

**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

Grudzień 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. Ogólnie informacje o systemie USOS	5
1.3. Cel pracy	6
1.4. Struktura pracy	6
1.5. Podział pracy	6
1.5.1. Maciej Góralski	6
1.5.2. Kacper Pawelec	7
1.5.3. Aliaksei Suvorau	7
1.5.4. Michał Swidziński	7
<b>2. Wymagania użytkowników</b>	9
2.1. Wymagania	9
2.2. Rozmowa z Dziekanem	9
2.3. Ankieta dla studentów	9
2.4. Rozmowa z Sekcją Studencką	11
<b>3. Specyfikacja</b>	13
3.1. Cel projektu	13
3.2. Definicje	13
3.2.1. kalendarz spotkań	13
3.2.2. słownik	13
3.2.3. słownik uzasadnień	13
3.2.4. Okres zapisów	13
3.2.5. Okres akceptacji	13
3.3. Definiowanie spotkania	14
3.4. Zapisy na spotkanie	14
3.5. Dodatkowe wymagania	14
3.6. Formularz	15
3.7. Szczegółowa specyfikacja użytkownika	15
3.7.1. Używanie USOSadm przez zalogowanego użytkownika z odpowiednimi uprawnieniami.	15
3.7.2. Używanie USOSweb przez zalogowanego użytkownika z odpowiednimi uprawnieniami.	16
3.8. Dodatkowe wymagania	17
3.9. Schemat bazy danych	17
3.10. Przykłady	19

<b>4. Implementacja</b>	21
4.1. USOS	21
4.2. USOSadm	21
4.2.1. Informacje ogólne	21
4.2.2. Aplikacja kliencka	22
4.2.3. Serwer	22
4.2.4. Komunikacja z bazą danych	22
4.2.5. Opis implementacji	22
4.2.6. Słownik uzasadnień	22
4.2.7. Kalendarz	22
4.2.8. Notatki	22
4.3. USOSweb	26
4.3.1. Informacje ogólne	26
4.3.2. Opis działania systemu USOSweb	26
4.3.3. Akcje	27
4.3.4. Szablony	28
4.3.5. Moduły pomocnicze	28
4.3.6. Wybór jednostki	28
4.3.7. Wybór spotkania	29
4.3.8. Moje spotkania	29
4.3.9. Problemy i ich rozwiązania	31
4.3.10. Zmiany w USOSWebie	32
4.3.11. Potencjalne ataki i zabezpieczenia	32
<b>5. Potencjalny dalszy rozwój</b>	33
5.1. Mobilny USOS	33
5.2. Spotkania innego rodzaju	33
<b>6. Dodatki</b>	35
6.1. Opis zawartości płyty CD	35
<b>Bibliografia</b>	37

# Rozdział 1

## Wprowadzenie

Niektóre problemy wymagają osobistego spotkania z Dziekanem Wydziału Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego (MIM). Niniejsza praca ma na celu usprawnienie procesu zapisów na spotkanie z Dziekanem.

### 1.1. Rozwiązanie dotychczasowe

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

1. Przyjść, zadzwonić albo wysłać wiadomość e-mail do Sekcji Studenckiej (Dziekanatu).
2. Podać imię, nazwisko, numer indeksu oraz powód wizyty.
3. Student otrzymuje odpowiedź od Sekcji Studenckiej. W przypadku kiedy problem studenta wymaga spotkania z Dziekanem otrzymuje datę wizyty oraz numer w kolejce na liście wizyt. W przypadku, kiedy problem studenta może być rozwiązany inaczej otrzymuje instrukcje, co ma robić.

Studenci preferują zapisywanie się osobiście ponieważ natychmiast otrzymują pewną odpowiedź. Wizyty w Sekcji Studenckiej wymagają od studenta poświęcenia dużo czasu i dostosowania się do krótkiego czasu pracy Sekcji. Przykładowym okresem, gdy szczególnie trudno jest uzyskać kontakt z Sekcją Studencką, jest okres przedłużania ważności legitymacji studenckich. Wtedy kolejki studentów potrafią być na tyle długie, że w rozsądnym czasie student nie będzie mógł wejść do sekretariatu. Istnieje zapotrzebowanie na zmniejszenie tłoku w Sekcji Studenckiej.

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.

Ponadto różne sposoby kontaktu powodują niesprawiedliwą kolejność wizyt.

### 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,

czyli Internetowa platforma pozwalająca studentom i pracownikom uczelni na dostęp do części funkcjonalności 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.

USOS składa się między innymi z:

- 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.

### **1.3. Cel pracy**

W ramach pracy powstał moduł do systemów USOSadm i USOSweb, dzięki któremu będzie możliwe umawianie, 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 oszczędzać czas Dziekana oraz studentów. Ponadto zmniejszy się zużycie papieru.

### **1.4. Struktura pracy**

Praca składa się z 6 rozdziałów i dodatków. Wymagania i uwagi 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 dalszy rozwój nowego modułu jest opisany w rozdziale 5, a dokumentacje dotyczące wytworzonej w ramach niniejszej pracy funkcjonalności są w rozdziale 6.

