaby funkcjonalność pozwalająca podpisywać pliki bezpośrednio w *Archiwum*. Mogłoby to znacznie ułatwić i przyspieszyć pracę z *Archiwum* i nie byłoby konieczności używania dodatkowej aplikacji do składania podpisu. Ze względów bezpieczeństwa podpisywanie musi odbywać się po stronie klienta (a nie serwera), w związku z tym byłoby to rozwiązać poprzez napisanie apletu, który uruchamiany byłby bezpośrednio ze strony WWW, ale w kontekście użytkownika (na komputerze klienta). Taki aplet mógłby wspomagać tworzenie zarówno podpisów kwalifikowanych (za pomocą kluczy prywatnych zapisanych na karcie kryptograficznej), jak i podpisów zwykłych (wówczas klucze mogłyby być trzymane na dysku użytkownika). Do generowania podpisów zwykłych wystarczy wykorzystać użytą już w *Archiwum* bibliotekę BouncyCastle, natomiast dla podpisów kwalifikowanych można spróbować skorzystać z bibliotek do podpisu, które wykorzystuje Unizeto w swoim oprogramowaniu (przy pomocy interfejsu Java Cryptography Architecture i implementacji napisanej przez Sun Microsystems [SUNPKCS11], która pozwala podłączyć się do dowolnej biblioteki PKCS#11 i przy jej pomocy generować podpisy) — to pozwoliłoby wykonywać podpisy z *Archiwum* tylko przez użytkowników systemu operacyjnego Windows (gdyż Unizeto posiada implementacje bibliotek PKCS#11 tylko dla Windowsa), ale w zasadzie nie ograniczy to w żaden sposób dotychczasowej „wieloplatformowości”, gdyż w obecnej sytuacji chcąc wykonać podpis przy pomocy certyfikatu oferowanego przez Unizeto, także musimy korzystać z oferowanej przez tę firmę aplikacji pod system Windows.

Trzeba jeszcze pamiętać, że chcąc przechowywać dokumenty w *Archiwum* przez dłuższy okres trzeba dbać o konserwację podpisów elektronicznych — poprzez cykliczne znakowanie czasem wszystkich dokumentów umieszczonych w *Archiwum*. Każde takie znakowanie przedłuża ważność podpisów pod dokumentami na 10 lat — w przypadku niestosowania usługi znakowania czasem, ważność podpisu elektronicznego wygasa w momencie wygaśnięcia ważności certyfikatu, który został użyty do złożenia tego podpisu (czyli taki podpis może być ważny maksymalnie przez 2 lata). Zapewne nie wszystkie dokumenty wymagają, aż tak długiego składowania, w związku z tym część dokumentów można co jakiś czas usuwać z *Archiwum* i tylko ważne dokumenty będą wówczas znakowane czasem.

## 5.7. Prezentacja danych

Za prezentację danych odpowiadają strony JSP, które na podstawie danych (wyników) udostępnianych przez *akcje*, generują odpowiedź w postaci odpowiedniej strony HTML. Każda strona jest budowana według jednego z dwóch szablonów.

Główne strony serwisu wykorzystują szablon *main.jsp* (zbudowanego na wzór szablonu strony znajdującego się w systemie USOSweb), który składa się z:

- górnego paska, na którym są umieszczone m.in. odnośniki do zmiany języka oraz zalogowania/wylogowania z systemu,
- lewej kolumny, zawierającej menu aplikacji,
- środkowej, największej części przeznaczonej na wypisywanie właściwych informacji,
- stopki.

Drugim szablonem jest *popup.jsp*, który jest wykorzystywany do wyświetlania stron HTML w otwieranych pomocniczo okienkach np. dla różnych wyszukiwarek (osób, przedmiotów, kierunków studiów itd.) i stron wyświetlających szczegóły o certyfikatach zarejestrowanych w *Archiwum*. Ten szablon nie zawiera żadnego nagłówka, menu aplikacji i stopki — umieszczony jest jedynie odnośnik *zamknij* (pozwalający szybko zamknąć okienko) znajdujący się w lewym górnym rogu.

### 5.7.1. Obsługa dwóch języków

Wszystkie strony mogą być wyświetlone w języku polskim i angielskim. Wersje językowe dla napisów nie pochodzących z bazy danych są zapisywane w plikach *\*.properties*. Są to pliki *global-messages\_pl.properties* i *global-messages\_en.properties* oraz zbiór par plików *package\_pl.properties* i *package\_en.properties* umieszczonych w różnych pakietach Javy. Do wyświetlania napisów w odpowiednim języku na stronach JSP wykorzystane zostało wsparcie dla internacjonalizacji, które daje Webwork. Wygodne jednak jest także korzystanie z internacjonalizacji w kodzie Javy — do tego celu wykorzystałem klasę *StringManager* dostępną w ramach projektu Apache [Apache], która na podstawie aktualnego języka, odczytuje napisy z odpowiedniego pliku.

### 5.7.2. AJAX

Kilka stron napisałem w oparciu o technologię AJAX — do tego celu wykorzystałem wsparcie, jakie udostępnia w tym zakresie Webwork. AJAX został użyty np. na stronie JSP generującej formularz WWW do tworzenia nowego dokumentu (dzięki temu podczas zmiany typu dokumentu nie trzeba przeładowywać całej strony, a jedynie poprzez asynchroniczne żądanie z serwera pobierane zostają dane na temat specyficznych danych dla wybranego typu dokumentu) oraz w panelu administratora na stronie pozwalającej rejestrować dla użytkowników nowe certyfikaty i stronie pozwalającej przydzielać użytkownikom do ról.



## Rozdział 6

# Instalacja i uruchomienie systemu

Ten rozdział zawiera szczegóły techniczne pomocne przy uruchomieniu aplikacji, a także opis ogólnych właściwości wykorzystanych narzędzi.

### 6.1. Wymagania *Archiwum* względem środowiska

Wymagania jakie musi spełniać środowisko, na którym ma być instalowane *Archiwum* zostały opisane w kolejnych punktach.

#### 6.1.1. Java

*Archiwum* jest napisane w języku Java, w związku z tym przed uruchomieniem konieczne jest zainstalowanie Java SE co najmniej w wersji 5 (podczas implementacji *Archiwum* była właśnie wykorzystywane Java 5).

#### 6.1.2. System operacyjny

W związku z tym, że *Archiwum* zostało napisane w Javie (która jest niezależna od platformy) możliwe jest uruchomienie tej aplikacji zarówno pod systemem Windows, jak i pod Linuksem.

#### 6.1.3. Apache Tomcat

Najlepiej jak *Archiwum* będzie uruchomiane na serwerze Apache Tomcat [Tomcat] (podczas tworzenia aplikacji był wykorzystywany Apache Tomcat w wersji 5.5.20), jednakże gdy znajdzie taka konieczności, to uruchomienie *Archiwum* na innym serwerze JEE również powinno się udać.

#### 6.1.4. Apache Ant

Apache Ant [Ant] jest darmowym narzędziem służącym do zautomatyzowania procesu budowy oprogramowania. W związku z tym nie jest ono wymagane do poprawnego działania aplikacji, a jedynie wykorzystywane i niezbędne do kompilacji *Archiwum*.

#### 6.1.5. MySQL

Jako serwer bazodanowy został wybrany projekt MySQL [MySQL], który również jest darmowy i sprawdzony w wielu aplikacjach, w tym także na Wydziale MIMUW. Zalecaną wersją MySQL jest wersja 4.1 lub wyższa.

## 6.2. Kompilacja *Archiwum*

W celu skompilowania aplikacji należy uruchomić odpowiednie komendy poprzez program *ant* na pliku *build.xml* (*ant* domyślnie pobiera komendy z pliku *build.xml*, w związku z czym nie jest konieczne ręczne wskazywanie tego pliku). Podczas kompilacji wykorzystywany jest także plik *build.properties* (znajdujący się w tym samym katalogu co *build.xml*), który powinien zawierać co najmniej jedną własność: *tomcat.home*, której wartością jest ścieżka do katalogu, w którym został zainstalowany Apache Tomcat. Dostępne są następujące komendy:

- *ant compile* — kompiluje pliki *\*.java* i wygenerowane pliki *\*.class* umieszcza w katalogu *build*,
- *ant war* (zależy od *compile*) — tworzy plik *Archiwum.war* (czyli gotowy plik z aplikacją, który może być wgrany do Tomcata) zawierający skompilowane klasy, strony JSP i pozostałe potrzebne pliki. Utworzony plik *Archiwum.war* także jest umieszczany w katalogu *build*,
- *ant deploy* (zależy od *war*) — utworzony plik *Archiwum.war* przegrywa do Tomcata (do katalogu *tomcat.home/webapps*),
- *ant clean* — usuwa katalog *build*,
- *ant javadoc* — generuje dokumentację javadoc dla wszystkich klas z *Archiwum* i umieszcza w katalogu *doc*.

## 6.3. Instalacja *Archiwum*

W celu zainstalowania *Archiwum* należy (zakładając, że odpowiednie programy z punktu 6.1 zostały zainstalowane):

- Przegrać plik *Archiwum.war* do katalogu *tomcat.home/webapps*, gdzie *tomcat.home* to katalog, w którym zainstalowano Tomcata.
- W pliku *tomcat.home/conf/server.xml* utworzyć kontekst przeznaczony dla aplikacji *Archiwum* i dodać w nim parametr określający nazwę katalogu domowego aplikacji (wewnątrz elementu *Host*). Przykładowa konfiguracja aplikacji w pliku *server.xml* to:

```
<Context path="/Archiwum" docBase="Archiwum.war" debug="0"
  reloadable="false">
  <Logger className="org.apache.catalina.logger.FileLogger"
    directory="logs" prefix="archiwum." suffix=".txt."
    timestamp="true"/>
  <Paramater name="homeDirectory" value="/home/Archiwum"
    override="false"/>
</Context>
```

- Utworzyć katalog domowy aplikacji (o takiej ścieżce jaka została podana w pliku *server.xml*), w którym muszą znajdować się:
  - katalog *logs* — będą w nim umieszczane wszystkie logi z *Archiwum*,
  - katalog *Pliki* — będą w nim umieszczane pliki umieszczane w *Archiwum*,

- plik *datasource.config* — zawiera informacje o połączeniu z bazą danych, powinny się w nim znajdować definicje 4 własności:
  - \* *database.type* — oznacza rodzaj bazy danych (obecnie dostępne tylko MySQL — należy wpisać wartość *mysql*),
  - \* *username* — nazwa użytkownika w bazie,
  - \* *password* — hasło dla użytkownika *username* w bazie,
  - \* *url* — adres do połączenia z bazą.

