

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

**Maciej Góralski**

Nr albumu: 346144

**Kacper Pawelec**

Nr albumu: 332436

**Aliaksei Suvorau**

Nr albumu: 374118

**Michał Swidziński**

Nr albumu: 371800

# Zapisy na spotkania w systemie USOS

Praca licencjacka  
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem  
**dr Janina Mincer-Daszkiewicz**  
Uniwersytet Warszawski

Wrzesień 2018

## **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 (autorów) 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

Podpisy autorów pracy

## **Streszczenie**

Praca polega na dodaniu funkcjonalności do systemów USOSadm i USOSweb. Funkcjonalność ta umożliwi zdalną rejestrację zapisów do Dziekana dla studentów, co zaoszczędzi czasu, zmniejszy kolejki do Sekcji Studenckiej i pozwoli na zbieranie informacji do celów statystycznych. Ponadto zdalna rejestracja będzie dużo wygodniejsza zarówno dla Dziekana, pracowników Sekcji Studenckiej jak i studentów.

## **Słowa kluczowe**

USOS, USOSadm, USOSweb, zapisy, spotkania, Dziekan, Sekcja Studencka

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

11.3 Informatyka

## **Klasyfikacja tematyczna**

Software and its engineering

Software creation and management

Software evolution



# Spis treści

<b>1. Wprowadzenie</b>	5
1.1. Rozwiązanie dotychczasowe	5
1.2. Ogólne informacje o systemie USOS	5
1.3. Cel pracy	6
1.4. Struktura pracy	6
1.5. Podział pracy	6
<b>2. Wymagania użytkowników</b>	7
2.1. Wymagania	7
2.2. Rozmowa z Dziekanem	7
2.3. Ankieta dla studentów	7
2.4. Rozmowa z Sekcją Studencką	8
<b>3. Specyfikacja</b>	9
3.1. Cel projektu	9
3.2. Zapisy	9
3.3. Cykl zapisów	9
3.4. Dodatkowe wymagania	10
3.5. Formularz	10
3.6. Szczegółowa specyfikacja użytkownika	10
3.6.1. Używanie USOSadm poprzez już zalogowanego użytkownika z odpowiednimi uprawnieniami.	10
3.6.2. Używanie USOSweb poprzez już zalogowanego użytkownika z odpowiednimi uprawnieniami.	11
3.7. Dodatkowe wymagania	12
3.8. Schemat bazy danych	12
3.9. Przykłady	14
<b>4. Implementacja</b>	17
4.1. USOS	17
4.2. USOSadm	17
4.2.1. Informacje ogólne	17
4.2.2. Aplikacja kliencka	18
4.2.3. Serwer	18
4.2.4. Komunikacja z bazą danych	18
4.2.5. Opis implementacji	18
4.2.6. Słownik uzasadnień	18
4.2.7. Kalendarz	18

4.2.8.	Notatki . . . . .	19
4.2.9.	Spotkania osób . . . . .	19
4.3.	USOSweb . . . . .	19
4.3.1.	Opis działania systemu USOSweb . . . . .	20
4.3.2.	Akcje . . . . .	20
4.3.3.	Szablony . . . . .	20
4.3.4.	Moduły pomocnicze . . . . .	21
4.3.5.	Problemy i ich rozwiązania . . . . .	21
4.3.6.	Zmiany w USOSWebie . . . . .	22
4.3.7.	Potencjalne ataki i zabezpieczenia . . . . .	22
<b>5.</b>	<b>Potencjalny rozwój . . . . .</b>	<b>25</b>
<b>6.</b>	<b>Dodatki . . . . .</b>	<b>27</b>
6.1.	Dokumentacja instalacyjna . . . . .	27
6.2.	Opis zawartości płyty CD . . . . .	27

# Rozdział 1

## Wprowadzenie

### 1.1. Rozwiązanie dotychczasowe

Obecnie student chcący zapisać się na spotkanie z Dziekanem musi przejść przez następujący proces:

1. Przyjść, zadzwonić albo wysłać e-mail do Sekcji Studenckiej.
2. Podać imię, nazwisko, numer indeksu oraz powód wizyty.
3. Student otrzymuje odpowiedź od Sekcji Studenckiej gdzie, w zależności od zasadności powodu, zawierając numer jaki zajmuje on na liście.

W zależności od wybranej formy komunikacji oraz okresu w jakim student się zapisuje, czas oczekiwania może być istotnie długi. Momentem gdy szczególnie trudno jest uzyskać kontakt z Sekcją Studencką jest okres przyklejania naklejek na legitymacje studenckie. Wtedy kolejki studentów potrafią być na tyle długie że w rozsądnym czasie student niekoniecznie będzie mógł wejść do sekretariatu, co jest najpewniejszą formą zapisania się.

Kluczowy jest tutaj fakt, że Sekcja Studencka na Wydziale MIM otwarta jest dla studentów tylko w poniedziałki, wtorki, czwartki i piątki przez okres 3 godzin dziennie. Ogranicza to czasowe możliwości studentów na zapisy. Co więcej, forma w jakiej przebiegają zapisy niepotrzebnie zajmuje czas studentów i pracowników Sekcji Studenckiej.

### 1.2. Ogólnie informacje o systemie USOS

USOS, czyli Uniwersytecki System Obsługi Studiów, powstał w wyniku zapotrzebowania na narzędzie informatyczne służące do zarządzania sprawami studiów na polskich uczelniach. Razem z sukcesem USOS powstała potrzeba żeby stworzyć platformę umożliwiającą studentom częściowy dostęp do możliwości jakie dostarcza USOS — w tym celu powstał USOSweb. Internetowa platforma pozwalająca studentom i pracownikom uczelni na dostęp do i pozwalająca na częściową modyfikację danych z USOS.