### **1.5. Podział pracy**

Podział pracy w projekcie wygląda następująco:

#### **1.5.1. Maciej Góralski**

Napisał większość pracy licencjackiej.



### **1.5.2. Kacper Pawelec**

Stworzył widoki słownik uzasadnień, kalendarz w USOSadm.

### **1.5.3. Aliaksei Suvorau**

Zaprojektował bazę danych, napisał skrypt tworzący bazę danych Oracle. Zaimplementował wszystkie funkcje USOSweb.

### **1.5.4. Michał Swidziński**

Napisał skrypt tworzący bazę danych MySQL. Stworzył widok notatki w USOSadm. Napisał część pracy licencjackiej.



## 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.
- Jak najwięcej pól w widokach spraw dla Dziekana oraz Sekcji powinno mieć domyślnie wypełnione wartości w celu przyspieszenia pracy
- Proces zapisywania się do Dziekana nie powinien być łatwy (np. trzeba szczegółowo opisać temat wizyty)
- Jednocześnie można być zapisanym tylko na jedno spotkanie.
- Dziekan powinien mieć możliwość zostawienia w systemie notatki ze spotkania.
- Powinno być możliwe usuwanie 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 internetową wśród studentów MIM, aby dowiedzieć się jakie byłyby ich preferencje względem niektórych funkcjonalności i czy pokrywają się z wymyślonymi przez nas rozwiązaniami. Ankietę wypełniło 170 studentów. Ich odpowiedzi wyglądały następująco:

- Jak wcześnie przed wizytą chciałbyś/chciałabyś zapisać się do dziekana? Odpowiedziało 169 studentów.
  - Tego samego dnia — 17 — 28%.
  - Na tydzień przed — 113 — 67%.
  - Na dwa tygodnie przed — 5 — 3%.
  - Na trzy tygodnie przed — 0 — 0%.
  - Na miesiąc przed — 3 — 2%.
  - Na dłużej niż miesiąc przed — 0 — 0%.
- W wypadku gdy limit dostępnych miejsc się zapełni, chciałbyś/chciałabyś? Odpowiedziało 170 studentów.
  - Zostać zapisanym/zapisaną na najbliższy dostępny termin — 17 — 10%.
  - Zostać zapisanym na następny termin wyznaczony przez siebie — 135 — 79%.
  - Nic, chcę zapisać się od nowa samodzielnie — 18 — 11%.
- Podając powód na wizytę u Dziekana, chcę: Odpowiedziało 170 studentów.
  - Napisać powód samemu — 11 — 6%.
  - Wybrać powód z listy dostępnych możliwych powodów — 16 — 9%.
  - Mieszankę powyższych odpowiedzi — 143 — 84%.
- Czy zapisując się do Dziekana bierzesz pod uwagę sytuację, gdy nie będziesz w stanie pojawić się na wizycie? Odpowiedziało 168 studentów.
  - tak — 51 — 30%.
  - nie — 117 — 70%.
- Jeśli tak, to z jakiego powodu i z jakim wyprzedzeniem przed wizytą będziesz o tym wiedzieć? Odpowiedziało 44 studentów. Między innymi tak:
  - „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”
  - „choroba, samodzielne rozwiązanie problemu przed wizytą, poprzedniego dnia”
- Dodatkowe uwagi. Jeśli masz pomysł na wygodny dla studenta i akceptowalny przez dziekana sposób zapisów, to go opisz. Pamiętaj, że system zapisów musi być odporny na: niepotrzebne rezerwowanie wielu miejsc przez jedną osobę, „okienka” w grafiku dziekana, generowanie zbędnego ruchu w USOSweb, wymóg wielokrotnego zaglądania do USOSweb przez studenta w celu sprawdzenia, czy zapisy zakończy się sukcesem. Odpowiedziało 31 studentów. Między innymi tak:
  - „Odnośnie pytania Ź jakim wyprzedzeniem przed wizytą chciałbyś/chciałabyś zapisać się do dziekana?": Chciałbym wybrać opcję 1a dzień przed". Nie było takiej, więc wybrałem "tego samego dnia".”