Przykładowa zawartość tego pliku wygląda następująco:

```
database.type=mysql
username=mm209268
password=mm209268
url=jdbc:mysql://localhost/mm209268
```

- plik *log4j.properties* — zawiera plik konfiguracyjny dla biblioteki log4j używanej do generowania logów (nie jest ten plik wymagany — wówczas żadne logi nie będą generowane przez *Archiwum*).

Przykładowy katalog domowy znajduje się razem ze źródłami *Archiwum* na dołączonej płycie CD — zamiast tworzyć od nowa wystarczy go skopiować w odpowiednie miejsce i zmienić tylko parametry połączenia z bazą danych.

- Wykonać na bazie skryptu *archiwum\_create.sql* tworzący strukturę odpowiednią strukturę danych wymaganą do działania *Archiwum*.
- Uruchomić aplikację Migrator (dla pliku konfiguracyjnego z dodatku C).

## 6.4. Uruchomienie *Archiwum*

Po prawidłowej instalacji wystarczy już tylko włączyć Tomcata — poprzez uruchomienie skryptu *startup.sh* (dla Linuksa) lub *startup.bat* (dla Windows). Te skrypty znajdują się w katalogu *tomcat.home/bin*. Aby sprawdzić działanie systemu, należy otworzyć przeglądarkę internetową i wpisać adres serwisu.

W przypadku, gdy instalacja nie przebiegła poprawnie, uruchomienie nie powiedzie się — na ekranie zostanie wypisany komunikat błędu (odpowiedni wyjątek Javy). Należy wówczas wyłączyć Tomcata — wykonywane jest to w zależności od systemu operacyjnego poprzez polecenie *shutdown.sh* lub *shutdown.bat*, postarać się usunąć błąd i ponownie uruchomić Tomcata. Wszystkie logi (informacje o błędach) można zawsze sprawdzić w pliku *logs/debug.log* umieszczonym w katalogu domowym aplikacji.

Domyślnie Tomcat jest uruchamiany na porcie 8080, jednak jeśli zajdzie taka potrzeba (np. ten port jest już przez kogoś innego używany) może zostać zmieniony w pliku *tomcat.home/conf/server.xml*.





# Rozdział 7

## Podręcznik użytkownika

Rozdział zawiera opis interfejsu programu i jego działania.

### 7.1. Interfejs zwykłego użytkownika

#### 7.1.1. Rozpoczęcie pracy z systemem

Aby rozpocząć pracę, należy otworzyć przeglądarkę internetową i wpisać adres serwisu. Wówczas powinna wyświetlić się strona logowania, wyglądająca tak jak na rys. 7.1. Należy wpisać swój identyfikator i hasło i nacisnąć przycisk *ZALOGUJ*. Jeśli wpisano poprawne dane, zostanie wyświetlona strona powitalna (taka jak na rys. 7.2, wpp. zostanie wyświetlony komunikat o błędnych danych).

Z menu po lewej stronie mamy do wyboru 5 pozycji:

- *STRONA POWITALNA* — przejście na stronę powitalną,
- *NOWY DOKUMENT* — przejście do ekranu dodawania nowego dokumentu do *Archiwum*,
- *MOJE DOKUMENTY* — przeglądanie swoich dokumentów, czyli dokumentów, które użytkownik dodał do archiwum oraz dokumentów, które są dedykowane temu użytkownikowi,
- *WYSZUKIWARKA* — wyszukiwarka dokumentów w *Archiwum*,
- *PRZEGLĄDANIE ARCHIWUM* — przeglądanie dokumentów w *Archiwum* po strukturze katalogów.

Na każdej stronie można także kliknąć we flagę w prawym górnym rogu, co spowoduje zmianę aktualnego języka. W sytuacji, gdy strona wyświetlana jest w języku polskim, pokazywana jest flaga brytyjska — kliknięcie na nią spowoduje przeładowanie strony i zmianę języka na angielski. Natomiast, gdy strona wyświetlana jest w języku angielskim, pokazywana jest flaga polska, a kliknięcie na nią powoduje przeładowanie strony i zmianę języka na polski. Na każdej stronie w prawym górnym rogu dostępny jest odnośnik *pomoc* pozwalający wyświetlić pomoc dotyczącą aktualnie wyświetlonej strony na ekranie.

#### 7.1.2. Nowy dokument

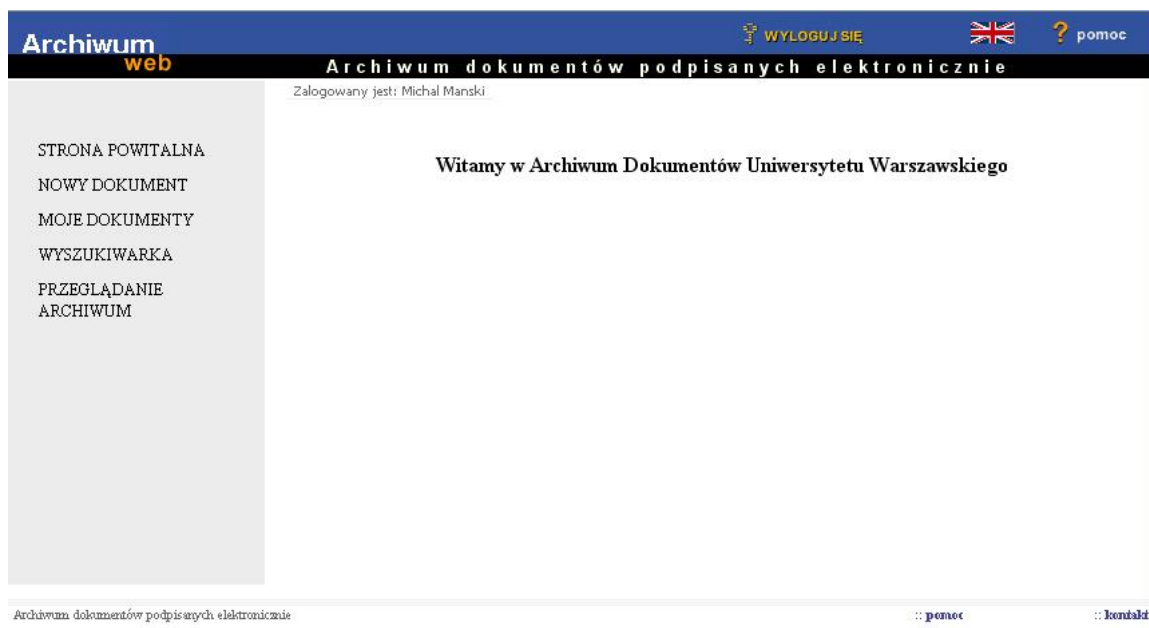
Po kliknięciu na pozycję w menu *NOWY DOKUMENT* użytkownik zostaje przekierowany do ekranu dodawania dokumentu (por. rys. 7.3). W celu dodania dokumentu należy uzupełnić pola (atrybuty) opisujące wstawiany dokument. Gdy obok nazwy pola wyświetlona jest czerwona gwiazdka, oznacza to, że dane pole jest obowiązkowe i niemożliwe będzie dodanie dokumentu bez ustalenia wartości



Copyright © 2005-2006 JA-SIG. All rights reserved.  
[Home Page](#) | [Wiki](#) | [Issue Tracker](#) | [Mailing Lists](#)

Powered by **JA-SIG CAS 3.0.7**

Rysunek 7.1: Logowanie do *Archiwum*



Rysunek 7.2: Strona powitalna *Archiwum*

tego pola. Na początku wybierana jest jednostka organizacyjna, z którą ma być związany tworzony dokument, w zależności od wybranej jednostki możemy mieć dostępne różne dokumenty do utworzenia (dzieje się tak w sytuacji, gdy użytkownik jest związany z więcej niż jednym wydziałem i w każdym z nich pełni inne role lub funkcje). W przypadku, gdy w ramach wybranej jednostki użytkownik nie ma prawa do dodawania żadnych dokumentów, na górze ekranu pojawi się komunikat „Nie masz uprawnień do tworzenia żadnych dokumentów w tej jednostce” i nie będzie możliwości dodania dokumentu żadnego typu. Posiadanie danej roli w jednostce X nie daje uprawnień do dodawania dokumentów dla podjednostki w jednostce X (aby posiadać takie uprawnienie, należy także w tej podjednostce posiadać odpowiednią rolę).

Dalej należy wybrać *rok akademicki* oraz podać opis dokumentu w języku polskim i angielskim, przy czym zamiast wpisywać ręcznie, można zaznaczyć, żeby opisy zostały wygenerowane automatycznie na podstawie pozostałych atrybutów dokumentu. Kolejnym polem do wyboru jest *typ dokumentu* — w zależności od wyboru typu dokumentu, na ekranie pojawią się inne pola opisujące dodawany dokument (na rys. 7.3 wybrany jest typ dokumentu *Dyplom* i w związku z tym widoczne są pola do wypełnienia związane z dyplomem).

W zależności od typu danego pola ustalenie jego wartości odbywa się w inny sposób. Dla pól tekstowych lub liczbowych, wartość jest wpisywana przez użytkownika ręcznie, dla pól, które są datami, wartość można zostać wpisana ręcznie lub może zostać wybrana z kalendarza (daty są zapisywane w formacie DD.MM.YYYY). Dla niektórych pól wartość jest ustalana poprzez wybór jednej wartości z listy wyboru. Natomiast dla bardziej skomplikowanych atrybutów ich wartość jest ustalana poprzez wybór z odpowiedniej wyszukiwarki. Przykładowo dla dokumentu, który jest dyplomem, atrybuty z wyszukiwarki ustala się dla pól *Student* i *Kierunek studiów*.

W celu otwarcia wyszukiwarki należy nacisnąć przycisk **WYBIERZ**. Otwarte zostanie nowe okienko, gdzie można ustalić kryteria wyszukiwania. Wszystkie wyszukiwarki wyglądają podobnie (jedna z nich — wyszukiwarka cykli dydaktycznych — została przedstawiona na rys. 7.4). Po odpowiednim ustaleniu warunków wyszukiwania, należy wcisnąć przycisk **SZUKAJ** — wówczas na ekranie pojawią się wyniki wyszukiwania według wybranych kryteriów, które można sortować po poszczególnych atrybutach. Na jednej stronie pojawia się 10 wyników, gdy liczba znalezionych wyników jest większa możliwe jest poruszanie się pomiędzy kolejnymi stronami wyników. W celu wybrania danego wyszukanego obiektu jako wartości atrybutu należy nacisnąć na odpowiedni odnośnik *Wybierz*. Dostępny jest także przycisk **NOWE WYSZUKIWANIE**, który pozwala na rozpoczęcie procesu wyszukiwania od nowa. W każdym momencie można zamknąć wyszukiwarę poprzez wcisnięcie w lewym górnym rogu odnośnika *zamknij*. Po kliknięciu *Wybierz*, okienko z wyszukiwarą jest zamykane, a wybrany obiekt zostaje wpisany w odpowiednie pole na formularzu dodawania nowego dokumentu. W celu anulowania wyboru należy wcisnąć przycisk **WYCZYŚĆ** — wówczas wartość danego pola zostanie wyczyszczona.

