

Frontend Aplikacji Webowej

Wykonał: Jeremi Sobierski

Instrukcja obsługi

Elementy wspólne

Plan zajęć:

Plan zajęć wyświetlany jest zawsze gdy konieczne jest wyświetlenie zajęć, zaznaczenie pewnych terminów, przenoszenie zajęć, wybór terminu lub ocena terminu. Gdy należy wybrać termin - naciśnij na jeden z zaznaczonych terminów
Wybór kilku terminów - naciśnij na termin aby dodać do wyborów, a następnie zatwierdź.

Przenoszenie zajęć:

Student - przeciągnij kafelek na jeden z zaznaczonych alternatywnych terminów

Administrator - przeciągnij kafelki na żądane terminy

Następnie zatwierdź przyciskiem pod planem

Ocena terminu - użyj przycisków (+/-) aby edytować ocenę. Następnie zatwierdź.

Aby podejrzeć zajęcia znajdujące się w ocenianym terminie, kliknij wystającą część kafelka pod licznikiem.

W aplikacji mobilnej - aby poruszać się między dniami planu zajęć, dotknij po prawej lub lewej stronie kolumny, bądź przeciągnij palcem w poziomym kierunku. Aby zmienić termin zajęć, kliknij kafelek, a następnie kliknij na termin docelowy.

Wybór grup:

Część funkcjonalności wymaga od użytkownika wybrania pewnej grupy studentów. Realizuje to pole tekstowe, w którym wyszukiwać można nazwy zajęć, przedmiotów lub grup. Po kliknięciu szukaj pod polem dodane zostaną wszystkie zajęcia, przedmioty oraz grupy których nazwa pasuje do wyszukiwanego tekstu. Guzikem X można je odznaczyć. W zbiorze studentów których dotyczyć dane działanie w aplikacji (np. dodanie ankiety) będzie każdy student należący do jednej z wybranych grup oraz każdy zapisany na jeden z wybranych przedmiotów lub zajęć.

Zakładki

W aplikacji znajdują się następujące zakładki:

Wspólne:

- Ankiety
- Aktualności
- Pracownicy

Tylko dla pracowników:

- Zmiany grup

Tylko dla administratora:

- Plan

Tylko dla studenta:

- Plan zajęć
- Rejestracja
- Oceny

Ankiety

W tej zakładce studenci mogą głosować na przypisane im ankiety, a pracownicy mogą zakładać nowe, kasować stare, oraz podglądać wyniki aktywnych ankiet.

Jako student, po wejściu w zakładkę widzimy listę wszystkich ankiet do których zostaliśmy dodani przez prowadzącego, i możemy za pomocą przycisku głosuj wziąć w nich udział. Ukazuje się plan zajęć, na którym w godzinach biorących udział w głosowaniu wyświetlają się liczniki, służące do głosowania - każdy z terminów można ocenić od 0 do 6 w kwestii dogodności. Następnie wciskając głosuj nasz wynik jest zapisywany.

Jako pracownik, możemy nacisnąć zakończ obok ankiety, aby usunąć niepotrzebną już ankietę. Możemy także wybrać wynik, aby wyświetlił się plan pokazujący liczbę głosów oddanych na każdą godzinę, oraz oznaczy zwycięski termin. Aby dodać ankietę, wciskamy Dodaj, a następnie w sposób opisany w pierwszej sekcji wybieramy interesującą nas grupę studentów. To im wyświetli się ankietą po jej utworzeniu. Następnie wybieramy na planie terminy, które mają wziąć udział w głosowaniu. Na planie zaznaczone na czerwono terminy oznaczają te, gdzie któryś z wybranych studentów ma zajęcia. Ułatwi wybór wspólnego wolnego terminu dla grupy studentów. Po wciśnięciu dodaj ankietą zostanie zapisana.

Aktualności

Tu wyświetla się lista wszystkich wydziałowych aktualności

Pracownicy

Jeśli chcemy znaleźć informacje na temat konkretnego pracownika, wyszukujemy jego imię lub nazwisko w wyświetlonym pasku wyszukiwania, a następnie wybieramy szukaj. Wybieramy interesującego nas ze znalezionych pracownika aby wyświetlić informacje.