- „Fajnie byłoby widzieć o której godzinie trzeba przyjść, a nie tylko wiedzieć że jest się 20 na liście.”
- „Moim zdaniem, byłoby wspaniale, gdyby osoba1 mogła w ciągu tygodnia przed wizytą wypisać się z listy, a na jej miejsce mogłaby dopisać się osoba2, która próbowała dostać się w tym terminie ale limit miejsc był już zapełniony (na przykład osoba2 mogłaby dostawać maila, że miejsce zostało zwolnione i można dopisać się).”
- „Ad pytanie 1 – 3-5 dni przed Ad pytanie 2 – chciałbym zobaczyć, jakie są najbliższe wolne terminy i wybrać któryś z nich jeśli pasuje. Zapisywanie na najbliższy z automatu jest bez sensu, bo może nie pasować). Przy braku terminów, lub zapisie na dalszy niż pożądanym, można by dodać opcję "powiadom mnie, jeśli zwolni się wcześniejszy termin". Po zwolnieniu terminu system mógłby wysłać powiadomienie mailem do kilku pierwszych osób z kolejki, można by wtedy przełożyć termin swojego spotkania na zwolniony – na zasadzie "kto pierwszy ten lepszy", w szczególności, kto kliknie w powiadomienie jako drugi zawsze może zająć poprzedni termin tego kto kliknął pierwszy :) Ad pytanie 3 – najsensowniejszą opcją wydaje mi się połączenie (1) wybór kategorii wizyty z listy ustalonej (i konfigurowalnej) przez Dziekana (tak żeby ułatwić mu to ustalanie wizyt, pisanie odpowiedzi, etc), (2) poniżej samodzielnie krótko (50-100 znaków?) opisać konkretny cel wizyty. P.S. gratuluję dobrego pomysłu na faktycznie użyteczny projekt ZPP :)”
- „Niech siedzi pani w Sekcji Studenckiej i zapisuje, żadne tam systemy informacyjne.”
- „Na wne jest strona, gdzie się zapisuje do pani prodziekana i działa bardzo dobrze. Na tej stronie wisi lista osób zapisanych na dyżur. Oczywiście w kolejności i przy zapisie wiadomo, którym się będzie. Max 30 osób pani przyjmuje i jeśli ma możliwość, to przyjmuje więcej. Jest lista rezerwowa dla zdesperowanych, którzy czekają cierpliwie jak skończy się lista podstawowa”
- „Przydałaby się możliwość wejścia na zwolnione wolne miejsce. Doświadczenia wizyt u prodziekana pokazują, że z pierwszych 10 osób pojawia się 3-5, a pozostałe miejsca są wolne - czasem przydatna byłaby opcja zapisania się rezerwowo na takie miejsce, np. w razie nagłej a pilnej sprawy. Wolę też w takiej sytuacji wejść kilka razy w USOSweb niż czekać tydzień na wizytę.”

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 w 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. Definicje

#### 3.2.1. kalendarz spotkań

Obiekt w którym definiuje się rodzaj spotkania, osobę, która prowadzi spotkanie, cykl dydaktyczny oraz jednostkę. Przykład: dziekan może w danym cyklu dydaktycznym prowadzić spotkania dziekańskie w jednym kalendarzu oraz konsultacje na dwóch różnych wydziałach w dwóch innych kalendarzach.

#### 3.2.2. słownik

Część danych przechowywanych w systemie USOS, takie że raczej rzadko zmieniają, a jednocześnie są wykorzystywane w wielu formularzach systemu. Przykładowo lista budynków i sal uczelni jest słownikiem, natomiast lista zajęć odbywanych w danej sali już nie [przewodnik].

#### 3.2.3. słownik uzasadnień

Słownik w którym są powody do zapisu na spotkanie oraz powody odrzucenia prośby o zapisanie.

#### 3.2.4. Okres zapisów

Przedział czasu w którym można składać prośby o zapisanie na spotkanie.

#### 3.2.5. Okres akceptacji

Przedział czasu w którym prośby o zapisanie na spotkanie są przeglądane i akceptowane lub odrzucane.

### 3.3. Definiowanie spotkania

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:

1. Początek okresu zapisów.
2. Koniec okresu zapisów.
3. Początek okresu akceptacji.
4. Koniec okresu akceptacji.
5. Termin spotkania.

### 3.4. Zapisy na spotkanie

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

1. Zdefiniowanie terminu spotkania.
2. Okres zapisów.
3. Okres akceptacji.
4. Spotkanie odbywa się.

### 3.5. Dodatkowe wymagania

Prośby o zapisanie się na spotkanie mogą zostać zaakceptowane, odrzucone lub mogą nie zmieścić się w limicie miejsc. Ci, których prośby nie zmieściły się w limicie, mają przez ograniczony czas możliwość przepisania swojego spotkania na inny termin, na który zapisy nie zostały jeszcze otwarte. Interesanci, których prośby odrzucono, 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 przeniesieni o jedno miejsce wyżej i, jeśli ktoś się na niej znajduje, pierwsze miejsce na liście rezerwowej zostanie przeniesione na listę główną.



### 3.6. 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.

### 3.7. Szczegółowa specyfikacja użytkowania

#### 3.7.1. Używanie USOSadm przez 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.

### **3.7.2. Używanie USOSweb przez zalogowanego użytkownika 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 końcu 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.