Dla wyszukiwarek takich danych jak studenci, pracownicy oraz przedmioty, które są ściśle związane z poszczególnymi jednostkami organizacyjnymi, wyniki wyszukiwania są ograniczane do wybranej dla danego dokumentu jednostki wraz z podjednostkami (zatem możliwe jest na przykład dodanie dokumentu dla Uniwersytetu Warszawskiego i przedmiotu wykładanego na Wydziale MIM).

Po ustaleniu wartości poszczególnych atrybutów, należy wybrać plik z dokumentem, który zostanie załadowany do *Archiwum* (można wpisać ręcznie jego lokalizację lub skorzystać z przycisku *Przeglądaj*, który pozwoli na wygodne wybranie pliku za pomocą graficznego interfejsu). Dla pliku należy także wskazać czy jest on podpisany za pomocą podpisu PGP, podpisu X.509, czy w ogóle nie jest podpisany podpisem elektronicznym. Jeśli dodawany plik jest podpisany, to podpis musi należeć do użytkownika, który ten plik umieszcza, i być wykonany za pomocą certyfikatu zarejestrowanego w *Archiwum*. W przypadku, gdy ten warunek nie jest spełniony wyświetlony zostanie komunikat informujący, że podany plik nie został poprawnie zweryfikowany i nie będzie możliwe umieszczenie w *Archiwum* takiego pliku. W celu zarejestrowania nowego certyfikatu, użytkownik powinien

## Nowy dokument

Jednostka organizacyjna*	Wydział Matematyki, Informatyki i Mechaniki ▾
Rok akademicki*	2007/2008 ▾
Opis (polski)*	<input type="text"/> <input type="checkbox"/> wygeneruj automatycznie
Opis (angielski)*	<input type="text"/> <input checked="" type="checkbox"/> wygeneruj automatycznie
Typ dokumentu	Dyplom ▾
Typ*	-- wybierz -- ▾
Typ dyplomu*	-- wybierz -- ▾
Student*	Imię <input type="text"/> Nazwisko <input type="text"/> WYBIERZ WYCZYŚĆ
Tytuł pracy*	<input type="text"/>
Kierunek studiów*	Kod <input type="text"/> Nazwa <input type="text"/> WYBIERZ WYCZYŚĆ
Plik*	<input type="text"/> Przeglądaj... bez podpisu ▾

### Załączniki

Tytuł*	Plik*	Typ podpisu	
<input type="text"/>	<input type="text"/> Przeglądaj...	brak ▾	USUŃ
DODAJ ZAŁĄCZNIK			

UTWÓRZ

Rysunek 7.3: Dodawanie nowego dokumentu

## Słownik cykli dydaktycznych

Kod	<input type="text"/>
Nazwa	<input type="text"/>
Początek	<input type="text"/>  <input type="text"/> 
Koniec	<input type="text"/>  <input type="text"/> 
<input type="button" value="SZUKAJ"/> <input type="button" value="WYCZYŚĆ"/>	

Rysunek 7.4: Wyszukiwarka cykli dydaktycznych

zgłosić się z odpowiednią prośbą do administratora.

Możliwe jest także umieszczenie w *Archiwum* oprócz podstawowego pliku z dokumentem, dowolnej liczby dodatkowych załączników. W tym celu należy kliknąć na przycisk *DODAJ ZAŁĄCZNIK*, co spowoduje pojawienie się pól do wybrania nowego załącznika. Należy wskazać tytuł załącznika oraz wybrać plik i typ podpisu (analogicznie jak podczas wybierania podstawowego pliku z dokumentem). Przycisk *USUŃ* pozwala usunąć dodany załącznik.

Gdy już wszystkie pola formularza zostały uzupełnione, należy wcisnąć przycisk *UTWÓRZ* na samym dole formularza, co spowoduje dodanie nowego dokumentu. W przypadku, gdy użytkownik zapomni wybrać wartości dla pewnego pola, które jest obowiązkowe, zostanie wyświetlony stosowny komunikat bezpośrednio nad przyciskiem *UTWÓRZ* i niemożliwe będzie dodanie dokumentu, zanim nie zostanie podana wartość tego pola.

Gdy dodawanie dokumentu zakończy się pomyślnie, użytkownik zostanie przekierowany na ekran *MOJE DOKUMENTY*, wpp. nastąpi powrót do ekranu dodawania dokumentu i na górze ekranu pojawi się stosowny komunikat informujący o błędzie, który uniemożliwia dodanie dokumentu.

### 7.1.3. Moje dokumenty

Na ekranie *MOJE DOKUMENTY* użytkownik ma do wyboru dwie zakładki: *DODANE PRZEZE MNIE* i *DEDYKOWANE MI* (domyślnie jest wyświetlana ta pierwsza) — odnośnik do tej, która jest aktywna, wyświetlana jest w kolorze brązowym. Pierwsza zakładka pozwala przeglądać dokumenty, które zostały dodane przez użytkownika, a druga pozwala przeglądać dokumenty, które są w pewien sposób z nami związane (i dzięki tej zakładce użytkownik ma do nich szybki dostęp), np. dokument, który jest pracą dyplomową, jest dedykowany studentom, którzy są autorami pracy oraz opiekunom tej pracy. Na każdy z wyświetlonych dokumentów można wejść klikając na odnośnik zawierający opis dokumentu. W przypadku, gdy użytkownik ma prawo do edycji wybranego dokumentu, zostanie przekierowany do ekranu edycji tego dokumentu, wpp. zostanie przekierowany do ekranu podglądu tego dokumentu. Rysunek 7.5 pokazuje przykładową zawartość zakładki *DODANE PRZEZE MNIE*.

### 7.1.4. Wyszukiwarka

W celu przejścia do wyszukiwarki dokumentów (rys. 7.6) należy wybrać opcję *WYSZUKIWARKA* w lewym menu. Możemy tam ustalić różne kryteria, według których będą wyszukiwane dokumenty. Nie ustalenie żadnych kryteriów spowoduje wyszukanie wszystkich dokumentów, do których prawo

## Moje dokumenty

DODANE PRZEZE MNIE • DEDYKOWANE MI

Opis ▲ ▼	Jednostka organizacyjna ▲ ▼	Dodany przez ▲ ▼	Data dodania ▲ ▼
Praca Licencjacka (Ewa1021 Kowalska1021, Michal Manski, Matematyka)	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	15.05.2007 08:58
Praca Licencjacka (Ewa100 Kowalska100, Informatyka)	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	08.05.2007 00:05
Dyplom (Magisterium, Ewa1021 Kowalska1021)	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	08.05.2007 00:03
Dyplom (Licencjat, Ewa1022 Kowalska1022)	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	07.05.2007 23:35

Rysunek 7.5: Ekran *Moje dokumenty*

odczytu posiada użytkownik. W przypadku określenia pewnych kryteriów ten zbiór wyników będzie odpowiednio ograniczony. W polu *Jednostka organizacyjna* można podać fragment tekstu, który musi zawierać nazwa jednostki w wyszukanych dokumentach. Drugim kryterium jest *Opis* — tutaj, także podajemy fragment tekstu, który ma być zawarty w opisie wyszukanych dokumentów. Dla obu tych kryteriów w zależności od ustawionego języka, wyszukiwanie będzie odbywało się po polskich lub angielskich nazwach jednostek oraz opisie w języku polskim lub angielskim.

Kolejnym kryterium jest *Data dodania*, pozwalającym wyszukać dokumenty dodane w danym przedziale czasowym (możliwe jest wybranie tylko jednej z tych dwóch dat). Ponadto można wyszukiwać według osoby, który dodała dokument. Można ręcznie wpisać fragment imienia i nazwiska, które odpowiednio ograniczą zbiór wyników, lub wybrać osobę ze słownika — wówczas zostaną wyszukane tylko te dokumenty, które zostały umieszczone przez tę osobę.

Ostatnim kryterium jest *Typ dokumentu* w postaci listy wielokrotnego wyboru — pozwala wyszukać dokumenty ściśle określonych typów.

Po naciśnięciu przycisku *SZUKAJ* zostaną wyświetlone odpowiednie wyniki (por. rys. 7.7). W sytuacji, gdy nie istnieje żaden dokument spełniający podane kryteria, zamiast wyników pojawi się ponownie ekran wyszukiwarki z odpowiednim komunikatem. Dla każdego z wyszukanych dokumentów można zobaczyć dokładne informacje o nim naciskając na odpowiedni odnośnik (w pierwszej kolumnie wyników) zawierający opis dokumentu. W przypadku posiadania praw edycji wybranego dokumentu, użytkownik zostanie przekierowany do ekranu edycji dokumentu, wpp. zostanie wyświetlony ekran zwykłego podglądu dokumentu.

### 7.1.5. Przeglądanie archiwum

Po wybraniu w lewym menu opcji *PRZEGLĄDANIE ARCHIWUM* użytkownik będzie miał możliwość poruszania się po strukturze katalogowej i przeglądania dokumentów znajdujących się w poszczególnych katalogach (por. rys. 7.8). Dokumenty są katalogowane według następującej zasady: dokument X jest umieszczany w katalogu *Główny/\$KATEGORIA/\$TYP/\$JEDNOSTKA/\$ROK*, gdzie:

- *\$TYP* to wartość atrybutu *Typ dokumentu* dokumentu X,
- *\$KATEGORIA* to nazwa kategorii dokumentów, do której należy typ dokumentu o nazwie *\$TYP* (w przypadku, gdy *\$KATEGORIA* jest podkategorią innej kategorii, podana ścieżka katalogów zostanie w naturalny sposób rozszerzona),

## Wyszukiwarka

Jednostka organizacyjna	<input type="text" value="Matematyki"/>
Opis	<input type="text"/>
Rok akademicki	<input type="text" value="2007/08"/> <input type="text" value="2006/07"/> <input type="text" value="2005/06"/>
Data dodania	<input type="text"/> <input type="text"/>
Osoba dodająca	Imię <input type="text"/> Nazwisko <input type="text"/> <input type="button" value="WYBIERZ"/> <input type="button" value="WYCZYŚĆ"/>
Typ dokumentu	<input type="text" value="Dyplom"/> <input type="text" value="Karta przebiegu studiów"/> <input type="text" value="Praca Dyplomowa"/> <input type="text" value="Podanie do dziekana"/> <input type="text" value="Rozliczenie pensum"/>
<input type="button" value="SZUKAJ"/>	

Rysunek 7.6: Wyszukiwarka dokumentów

## Wyszukiwarka

<input type="button" value="◀"/> <input type="button" value="◀◀"/> Wyniki 1..5 z 5 <input type="button" value="▶▶"/> <input type="button" value="▶"/>			
Opis ▲▼	Jednostka organizacyjna ▲▼	Dodany przez ▲▼	Data dodania ▲▼
<a href="#">Praca Magisterska (Jan10 Kowalski10, Informatyka)</a>	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	05.06.2007 23:44
<a href="#">Praca Licencjacka (Ewa1054 Kowalska1054, Matematyka)</a>	Wydział Matematyki, Informatyki i Mechaniki	Administrator Wydziału MIM	05.06.2007 23:41
<a href="#">Dyplom (Licencjat, Ewa1022 Kowalska1022)</a>	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	03.06.2007 23:30
<a href="#">Praca Magisterska (Ewa1021 Kowalska1021, Analityka)</a>	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	03.06.2007 01:02
<a href="#">Praca Magisterska (Ewa1019 Kowalska1019, Archeologia)</a>	Wydział Matematyki, Informatyki i Mechaniki	Pracownik Dziekanatu	02.06.2007 23:41

Rysunek 7.7: Wyniki wyszukiwarki dokumentów

## Przeglądanie archiwum

The screenshot displays a web interface for document archiving. On the left, a table lists documents with columns for description, author, and date. On the right, a tree view shows the folder structure, including 'Dyplomy' and 'Studia'.