USOS dostarcza takie usługi jak:

1. Rekrutacja na studia i immatrykulacja.
2. Elektroniczne Legitymacje Studenckie.
3. Przygotowywanie oferty dydaktycznej.

4. Zarządzanie tokiem studiów.
5. Podania, stypendia i ankiety.
6. Akademiki i płatności

oraz inne.

### 1.3. Cel pracy

W ramach pracy powstanie moduł do systemów USOSadm i USOSweb, dzięki któremu będzie możliwe tworzenie, kontrola i archiwizacja wizyt studentów u Dziekana. W USOSweb studenci i osoby spoza wydziału będą mogli zapisywać się na spotkania z Dziekanem w odgórnie ustalonych terminach. Poza ułatwieniem rejestracji i zwiększeniem jej dostępności, system ten będzie również maksymalizować wykorzystanie czasu Dziekana oraz studentów.

### 1.4. Struktura pracy

Praca składa się z 6 rozdziałów i dodatków. Uwagi i wymagania potencjalnych użytkowników zostały opisane w rozdziale 2. W rozdziale 3 znajduje się specyfikacja nowych funkcjonalności. W rozdziale 4 jest zawarty opis technologii użytych do implementacji oraz jej szczegóły. Potencjalny rozwój nowego modułu jest opisany w rozdziale 5, a dokumentacje dotyczące naszej funkcjonalności są w rozdziale 6.

### 1.5. Podział pracy

Podział pracy w projekcie wygląda następująco:

- Maciej Góralski — praca nad USOSadm w Java oraz pisanie pracy licencjackiej,
- Kacper Pawelec — praca nad USOSadm w Java,
- Aliaksei Suvorau — praca nad USOSweb w PHP i MySQL,
- Michał Swidziński — praca nad USOSweb w PHP i MySQL.



## Rozdział 2

# Wymagania użytkowników

### 2.1. Wymagania

Przed przystąpieniem do projektowania systemu postanowiliśmy zapytać potencjalnych użytkowników jakie mają wymagania względem systemu. Zidentyfikowaliśmy trzy główne grupy użytkowników które zapytaliśmy o zdanie: Dziekana, studentów oraz pracowników Sekcji Studenckiej Dziekanatu.

### 2.2. Rozmowa z Dziekanem

Dziekana, będącego jednym z głównych odbiorców modułu, jako pierwszego zapytaliśmy o wymagania względem projektu. Najistotniejsze z nich to:

- Powinien istnieć mechanizm ręcznej i automatycznej akceptacji próśb o spotkanie.
- Moduł musi dawać możliwość konfiguracji limitów czasowych oraz limitów ilości osób.
- Jednocześnie można być zapisanym tylko na jedno spotkanie.
- Dziekan powinien mieć możliwość zostawienia w systemie notatki ze spotkania.
- Powinno być możliwe usowanie informacji o starych spotkaniach, z zachowaniem notatek.

Wymagania jakie przedstawił Dziekan w dużej mierze uformowały kształt jaki ten projekt przyjął. Na ich podstawie rozwijaliśmy lub dodawaliśmy do modułu funkcjonalności.

### 2.3. Ankieta dla studentów

W trakcie projektowania modułu przeprowadziliśmy ankietę wśród studentów aby dowiedzieć się jakie byłyby ich preferencje względem niektórych funkcjonalności i czy pokrywają się z wymyślonymi przez nas rozwiązaniami. Ich odpowiedzi wyglądały następująco:

- Jak wcześnie przed wizytą chciałbyś/chciałabyś zapisać się do dziekana?
  - Tego samego dnia — 28.2%.
  - Na tydzień przed — 66.47%.
  - Na dwa tygodnie przed — 2.94%.
  - Na trzy tygodnie przed — 0.00%.

- Na miesiąc przed — 1.76%.
- Na dłużej niż miesiąc przed — 0.00%.
- W wypadku gdy limit dostępnych miejsc się zapełni, chciałbyś/chciałabyś:
  - Zostać zapisanym/zapisaną na najbliższy dostępny termin — 10%.
  - Zostać zapisanym na następny termin wyznaczony przez siebie — 79.41%.
  - Nic, chcę zapisać się od nowa samodzielnie — 10.59%.
- Podając powód na wizytę u Dziekana, chcę:
  - Napisać powód samemu - 6.47%.
  - Wybrać powód z listy dostępnych możliwych powodów - 9.41%.
  - Mieszaną powyższych odpowiedzi - 84.12%.
- Czy zapisując się do Dziekana bierzesz pod uwagę możliwość w której nie będziesz w stanie pojawić się na wizycie wiedząc, że to może grozić trudnościami przy kolejnych zapisach? Jeśli tak, to jak sobie wyobrażasz optymalny system zapisów? Wszelkie inne uwagi podaj również w tym polu.
  - “Chcę móc odwołać bez podania powodu“
  - “Zazwyczaj z powodu nagłych wypadków / choroby - 1-4 dni przed / rano tego samego dnia“
  - “wypadki losowe, 2-3 dni przed“
  - “24h brzmi sensownie“

Większość studentów potwierdziła, że zaprojektowane przez nas rozwiązanie będzie dla nich wygodne i funkcjonalne.