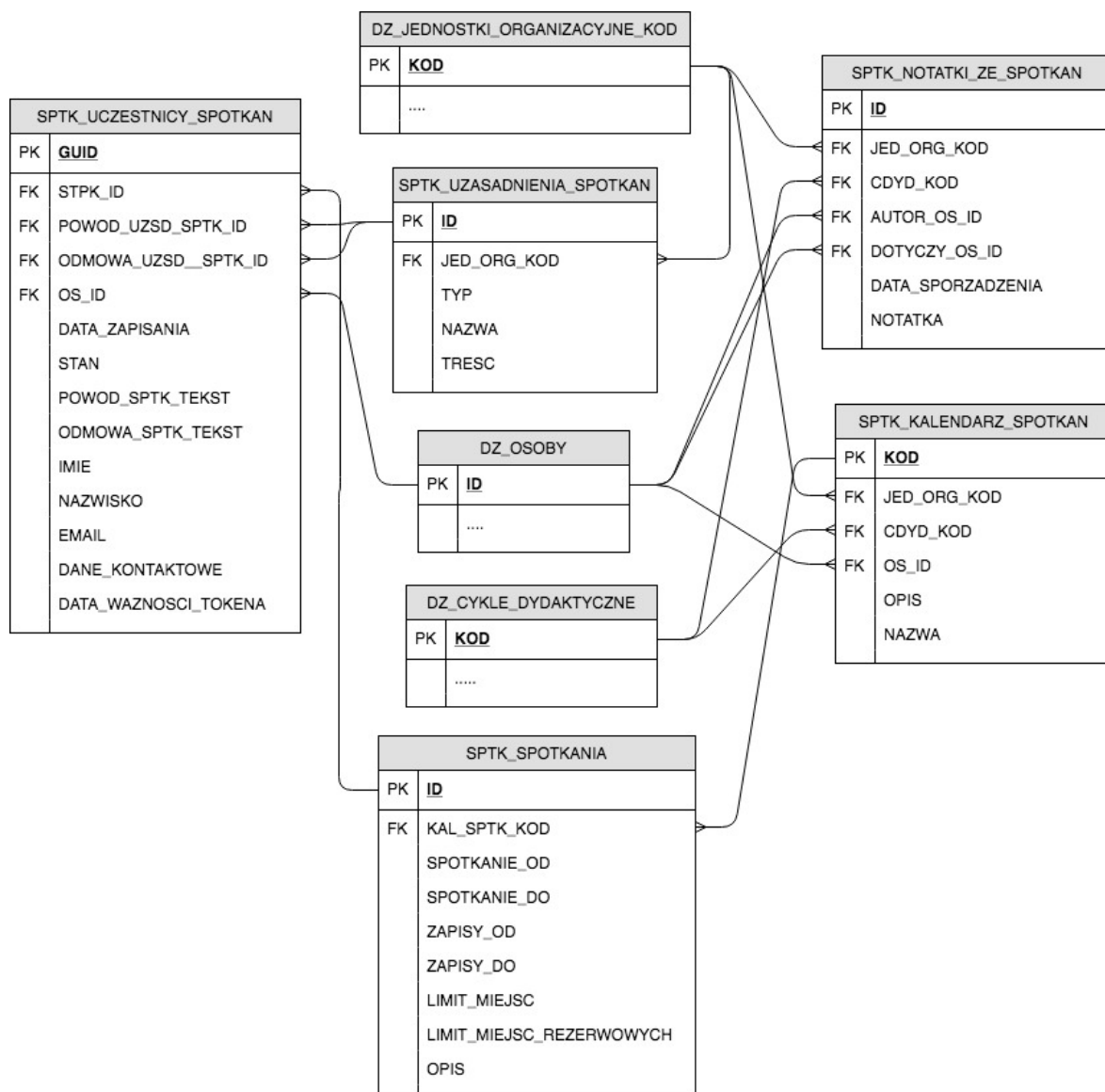
## 3.8. Dodatkowe wymagania

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).

## 3.9. Schemat bazy danych

W wyniku wymagań i potrzeb implementacyjnych, utworzyliśmy następujący schemat bazy danych, pokazany na rysunku 3.1. Kluczowe jej tabele zawierają następujące informacje:

- SPTK\_UCZESTNICZY\_SPOTKAN
  - Podstawowe dane uczestników spotkania.
  - Id spotkania.
  - Powód wraz z uzasadnieniem.
  - Odmowę wraz z uzasadnieniem.
  - Stan spotkania.
- SPTK\_SPOTKANIA
  - Kalendarz spotkań.
  - Daty spotkania i zapisów.
  - Limit miejsc.
  - Opisu.
- SPTK\_NOTATKI\_ZE\_SPOTKAN
  - Jednostka organizacyjna.
  - Cykl dydaktyczny.
  - Autor.
  - Osoba której dotyczy.
  - Data utworzenia.
  - Notatka.



Rysunek 3.1: Schemat bazy danych.

### 3.10. 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, rozwijanym przez lata. Składa się on obecnie 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.
- USOSweb – moduł przeznaczony dla studentów.
- Baza danych Oracle – centralna baza danych USOS. Posiada bogatą wewnętrzną logikę odpowiadającą między innymi za dostęp do danych. Bezpośrednio kontaktuje się z nią USOSadm.
- Baza danych MySQL – Baza danych z którą łączy się USOSweb. W większości kopia bazy danych Oracle.
- 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 spotkania 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 i dodawanie notatek ze spotkań osób.

#### **4.2.6. Słownik uzasadnień**

Widok Słownika Uzasadnień jest bardzo prosty w budowie i obsłudze.

#### **4.2.7. Kalendarz**

TODO: Wstawić screena i napisać co trzeba klikać

#### **4.2.8. Notatki**

Widok notatek w sumie też.





Dodaj uzasadnienie

?

