

Dokumentacja koncepcyjna

*| Projekt: Interaktywny system*

*wspomagania układania*

*rozkładu zajęć*

Roger Barlik

Krystian Ujma

Krzysztof Wróbel

Krzysztof Nowakowski

Spis treści

[1. Wstęp 3](#_Toc419229657)

[2. HTML 5 3](#_Toc419229658)

[2.1 Opis ogólny 3](#_Toc419229659)

[2.2. Kompatybilność 3](#_Toc419229663)

[2.3. JavaScript 3](#_Toc419229664)

[2.4. Multimedia 4](#_Toc419229665)

[2.5. Element Canvas 4](#_Toc419229666)

[3. Architektura systemu 4](#_Toc419229667)

[3.1 Przepływ danych 4](#_Toc419229668)

[3.2 Diagram sekwencji 5](#_Toc419229669)

[3.3 Moduł backend 5](#_Toc419229670)

[3.3.1 Warstwa kontrolerów 6](#_Toc419229671)

[3.3.2 Warstwa Modelu 6](#_Toc419229672)

[3.4 Moduł frontend 6](#_Toc419229673)

[4. Spis funkcji 7](#_Toc419229674)

[4.1 Wykonywanie operacji na wykładowcach 7](#_Toc419229675)

[4.2 Wykonywanie operacji na awatarach 8](#_Toc419229676)

[4.3 Wykonywanie operacji na grupach studenckich 8](#_Toc419229677)

[4.4 Wykonywanie operacji na liście agregatów 9](#_Toc419229678)

[4.5 Wyświetlanie widoków 9](#_Toc419229679)

[4.6 Wykonywanie operacji na szablonach 10](#_Toc419229680)

[4.7 Wykonywanie operacji na skrzynce widomości i alertów 10](#_Toc419229681)

[4.8 Wykonanie operacji przeciągania 11](#_Toc419229682)

[5. Bibliografia 11](#_Toc419229683)

# Wstęp

Tworzona aplikacja zajmuję się obsługą użytkownika i wspieranie go w układaniu planów zajęć. W tym celu zaimplementowane zostały w niej funkcjonalności ułatwiające zarządzaniem planem w czasie układania biorąc pod uwagę preferencje użytkownika. Aplikacja składa się z dwóch modułów odpowiadających za interakcje z użytkownikiem i zapewnienie interfejsu graficznego oraz przechowywanie danych i obsługę dotyczących ich zapytań. Moduł odpowiadający za widoczną część aplikacji został stworzony w technologii HTML5, który został szerzej opisany w następnym rozdziale. Ukryta przed użytkownikiem część składa się z bazy danych Postgres, frameworku Hibernate oraz Spring.

# HTML 5

## Opis ogólny

Język HTML 5 jest rozwinięciem języka HTML 4 oraz XHTML 1. Został opracowany przez grupę roboczą WHATWG(*Web Hypertext Application Technology Working Group*) oraz W3C.

Nowa odmiana języka pozwala stosować taktyki które zostały zakazane w poprzednich wersjach. Poprawia obsługę błędów, co pozwala na wyświetlanie stron tak samo we wszystkich przeglądarkach. Umożliwia też odtwarzanie multimediów bez

konieczności używania wtyczek do przeglądarek wrodzaju wtyczki Flash. Wprowadza też istną powódź obsługiwanych przez język JavaScript właściwości, które nadają stronom interaktywność porównywalną z dedykowanym oprogramowaniem systemowym.



## Kompatybilność

Standard HTML 5 jest ciągle dość młodym standardem, dlatego nie wszystkie używane obecnie wersję przeglądarek go obsługują. Faktem dodatkowo komplikującym jest fakt ze HTML 5 składa się z grupy niezależnych standardów, które moją różną popularność i z tego powodu ich obsługa przez przeglądarki się różni. Poniżej przedstawiona jest lista przeglądarek które obsługują dużą część standardu HTML5:

• Internet Explorer 9

• Firefox 3.5

• Googlc Chrome 8

• Safari 4

• Opera 10.5

Oczywiście kolejne wersje tych przeglądarek też obsługują standard HTML5

## JavaScript

W standardzie HTML5 JavaScript stał się domyślnie używanym językiem skryptowym i niepotrzebna jest już deklaracja jego użycia. W nowym standardzie nastąpiła też większa integracja z językiem skryptowym co pozwoliło na nowe możliwości takie jak: elementy canvas, odtwarzacze multimediów, wsparcie globolokalizaji oraz składowanie danych lokalnie.