Opis	Dodany przez	Data dodania
Praca Magisterska (Jan10 Kowalski10, Informatyka)	Pracownik Dziekanatu	05.06.2007 23:44
Praca Licencjacka (Ewa1054 Kowalska1054, Matematyka)	Administrator Wydziału MIM	05.06.2007 23:41
Praca Magisterska (Ewa1019 Kowalska1019, Archeologia)	Pracownik Dziekanatu	02.06.2007 23:41

Navigation tree structure:

- Główny
  - Dyplomy
    - Dyplom
    - Praca Dyplomowa
      - Wydział Matematyki, Informatyki i Mechaniki
        - 2005/06
        - 2006/07
  - Studia
    - Albumy
    - Egzaminy

Rysunek 7.8: Przeglądanie *Archiwum*

- $\$JEDNOSTKA$  to wartość atrybutu *Jednostka organizacyjna* dokumentu  $X$ ,
- $\$ROK$  to wartość atrybutu *Rok akademicki* dokumentu  $X$ .

Po lewej stronie wyświetlana jest lista dokumentów, a po prawej rozwijalne drzewo. Aby rozwinąć daną gałąź drzewa należy kliknąć na odpowiedni przycisk oznaczony symbolem „+”, aby gałąź schować należy kliknąć na przycisk z symbolem „-”. Po kliknięciu na daną nazwę katalogu, po lewej stronie pojawi się odpowiednia lista dokumentów. Po kliknięciu na opis danego dokumentu użytkownik zostanie przeniesiony do ekranu edycji dokumentu lub ekranu podglądu dokumentu w zależności od posiadanych uprawnień.

### 7.1.6. Podgląd dokumentu

Na ekranie podglądu dokumentu, użytkownik może obejrzeć wartości metadanych przypisanych do danego dokumentu (por. rys. 7.9). Ponadto w dolnej części ekranu wyświetlane są informacje o głównym pliku tego dokumentu oraz dodatkowych załącznikach (o ile takie istnieją). Po kliknięciu na odnośnik *Podgląd* obok danego załącznika, użytkownik zostanie przeniesiony do ekranu podglądu tego załącznika. W przypadku posiadania prawa do pobierania plików widoczne będą także odnośniki *Pobierz*. Po kliknięciu w taki odnośnik, użytkownik będzie miał możliwość pobrania odpowiedniego pliku.

W pewnych sytuacjach może się okazać, że nie będzie możliwe poprawne wyświetlenie informacji o wszystkich metadanych danego dokumentu — pojawi się wówczas następujący komunikat: „Niektóre wartości atrybutów dokumentu nie mogą zostać odczytane”. Nie jest to błąd aplikacji, a jedynie błąd wynikający z niespójności bazy danych na skutek usunięcia z bazy USOS pewnych informacji. Takie przypadki zdarzają się jednak niezmiernie rzadko i można je ignorować.

### 7.1.7. Edycja dokumentu

Na ekranie edycji dokumentu (rys. 7.10) użytkownik może obejrzeć wartości metadanych przypisanych do danego dokumentu (tak jak na ekranie podglądu dokumentu) oraz dodatkowo zmienić wartości tych metadanych (w analogiczny sposób jak na ekranie dodawania nowego dokumentu). Po naciśnięciu przycisku *ZAPISZ ZMIANY* wprowadzone zmiany w odpowiednich polach zostaną zapisane do bazy. Flaga *wygeneruj automatycznie* jest domyślnie wyłączona zarówno dla opisu w języku polskim, jak i opisu w języku angielskim. Zatem chcąc, aby po zapisaniu zmian opisy automatycznie się zaktualizowały należy zaznaczyć te flagi.



## Podgląd dokumentu

Jednostka organizacyjna	Wydział Matematyki, Informatyki i Mechaniki		
Typ dokumentu	Praca Dyplomowa		
Dodany przez	Pracownik Dziekanatu		
Data dodania	05.06.2007 23:44		
Rok akademicki	2006/07		
Opis	Praca Magisterska (Jan10 Kowalski10, Informatyka)		
Typ dyplomu	Praca Magisterska		
Studenci	<input type="text" value="Jan10 Kowalski10"/> <input type="text" value="Ewa102445 Kowalska102445"/>		
Kierujący pracą	<input type="text" value="Julian Kalinowski"/>		
Kierunek studiów	Kod	<input type="text" value="IN"/>	
	Nazwa	<input type="text" value="Informatyka"/>	
Praca	Tytuł pracy	<input type="text" value="Adaptacja weryfikatora ESC/ Java2 na p"/>	
	Data złożenia	<input type="text" value="28.09.06"/>	
	Data zatwierdzenia	<input type="text" value="13.09.04"/>	

### Plik

Wersja	Nazwa pliku	Autor	Podpis	Rozmiar	Data	
1	praca.pdf.sig	Pracownik Dziekanatu	X.509	114302	05.06.2007	<a href="#">Podgląd</a>

Rysunek 7.9: Podgląd dokumentu

W dolnej części ekranu wyświetlone są informacje o głównym pliku oraz dodatkowych załącznikach zawartych w tym dokumencie (wyświetlane są tylko informacje o najnowszych wersjach plików). W przypadku posiadania prawa do pobierania plików widoczny będzie odnośnik *Pobierz* obok podstawowego pliku i wszystkich załączników. Po kliknięciu w ten odnośnik, użytkownik będzie miał możliwość pobrania odpowiedniego pliku.

Po kliknięciu na odnośnik *Podgląd* obok danego załącznika, użytkownik zostanie przeniesiony do ekranu podglądu tego załącznika.

W przypadku posiadania prawa do modyfikacji załączników, będzie możliwe dodanie nowych załączników. Aby to zrobić należy wcisnąć przycisk *DODAJ ZAŁĄCZNIK* oraz wskazać plik i podać odpowiedni tytuł i typ podpisu. Przycisk *USUŃ* powoduje usunięcie odpowiedniego załącznika z listy. W momencie naciśnięcia przycisku *ZAPISZ ZAŁĄCZNIKI* wybrane załączniki zostaną wysłane i zapisane w *Archiwum*.

### 7.1.8. Podgląd załącznika dokumentu

Na ekranie podglądu załącznika można obejrzeć dokładne informacje o wszystkich wersjach danego załącznika oraz (o ile posiada się uprawnienie do modyfikacji załączników tego dokumentu) dodać nową wersję załącznika ((por. rys. 7.11).

Dla wszystkich wersji jest wyświetlana m.in. informacja o typie podpisu, którym został podpisany ten dokument (ewentualnie informacja o braku podpisu). Można także zobaczyć szczegóły podpisu — po kliknięciu na odnośnik *Pokaż* pojawiają się dodatkowe informacje o podpisie i certyfikacie podpisującego (por. rys. 7.12). Kliknięcie na odnośnik *Schowaj* spowoduje ponowne ukrycie tych informacji. Istnieje także możliwość obejrzenia szczegółów certyfikatu poprzez odnośnik *Zobacz certyfikat* (w zależności od typu podpisu wyświetlone zostanie okienko opisujące odpowiedni certyfikat X.509 lub klucz publiczny PGP) lub w przypadku, gdy jest to certyfikat X.509 także pobrania go poprzez odnośnik *Pobierz certyfikat*.

Uprawnieni użytkownicy mogą pobrać poszczególne pliki. W sytuacji, gdy dany plik nie jest podpisany pojawia się odnośnik *Pobierz*, natomiast, gdy plik jest podpisany, pojawiają się dwa odnośniki *Pobierz z podpisem* oraz *Pobierz bez podpisu*. Ten pierwszy oznacza pobranie pliku w takiej formie w jakiej został dodany do *Archiwum*, ten drugi natomiast pozwala pobrać oryginalny plik bez podpisu (czyli taki, którego zawartość można obejrzeć w odpowiednim programie, np. plik *.doc* w programie Microsoft Word, gdyż zawartości podpisanego pliku, np. z rozszerzeniem *.sig*, przeważnie nie da się odczytać w żadnym programie).

Dla dodatkowych załączników można także zmienić ich tytuł oraz można usunąć całkowicie dany załącznik poprzez przycisk *USUŃ ZAŁĄCZNIK* (jak ma się prawa modyfikacji tego załącznika).

### 7.1.9. Podgląd klucza publicznego PGP

Na ekranie podglądu klucza publicznego PGP (rys. 7.13) wyświetlana jest informacja, na kogo jest ten klucz zarejestrowany oraz podstawowe informacje o tym kluczu, takie jak czas utworzenia klucza, odcisk palca i cały klucz publiczny PGP zakodowany w Base64.

### 7.1.10. Podgląd certyfikatu X.509

Na ekranie podglądu certyfikatu X.509 (rys. 7.14) wyświetlana jest informacja na kogo jest ten certyfikat zarejestrowany oraz podstawowe informacje o tym certyfikacie takie jak na kogo został wystawiony, wystawca certyfikatu czy okres ważności certyfikatu. Ponadto możliwe jest pobranie tego certyfikatu poprzez odnośnik *Pobierz Certyfikat*.