## 2.4. Rozmowa z Sekcją Studencką

Zapytaliśmy o zdanie również pracowników Sekcji Studenckiej. Ich wymagania wyglądały następująco:

- System powinien być wygodny i prosty w obsłudze.
- Jak najwięcej opcji i pól powinno mieć domyślne wartości.
- Powinna istnieć opcja automatycznej akceptacji spotkań.
- Limity dostępnych miejsc, czasowe i inne powinny być modyfikowalne.

Odpowiedź pracowników Sekcji Studenckiej wyraźnie podkreśliła wagę jaką ma wygoda i łatwość użytkowania. W związku z tym postanowiliśmy położyć dodatkowy nacisk na te cechy systemu.

## Rozdział 3

# Specyfikacja

### 3.1. Cel projektu

Celem projektu jest dodanie do aplikacji USOSadm i USOSweb funkcjonalności zapisów na spotkania z Dziekanem. System dzieli się na dwie części: moduł USOSweb dla osoby zapisującej się na spotkanie i moduł USOSadm dla personelu obsługującego spotkania.

### 3.2. Zapisy

Zapisy na spotkania będą realizowane poprzez kalendarz spotkań, ich przebieg będzie wyglądał następująco:

- Zdefiniowanie terminu spotkania.
- Okres zapisów: interesanci składają prośby o spotkanie w formie formularzy.
- Okres akceptacji: pracownicy przeglądają formularze i akceptują lub odrzucają prośby.
- Spotkanie odbywa się.

### 3.3. Cykl zapisów

W cyklu zapisów można wyróżnić następujące kluczowe informacje:

- Data i godzina spotkania .
- Termin początku i końca okresu zapisów.
- Termin początku i końca okresu akceptacji.
- Limit miejsc na spotkanie.
- Limit miejsc rezerwowych na spotkanie.
- Jednostkę, której dotyczy spotkanie.
- Osobę, której dotyczy spotkanie

Porządek chronologiczny terminów:

- Początek okresu zapisów.

- Koniec okresu zapisów.
- Początek okresu akceptacji.
- Koniec okresu akceptacji.
- Termin spotkania.

### 3.4. Dodatkowe wymagania

Interesanci po zapisaniu się na spotkanie mogą zostać zaakceptowani, odrzuceni lub mogą nie zmieścić się w limicie miejsc. Ci, którzy nie zmieścili się w limicie, mają przez ograniczony czas możliwość przepisania swojego spotkania na inny termin, na który zapisy nie zostały jeszcze otwarte. Odrzuceni interesanci nie mają takiej możliwości.

Limit miejsc rezerwowych ma służyć maksymalnemu wykorzystaniu czasu na spotkania. W przypadku gdy osoba potwierdzająca zapisy studenta odrzuci jego prośbę o spotkanie z Dziekanem, zostanie ona usunięta z kolejki, wszyscy studenci niżej na liście głównej przeniesi o miejsce wyżej i, jeśli ktoś się na niej znajduje, pierwsze miejsce na liście rezerwowej zostanie przeniesione na listę główną. .

### 3.5. Formularz

Formularz zapisów zawiera następujące informacje:

- Imię i nazwisko osoby zapisującej się.
- Numer indeksu, jeśli jest studentem.
- Powód spotkania.
- Spotkanie, którego dotyczy formularz.
- Czy formularz jest “nowy” w aktualnym cyklu zapisów.
- Informacja czy spotkanie zostało zaakceptowane, odrzucone czy nierozpatrzone.
- Informacja czy spotkanie się odbyło.
- Uzasadnienie odrzucenia.

Status “nowości” formularz zyskuje w chwili utworzenia. Tracony jest podczas pierwszego przeglądania przez personel. Służy szybkiemu docieraniu do nowych formularzy, w przypadku wielokrotnego przeglądania.

### 3.6. Szczegółowa specyfikacja użytkowania

#### 3.6.1. Używanie USOSadm poprzez już zalogowanego użytkownika z odpowiednimi uprawnieniami.

1. Utworzenie kalendarza spotkań.
  - 1.1. Podanie danych nowego kalendarza:
    - Nazwa kalendarza.
    - Cykl dydaktyczny.
    - Jednostka w której odbywa się spotkanie.

- Osoba z którą odbywa się spotkanie.
- 1.2. Załadowanie danych z już istniejącego kalendarza.
- 2. Utworzenie spotkań, dla każdego podanie:
  - Początek i koniec okresu zapisów.
  - Początek i koniec spotkania.
  - Początek i koniec okresu akceptacji.
- 3. Utworzenie listy powodów odrzuceń, dla każdego podając:
  - Jednostkę organizacyjną.
  - Nazwę.
  - Treść powodu.
- 4. Udostępnienie kalendarza poprzez zmianę stanu na "Dostępne".
- 5. Okres akceptacji.
  - 5.1. Wyświetlenie list studentów.
    - 5.1.1. Sortowanie listy po:
      - Imieniu.
      - Nazwisku.
      - Peselu.
      - Indeksie.
  - 5.2. Akceptacja studenta.
  - 5.3. Odrzucenie studenta.

### **3.6.2. Używanie USOSweb poprzez już zalogowane użytkowników z odpowiednimi uprawnieniami.**

1. Wejście w widok "Dla studentów".
2. Przejście w zakładkę "Moje jednostki" lub "Wszystkie jednostki".
3. Wybranie jednostki, z możliwością sortowania po:
  - Kodzie jednostki.
  - Nazwie jednostki.
  - Liczbie dostępnych spotkań.
4. Przeglądanie dostępnych spotkań wybranej jednostki:
  - 4.1. Możliwość przeglądania nieaktywnych terminów z dostępnych spotkań.
  - 4.2. Dostęp do informacji o:
    - Dacie spotkania.
    - Godzinach w jakich odbywa się spotkanie
    - Początku i końca okresu zapisów.
    - Stanie zapisów.