## Multimedia

Nowy element <video> jest odpowiedzialny za obsługę filmów oraz innych multimediów. Został on zaprojektowany aby zastąpić używane dotąd pluginy jak Apple Quick Time oraz Adobe Flash. Element pozwala na odtwarzanie filmów bez potrzeby użycia skryptów do detekcji rodzaju przeglądarki. Możliwe jest wskazanie kilku plików filomowych o różnych formatach i przeglądarka sama wybierze ten przez nią obsługiwany.

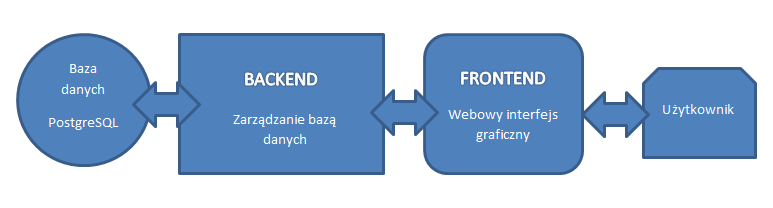
## Element Canvas

Element <canvas> jest nowym elementem wprowadzonym w standardzie HTML5. Element ten jest podobny do klasy canvas w JAVA Spring. Element pozwala na stworzenie obszaru bitmapy na którym następnie można rysować dowolne figury przy użyciu języka JavaScript.

# Architektura systemu

Aplikacja składa się z dwóch głównych modułów odpowiadających za obsługę użytkownika oraz obsługę bazy danych.

* Moduł backend - odpowiada za obsługę bazy danych i komunikację z front endem, umieszczona została tu logika biznesowa
* Moduł frontend – interfejs webowy dostępny przez przeglądarkę, odpowiada za komunikację z użytkownikiem

System jest zbudowany zgodnie z modelem MVC(Model - View - Controller) który zakłada podział aplikacji na trzy Moduły: model zawierający logikę biznesową, widok odpowiedzialny za oraz wyświetlania danych, sterownik który ma zapewnić interfejs do komunikacji z użytkownikiem. Moduły do komunikacji używają interfejsu REST Pozwala on dzięki jednorodnemu interfejsowi na późniejszą rozbudowę systemu o inne moduły oraz prostą oraz szybką komunikację pomiędzy częścią kliencką a serwerem przy pomocy obiektów JSON. Poniżej przedstawiamy ogólny schemat systemu: 

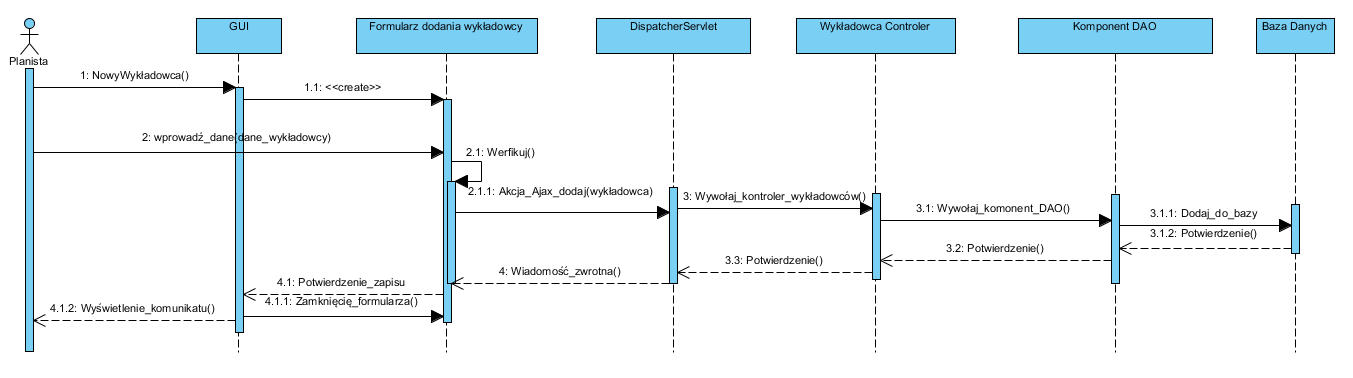
## Przepływ danych

Przepływ danych pomiędzy modułami i bazą danych wygląda następująco:

1. Informacja o wybranej akcji trafia do kontrolera menu który w razie wybrania operacji dodania wyświetla odpowiedni formularz w celu zebrania danych.
2. Wysyłane jest zapytanie REST do strony serwerowej
3. DispatcherServlet po stronie serwerowej tworzy obiekt entity i przekazuje go do kontrolera wykładowców
4. Kontroler przesyła żądanie do odpowiedniej usługi
5. Usługa wywołuje odpowiednią metodę z warstwy DAO
6. Warstwa DAO operuje na bazie danych
7. Generowany jest widok dla użytkownika

## Diagram sekwencji

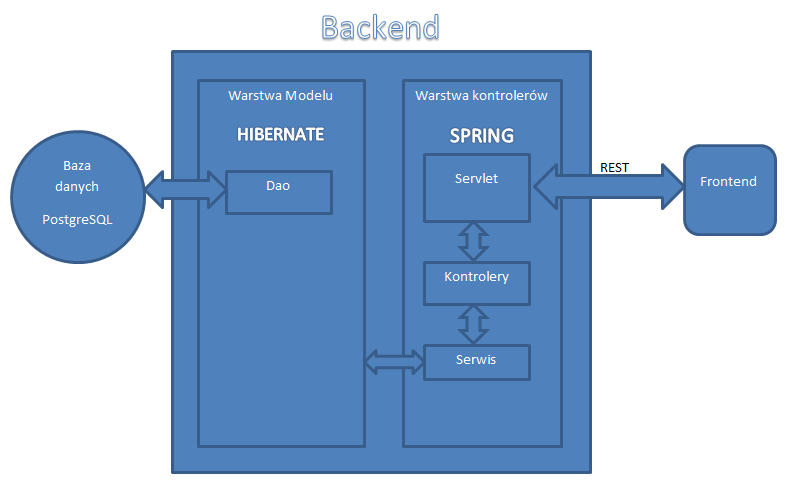
Poniżej przedstawiony jest diagram sekwencji dla operacji dodania nowego wykładowcy



## Moduł backend

Moduł serwerowy składa z logiki biznesowej napisanej w języku Java zbudowanej przy użyciu frameworku hibernate. Drugą częścią modułu jest warstwa kontrolerów zbudowana zgodnie z frameworkiem Spring. Dane użytkowane przez system są przechowywane w relacyjnej bazie danych PostgreSQL.

Sposób działania modułu back end wygląda następująco: Od strony front endu przesyłane jest zapytanie klienckie typu REST. Dane przesyłane do bazy są w formacie JSON. Zapytanie trafia do warstwy kontrolerów(Spring), gdzie jest kierowana do konkretnego kontrolera przez dispartcher servlet. Kontroler przekazuje następnie odpowiednią akcję do serwisu, który komunikuje się z warstwą modelu (hibernate), gdzie wykonywane są odpowiednie operacje na bazie. Ostatecznie dane powrotne są odpowiednio opakowywane przez serwer i odsyłane do front endu. Poniżej został przedstawiony model back endu i jego sposobu komunikacji.



### Warstwa kontrolerów

Warstwa kontrolerów oparta na frameworku Spring odpowiedzialna jest za reagowanie na przychodzące zapytania od klienta, tworzenie odpowiedzi protokołu http i odsyłanie jej do klienta. Centralnym elementem architektury Spring jest klasa DispatcherServlet, implementująca mechanizm przekierowania obsługi żądania HTTP do odpowiedniego kontrolera. Rolę zarządcy pełni zdefiniowany w pliku konfiguracyjnym web.xml serwlet będący instancją klasy DispatcherServlet. Dla każdego z rodzaju zasobów używana jest oddzielna klasa kontrolera. Każdy z kontrolerów posiada będzie odpowiednie metody odpowiedzialne za wykonanie operacji na zestawie danych. W kontrolerach konstruowane są odpowiedzi REST dla front endu, na podstawie danych dostarczonych przez logikę biznesową.