Zmiany grup

Tu wyświetla się lista wszystkich oczekujących próśb o zmianę grup, dotyczących przedmiotów które koordynujemy. Odpowiednimi przyciskami można je zatwierdzać bądź odrzucać.

Plan Administratora

Zanim wyświetli się plan należy wybrać grupę studentów (w opisany w pierwszej sekcji sposób), aby wyświetliły się ich zajęcia. Dzięki temu nie wyświetlają się wszystkie zajęcia na raz, co by czyniło plan nieczytelny i niewygodny w obsłudze. Następnie możemy przeciągać kafelki aby zmieniać godziny w jakich odbywają się konkretne zajęcia. Na dole strony znajduje się też guzik Ułóż plan, który po potwierdzeniu spowoduje zmianę wszystkich godzin zajęć na takie, aby możliwie uniknąć kolizji zakładając aktualny stan zapisania studentów na zajęcia.

Plan Zajęć (studenta)

W tej zakładce możemy zobaczyć nasz plan zajęć. Dodatkowo wybierając pod planem Zmiana Grup można zmieniać terminy zajęć na alternatywne. Zmiany będą wyszarzone dopóki nie zostaną zatwierdzone przez koordynatora przedmiotu. Wybierając pod planem Dogodność godzin możemy zmieniać nasze preferencje co do każdego terminu. Ocena od 0 do 6 wyznacza na ile dogodne by było dla nas mieć zajęcia w danym terminie. Oceny te będą wykorzystane potem aby oznaczyć podczas rejestracji każde zajęcia względem ich dogodności dla nas, oraz jako domyślny głos w ankiecie. W przyszłości można by także brać je pod uwagę podczas układania planu.

Rejestracja

Wyświetla pole wyszukiwania oraz listę zapisów na przedmioty oraz zajęcia. Za pomocą pola wyszukiwania można filtrować wyświetlane wyniki po nazwie. Rejestracja na zajęcia wizualnie różni się od rejestracji na przedmioty widocznym obok guzika indykatorem, wskazującym jak dobrze dostępne terminy zajęć pasują do preferencji godzinowych wybranych w zakładce Plan Zajęć. Aby zarejestrować się, należy wcisnąć guzik zarejestruj, a następnie, w przypadku zajęć, dokonać wyboru termin zajęć w wyświetlonym planie zajęć.

Oceny

Wyświetla listę ocen końcowych z wszystkich przedmiotów na jakie zapisany jest dany student.

Wersja mobilna

W wersji aplikacji wyświetlanej na wąskim ekranie (smartfonie), poza opisanym wyżej nieco innym interfejsem planu zajęć, zakładki nawigacyjne są domyślnie ukryte. Po wciśnięciu guzika menu w lewym górnym rogu wyświetla się menu nawigacyjne, które po wybraniu zakładki ponownie znika.

Implementacja

Aplikacja została napisana w frameworku React, w związku z czym opis implementacji skupi się na głównych komponentach aplikacji oraz Hookach.

Korzeń

Korzeniem jest komponent `<App/>`, odpowiedzialny za wyświetlanie paska nawigacji, głównego widoku oraz stopki. Posiada też on reduktor stanu z metodą `dispatch`, która poprzez kontekst przekazywana jest w dół drzewa, co umożliwia komponentom potomnym na zmianę stanu aplikacji, np aktualnego widoku.

Widoki

Każdemu widokowi w aplikacji odpowiada oddzielny komponent - zawiera on wszystkie widoczne w głównej części strony elementy, oraz odpowiedzialny jest za przechowywanie wspólnych dla tego widoku elementów stanu. Ustawiane są jako aktywny widok za pomocą akcji aplikacji `"setContent"`.