- Ilości zajętych miejsc.
5. Zapisanie na spotkanie:
    - 5.1. Wybór spotkania.
    - 5.2. Wybranie powodu z listy dostępnych powodów.
    - 5.3. Wypełnienie pisemnego powodu.
  6. Przeglądanie spotkań w zakładce "Moje spotkania"
    - 6.1. Możliwość sortowania po:
      - Opisie spotkania.
      - Kodzie jednostki.
      - Dacie spotkania.
      - Stanie zapisu:
        - A – zaakceptowane.
        - N – nierozpatrzone.
        - O – odrzucone.
  7. Wypisanie ze spotkania:
    - 7.1. Znalezienie spotkania.
    - 7.2. Kliknięcie koszyka wypisania.
    - 7.3. Potwierdzenie decyzji.

### 3.7. Dodatkowe wymagania

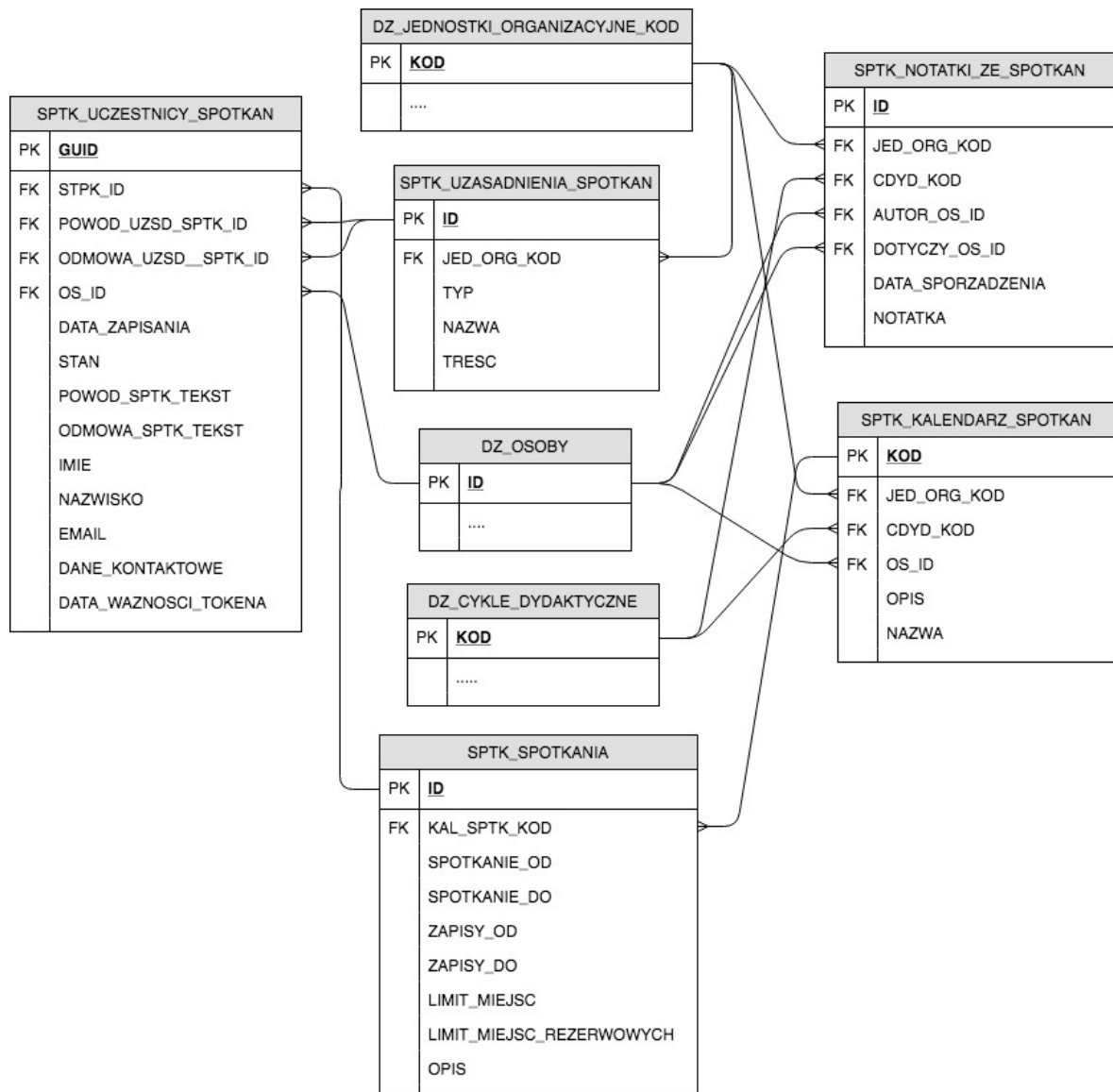
1. Interfejs powinien mieć zintegrowane, zrozumiałe instrukcje użytkowania.
2. Interesanci nie mogą mieć dostępu do danych zapisów innych interesantów.
3. Korzystanie z aplikacji wymaga konta w uczelnianym Centralnym Systemie Uwierzytelniającym (CAS, z ang. Central Authentication Service).