Pomoc

Jednostka organizacyjna: \*

Kliknij przycisk, aby wybrać

Wybierz

Typ: \*

Nazwa: \*

Nazwa w języku angielskim:

Powód spotkania/odmowy spotkania:

Pozostało znaków:1000

Powód spotkania/odmowy spotkania po angielsku:

Rysunek 4.2: Formularz dodawania uzasadnienia zapisu na spotkanie.

Uniwersytet TESTMR - System Uwierzytelniania Zalogowany użytkownik: Renata Umińska | wyloguj się

Główne Osoby Pracownicy Studenci Płatności Słowniki Rejestracje żetonowe USOSmail Wymiana międzynarodowa mobilny USOS Administracja Serwis

Studenci Spotkania Notatki Zalogowany użytkownik: JMD (Rola: PELNA)

Przejdź do

Wybierz osobę wpisując PESEL, indeks bądź nazwisko i imię Numer stykowy Numer bezstykowy

Wybierz jednostkę programu Wybierz program Wybierz etap Wybierz cykl dydaktyczny -- brak filtra -- Wybierz miejsce odbywania próg Określ filtry

Lp.	Nazwisko	Imię	PESEL	Indeks główny	Program główny	Jednostka oso
1	000aaa	xxx				01010101
2	A	Yue	33253070304	1234271846	D-BWZ (Studenci, Obcokrajowcy, Erasmus, krótkoterminowi)	04010000
3	Abad Serra	Antoni	38301797813	1234304909	S2-HS (Historia sztuki, stacjonarne, drugiego stopnia)	31050000

Pierwsza << < 1 2 3 4 5 > >> Ostatnia 3

+ Dodaj Usuni wszystkie notatki o osobie

Notatki

Lp.	Cykl dydaktyczny	Jednostka organiz	Data	Autor	Notatka
1	2018Z	10000000	22.11.2018 22:47	Umińska Renata Maria	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore e...
2	2018L	01010000	16.11.2018 16:39	Umińska Renata Maria	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore e...
3	2018	01001000	16.11.2018 16:40	Umińska Renata Maria	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore e...

Pierwsza << < > >> Ostatnia 10

Logo uczelni

Informacje o uczelni

Uniwersytet TESTMR

Testowa 00-123

tel: (12) 345 67 89

www: http://www.test.edu.pl

Informacje o aplikacji

Uniwersytecki System Obsługi Studiów dla pracowników administracji (USOSadm)

Wersja 6.4.0-2 (b5f6211) - 21.12.2018

www: http://www.usos.edu.pl/

© 2019 Międzyuniwersyteckie Centrum Informatyzacji. Wszelkie prawa zastrzeżone.

Rysunek 4.3: Widok notatek ze spotkań.

Dodaj notatkę

? Pomoc

Autor: Umińska Renata Maria

Dotyczy: Abad Serra Antoni

Data: 07.01.2019 12:14

Cykl dydaktyczny: \* Kliknij przycisk, aby wybrać Wybierz

Jednostka organizacyjna: \* Kliknij przycisk, aby wybrać Wybierz

Notatka: \*

Pozostało bajtów: 4000

\* - pola wymagane

Rysunek 4.4: Formularz dodawania notatki ze spotkania.

## 4.3. USOSweb

### 4.3.1. Informacje ogólne

Implementacja funkcjonalności w USOSweb polegała na jego rozbudowaniu przy pomocy najlepszych w danej chwili praktyk programowania przyjętych przez zespół tworzący i wspierający system USOSweb. Zachowując spójność z pozostałymi częściami systemu, stworzyliśmy moduł pozwalający studentowi na:

- Wyświetlanie informacji o jednostkach oferujących spotkania.
- Wybór jednostki w ramach której odbywa się interesujące studenta spotkanie.
- Wyświetlanie kalendarza spotkań wybranej osoby lub jednostki.
- Wyświetlanie terminów spotkań dla każdego kalendarza wraz ze wszystkimi niezbędnymi informacjami (zajętość miejsc, stan zapisów...).
- 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 (czyli takich, na które student jest albo był zapisany).

Dużą uwagę zwróciliśmy na kontrolę dostępów. Czyli pozwalamy studentowi wykonywać tylko te akcje z danymi (na przykład zapisami na spotkania), do których w danej chwili on ma dostęp. W tym celu przy każdym zapytaniu do serwera, wysyłanym z aplikacji klienckiej albo przeglądarki, sprawdzamy, czy użytkownik jest zalogowany oraz czy ma dostęp do danej akcji. Nie zapomnieliśmy również o bezpieczeństwie. Moduł posiada zabezpieczenie przeciwko potencjalnym atakom, więcej na ten temat w sekcji 4.3.11. Staraliśmy maksymalizować użycie gotowych modułów, stworzonych w USOSwebie, minimalizując tym samym duplikowanie kodu i redundancję całego systemu. Często mieliśmy sytuacje, kiedy trzeba było rozbudować jakąś część systemu, żeby ona w pełni załatwiała nasze potrzeby i umożliwiała wygodne korzystanie w przyszłości.

### 4.3.2. Opis działania systemu USOSweb

Jak każda aplikacja webowa system USOSweb składa się z następujących głównych składowych.