## Edycja dokumentu

Jednostka organizacyjna	Wydział Matematyki, Informatyki i Mechaniki		
Typ dokumentu	Dyplom		
Dodany przez	Pracownik Dziekanatu		
Data dodania	03.06.2007 23:30		
Rok akademicki*	2007/08		
Opis (polski)*	Dyplom (Licencjat, Ewa1022 Kowalska1022) <input type="checkbox"/> wygeneruj automatycznie		
Opis (angielski)*	Diploma (Bachelor, Ewa1022 Kowalska1022) <input type="checkbox"/> wygeneruj automatycznie		
Typ*	Dyplom		
Typ dyplomu*	Licencjat		
Student*	Imię	Ewa1022	
	Nazwisko	Kowalska1022	
		<input type="button" value="WYBIERZ"/> <input type="button" value="WYCZYŚĆ"/>	
Tytuł pracy*	Analiza statyczna rozszerzonych automatów czasowych		
Kierunek studiów*	Kod	MT	
	Nazwa	Matematyka	
		<input type="button" value="WYBIERZ"/> <input type="button" value="WYCZYŚĆ"/>	

### Plik

Wersja	Nazwa pliku	Autor	Podpis	Rozmiar	Data		
2	plik.pdf.pgp	Pracownik Dziekanatu	PGP	105950	06.06.2007	<a href="#">Pobierz</a>	<a href="#">Podgląd</a>

### Załączniki

Rysunek 7.10: Edycja dokumentu

## Podgląd załącznika

Typ:	Główny plik
Dodany przez:	Pracownik Dziekanatu

[POWRÓT DO DOKUMENTU](#)

### Wersje pliku

Wersja	1
Data włożenia	08.05.2007 00:04
Nazwa pliku	ref2.pdf.sig
Dodany przez	Pracownik Dziekanatu
Typ podpisu	X.509
Szczegóły podpisu	<a href="#">Pokaż</a>
Rozmiar	342551
	<a href="#">Pobierz z podpisem</a> <a href="#">Pobierz bez podpisu</a>

### Nowa wersja pliku

<input type="text"/>	<input type="button" value="Przełączaj..."/>	<input type="button" value="bez podpisu"/> <input type="button" value="v"/>
----------------------	--	---

[DODAJ](#)

Rysunek 7.11: Podgląd załącznika

Szczegóły podpisu	<a href="#">Schowaj</a>																
	<table><tr><td>Podpisujący</td><td>Pracownik Dziekanatu</td></tr><tr><td colspan="2">Informacje o certyfikacie podpisującego:</td></tr><tr><td>Podmiot</td><td>Michał Mański</td></tr><tr><td>Wystawca</td><td>CERTUM QCA</td></tr><tr><td>Ważny od</td><td>19.10.2006 15:48</td></tr><tr><td>Ważny do</td><td>18.10.2008 15:48</td></tr><tr><td colspan="2"><a href="#">Zobacz certyfikat</a></td></tr><tr><td colspan="2"><a href="#">Pobierz certyfikat</a></td></tr></table>	Podpisujący	Pracownik Dziekanatu	Informacje o certyfikacie podpisującego:		Podmiot	Michał Mański	Wystawca	CERTUM QCA	Ważny od	19.10.2006 15:48	Ważny do	18.10.2008 15:48	<a href="#">Zobacz certyfikat</a>		<a href="#">Pobierz certyfikat</a>	
Podpisujący	Pracownik Dziekanatu																
Informacje o certyfikacie podpisującego:																	
Podmiot	Michał Mański																
Wystawca	CERTUM QCA																
Ważny od	19.10.2006 15:48																
Ważny do	18.10.2008 15:48																
<a href="#">Zobacz certyfikat</a>																	
<a href="#">Pobierz certyfikat</a>																	

Rysunek 7.12: Szczegóły podpisu załącznika

## Publiczny Klucz PGP

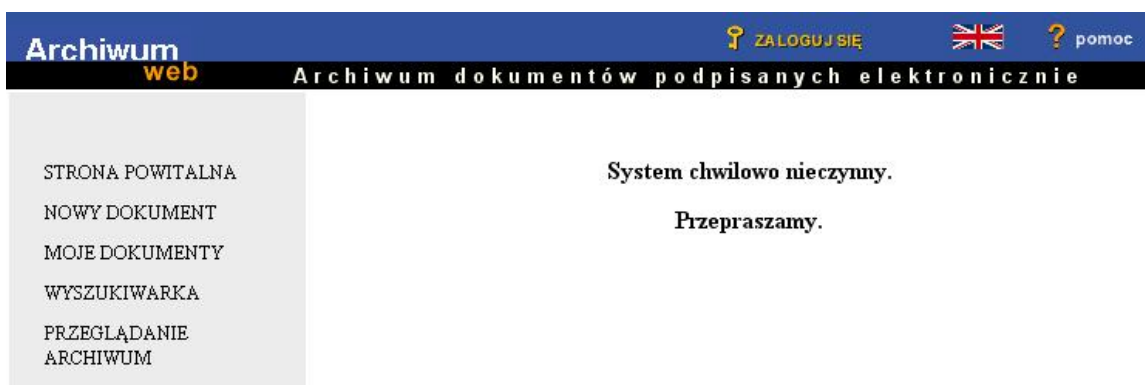
Zarejestrowany dla	Administrator Wydziału MIM
ID użytkownika	Michał Manski <kassini@interia.pl>
Algorytm	RSA
Czas utworzenia	03.04.07
Odcisk palca	246caf13d787ab2f711b24ae041da4de
Klucz zakodowany w Base64	<pre> -----BEGIN PGP PUBLIC KEY BLOCK----- Version: PGPfreeware 6.5.8 for non-commercial use &lt;http://www.pgp.com&gt;  mQENADYSg/YAAAEIAKI8mtJ3lCU1qDq4P+bJzuxgjaQNaaIeTWMBLIAjLmv3EKI8 i75gVYuAdPJvbnSIsQjYF+17AUV+KO3k0zXUQHNI2BdB+/uuIAkxm4ENivqV60YI mHMxkIB++CDDAEI7xUOKD1aWh7JbKlhcOb3qZLCgw1CEK3YuWnt6aD9UU6sHzvCc KN5ma105rFNwduiIdFJL7kCyWv+W+o1XOpAyzYIYHYUXAxfEssX9d1HpfNs0rfrc LAhSfGroOc7AMA9stYwdAT6pf+oFn+WuHaVAnNea9Tg0BYFO3RvWvOZokgAlDw9Uy QUWhqhO2IREDUPsiYxcXVbZ2v8VerWbtLPcIV1EABRG0Ik1pY2hhbCBNYW5za2kg PGthc3NpbmlAaW50ZXJpYS5wbD6JARUDBRBGEoP2rWbtLPcIV1EBAVAk8/9mjtK sbOFxyvuQUrdqMzeWzPmz1vhh1YPrwXhix9fgk4EvWCB/RLJsc6Un2CKrqDkRAF ARjC76jLTXsj53c0I9f/YvBSXin7se+wqW01A46TpkISj2oc17P/7in5IuHsv0z7 zY/M61oqxLzgyE5yobVWjwIRYBsmLUQMjNBg39Um0j+uYaGNdRdLIIdFy6KhIOUhn kxaIKjPz7PyXZtNrk6CbIXQxT5XL9lygmNy/zn7XyAYKgDeKfzq67aFSnOL5npxv h3/4GD0efe6e/Ize337T0ddNJYtDPZPVKPLnEj+uIm+06DQWNYV12B86oNdyFFgq hf9UkjDsQFmkfx5W =bkfr -----END PGP PUBLIC KEY BLOCK----- </pre>

Rysunek 7.13: Podgląd klucza publicznego PGP

## Certyfikat X.509

Zarejestrowany dla	Pracownik Dziekanatu
Numer seryjny	8694
Algorytm podpisu	SHA1withRSA
Podmiot	C=PL, SERIALNUMBER=PESEL:83080808579, SURNAME=Mański, GIVENNAME=Michał, CN=Michał Mański
Wystawca	SERIALNUMBER=Nr wpisu: 1, CN=CERTUM QCA, O=Unizeto Technologies S.A., C=PL
Ważny od	19.10.2006 15:48
Ważny do	18.10.2008 15:48
	<a href="#">Pobierz certyfikat</a>

Rysunek 7.14: Podgląd certyfikatu X.509



Rysunek 7.15: Blokada systemu

### 7.1.11. Blokada systemu

W sytuacji, gdy aplikacja zostanie zablokowana przez administratora systemu, nie będzie możliwe wykonywanie żadnych funkcji w *Archiwum* aż do zakończenia planowanego czasu lub zdjęcia blokady przez administratora. Wówczas zawsze zamiast odpowiednich stron, będzie wyświetlany komunikat informujący o tej sytuacji (rys 7.15).

Dodatkowo na każdej stronie u góry ekranu przez 5 minut przed rozpoczęciem blokady systemu będzie wyświetlana informacja o planowanym zamknięciu serwisu. Osoby zalogowane powinny wówczas czym prędzej zakończyć swoje zadania i wylogować się.

## 7.2. Interfejs administratora jednostki

Administrator danej jednostki ma dostęp do wszystkich tych samych funkcji co zwykły użytkownik. Dodatkowo pojawia mu się w lewym menu pozycja *ADMINISTRACJA*. Po kliknięciu w tę pozycję następuje przekierowanie do panelu administratora. Do wyboru są tam 3 zakładki:

- *TYPY DOKUMENTÓW* — zarządzanie zarejestrowanymi w *Archiwum* typami dokumentów,
- *CERTYFIKATY* — zarządzanie certyfikatami użytkowników,
- *UŻYTKOWNICY* — przypisywanie użytkowników do ról.

W kolejnych punktach opisano funkcjonalności powyższych zakładek.

### 7.2.1. Typy dokumentów

W zakładce *TYPY DOKUMENTÓW* wyświetlana jest lista wszystkich zarejestrowanych w *Archiwum* typów dokumentów (rys. 7.16). Po kliknięciu na nazwę kodową danego typu dokumentu, użytkownik jest przekierowywany do ekranu edycji tego typu. Ponadto możliwe jest dodanie typu dokumentu poprzez wskazanie pliku XML opisującego nowy typ XML i kliknięcie przycisku *DODAJ* u dołu ekranu.

Gdy użytkownik wejdzie na ekran edycji danego typu dokumentu (rys. 7.17), może załadować zaktualizowany plik XML opisujący ten typ. Trzeba pamiętać, że jeśli aktualizacja wymaga zmiany struktury odpowiedniej tabeli w bazie to musi ona zostać ręcznie wykonana — system sam takiej zmiany nie dokona i w przypadku niewykonania ręcznie takiej aktualizacji mogą pojawić się niespodziewane błędy podczas działania *Archiwum*.