### 3.8. Schemat bazy danych

W wyniku wymagań i potrzeb implementacyjnych, utworzyliśmy następujący schemat bazy danych, pokazany w obrazie 3.1. Kluczowe jej tabele mają następujący skład:

- SPTK\_UCZESNITY\_SPOTKAN
  - Podstawowych danych uczestników spotkania.
  - Id spotkania.
  - Powód wraz z uzasadnieniem.
  - Odmowę wraz z uzasadnieniem.
  - Stan spotkania.
- SPTK\_SPOTKANIA

- Kalendarza spotkań.
- Dat spotkania i zapisów.
- Limitów miesjc.
- Opisu.
- SPTK\_NOTATKI\_ZE\_SPOTKAN
  - Jednostki organizacyjnej.
  - Cyklu dydaktycznego.
  - Autora.
  - Osoby której dotyczy.
  - Daty utworzenia.
  - Notatki.



Rysunek 3.1: Schemat bazy danych.

### 3.9. Przykłady

Przykładowe scenariusze użycia modułu. Dla prostoty przyjęto dzień jako niepodzielną jednostkę czasu.

Tablica 3.1: Lista Rezerwowa

Dzień	Student	Spotkanie1
1		Definicja
2	Dostępne spotkania: Spotkanie1. Zapisanie na Spotkanie1, miejsce na liście rezerwowej.	Zapisy
3	Kilka miejsc w kolejce zostaje zwolnionych, dostaje się na spotkanie	Akceptacja
4		Spotkanie

Tablica 3.2: Token przeniesienia

Dzień	Student	Spotkanie1	Spotkanie2
1		Definicja	Definicja
2	Dostępne spotkania: Spotkanie1. Zapisanie na Spotkanie1, miejsce na liście rezerwowej	Zapisy	
3	Nie dostaje się na spotkanie, pozostaje w rezerwie.	Akceptacja	
4	Dostaje token przeniesienia. Dostępne spotkania: Spotkanie2. Przenosi się na Spotkanie2	Spotkanie	
5			Zapisy
6	Dostaje się na spotkanie		Akceptacja
7			Spotkanie



Tablica 3.3: Token przeniesienia pozwala na przeniesienie tylko do nieotwartego spotkania

Dzień	Student	Spotkanie1	Spotkanie2	Spotkanie3
1		Definicja	Definicja	
2	Dostępne spotkania: Spotkanie1. Zapisanie na Spotkanie1, miejsce na liście rezerwowej.	Zapisy		
3	Nie dostaje się na spotkanie, pozostaje w rezerwie.	Akceptacja		
4	Dostaje token przeniesienia. Dostępne spotkania: brak	Spotkanie		
5	Token wygasa		Zapisy	Definicja
6			Akceptacja	Zapisy

Tablica 3.4: Wiele tur jednocześnie

Dzień	Student	Spotkanie1	Spotkanie2
1		Definicja	Definicja
2	Dostępne spotkania: Spotkanie1. Zapisanie na Spotkanie1, miejsce na liście rezerwowej.	Zapisy	Zapisy
3	Nie dostaje się na spotkanie, pozostaje w rezerwie.	Akceptacja	Zapisy
4	Dostaje token przeniesienia. Dostępne spotkania: brak. Dostępne spotkania: Spotkanie2. Zapisanie na Spotkanie2	Spotkanie	Zapisy
5	Token wygasa		Akceptacja
6			Spotkanie



## Rozdział 4

# Implementacja

### 4.1. USOS

USOS jest skomplikowanym systemem, powstającym przez lata. Składa się on aktualnie z wielu komponentów odpowiadających za różne zbiory funkcjonalności potrzebnych do obsługi studiów. Ze względu na zróżnicowane okresy powstawania oraz przeznaczenie, różnią się między sobą znacząco technologią wykonania, a zasada współdziałania całego systemu jest bardzo skomplikowana. Składowe, z którymi mieliśmy styczność podczas naszej pracy, to:

- USOSadm w Javie – moduł przeznaczony dla pracowników administracji, pozwalający na cyfryzację typowych zadań administracyjnych, jak zarządzanie rejestracjami, płatnościami, wymianą międzynarodową etc.
- USOSWeb – moduł przeznaczony dla studentów napisany w PHP. Udostępnia podstawowe informacje o toku studiów studenta oraz funkcje zdalnego załatwiania wielu formalności, jak rejestracja na przedmioty czy składanie podań.
- Baza danych Oracle – centralna baza danych USOS. Posiada bogatą wewnętrzną logikę odpowiadającą między innymi za dostęp do danych. Bezpośrednio kontaktuje się z nią USOSadm.
- Migrator – moduł odpowiedzialny za synchronizację baz danych. Ponieważ ruch w systemie USOS jest rozproszony pomiędzy wiele modułów i baz danych, Migrator dba o zachowanie spójności danych z centralną bazą wykonując częste, okresowe migracje.

### 4.2. USOSadm

#### 4.2.1. Informacje ogólne

Celem prac w obrębie USOSadm było zaimplementowanie funkcjonalności przy wykorzystaniu przyjętych praktyk programistycznych. Jej zakres obejmował następujące mechanizmy:

- Definiowania nowych kalendarzy spotkań oraz spotkań z pracownikami uczelni.
- Zapisu i akceptowania zapisów na spokania studentów.
- Sporządzania notatek ze spotkań.
- Definiowania listy najczęstszych powodów zapisów na spotkania.

- Mechanizm tokenowy, pozwalający odrzuconym studentom zapisać się na kolejne spotkanie z uprzywilejowaną pozycją.

#### **4.2.2. Aplikacja kliencka**

Aplikacja kliencka USOSadm to część aplikacji wykonywana po stronie klienta. Składa się m.in. z kodu HTML i skryptów JavaScript. Napisana została przy pomocy frameworka JavaServer Faces z wykorzystaniem biblioteki komponentów RichFaces.