- Część serwerowa - program działający na zdalnym komputerze podłączonym do sieci Internet (w USOSweb-ie jest zaimplementowany w PHP)
- Aplikacja kliencka - program, działający na komputerze użytkownika, łączący się z aplikacją serwerową (zaimplementowany w JS)

Do zrozumienia dalszego opisu implementacji części systemu USOSweb niezbędne są następujące definicje

- Akcja - klasa obiektów, reprezentujących interakcje części klienckiej i serwerowej. Każda zapytanie na serwer (GET, POST, ...) powinno mieć zdefiniowaną akcję, która opisuje zachowanie serwera dla danego zapytania. W sensie programistycznym każda osobna akcja jest klasą (w języku PHP) która naśladuje klasę Action.
- Użytkownik - student albo pracownik, mający konto w systemie CAS, korzystający z systemu USOSweb
- Zalogowany użytkownik - użytkownik, który wykonał procedurę logowania się. Kontekst zalogowanego użytkownika zawiera identyfikator, imię i nazwisko, oraz wiele innych pomocniczych informacji.

Aplikacja serwerowa oparta jest o kontroler, zaimplementowany w języku PHP, który dostaje w parametrach GET opis akcji, którą użytkownik chce 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 stworzymy jako osobne statyczne albo niestyczne klasy. Każdy moduł ma zestaw swoich akcji. Poza tym aplikacja serwerowa posiada statyczne klasy, znajdujące się w folderze Utils, będące zbiorem pomocniczych narzędzi (np. funkcji albo stałych).

#### 4.3.3. 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

#### 4.3.4. Szablony

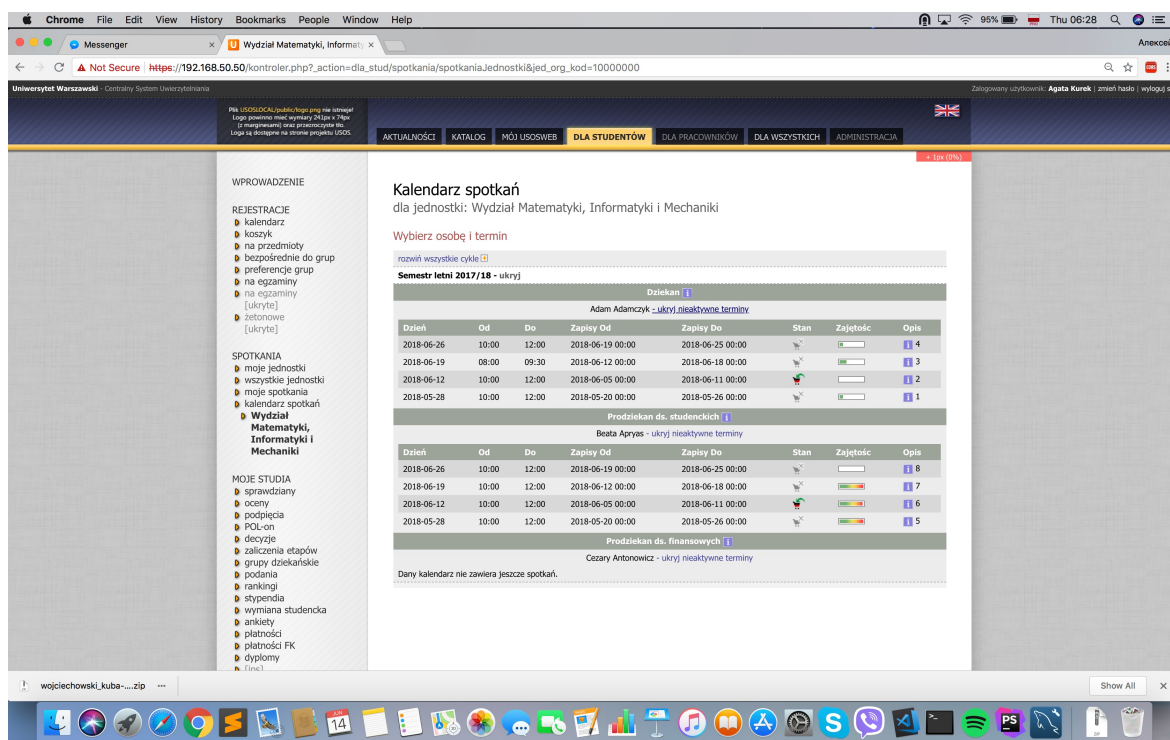
USOSweb korzysta z szablonów smarty, które są powiązane z akcjami za pomocą plików konfiguracyjnych i zapewniają odpowiednią prezentację danych, które powstały w wyniku danej akcji. W module spotkań utworzyliśmy szablony:

- FormularzSpotkania: formularz, który wyświetla się w oknie modalnym i pozwala na wykonanie przez studenta dozwolonej akcji związanej ze spotkaniem
- SpotkaniaJednostki: rozwijana lista cykli, gdzie domyślnie rozwinięte są tylko aktywne cykle. Każdy cykl zawiera listę kalendarzy spotkań. Dla każdego kalendarza wyświetlany zostaje opis osoby prowadzącej spotkanie 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.
- WybórJednostki: lista jednostek z informacją o ilości spotkań w danej jednostce.