Plan zajęć

Komponent planu zajęć jest najbardziej rozbudowanym komponentem całej aplikacji. Wykorzystywany jest nie tylko przez widok planu zajęć, ale także np. przez ankiety czy zapisy. Jako właściwość przekazujemy do planu jego typ funkcjonowania: podgląd, wybór (zaznaczanie terminów), zmiana (przeciąganie kafelków), lub ocena (terminów za pomocą wyświetlanych liczników). Przekazujemy też opcjonalnie początkowy plan, wybory oraz oceny terminów. Jeśli zostaną one zmienione cały komponent ulegnie przeładowaniu. Przekazujemy też listę akcji, zawierającą nazwy i callbacki. Plan wyświetli na dole guziki oznaczone nazwami, które po wciśnięciu wywołują odpowiedni callback z mutacją stanu (zmianami jakie użytkownik wprowadził w planie, kafelki które zaznaczył), zależną od trybu funkcjonowania planu. Jest także zaimplementowany opcjonalny mechanizm imperatywnego wydobywania zmutowanego stanu. Struktura komponentu opiera się o komponenty: `<ClassBox/>` `<Collision/>` `<Class/>` i `<Rater/>`. `ClassBox` odpowiedzialny jest za termin zajęć. Reaguje na zdarzenia związane z wyborem terminu. Jeśli w danym terminie znajdują się zajęcia, to zawiera on odpowiedni komponent `Class`. Jeśli zajęć jest więcej niż jedno, to opakowuje je w komponent `Collision`.

Pole wyszukiwania

Pole wyszukiwania zaimplementowane jest jako niezależny komponent, aby można było z niego korzystać w wielu różnych widokach. Przekazywane poprzez właściwości są lista propozycji (opcjonalnie) oraz callback który otrzymuje wyszukaną wartość.

Wyszukiwanie grup

Opisywanie wyżej wybieranie grup studentów realizuje uniwersalny komponent `<GroupSelector/>`. Zawiera on głównie pole do wyszukiwania i aktywne wybory, realizując opisaną w instrukcji funkcjonalność.

Pop-up

Aby ułatwić wyświetlanie okienek typu pop-up zaimplementowany został specjalny hook - `usePopup`. Zwraca on element który należy umieścić w zwracanym elemencie oraz funkcję `show`. Gdy funkcja `show` wywołana zostanie z funkcją handler => element jako parametrem, wyświetlane jest okienko pop-upu z zadaniem elementem, oraz zwracany jest Promise. Handler ma być wywołany przez element gdy użytkownik zakończy z nim pracę, z danymi jakie otrzymał element jako parametrem. Promise jest wtedy spełniony z danymi jakie wprowadził do okienka użytkownik. Jeśli użytkownik zamknie okno, Promise ulega odrzuceniu. Daje to bardzo wygodny interfejs do pobierania danych od użytkownika za pomocą tych okienek, z wykorzystaniem już istniejących komponentów.

Pomoc

Jako że prawie każdy widok wyświetla pomoc, stworzony został hook `useHelp`, który przyjmuje jako parametr tekst pomocy, i realizuje wyświetlanie w rogu aplikacji guzika otwierającego pomoc z zadaniem tekstem.

Komunikacja z Backendem

Wszystkie zapytania do backendu zostały umieszczone w pliku `Remote.js` w formie funkcji asynchronicznych. Dodatkowo udostępniony jest hook `useRemoteData`, obsługujący ich interfejs.

Aplikacja Desktopowa

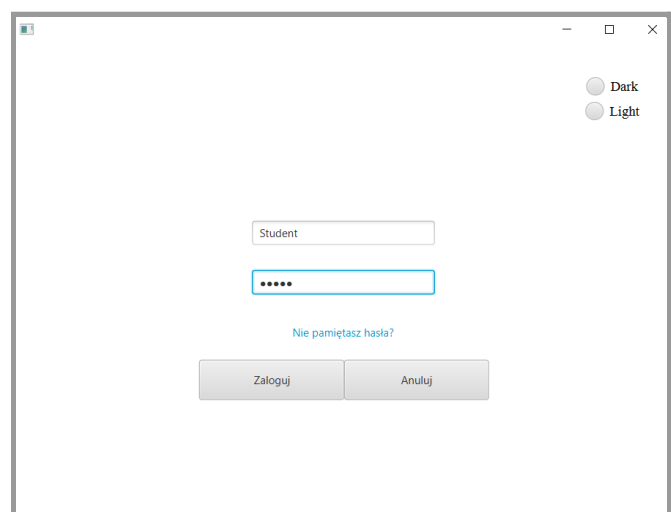
Wykonali: Michał Kopeć, Wiktor Pytlewski