# Administracja

TYPY DOKUMENTÓW • CERTYFIKATY • UŻYTKOWNICY

## Istniejące typy dokumentów

Nazwa kodowa	Nazwa dla użytkownika	Tabela w bazie
decyzja_administracyjna	Decyzja administracyjna w sprawie studenta	ad_doc_decyzja_administracyjna
dyplom	Dyplom	ad_doc_dyplom
karta_egzam	Karta egzaminacyjna	ad_doc_karta_egzam
karta_przebiegu_studiow	Karta przebiegu studiów	ad_doc_karta_przebiegu_studiow
ksiega_albumow	Księga Albumów	ad_doc_ksiega_albumow
ksiega_dyplomow	Księga Dyplomów	ad_doc_ksiega_dyplomow
lista_stypendialna	Lista stypendialna	ad_doc_lista_stypendialna
podanie_do_dziekana	Podanie do dziekana	ad_doc_podanie_do_dziekana
praca_dyplomowa	Praca Dyplomowa	ad_doc_praca_dyplomowa
protokol_egzam	Protokół egzaminacyjny	ad_doc_protokol_egzam
rozliczenie_pensum	Rozliczenie pensum	ad_doc_rozliczenie_pensum

## Nowy typ dokumentu

Wybierz plik*	<input type="text"/>	Przełóżaj...
	DODAJ	

Rysunek 7.16: Zakładka *TYPY DOKUMENTÓW*

# Administracja

TYPY DOKUMENTÓW • CERTYFIKATY • UŻYTKOWNICY

## Edycja typu dokumentu

Nazwa kodowa	ksiega_albumow
Nazwa dla użytkownika	Księga Albumów
Tabela w bazie	ad_doc_ksiega_albumow
Nowy plik XML	<input type="text"/> Przełóżaj... AKTUALIZUJ
UWAGA:	Jeśli zaktualizowany plik XML wymaga dokonania zmiany w strukturze bazy danych, to należy wykonać ją ręcznie!

Rysunek 7.17: Edycja typu dokumentu

Na ekranie edycji typu możliwe jest także usunięcie danego typu poprzez przycisk *USUŃ* (o ile nie istnieje żaden dokument tego typu w *Archiwum*, wpp. podczas próby usunięcia pojawi się stosowny komunikat), po naciśnięciu go pojawi się komunikat z prośbą o potwierdzenie wykonania tej operacji. Po pomyślnym usunięciu typu nastąpi powrót do głównej strony zakładki *TYPY DOKUMENTÓW*.

### 7.2.2. Certyfikaty

Zakładka *CERTYFIKATY* służy do przeglądania i rejestrowania certyfikatów dla poszczególnych użytkowników (rys. 7.18). Na początku należy wybrać z wyszukiwarki osobę, dla której chcemy wyświetlić certyfikaty (wyszukiwarka otwiera się za pomocą przycisku *WYBIERZ*). Po wybraniu osoby z wyszukiwarki (czyli kliknięciu odnośnika *Wybierz* obok wybranej osoby) na ekranie pojawiają się certyfikaty X.509 oraz klucze publiczne PGP zarejestrowane dla tej osoby (w jednej tabelce pokazywane są informacje o certyfikatach X.509, a w innej o kluczach publicznych PGP — w przypadku braku takich kluczy lub certyfikatów odpowiednie tabelki nie zostaną w ogóle wyświetlone). Wyświetlają się także dwa formularze pozwalające zarejestrować dla tego użytkownika nowy certyfikat X.509 i klucz publiczny PGP.

W celu zarejestrowania klucza publicznego PGP należy w pole (umieszczone pod napisem *Nowy klucz publiczny:*) wpisać nowy klucz publiczny i nacisnąć znajdujący się obok przycisk *DODAJ*. W przypadku pomyślnego dodania tego klucza zostanie wyświetlony komunikat informujący o tym oraz w tabelce wyświetlającej klucze publiczne PGP zarejestrowane dla tej osoby pojawi się nowy wpis z właśnie dodanym kluczem. Natomiast w przypadku, gdy podano błędny klucz (tzn. *Archiwum* nie jest w stanie rozpoznać wpisanego tekstu jako klucz publiczny PGP), pojawi się stosowny komunikat w kolorze czerwonym.

W celu zarejestrowania certyfikatu X.509 należy w pole (umieszczone pod napisem *Nowy certyfikat:*) wpisać nazwę pliku (lub wybrać za pomocą graficznej przeglądarki plików) zawierającego certyfikat, po czym nacisnąć znajdujący się obok przycisk *DODAJ*. W przypadku pomyślnego dodania tego certyfikatu zostanie wyświetlony komunikat informujący o tym oraz w tabelce wyświetlającej certyfikaty X.509 pojawi się nowy wpis z właśnie dodanym certyfikatem. Natomiast w przypadku, gdy podano błędny certyfikat (tzn. *Archiwum* nie jest w stanie rozpoznać wybranego pliku jako certyfikatu X.509) pojawi się stosowny komunikat w kolorze czerwonym.

### 7.2.3. Użytkownicy

Zakładka *UŻYTKOWNICY* służy do przydzielania poszczególnych użytkowników do ról (rys. 7.19). Na początku należy wybrać z wyszukiwarki osobę, dla której przypisanie do ról chcemy zmienić (wyszukiwarka otwiera się za pomocą przycisku *WYBIERZ*). Po wybraniu osoby z wyszukiwarki (czyli kliknięciu odnośnika *Wybierz* obok wybranej osoby) na ekranie pojawi się lista ról, do których dana wybrana osoba jest aktualnie przypisana (role są nadawane w ramach jednostek organizacyjnych, zatem możliwe jest, aby jedna osoba była przypisana dwa razy do tej samej roli, ale w różnych jednostkach).

W celu usunięcia osoby z danej roli należy kliknąć na odpowiedni przycisk *USUŃ*. Po pomyślnym wykonaniu operacji zostanie wyświetlony stosowny komunikat. W celu przypisania osoby do nowej roli należy na formularzu na dole strony w polu *Rola* wybrać odpowiednio rolę i w polu *Jednostka organizacyjna* odpowiednią jednostkę, po czym nacisnąć przycisk *PRZYPI SZ DO ROLI*. W przypadku pomyślnego wykonania tej operacji zostanie wyświetlony komunikat „Przypisano użytkownika do nowej roli”, natomiast w sytuacji, gdy użytkownik już był przypisany do danej roli pojawi się odpowiedni komunikat informujący o tym.



# Administracja

TYPY DOKUMENTÓW • CERTYFIKATY • UŻYTKOWNICY

Osoba	Pracownik Dziekanatu	WYBIERZ
-------	----------------------	---------

## Publiczne Klucze PGP

ID użytkownika	Odcisk palca	
Michał Mański <kassini@interia.pl>	246caf13d787ab2f711b24ae041da4de	Podgląd

Nowy klucz publiczny:	
<input type="text"/>	DODAJ

## Certyfikaty X.509

Podmiot	Wystawca	Ważność		
Michał Mański	CERTUM QCA	19.10.2006 15:48 - 18.10.2008 15:48	Pobierz	Podgląd

Nowy certyfikat:		
<input type="text"/>	Przełóżaj...	DODAJ

Rysunek 7.18: Zakładka CERTYFIKATY

Zalogowany jest: Administrator Wydziału MIM

## Administracja

TYPY DOKUMENTÓW • CERTYFIKATY • UŻYTKOWNICY

Osoba	Ewa2007 Kowalska2007	WYBIERZ
-------	----------------------	---------

### Role

Rola	Jednostka organizacyjna	
Pracownik kvestury	Wydział Matematyki, Informatyki i Mechaniki	USUŃ
Pracownik	Wydział Matematyki, Informatyki i Mechaniki	
Pracownik kvestury	Wydział Chemii	

### Nowa rola

Rola*	Administrator
Jednostka organizacyjna*	Wydział Matematyki, Informatyki i Mechaniki

PRZYPIŚZ DO ROLI

Rysunek 7.19: Zakładka UŻYTKOWNICY

Możliwe jest usuwanie i przydzielanie użytkowników tylko do ról, które są nadawane po stronie *Archiwum* (czyli do następujących: *Administrator*, *Pracownik BSS*, *Pracownik dziekanatu*, *Pracownik kwestury*) — nie ma możliwości przydzielenia do ról, która są migrowane z systemu USOS. Ponadto użytkownik może nadawać/usuwać przypisanie do ról tylko w obrębie tych jednostek, w których jest administratorem (administrator całego systemu ma to uprawnienie we wszystkich jednostkach organizacyjnych).

### 7.3. Interfejs administratora systemu

Administrator systemu ma dostęp do tych samych funkcji co administrator jednostki, przy czym może je wykonywać w obrębie wszystkich jednostek organizacyjnych. Dodatkowo w panelu administratora dostępne są 2 inne zakładki:

- *ROLE* — zarządzanie rolami w *Archiwum*,
- *INNE* — pozostałe funkcjonalności.

W kolejnych punktach opisano funkcjonalności tych zakładek.

#### 7.3.1. Role

Zakładka *Role* pozwala na dodanie nowej roli w systemie (por. rys. 7.20). Aby to zrobić, należy na formularzu w dolnej części ekranu podać nazwę kodową tej roli (powinna zawierać tylko litery alfabetu angielskiego, cyfry oraz znaki ' \_ '), nazwę roli w języku polskim i nazwę roli w języku angielskim, po czym nacisnąć przycisk *DODAJ*. W przypadku, gdy istnieje już inna rola o podanej nazwie kodowej, zdefiniowana rola nie zostanie dodana i będzie wyświetlony stosowny komunikat.

Po kliknięciu na odnośnik zawierający nazwę kodową danej roli, użytkownik zostanie przekierowany do ekranu edycji odpowiedniej roli (por. rys. 7.21). Można tam zmienić nazwę roli w języku polskim i angielskim oraz ustawiać globalne uprawnienia dla danej roli. Zmiany zostaną zapisane po wciśnięciu na przycisk *ZAPISZ ZMIANY*.

Możliwe jest tam także usunięcie roli, która nie jest migrowana z systemu USOS, poprzez przycisk *USUŃ ROLE* (o ile żaden użytkownik nie jest do niej przypisany, wpp. podczas próby usunięcia pojawi się stosowny komunikat), po naciśnięciu go pojawi się komunikat z prośbą o potwierdzenie wykonania tej operacji, gdyż przy usuwaniu ról należy być bardzo ostrożnym, bo np. pewien typ dokumentu może używać tej roli do definiowania uprawnień i wówczas mogą pojawić się w systemie niespodziewane błędy np. podczas dodawania nowego dokumentu. Po pomyślnym usunięciu roli nastąpi powrót do głównej strony zakładki *ROLE*.