#### **4.2.3. Serwer**

Serwer USOSadm odpowiedzialny jest za większą część logiki aplikacji. Napisany został w Javie. Część logiki wykonywana jest po stronie bazy danych Oracle.

#### **4.2.4. Komunikacja z bazą danych**

Komunikacja z bazą danych została zrealizowana przy pomocy własnego, bogatego API zrealizowanego przy pomocy technologii Hibernate. Technologia Hibernate pozwala na odwzorowanie danych z bazy danych za pomocą odpowiednio spreparowanych obiektów w Javie.

#### **4.2.5. Opis implementacji**

Funkcjonalność została wydzielona w oddzielnym module do USOSadm, "Spotkania". Składa się łącznie z 4 widoków:

- Słownik uzasadnień – pozwala zdefiniować najczęstsze uzasadnienia zapisów na spotkania jak i odrzucenia spotkań.
- Kalendarz Spotkań – pozwala na definicję kalendarzy spotkań, spotkań, zapisywanie studentów oraz akceptowanie zapisów na spotkania.
- Notatki – pozwala na przeglądanie notatek ze spotkań osób oraz dodawanie nowych.
- Spotkania osób – udostępnia funkcjonalność potrzebną do przeprowadzenia spotkania, taką jak przeglądanie spotkań danej osoby, przeglądanie zapisów na spotkanie oraz notatek z poprzednich spotkań danej osoby.

#### **4.2.6. Słownik uzasadnień**

Widok słownika uzasadnień ma prostą strukturę. Pozwala na dodawanie i edycję uzasadnień spotkań i odrzucenia spotkań.

#### **4.2.7. Kalendarz**

Kalendarz spotkań Widok kalendarza spotkań składa się z trzech części:

- Kalendarz spotkań – pozwala na dodawanie i edycję kalendarzy spotkań. Kalendarz odpowiada wybranemu rodzajowi spotkań dla wybranej osoby, w wybranym cyklu dydaktycznym, w wybranej jednostce. Przykładowo, dziekan może w danym cyklu dydaktycznym prowadzić spotkania dziekańskie w jednym kalendarzu oraz konsultacje na dwóch różnych wydziałach w dwóch innych kalendarzach.

- Spotkania – pozwala na definiowanie i edycję spotkań dla konkretnego kalendarza. Z uwagi na cykliczną naturę spotkań, przewidziano możliwość stworzenia spotkania bazującego na kopii poprzednio zdefiniowanego spotkania.
- Zaspiy – pozwala na zapis studentów na spotkanie jak i edycję zapisów. Przewidziano możliwość automatycznej akceptacji zapisów do wyczerpania limitu. Przewidziano również możliwość nieformalnego zamknięcia rozpatrywania zapisów, które polega na wysłaniu maili do zapisanych studentów z informacją o wyniku rozpatrywania ich zapisu.

#### 4.2.8. Notatki

Widok notatek pozwala na przeglądanie starych notatek wybrajen osoby, jak i dodawanie nowych. Jego struktura nie jest skomplikowana.

#### 4.2.9. Spotkania osób

Widok spotkania osób udostępnia podstawowa funkcjonalność potrzebną do przeprowadzenia spotkania. Ma on bardziej złożoną budowę. Pierwsza część pozwala nam wybrać odpowiedniego pracownika, którego spotkanie przeprowadzamy. W domyśle: siebie. W kolejnej części naturalnie musimy wybrać kalendarz spotkań, z którego chcemy przeprowadzić spotkanie oraz konkretne spotkanie. Kolejna część pozwala nam obsługiwać pojedyncze osoby. Polega to na udostępnieniu listy zapisanych osób, jak i wygodny przegląd ich notatek z poprzednich spotkań. Udostępniono też możliwość dodania nowej notatki.

### 4.3. USOSweb

Implementacja funkcjonalności w USOSweb polega na rozbudowaniu go, używając najlepszych w danej chwili praktyk programowania przyjętych przez zespół tworzący USOSweb. Zachowując spójność z pozostałym kodem systemu, podjęliśmy próbę stworzenia modułu, pozwalającego studentowi na:

- Wybór jednostki w ramach której odbywa się interesujące studenta spotkanie.
- Wyświetlanie kalendarza spotkań wybranej osoby lub jednostki.
- Wyświetlanie terminów spotkań dla każdego kalendarza.
- Wykonywanie dostępnych akcji związanych ze spotkaniem:
  - Zapisywanie się na listę główną lub rezerwową.
  - Wypisywanie się.
  - Podgląd aktualnego stanu zapisu na spotkanie, w tym informacje takie jak liczba dostępnych miejsc.
- Wyświetlanie spotkań związanych ze studentem.

Dużą uwagę zwróciliśmy na niepozwolenie studentowi na wykonywanie akcji do których w danej chwili nie ma on dostępu. W tym celu przy każdym zapytaniu do serwera które powodowałoby zmianę w bazie danych - sprawdzamy czy dany student ma dostęp do tej akcji. Oczywiście moduł został zabezpieczony od potencjalnych ataków, więcej na ten temat w sekcji 4.3.7.