Założenia

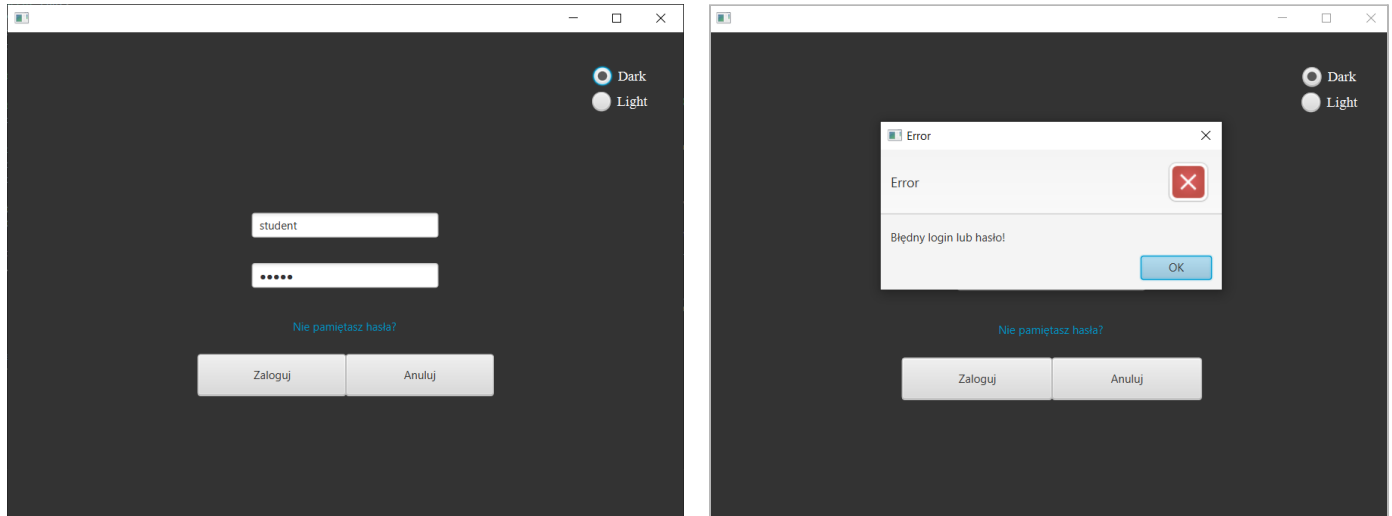
Aplikacja desktopowa jest przeznaczona wyłącznie dla Studentów.

Logowanie

Logowanie wykorzystuje zapytania http, w celu uzyskania informacji o poprawności danych logowania. Jeśli podane dane okażą się błędne zostanie wyświetlona informacja o



niepoprawnych danych logowania. Na ekranie logowania w prawym górnym rogu znajduje się przełącznik służący do przełączania trybu pomiędzy trybem ciemnym, a jasnym.



Strona główna

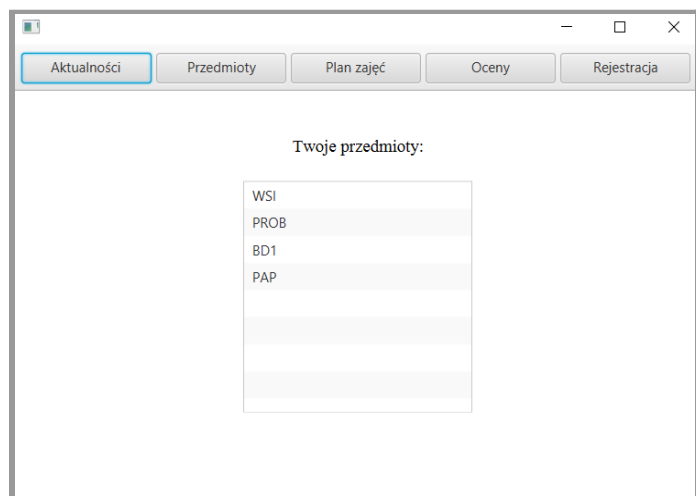
Strona główna aplikacji posiada na górze ekranu pasek służący do przełączania się między poszczególnymi zakładkami takimi jak: aktualności, przedmioty, plan zajęć, oceny, rejestracja.