### Warstwa Modelu

Warstwa modelu odpowiedzialna będzie za wydobycie z bazy danych odpowiedniego

zasobu, wykonaniu na tych danych określonych operacji i zwrócenie ich do kontrolera. W tej warstwie działa framework Hibernate który jest odpowiedzialny za mapowanie bazy relacyjnej na obiekty oraz komponent DAO(Data Access Object) zapewniający jednolity interfejs służący do komunikacją pomiędzy aplikacją a źródłem danych.

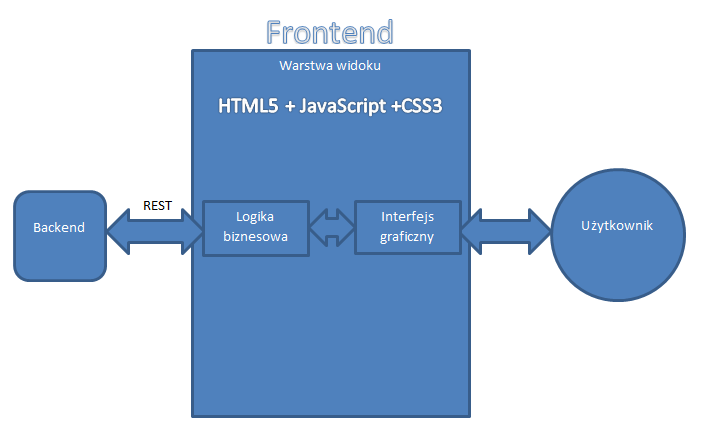
## Moduł frontend

Moduł od strony klienckiej jest interfejsem webowym dostępny przez przeglądarkę przeznaczonym dla użytkownika końcowego. Podstawowymi jego zadaniami są: zapewnieni interfejsu graficznego pobieranie danych od użytkowników, reagowanie na działania użytkownika oraz wyświetlanie danych otrzymanych od serwera.

Moduł ten został stworzony przy pomocy standardu HTML5, który pozwala na stworzenie w pełni funkcjonalnej aplikacji internetowej używając przy jej tworzeniu języka HTML oraz języka skryptowego JavaScript. Za wygląd interfejsu odpowiedzialny jest standard CSS3.

Moduł komunikuje się z częścią serwerową przy pomocy protokołu REST

Zarys modułu:



# Spis funkcji

## Wykonywanie operacji na wykładowcach

Planista ma możliwość wykonywania operacji na liście wykładowców umieszczonych w systemie. Możliwe operacje do przeprowadzenia na wykładowcach:

* Wyświetlenie listy wykładowców

Poprzez naciśnięcie na przycisk „Wykładowcy” na pasku menu istnieje możliwość wyświetlenia listy wszystkich wykładowców istniejących w systemie.

* Stworzenie nowego wykładowcy

Z poziomu listy wykładowców możliwe jest stworzenie nowego wykładowcy poprzez przycisk „Nowy wykładowca” .

* Zmiana danych wykładowcy

Z poziomu listy wykładowców jest możliwa edycja wybranego wykładowcy poprzez kliknięcie na nim oraz naciśnięcie przycisku „Edytuj”, otwiera to nowe okno w którym możemy zmienić wszystkie dane wykładowcy.

* Usunięcie wykładowcy

Z poziomu listy wykładowców można usunąć wykładowcę z sytemu, w tym celu należy kliknąć na wybranym wykładowcy i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na awatarach

Planista ma możliwość wykonywania operacji na liście awatarów umieszczonych w systemie. Możliwe operacje do przeprowadzenia na awatarach:

* Wyświetlenie listy awatarów

Poprzez naciśnięcie na przycisk „Awatarzy” na pasku menu istnieje możliwość wyświetlenia listy wszystkich awatarów istniejących w systemie.

* Stworzenie nowego awatara

Z poziomu listy awatarów możliwe jest stworzenie nowego awatara poprzez przycisk „Nowy awatar” . Możliwe jest nie wypełnianie wszystkich pól awatara podczas jego tworzenia.