#### 4.3.1. Opis działania systemu USOSweb

USOSweb oparty jest o kontroler zaimplementowany w języku PHP, który dostaje w parametrach GET opis akcji, którą użytkownik chciałby wykonać. Każda akcja zaimplementowana jest jako osobna klasa języka PHP (akcja obsługuje całą logikę, ona tworzy wynikowe dane i ew. powoduje efekty uboczne). Za pomocą plików konfiguracyjnych actions.xml akcje wiążą się z szablonami (szablony odpowiadają za format wyświetlania danych, które są wynikiem akcji). Akcję mogą, a czasami muszą, korzystać z dodatkowych modułów, które tworzymy jako osobne statyczne albo niestacyjne klasy.

#### 4.3.2. Akcje

Podczas implementacji przez nas modułu spotkań, zaimplementowane zostały następujące akcje:

- **MojeSpotkaniaAction**  
Odpowiada za wyświetlanie spotkań, związanych z zalogowanym studentem
- **SpotkaniaJednostkiAction**  
Odpowiada za wyświetlanie kalendarzy spotkań wraz z terminami tych spotkań oraz związanych z nimi informacji w wybranej jednostce
- **WyborJednostkiAction**  
Odpowiada za wyświetlanie jednostek i informacji o ilości spotkań w każdej z jednostek. Przyjmuje dodatkowy parametr w GET, który włącza lub wyłącza ograniczenie listy jednostek tylko do tych, które są związane ze studentem
- **ZapisNaSpotkanieAction**  
Odpowiada za tworzenie formularza, pozwalającego na wykonywanie przez studenta dostępnych akcji z danym spotkaniem. Akcja przyjmuje w GET informację o spotkaniu, czyli identyfikator spotkania z bazy, i tworzy odpowiedni formularz dla studenta w zależności od tego, co zalogowany student może zrobić ze wskazanym spotkaniem w danym momencie.
- **ZmianaZapisuNaSpotkanieAction**  
Akcja odpowiada za zmianę stanu zapisu zalogowanego studenta na dane spotkanie. Jest to jedyna akcja w danym module, która zmienia dane w bazie danych. Dana akcja przyjmuje parametry zapytania w postaci POST i sprawdza CSRF token w celu zabezpieczenia się przed atakami CSRF. Dla każdego zapytania przed modyfikowaniem danych w bazie na początku jest prowadzony proces tak zwanej walidacji w celu uniemożliwienia wykonania niepoprawnych zmian

#### 4.3.3. Szablony

USOSweb korzysta z szablonów smarty, które wiążą się z akcjami za pomocą plików konfiguracyjnych i zapewniają odpowiednią reprezentację danych, które powstały w wyniku danej akcji. W module spotkań utworzyliśmy szablony:

- **FormularzSpotkania:** formularz, który wyświetla się w oknie modalnym i pozwala na wykonanie przez studenta dozwolonej akcji związanej ze spotkaniem

- **SpotkaniaJesdnostki**: rozwijana lista cykli, gdzie domyślnie rozwinięte są tylko aktywne cykle. Każdy cykl zawiera listę kalendarzy spotkań. Dla każdego kalendarza wyświetlany zostaje opis osoby prowadzącej spotkania i nazwa kalendarza spotkań. Zawiera on również listę dostępnych terminów i przycisk, pozwalający na pokazanie także niedostępnych terminów.
- **MojeSpotkania**: lista spotkań związanych ze studentem. Widok jest identyczny z widokiem spotkań jednostki, ale pokazywane są tylko spotkania i kalendarze związane z danym studentem, czyli takie, na które on jest albo był zapisany. Nie ma żadnych ograniczeń na jednostkę w której dane spotkanie się odbywa.
- **Spotkanie**: szablon pomocniczy używany w szablonie FormularzSpotkania.
- **WyborJednostki**: lista jednostek z informacją o ilości spotkań w danej jednostce.

#### 4.3.4. Moduły pomocnicze

Często akcje w USOSweb wykonują te same czynności: pobranie kontekstu zalogowanego użytkownika lub nazwy jednostki na podstawie kodu, albo utworzenie struktury pozwalającej na wyświetlanie tabeli z danymi na podstawie zapytania. W naszym rozwiązaniu zostały utworzone następujące moduły pomocnicze:

- **CyklDydaktyczny**: klasa jest modelem cyklu dydaktycznego. Zawiera jedną statyczną metodę pozwalającą na pobranie z bazy danych informacji o wszystkich kalendarzach spotkań w cyklu dla danej jednostki wraz z terminami spotkań i kontekstem każdego osobnego spotkania dla użytkownika. Funkcja ta korzysta ze stałej liczby zapytań do bazy danych.
- **KalendarzSpotkan**: klasa jest modelem KalendarzuSpotkań i posiada metodę statyczną, pozwalającą na pobieranie z bazy danych informacji o wszystkich kalendarzach spotkań i spotkaniach dla każdego z kalendarzy, wykonując 2 zapytania do bazy danych.
- **SpotkaniaUtils**: zestaw funkcji pomocniczych
- **SpotkanieOsoba**: klasa jest modelem opisującym stan spotkania w kontekście danej osoby. Ma dużą ilość niestatycznych funkcji pomocniczych, wykorzystywanych w wielu miejscach modułu spotkań

#### 4.3.5. Problemy i ich rozwiązania

##### Pobranie nietypowej drzewiastej struktury z danymi o spotkaniach

Widok spotkań danej jednostki oraz widok spotkań studenta został zaprojektowany w taki sposób, że dla jego realizacji potrzebowaliśmy dostać w wyniku akcji drzewiastą strukturę następującej postaci:

- Pierwszy poziom: Cykle dydaktyczne z informacją pomocniczą (nazwa, stan, kod).
- Drugim poziom: Kalendarze spotkań, gdzie każdy kalendarz wskazuje na odpowiedni cykl dydaktyczny, oraz informacje z nimi związane (nazwa, osoba),
- Trzeci poziom: Spotkania (terminy) z informacją o stanie danego spotkania w kontekście zalogowanego studenta (stan zapisu, stan spotkania, ilość zapisanych osób na liście główną oraz rezerwową, data i czas rozpoczęcia i zakończenia spotkania itd.)