Aktualności

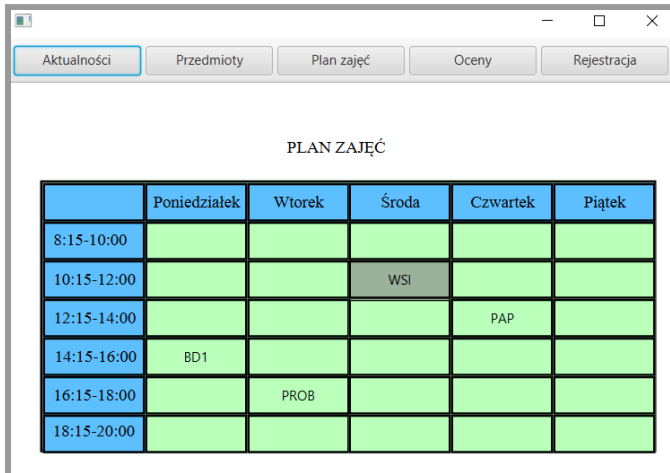
Zakładka z aktualnościami jest w tym momencie jedynie fundamentem pod rozbudowę i jest to jedna z opcji dalszej rozbudowy aplikacji.

Przedmioty

Strona przedmiotów wyświetla informacje na jakie zajęcia uczęszcza student. Jest to lista poprzez którą można przenieść się na stronę przedmiotu poprzez kliknięcie na jego nazwę.



Plan zajęć



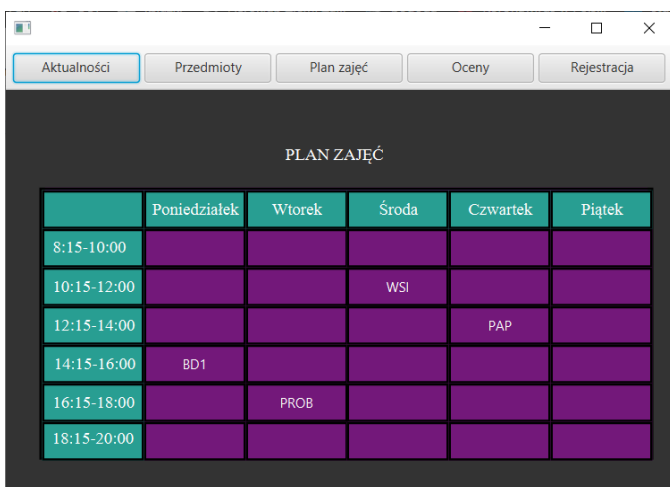
The screenshot shows a web application window titled 'PLAN ZAJĘĆ'. It has a navigation bar with buttons: 'Aktualności', 'Przedmioty', 'Plan zajęć' (selected), 'Oceny', and 'Rejestracja'. The main content area displays a table with columns for days of the week (Poniedziałek, Wtorek, Środa, Czwartek, Piątek) and rows for time slots (8:15-10:00, 10:15-12:00, 12:15-14:00, 14:15-16:00, 16:15-18:00, 18:15-20:00). The table contains the following data:

	Poniedziałek	Wtorek	Środa	Czwartek	Piątek
8:15-10:00					
10:15-12:00			WSI		
12:15-14:00				PAP	
14:15-16:00	BD1				
16:15-18:00		PROB			
18:15-20:00					

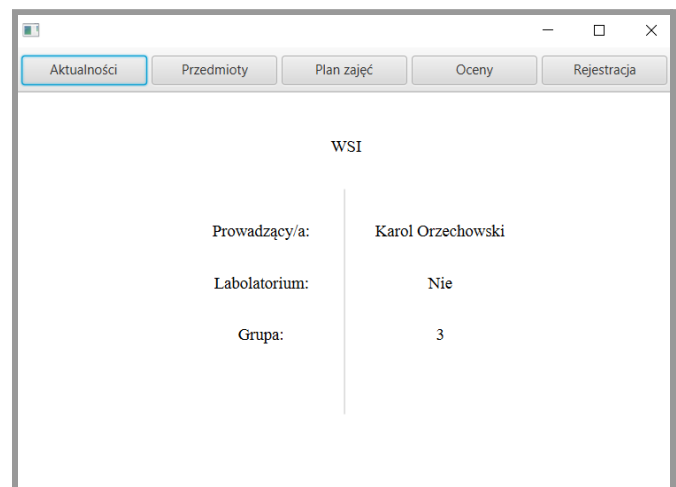
Wyświetlane są zajęcia studenta w 30 możliwych przedziałach czasowych.