#### 4.3.5. 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 Kalendarza Spotkań i posiada metodę statyczną, pozwalającą na pobranie z bazy danych informacji o wszystkich kalendarzach spotkań i spotkaniach dla każdego z kalendarzy. wykonuje 2 zapytania do bazy danych.
- SpotkanieOsoba: klasa jest modelem opisującym stan spotkania w kontekście danej osoby. Ma dużą ilość niestatycznych funkcji pomocniczych, wykorzystywanych w wielu miejscach modułu spotkań
- SpotkaniaUtils: zestaw funkcji pomocniczych



Rysunek 4.5: Kalendarz spotkań.

#### 4.3.6. Wybór jednostki

TODO: Wstawić screena i napisać co trzeba klikać (Może to wywalić? - Nie rozumiem tego pytania (Olek))

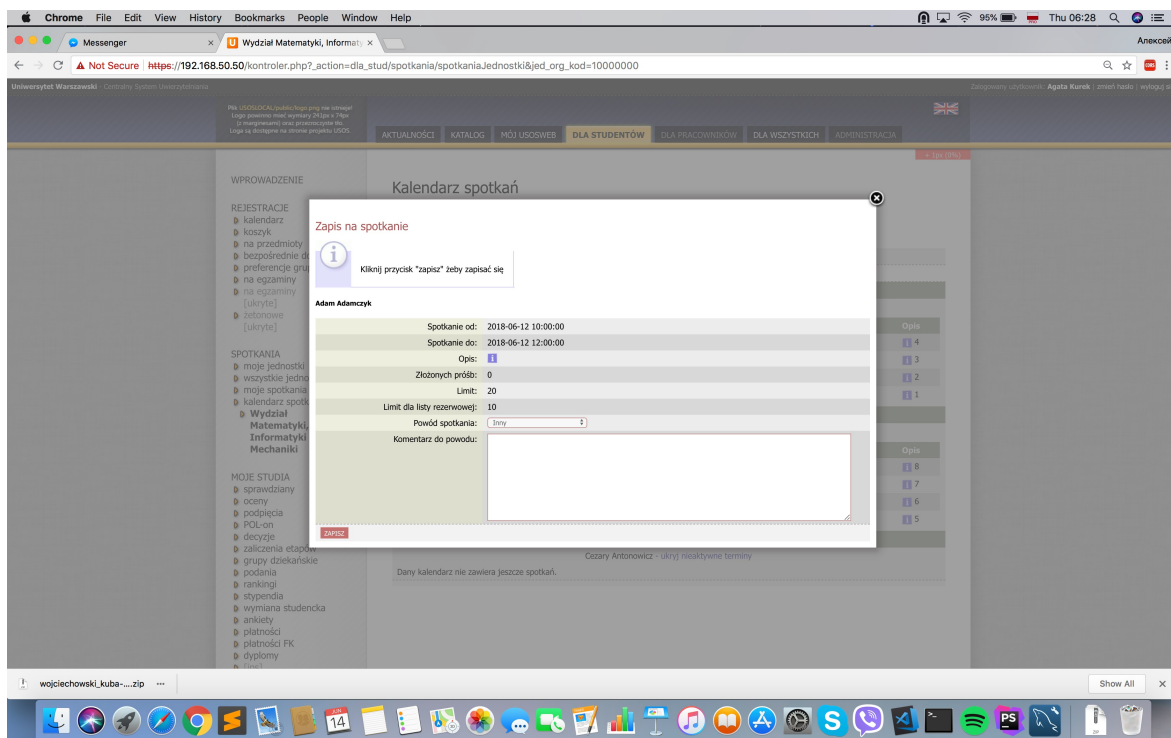
#### 4.3.7. Wybór spotkania

Na ekranie 4.5 są widoczne trzy kalendarze spotkań. Jeden z nich nie ma zdefiniowanego żadnego spotkania. Pozostałe dwa mają po jednym spotkaniu, na które można się zapisać. Aby zapisać się należy kliknąć odpowiedni przycisk.