#### 7.3.2. Inne

Zakładka *INNE* służy do wykonywania przez administratora systemu pozostałych ważnych funkcji w *Archiwum* (rys. 7.22). Obecnie jest tu tylko możliwość zablokowania systemu (np. na czas migracji danych). Aby to zrobić należy ustalić przedział czasowy, w której aplikacja ma być zablokowana (poprzez wybór dwóch dat z kalendarza) oraz nacisnąć przycisk *ZAPISZ*. W celu anulowania blokady lub zmiany daty początkowej i końcowej należy zmodyfikować ustalone wcześniej daty (pozostawienie pustych pól spowoduje anulowanie blokady).

## Istniejące role

Nazwa kodowa	Nazwa (pl)	Nazwa (en)
admin	Administrator	Administrator
dziekan	Dziekan	Dean
pracownik	Pracownik	Employee
pracownik_bss	Pracownik BSS	Employee of the central student's office
pracownik_dydaktyczny	Pracownik dydaktyczny	Didactic employee
pracownik_dziekanatu	Pracownik dziekanatu	Employee of a student's office
pracownik_kwestury	Pracownik kwestury	Employee of the finance office
student	Student	Student

## Nowa rola

Nazwa kodowa*	<input type="text"/>
Nazwa (pl)*	<input type="text"/>
Nazwa (en)*	<input type="text"/>
	<input type="button" value="DODAJ"/>

Rysunek 7.20: Zakładka *ROLE*

## Rola

Nazwa kodowa	pracownik_dziekanatu
Nazwa (pl)	<input type="text" value="Pracownik dziekanatu"/>
Nazwa (en)	<input type="text" value="Employee of a student's office"/>
Migrowana z USOS-a	Nie

## Uprawnienia

<input checked="" type="checkbox"/> dodawanie dokumentów
<input checked="" type="checkbox"/> edycja dokumentu
<input checked="" type="checkbox"/> dodawanie i modyfikacja załączników dokumentu

Rysunek 7.21: Edycja roli

# Administracja

TYPY DOKUMENTÓW • CERTYFIKATY • ROLE • UŻYTKOWNICY • INNE

## Blokada systemu

Początek	<input type="text"/>	
Koniec	<input type="text"/>	
	<input type="button" value="ZAPISZ"/>	

Rysunek 7.22: Zakładka *INNE*



## Rozdział 8

# Podsumowanie

W dzisiejszym świecie podpisy elektroniczne mogą nam bardzo ułatwić życie i przyspieszyć wykonywanie wielu czynności prawnych, ale korzystając z nich napotykamy na nowe problemy, o których trzeba pamiętać. W szczególności trzeba dbać o konserwację podpisów elektronicznych, bo w przeciwnym razie dokument podpisany elektronicznie straci ważność. Jest kilka różnych typów podpisów, a dodatkowo dwa równoważne podpisy elektroniczne można zapisać w wielu niezgodnych formatach, co również może przysparzać problemów, dlatego też powinno dążyć się do ustandaryzowania stosowanych podpisów.

W związku z ciągle narastającą potrzebą przechowywania różnych dokumentów, archiwa elektroniczne są bardzo przydatne i coraz częściej niezbędne w wielu sytuacjach, a podpisy elektroniczne mogą właśnie sprawić, że takie archiwa będą bezpieczne i dokumenty w nich przechowywane będą miały moc równoważną dokumentom przechowywanym w postaci papierowej. Na Uniwersytecie Warszawskim także przydatne byłoby bezpieczne archiwum elektroniczne. Mam nadzieję, że powstałe w ramach tej pracy *Archiwum* spełni swoje zadanie i przyczyni się do powstania w przyszłości w pełni funkcjonalnego bezpiecznego archiwum dokumentów elektronicznych, które będzie powszechnie używane na UW eliminując tym samym archiwa papierowe.

### 8.1. Zrealizowane założenia

*Archiwum* w obecnej postaci umożliwia przechowywanie dokumentów elektronicznych zarówno niepodpisanych, jak i podpisanych podpisem zwykłym w standardzie PGP oraz podpisem zwykłym lub kwalifikowanym w standardzie X.509 zapisanym w formacie *CMS*. Przechowywane dokumenty są opisywane metadanymi wprowadzanymi przez użytkownika (część atrybutów jest opisywana za pomocą danych migrowanych z systemu *USOS*) i możliwe jest przeszukiwanie wcześniej umieszczonych dokumentów według różnych kryteriów.

Dokumenty podzielone są na różne typy i możliwe jest rejestrowanie nowych typów dynamicznie podczas działania *Archiwum*, co pozwala łatwo rozszerzać funkcjonalność systemu o przechowywanie dokumentów nowego rodzaju. Ponadto istnieje elastyczny system uprawnień, definiujący dostęp do dokumentów na poziomie typu dokumentu, dzięki czemu dla każdego dokumentu mamy pewność, że tylko uprawnieni użytkownicy będą mieli prawo do jego odczytu lub modyfikacji.

## 8.2. Możliwe rozszerzenia

### 8.2.1. Podpisy elektroniczne

Obecnie weryfikowana jest tylko poprawność podpisu złożonego pod dokumentem, brakuje natomiast sprawdzania, czy został on złożony certyfikatem, który był ważny w momencie składania podpisu. Ponadto aktualną funkcjonalność podpisów w *Archiwum* można rozszerzyć o następujące funkcje:

- możliwość umieszczania dokumentów podpisanych przez kogoś innego niż osoba go dodająca,
- weryfikacja wszystkich podpisów pod dokumentem podpisanym przez więcej niż jedną osobę (obecnie weryfikowany jest tylko podpis osoby umieszczającej dokument),
- możliwość podpisywania plików bezpośrednio z *Archiwum* np. poprzez aplet,
- przyjmowanie plików podpisanych we wszystkich popularnych formatach (obecnie akceptowane są tylko podpisy PGP i podpisy X.509 w formacie *CMS*),
- możliwość rejestrowania własnych certyfikatów przez zwykłych użytkowników (obecnie tylko administratorzy mają prawo do rejestrowania certyfikatów).

### 8.2.2. Inne funkcje

Dodatkowo przydatne byłoby zaimplementowanie w *Archiwum* następujących funkcjonalności.

- System dzienników, pozwalający na logowanie do plików informacji o wykonywanych operacjach przez poszczególnych użytkowników — kto kiedy się logował, jakie dokumenty/pliki umieszczał itp. Obecnie są logowane tylko błędy i inne informacje na potrzeby programistów.
- Zaawansowane wyszukiwanie dokumentów. Obecnie możliwe jest tylko wyszukiwanie dokumentów po atrybutach wspólnych dla wszystkich dokumentów, natomiast przydatne byłyby wyszukiwarki filtrujące dokumenty po specyficznych atrybutach w obrębie jednego typu dokumentu
- Sposób aktualizacji typu dokumentu. Czasem może wystąpić potrzeba aktualizacji istniejącego już typu dokumentu, być może wymagająca tylko zmian kilku nazw atrybutów czy wartości wyliczeniowych, więc przydatna byłaby możliwość wyklikania tych zmian poprzez interfejs WWW (obecnie trzeba ładować cały zaktualizowany plik XML).
- Buforowanie uprawnień do poszczególnych dokumentów. Mogłoby to być pomocne zważywszy, że dokumentów w takim archiwum ciągle przybywa i tabela w bazie trzymające informacje o tych uprawnieniach będzie szybko rosła i w związku z tym dostęp do tych uprawnień może być czasochłonny. Buforowane uprawnienia powinny być trzymane w pewnej kolejce priorytetowej, aby najdłużej pamiętane były te uprawnienia, którą są najczęściej sprawdzane.

## 8.3. Podziękowania

Pragnę w szczególności podziękować Pani Doktor Janinie Mincer-Daszkiewicz za opiekę nad projektem, poświęcony czas oraz wszystkie cenne wskazówki dotyczące niniejszej pracy.



## Dodatek A

# Opis zawartości płyty CD

- Niniejsza praca magisterska w formacie PDF wraz ze źródłami w formacie  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Kod źródłowy *Archiwum*,
- *Archiwum* w postaci skompilowanej w pliku *Archiwum.war*,
- Pliki XML opisujące typy dokumentów,
- Skrypt tworzący widoki po stronie systemu USOS na potrzeby *Archiwum*,
- Plik konfiguracyjny migratora.



## Dodatek B

# Skrypt tworzący widoki po stronie systemu USOS na potrzeby *Archiwum*

```
create or replace view ad_studenci as
select distinct t_prgos.os_id, t_jedn.jed_org_kod
  from dz_programy_osob t_prgos,
       dz_jed_org_programow t_jedn
where t_jedn.prg_kod = t_prgos.prg_kod
      and t_jedn.administracja = 'T';
```

```
create or replace view ad_pracownicy as
select t_prac.os_id, t_prac.aktywny, t_os.jed_org_kod
  from dz_pracownicy t_prac,
       dz_osoby t_os
where t_os.id = t_prac.os_id;
```

```
create or replace view ad_role as
select distinct t_prgos.os_id, t_jedn.jed_org_kod, 'student' rola
  from dz_programy_osob t_prgos,
       dz_jed_org_programow t_jedn
where t_jedn.prg_kod = t_prgos.prg_kod
      and t_jedn.administracja = 'T'
      and exists (select 1 from dz_etapy_osob t_etpos where
t_etpos.prgos_id = t_prgos.id and status_zaliczenia in ('X', 'W',
'D'))
      and not exists (select 1 from dz_historia_skr t_hs where
t_hs.prgos_id = t_prgos.id and t_hs.czy_anulowane = 'N')
union all
select t_prac.os_id, t_os.jed_org_kod, 'pracownik_naukowy'
  from dz_pracownicy t_prac,
       dz_osoby t_os
where t_os.id = t_prac.os_id
      and t_prac.aktywny = 'T'
      and exists (select 1 from dz_prac_zatr t_zatr where t_zatr.prac_id
= t_prac.id and t_zatr.umowa_pocz <= sysdate and nvl
(t_zatr.umowa_kon, sysdate + 1) >= sysdate)
```

```

union all
select t_prac.os_id, t_os.jed_org_kod, 'pracownik'
  from dz_pracownicy t_prac,
       dz_osoby t_os
where t_os.id = t_prac.os_id
  and exists (select 1 from dz_prac_zatr t_zatr where t_zatr.prac_id
= t_prac.id and t_zatr.umowa_pocz <= sysdate and nvl
(t_zatr.umowa_kon, sysdate + 1) >= sysdate)
union all
select t_prac.os_id, t_os.jed_org_kod, 'dziekan'
  from dz_pracownicy t_prac,
       dz_osoby t_os
where t_os.id = t_prac.os_id
  and exists (select 1 from dz_prac_zatr t_zatr where t_zatr.prac_id
= t_prac.id and t_zatr.umowa_pocz <= sysdate and nvl
(t_zatr.umowa_kon, sysdate + 1) >= sysdate)
  and exists (select 1 from dz_pelnione_funkcje t_fun,
dz_funkcje_zatr t_f where t_fun.prac_id = t_prac.id and upper
(t_f.nazwa) like '%DZIEKAN%' and t_f.kod = t_fun.funkz_kod and
t_fun.data_pocz <= sysdate and nvl (t_fun.data_kon, sysdate + 1) >=
sysdate);