Po najechaniu kursorem na odpowiednie zajęcia i kliknięciu w nie, aplikacja przenosi użytkownika do strony przedmiotu, zawierającej informacje o tym przedmiocie takie jak:

- Informacja o prowadzącym zajęcia
- Informacja czy przedmiot prowadzony jest z wraz z laboratoriami/ćwiczeniami
- Numer grupy zajęciowej



This screenshot shows the same 'PLAN ZAJĘĆ' window as above, but with a dark theme. The table data is identical to the previous one.

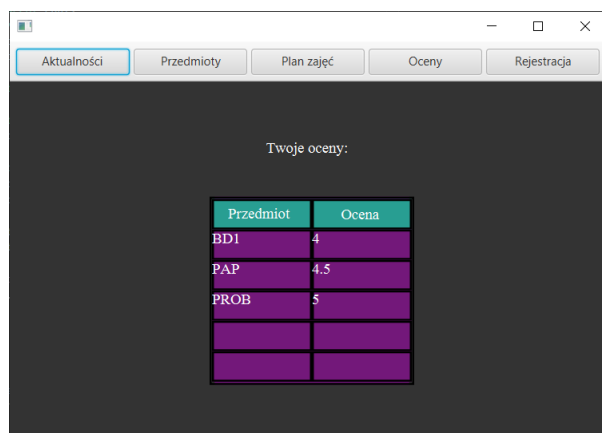


The screenshot shows a web application window titled 'WSI'. It has the same navigation bar as the previous windows. The main content area displays the following information:

WSI	
Prowadzący/a:	Karol Orzechowski
Laboratorium:	Nie
Grupa:	3

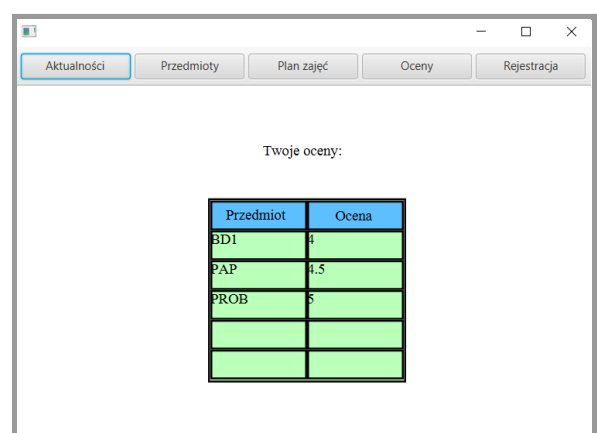
Oceny

Zakładka oceny wyświetla oceny końcowe z poszczególnych przedmiotów, na które student jest zapisany.



The screenshot shows a web application window titled 'Twoje oceny:'. It has the same navigation bar. The main content area displays a table with the following data:

Przedmiot	Ocena
BD1	4
PAP	4.5
PROB	5



This screenshot shows the same 'Twoje oceny:' window as above, but with a light theme. The table data is identical to the previous one.

Instrukcja obsługi

Aplikacja wymaga włączonego serwera, który odpowie na wysyłane zapytania **http** o autoryzację użytkownika. Jeżeli podana nazwa użytkownika i hasło są nieprawidłowe, to zostanie wyświetlone okienko z informacją o błędzie. Po zalogowaniu użytkownik ma dostęp do wszystkich wyżej wymienionych funkcjonalności.

Baza Danych

Wykonali: Michał Kopeć, Wiktor Pytlewski

Tabele bazy:

- **Location** - tabela zawierająca adresy, które odpowiadają adresowi wydziału, adresowi uczelni, czy adresowi zamieszkania.
- **Department** - tabela wydziałów, zawierająca jego nazwę oraz referencje do tabeli adresów.
- **Institutions** - tabela instytutów, posiada nazwę oraz referencję do wydziału pod jaki podlega.
- **People** - tabela zawierająca informacje o osobach w naszej bazie.
- **Student** - tabela studentów, zawiera takie informacje jak pesel, numer indeksu studenta, czy semestr studiów na którym obecnie się znajduje.
- **Worker** - tabela pracownika, posiada między innymi informacje o stopniu naukowym danego pracownika, czy instytucie w jakim pracuje.
- **Classroom** - tabela klasy, zawierającej numer, czy przybliżoną lokalizację.
- **University** - tabela uniwersytetów wraz z informacjami o nich.
- **Subject** - tabela zawierająca informacje o prowadzonych przedmiotach.
- **Time-slots** - tabela definiująca przedziały czasowe.
- **Classes** - tabela zawierająca informacje o prowadzonych zajęciach.
- **Applications** - tabela zawierająca informacje o aplikacjach dostępnych dla studentów.
- **Stud_subjects** - tabela łącząca studentów z ich przedmiotami.