* Zmiana danych awatara

Z poziomu listy awatarów jest możliwa edycja wybranego awatara poprzez kliknięcie na nim oraz naciśnięcie przycisku „Edytuj”, otwiera to nowe okno w którym możemy zmienić wszystkie dane awatara.

* Zmiana wykładowców przypisanych do awatara

W oknie edycji awatara jest możliwość zmiany wykładowców przypisanych do awatara poprzez zaznaczenia ich w liście i naciśnięcie przycisku „Usuń” lub naciśnięcie przycisku „Dodaj”, który otworzy listę z dostępnymi wykładowcami.

* Usunięcie awatara

Z poziomu listy awatarów można usunąć awatara z sytemu, w tym celu należy kliknąć na wybranym awatarze i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na grupach studenckich

Planista ma możliwość wykonywania operacji na liście grup studenckich umieszczonych w systemie. Możliwe operacje do przeprowadzenia na grupach studenckich:

* Wyświetlenie listy grup studenckich

Poprzez naciśnięcie na przycisk „grupy studenckie” na pasku menu istnieje możliwość wyświetlenia listy wszystkich grup studenckich istniejących w systemie.

* Stworzenie nowej grupy studenckiej

Z poziomu listy grup studenckich możliwe jest stworzenie nowego awatara poprzez przycisk „Nowa grupa studencka” .

* Usunięcie grupy studenckiej

Z poziomu listy grup studenckich można usunąć grupę studencką z systemu, w tym celu należy kliknąć na wybranej grupie studenckiej i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na liście agregatów

Planista ma możliwość wykonywania operacji na liście agregatów umieszczonych w systemie. Możliwe operacje do przeprowadzenia na agregatach:

* Wyświetlenie listy agregatów

Poprzez naciśnięcie na przycisk „grupy studenckie” na pasku menu istnieje możliwość wyświetlenia listy wszystkich grup studenckich istniejących w systemie.

* Filtrowanie agregatów

Poprzez wybór jednego z możliwych sposobów filtracji jest możliwe wyświetlenie tylko interesującej nas części agregatów. Używamy tej opcji poprzez wybór możliwości z listy rozwijanej filtrów oraz kliknięcia przycisku „Filtruj”.

* Stworzenie nowego agregatu

Z poziomu listy agregatów możliwe jest stworzenie nowego awatara poprzez przycisk „Nowy agregat” . Możliwe jest nie wypełnianie wszystkich pól agregata podczas jego tworzenia.

* Edycja danych agregatu

Z poziomu listy agregatów jest możliwa edycja wybranego agregatu poprzez kliknięcie na nim oraz naciśnięcie przycisku „Edytuj”, otwiera to nowe okno w którym możemy zmienić wszystkie dane agregatu.

* Usunięcie agregatu

Z poziomu listy agregatów można usunąć agregata z sytemu, w tym celu należy kliknąć na wybranym agregacie i nacisnąć przycisk „Usuń”.

## Wyświetlanie widoków

Planista ma możliwość wyświetlania różnych widoków kalendarza w różnych konfiguracjach. Możliwe interakcje z widokiem.

* Wyświetlenie widoku

Z poziomu widoku głównego możliwe jest wyświetlenie wybranego widoku, w tym celu należy z listy rozwijanej widoków wybrać intersujący nas widok. Nowy widok wyświetli się w oknie.

* Przełączenie widoku

Z poziomu widoku głównego możliwa jest zmiana jednego z aktualnie wyświetlanych widoków. W celu zmiany widoku należy wybrać nowy widok z listy rozwijanej widoków. Wybrany widok zastąpi aktualnie wyświetlany widok.

* Zmiana zakresu wyświetlanych dat

Z poziomu widoku głównego możliwa jest zmiana zakresu wyświetlanych dni. Do zmiany służą ikony: „Widok Miesiąca” zmieni widok tygodniowy na widok miesiąca, „Widok Tygodnia” zmieni widok w odwrotny sposób, strzałka „Następny” przeniesie zakres odpowiednio o tydzień lub miesiąc do przodu, strzałka „Poprzedni ” cofnie zakres o tydzień lub miesiąc, „Wybrany Zakres” pozwoli na przełączenie się na wybrany tydzień lub miesiąc.