```

## Dodatek C

# Konfiguracja migratora

Konfiguracji migratora dla *Archiwum*.

```
<?xml version="1.0"?>
<mapping>
  <tables>
    <table from-name="MV_OSOBY" to-name="ad_person">
      <field from-name="ID" to-name="id" />
      <field from-name="PESEL" to-name="pesel" />
      <field from-name="IMIE" to-name="firstname" />
      <field from-name="IMIE2" to-name="secondname" />
      <field from-name="NAZWISKO" to-name="lastname" />
      <field from-name="EMAIL" to-name="email" />
      <primary-key>
        <field name="ID" />
      </primary-key>
      <dependencies>
      </dependencies>
    </table>

    <table from-name="MV_KONTA" to-name="ad_account">
      <field from-name="OS_ID" to-name="personId" />
      <field from-name="LOGIN" to-name="login" />
      <cond field-name="INST_WWW_KOD">
        <value>CUS_WWW</value>
      </cond>
      <primary-key>
        <field name="OS_ID" />
      </primary-key>
      <dependencies>
      </dependencies>
    </table>

    <table from-name="MV_KIERUNKI_STUDIOW" to-name="ad_field_of_study">
      <field from-name="KOD" to-name="code" />
      <field from-name="OPIS" to-name="name_pl" />
      <field from-name="DESCRIPTION" to-name="name_en" />
    </table>
  </tables>
</mapping>
```

```

    <primary-key>
      <field name="KOD" />
    </primary-key>
    <dependencies>
    </dependencies>
  </table>

<table from-name="MV_JEDNOSTKI_ORGANIZACYJNE" to-name="ad_faculty">
  <field from-name="KOD" to-name="code" />
  <field from-name="JED_ORG_KOD" to-name="parent_code" />
  <field from-name="OPIS" to-name="name_pl" />
  <field from-name="OPIS_ANG" to-name="name_en" />
  <cond field-name="CZY_DYDAKTYCZNA">
    <value>T</value>
  </cond>
  <primary-key>
    <field name="KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="MV_PRZEDMIOTY" to-name="ad_subject">
  <field from-name="KOD" to-name="code" />
  <field from-name="JED_ORG_KOD" to-name="faculty_code" />
  <field from-name="NAZWA" to-name="name_pl" />
  <field from-name="NAME" to-name="name_en" />
  <primary-key>
    <field name="KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="MV_PROGRAMY" to-name="ad_study_program">
  <field from-name="KOD" to-name="code" />
  <field from-name="TRYB_STUDIOW" to-name="studies_mode_pl" />
  <field from-name="RODZAJ_STUDIOW" to-name="studies_mode_pl" />
  <field from-name="TRYB_STUDIOW_ANG" to-name="studies_type_en" />
  <field from-name="RODZAJ_STUDIOW_ANG" to-name="studies_type_en" />
  <field from-name="OPIS" to-name="description_pl"/>
  <field from-name="DESCRIPTION" to-name="description_en"/>
  <primary-key>
    <field name="KOD" />
  </primary-key>

  <dependencies>
  </dependencies>
</table>

```

```

<table from-name="MV_CYKLE_DYDAKTYCZNE" to-name="ad_didactic_period">
  <field from-name="OPIS" to-name="name_pl" />
  <field from-name="DESCRIPTION" to-name="name_en" />
  <field from-name="DATA_DO" to-name="endDate" />
  <field from-name="KOD" to-name="code" />
  <field from-name="DATA_OD" to-name="startDate" />
  <field from-name="TCDYD_KOD" to-name="type" />
  <primary-key>
    <field name="KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="MV_ETAPY" to-name="ad_stage">
  <field from-name="DESCRIPTION" to-name="description_en" />
  <field from-name="OPIS" to-name="description_pl" />
  <field from-name="KOD" to-name="code" />
  <primary-key>
    <field name="KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="MV_PRZEDMIOTY_CYKLI" to-name="ad_subject_in_period">
  <field from-name="PRZ_KOD" to-name="subject_code" />
  <field from-name="CDYD_KOD" to-name="period_code" />
  <primary-key>
    <field name="CDYD_KOD" />
    <field name="PRZ_KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="MV_PRACE_CERT" to-name="ad_thesis">
  <field from-name="ID" to-name="id" />
  <field from-name="PRZ_KOD" to-name="subject_code" />
  <field from-name="JED_ORG_KOD" to-name="faculty_code" />
  <field from-name="TYTUL" to-name="title_pl" />
  <field from-name="TYTUL_ANG" to-name="title_en" />
  <field from-name="DATA_ZLOZENIA" to-name="submissionDate" />
  <field from-name="DATA_ZATW_TEMATU" to-name="commitionDate" />
  <field from-name="SLOWA_KLUCZOWE" to-name="keywords" />
  <field from-name="STRESZCZENIE" to-name="description" />
  <primary-key>
    <field name="ID" />
  </primary-key>

```

```

    </primary-key>
    <dependencies>
    </dependencies>

</table>

<table from-name="AD_STUDENCI" to-name="ad_student">
  <field from-name="OS_ID" to-name="personId"/>
  <field from-name="JED_ORG_KOD" to-name="facultyCode"/>
  <primary-key>
    <field name="OS_ID" />
    <field name="JED_ORG_KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="AD_PRACOWNICY" to-name="ad_employee">
  <field from-name="OS_ID" to-name="personId"/>
  <field from-name="JED_ORG_KOD" to-name="facultyCode"/>
  <field from-name="AKTYWNY" to-name="didactic"/>
  <primary-key>
    <field name="OS_ID" />
    <field name="JED_ORG_KOD" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>

<table from-name="AD_ROLE" to-name="ad_role_user_usos">
  <field from-name="OS_ID" to-name="user_id"/>
  <field from-name="JED_ORG_KOD" to-name="faculty_code"/>
  <field from-name="ROLA" to-name="role_id"/>
  <primary-key>
    <field name="OS_ID" />
    <field name="JED_ORG_KOD" />
    <field name="ROLA" />
  </primary-key>
  <dependencies>
  </dependencies>
</table>
</tables>
</mapping>

```



# Bibliografia

- [Ant] Strona główna projektu Apache Ant — <http://ant.apache.org/>
- [Apache] Strona główna Apache Software Foundation — <http://www.apache.org>
- [APD] Archiwum Prac Dyplomowych Uniwersytetu Warszawskiego — <https://apd.uw.edu.pl/>
- [BC] The Legion of the Bouncy Castle Java Cryptography APIs — <http://www.bouncycastle.org/java.html>
- [CAS] JA-SIG Central Authentication Service Home Page — <http://www.ja-sig.org/products/cas/>
- [CERTUM] CERTUM Powszechne Centrum Certyfikacji — <http://www.certum.pl>
- [DBDes] Strona domowa projektu DBDesigner — <http://fabforce.net/dbdesigner4/>
- [JCA] Java Cryptography Architecture (JCA) Reference Guide — <http://java.sun.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html>  
Java Cryptography Architecture API Specification & Reference <http://java.sun.com/j2se/1.5.0/docs/guide/security/CryptoSpec.html>
- [Eclipse] Strona domowa projektu Eclipse — <http://www.eclipse.org/>
- [Hibernate] Strona domowa biblioteki Hibernate — <http://www.hibernate.org>
- [JavaEE] An Introduction to the Java EE Platform — <http://java.sun.com/javaee/5/docs/firstcup/doc/toc.html>  
Java EE 5 SDK API Specifications <http://java.sun.com/javaee/5/docs/api/>
- [KIR] Krajowa Izba Rozliczeniowa — <http://www.kir.com.pl>
- [LaTeX2e] Tobias Oetiker, Hubert Partl, Irene Hyna i Elisabeth Schlegl — The Not So Short Introduction to LATEX — <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- [Migrator] R. Cieplak, *Uniwersytecki System Obsługi Studiów. Migracja danych*, Praca magisterska, Instytut Informatyki Uniwersytetu Warszawskiego, 2004
- [MySQL] Strona domowa bazy danych MySQL — <http://www.mysql.com>
- [PGP] Strona o PGP (Pretty Good Privacy) w Wikipedii — [http://en.wikipedia.org/wiki/Pretty\\_Good\\_Privacy](http://en.wikipedia.org/wiki/Pretty_Good_Privacy)  
OpenPGP Alliance — <http://www.openpgp.org/>  
OpenPGP Message Format (RFC 2440)— <http://rfc.net/rfc2440.html>

- [RFC2315] RFC 2315 — PKCS#7: Cryptographic Message Syntax Version 1.5 — <http://www.ietf.org/rfc/rfc2315.txt>
- [RFC3852] RFC 3852 — Cryptographic Message Syntax (CMS) — <http://www.ietf.org/rfc/rfc3852.txt>
- [SIGILLUM] Sigillum Polskie Centrum Certyfikacji Elektronicznej — <http://www.sigillum.pl>
- [Struts] Strona domowa projektu Struts — <http://struts.apache.org>
- [SUNPKCS11] Java PKCS#11 Reference Guide — <http://java.sun.com/j2se/1.5.0/docs/guide/security/p11guide.html>
- [USOS] Uniwersytecki System Obsługi Studiów — <http://usos.mimuw.edu.pl>
- [USOSweb] Internetowy moduł USOS, np. — <http://usosweb.mimuw.edu.pl>
- [Tomcat] Strona domowa kontenera aplikacji webowych Apache Tomcat — <http://tomcat.apache.org>
- [Uniwex] Strona projektu Eurowex, którego zadaniem jest dostosowanie systemu Uniwex do potrzeb europejskich uniwersytetów — <http://www.eurowex.org/en>
- [Unizeto] Unizeto Technologies SA — <http://www.unizeto.pl>
- [Ust01] Ustawa z dnia 18 września 2001r. o podpisie elektronicznym — <http://isip.sejm.gov.pl/servlet/Search?todo=open&id=WDU20011301450>
- [WW] Strona domowa narzędzia Webwork — <http://www.opensymphony.com/webwork>
- [X509] Strona o standardzie X.509 w Wikipedii — <http://en.wikipedia.org/wiki/X509>
- [XaDES] XML Advanced Electronic Signatures (XAdES) — <http://www.w3.org/TR/XAdES/>