- **Time_slots_ratings** - tabela zawierająca oceny wszystkich możliwych przedziałów czasowych.
- **Polls** - tabela zawierająca informacje o ankiecie.
- **Poll_time** - tabela definiująca możliwe przedziały czasowe w ankiecie.
- **Poll_time_ratings** - tabela zawierająca oceny każdego przedziału czasowego w danej ankiecie.
- **Request_change_group** - tabela zawierająca prośby o przeniesienie do innej grupy zajęciowej.
- **Users** - tabela łącząca z loginy użytkowników z odpowiadającymi im peselami osób.
- **Authorities** - tabela zawierająca uprawnienia użytkowników.
- **Stud_classes** - tabela łącząca studenta z zajęciami, na które jest zapisany.

Zapytania do bazy danych

Backend aplikacji wysyła poniższe zapytania SQL w celu uzyskania informacji z poszczególnych tabel bazy danych.

COURSES

Otrzymując login użytkownika zwraca listę zawierającą numer identyfikacyjny oraz nazwę przedmiotu.

CLASSES

Otrzymując login użytkownika zwraca listę zawierającą numer identyfikacyjny zajęć, nazwę przedmiotu, rodzaj, numer pokoju oraz czas w którym się odbywając.

ALTERNATIVES

Otrzymując numer identyfikacyjny zajęć zwraca wszystkie alternatywne terminy, na które student może się zapisać.

RATINGS

Otrzymując login użytkownika zwraca listę ocen wszystkich przedziałów czasowych tego użytkownika.

ADD RESCHEDULE

Do tabeli próśb o zmianę zajęć dodaje prośbę użytkownika o zmianę terminu zajęć

SET RATING

Otrzymując login użytkownika, przedział czasowy, oraz ocenę tego przedziału dodaje wiersz z takimi danymi do tabeli ocen przedziałów czasowych.

RESCHEDULE

Otrzymując numer identyfikacyjny prośby o zmianę, aktualizuje tabele zajęć użytkownika.

DEL REQUEST

Otrzymując numer identyfikacyjny prośby o zmianę usuwa ją z tabeli próśb o zmianę terminu zajęć.

DEL POLL

Otrzymując numer identyfikacyjny ankiety usuwa ją z tabeli ankiet.

ADD POLL

Otrzymując login użytkownika oraz nazwę nowej ankiety, dodaje wiersz z takimi danymi do tabeli zawierającej ankiety.

ADD SLOT TO POLL

Dodaje przedział czasowy do możliwych przedziałów czasowych w ankiecie.

REGISTER

Otrzymując login użytkownika oraz numer identyfikacyjny przedmiotu dodaje wiersz do tabeli zajęć użytkownika.

GET REGISTER

Zwraca wszystkie możliwe zajęcia na które student może się zapisać.

POLL RESULT

Zwraca wyniki ankiety.

SET POLL RATING

Otrzymując login użytkownika, przedział czasowy, numer identyfikacyjny ankiety, oraz ocenę dodaje wiersz z takimi danymi do tabeli ocen slotów w ankietach

GRADES

Otrzymując login użytkownika, zwraca nazwę przedmiotu i otrzymaną ocenę.

REQUESTS

Otrzymując login użytkownika, który jest prowadzącym zajęcia, zwraca wszystkie skierowane do niego prośby o zmianę grupy zajęciowej.

POLLS

Otrzymując login użytkownika zwraca listę z nazwą ankiety oraz jej numerem identyfikacyjnym.

POLL SLOTS

Otrzymując login użytkownika oraz numer identyfikacyjny ankiety zwraca listę zawierającą przedział czasowy oraz jego ocenę.