## Wykonywanie operacji na szablonach

Planista ma możliwość wykonywania operacji na szablonach stworzonych w systemie. Możliwe operacje do przeprowadzenia na szablonach:

* Wyświetlenie listy agregatów

Poprzez naciśnięcie na przycisk „szablony” na pasku menu istnieje możliwość wyświetlenia listy wszystkich grup studenckich istniejących w systemie.

* Stworzenie szablonu

Z poziomu listy szablonów możliwe jest stworzenie nowego awatara poprzez przycisk „Nowy szablon” .

* Edycja zakresu szablonu

Z poziomu listy szablonów możliwa jest edycja zakresu szablonu, osiągamy to poprzez wybranie interesującego nas szablonu a następnie kliknięcie przycisku „Edytuj”, który otworzy formularz gdzie będziemy mieli możliwość m.in. zmiany zakresu dni jakie obejmuje szablon

* Usunięcie szablonu

Z poziomu listy szablonów można usunąć szablon z systemu, w tym celu należy kliknąć na wybranym szablonie i nacisnąć przycisk „Usuń”.

## Wykonywanie operacji na skrzynce widomości i alertów

Planista ma możliwość wykonywania operacji na wiadomościach i alertach wyświetlonych w skrzynce. Możliwe operacje do przeprowadzenia:

* Przeczytanie wiadomości

Z poziomu skrzynki widomości i alertów możliwe będzie wyświetlenie widomości od wykładowcy. W tym celu należy kliknąć na wybranej wiadomości, otworzy to pole z pełną widomością

* Usunięcie wiadomości

Z poziomu skrzynki widomości i alertów możliwe będzie usunięcie wiadomości. W tym celu należy kliknąć na wybranej wiadomości oraz nacisnąć przycisk „Usuń”

* Przeczytanie szczegółów alertu

Z poziomu skrzynki widomości i alertów możliwe będzie wyświetlenie szczegółów. W tym celu należy kliknąć na wybranej wiadomości, otworzy to nowe pole z treścią alertu.

* Zignorowanie alertu

Z poziomu skrzynki widomości i alertów możliwe będzie zignorowanie alertu. W tym celu należy kliknąć na wybranej wiadomości oraz nacisnąć przycisk „Usuń”. Usunie to alert ze skrzynki

## Wykonanie operacji przeciągania

System umożliwia wykonywania operacji przeciągania na agregatach w celu ich przemieszczania, łączenia lub wykonywania innych operacji. Możliwe operacje przeciągania:

* Przeciąganie pomiędzy polami kalendarza lub kalendarzy

Z poziomu widoku możliwe jest dowolne przeciąganie agregatów pomiędzy polami za pomocą metody przytrzymania i upuszczenia. Agregat po upuszczeniu zakotwiczy się w najbliższym polu.

* Przeciąganie z zasobnika agregatów na kalendarz

Z poziomu widoku możliwe jest przeciąganie agregatu z zasobnika agregatów na kalendarz, spowoduje to stworzenie nowego egzemplarza przeciąganego agregatu i zakotwiczeniu go na najbliższym polu.

* Przeciąganie z kalendarza na zasobnik

Z poziomu widoku możliwe jest przeciąganie agregatu z kalendarza na zasobnika agregatów, spowoduje to usunięcie przeciąganego egzemplarza agregatu i przywrócenie go do puli agregatów w zasobników.

* Zamiana agregatów miejscami w ramach tego samego typu kalendarza

Z poziomu widoku możliwe jest dowolne przeciąganie jednego agregatu na drugi w ramach tego samego typu kalendarza za pomocą metody przytrzymania i upuszczenia. Po upuszczeniu agregatu wyświetli się komunikat w celu potwierdzenia chęci zamiany agregatów miejscami. Po potwierdzeniu agregaty zamienią się miejscami

# Bibliografia

<http://dev.w3.org/html5/html-author/> - Dokumentacja standardu html5

<http://www.w3.org/TR/#tr_CSS> – Dokumentacja standardu CSS3

<http://hibernate.org/> - Główna strona frameworku Hibernate

<https://spring.io/> - Główna strona frameworku Spring