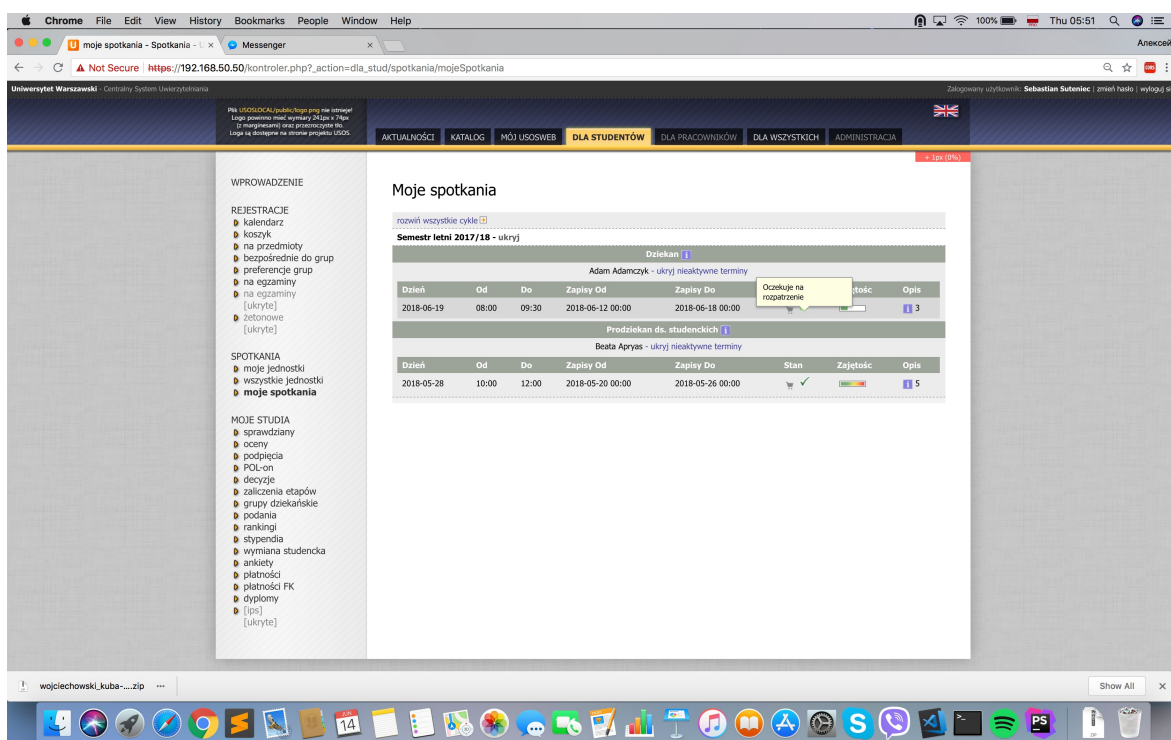
ekran 4.6: Należy wybrać powód zapisu z listy, dodać komentarz tekstowy i kliknąć przycisk „ZAPISZ”

#### 4.3.8. Moje spotkania

Na ekranie 4.7 widać spotkania, które student złożył prośby o zapisanie. Można sprawdzić, czy prośba została zaakceptowana.



Rysunek 4.6: Zapis na spotkanie.



Rysunek 4.7: Moje spotkania.



#### 4.3.9. 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 osób zapisanych 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. Relacyjne bazy danych SQL pozwalają na pobranie informacji wyłącznie w postaci tabel. Więc powstało poniższe pytanie.

##### W jaki sposób wyciągnąć taką drzewiastą strukturę z bazy danych w optymalny sposób?

- Można pobierać listę cykli, potem dla każdego cyklu listę kalendarzy i dla każdego kalendarza - listę spotkań. Co prawda, 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.
- Najlepszym 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.

Oczywiście wybraliśmy ostatni sposób, tym samym wykonując minimalną niezbędną liczbą zapytań do bazy danych, co znacząco zmniejsza czas oczekiwania aplikacji serwerowej i przyspiesza działanie całej aplikacji.

##### Podobne akcje

W module spotkań mieliśmy dwie 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 dwa języki: angielski oraz polski. Dla wsparcia wielojęzyczności w naszym module korzystaliśmy z wbudowanego w szablony smarty znacznika {t}, a w modelu bazy danych dla pól tekstowych takich jak nazwa albo opis tworzyliśmy duplikat w języku angielskim.

### **4.3.10. 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`.

#### **Błąd w smarty**

W trakcie implementowania natknęliśmy się na błąd w implementacji znacznika {textarea} w szablony 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.

### **4.3.11. 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 dalszy rozwój

### 5.1. Mobilny USOS

W ankiecie dla studentów pojawiły się odpowiedzi, że studenci chcieliby znać godzinę spotkania, a nie tylko numer na liście. Projekt będzie można rozszerzyć o integrację z mobilnym USOS. Potrzebne będzie stworzenie mechanizmów do kontroli stanu spotkania i przepływu studentów w czasie rzeczywistym. Dodanie akcji Dziekana, w której określałby, które spotkanie się rozpoczęło/zakończyło oraz odpowiednich funkcji w mobilny USOS mogłoby umożliwić śledzenie stanu spotkania przez studentów. Na przykład student jest na zajęciach, sprawdza przebieg spotkania w aplikacji mobilnej i wie, w którym momencie musi iść na spotkanie.

### 5.2. Spotkania innego rodzaju

W rozdziale 1 zauważono, że system powstał po to, aby prowadzić zapisy na spotkania z Dziekanem MIM. Dalszy rozwój systemu mógłby dopasować system również do organizacji innych wydarzeń. Wymagałoby to możliwości obsługi spotkań poprzez pracowników naukowych z poziomu USOSweb. Możliwe byłoby tworzenie wydarzeń takich jak:

- konsultacje z dydaktykami,
- spotkania studenckie lub okolicznościowe np. święta Bożego Narodzenia.



## Rozdział 6

# Dodatki

### 6.1. Opis zawartości płyty CD

TODO



# Bibliografia

[przewodnik] Jan Rudziński, *Przewodnik po systemie*, Między uniwersyteckie Centrum Informatyzacji, 2010, <https://www.usos.edu.pl/sites/default/files/pl-usos-przewodnik.pdf>.