Używając instrumentów językowych dostępnych w PHP taką strukturę najwygodniej przedstawić jako listę obiektów cykli, gdzie każdy z nich ma pole "kalendarzeSpotkan", będące listą obiektów kalendarzy spotkań w danym cyklu. Każdy z tych obiektów miałby pole spotkania, będące listą obiektów klasy `SpotkanieOsoba` i zawierał całą informację pomocniczą. Tylko jaki sposób wyciągnąć taką drzewiastą strukturę z bazy danych w optymalny sposób? Relacyjne bazy danych SQL pozwalają na pobranie informacji w postaci tabel. Można by pobierać listę cykli, potem dla każdego cyklu listę kalendarzy i dla każdego kalendarza - listę spotkań. Co prowadzi, to spowodowałoby gigantyczną ilość zapytań. Innym rozwiązaniem mogło być pobranie całej drzewiastej struktury w jednym zapytaniu. Jako że relacyjna baza danych z której korzystamy zwraca dane tylko w postaci tabel można wywnioskować, że pobieralibyśmy gigantyczną ilość zbędnych i duplikujących się danych o cyklach i kalendarzach. Wybraniem przez nas rozwiązaniem było wykonanie trzech zapytań: pobieranie cykli i informacji o nich, pobieranie kalendarzy wraz z informacjami o nich oraz pobranie spotkań. Po pobraniu danych wiążemy je: tworzymy opisaną wyżej drzewiastą strukturę, iterując po każdej z list i używając informacji zawartej w polach z kodem cyklu albo kodem kalendarza.

### **Podobne akcje**

W module spotkań mieliśmy 2 pary bardzo podobnych akcji: akcje spotkań studenta i spotkań w jednostce, oraz akcje wyświetlania jednostek użytkownika i akcje wyświetlania wszystkich jednostek. W przypadku każdej z tych par utworzyliśmy implementację ich wspólnej części i korzystaliśmy z niej, zapobiegając duplikowaniu kodu.

### **Wielojęzyczność**

System USOSweb wspiera 2 języki: angielski oraz polski. Dla wsparcia wielojęzyczności w naszym module korzystaliśmy z wbudowanego w szablon Smarty customowego tag-u `{t}` oraz w modelu bazy danych dla pól tekstowych takich jak nazwa albo opis tworzyliśmy duplikat w języku angielskim.

#### **4.3.6. Zmiany w USOSWebie**

##### **Wyjątki**

Przy wysyłaniu danych do serwera w postaci JSON chcieliśmy dostać informację o ewentualnym błędzie również w postaci JSON. W tym celu rozbudowaliśmy mechanizm wyjątków używany w USOSweb dodając możliwość zwracania wyniku wyjątku w postaci JSON wskazując odpowiednią opcję w momencie podnoszenia wyjątku, który jest obiektem klasy `ActionError`.

##### **Bug w Smarty**

W trakcie implementowania natknęliśmy się na błąd w implementacji customowego tagu `{textarea}` w szablonach Smarty. Po wskazaniu atrybutu `limit` w danym tagu, będąc w kontekście okna modalnego cała zawartość strony zniknęła. Zmieniliśmy kod JavaScript obsługujący wypisywanie limitu oraz pozostałych dostępnych liter, naprawiając ten błąd.

#### **4.3.7. Potencjalne ataki i zabezpieczenia**

##### **CSRF**

W celu zapobiegania potencjalnym atakom CSRF w jedynym miejscu w którym zmieniamy dane dodaliśmy sprawdzenie tokenu CSRF.



**SQL injection** Każdy parametr będący częścią zapytania do bazy jest escape-owany w celu zapobiegania atakom typu SQL-injection.



## Rozdział 5

# Potencjalny rozwój

Projekt będzie można potencjalnie rozszerzyć o integrację z mobilnym USOS. Obecna architektura pozwoli na zbudowanie na niej kontroli stanu i przepływu studentów w czasie rzeczywistym, poprzez:

- Akcje Dziekana zaznaczającego studentów których spotkanie się już odbyło.
- Akcje studentów, zaznaczające czy będą na spotkaniu i czy się na nie spóźnią.

Integracja tych rzeczy pozwoli na zminimalizowanie czasu jaki studenci muszą poświęcić czekając na swoją kolej, tym samym pozwalając Dziekanowi na kontrole przepływu studentów, dając możliwość przewidywania czasu zakończenia spotkania. Potencjalnymi rozszerzeniami są także:

- Możliwość obsługi spotkań poprzez pracowników naukowych z poziomu USOSweb.
- Obsługa konsultacji z wykładowcami/ćwiczeniowcami.
- Wykorzystanie modułu do tworzenia spotkań studenckich lub okolicznościowych, jak np. święta Bożego Narodzenia na wydziale.

Dodatkowo, możnaby połączyć spotkania z już istniejącymi w USOS funkcjonalnościami takimi jak podania, płatności czy wymiany studenckie. Osiągnąć to można chociażby poprzez pobieranie do spotkań większej ilości danych z baz danych, oraz rozszerzenie spotkań o podtypy spotkań.



## Rozdział 6

# Dodatki

6.1. Dokumentacja instalacyjna

6.2. Opis zawartości płyty CD