Backend

Wykonał: Karol Orzechowski

Wstęp

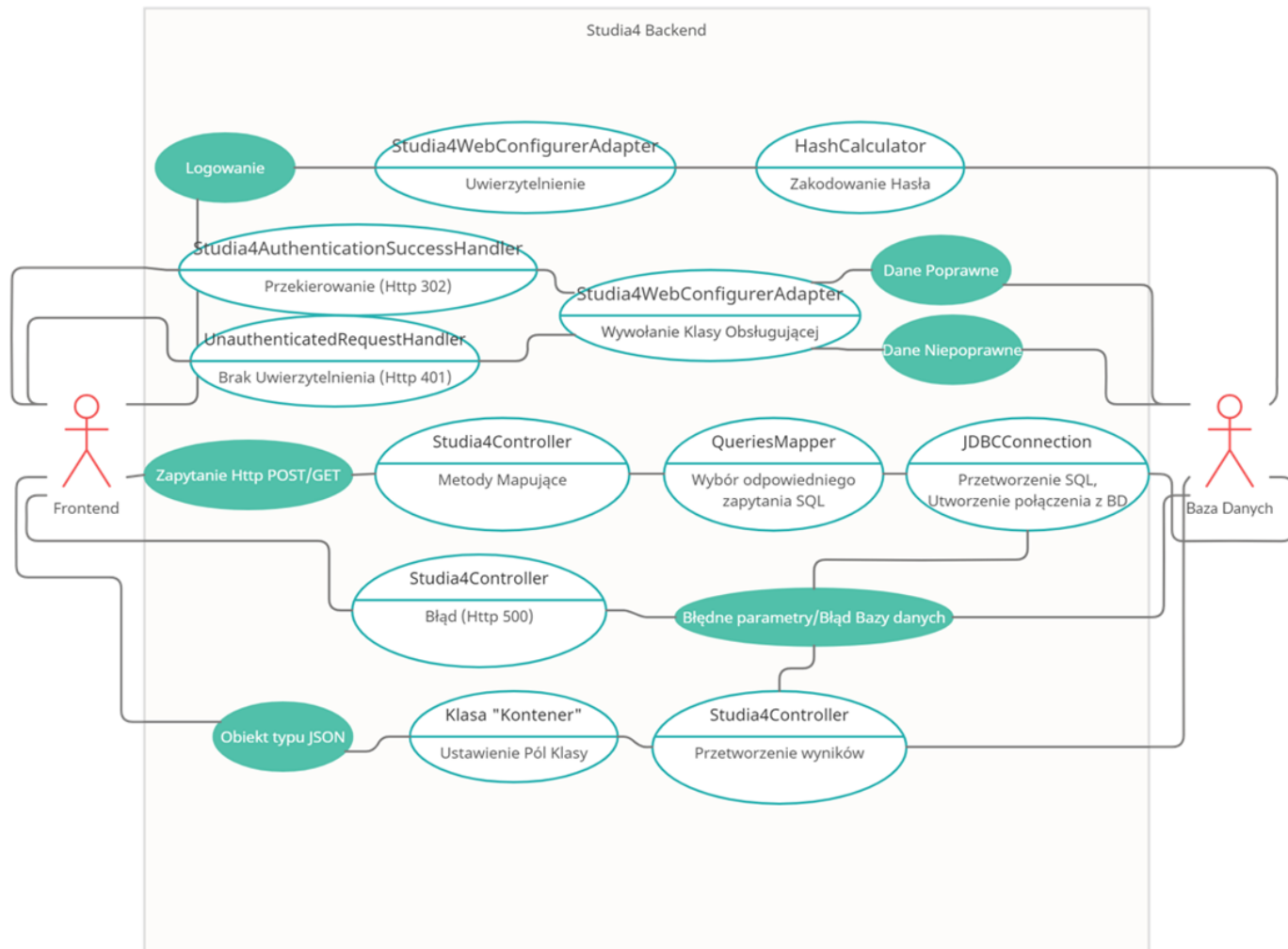
Backend został napisany w formie RESTful api – to znaczy odpowiada on za obsługę zapytań http, wysyłając odpowiednie obiekty typu json zamiast odpowiedniej strony. Serwer działa z użytkownikiem za pomocą utworzonej sesji http. Drugą rolą serwera jest również wystawianie strony internetowej na świat, znajdującej się w jego plikach statycznych. Przetwarzanie zapytań odbywa się z połączeniem z bazą danych Oracle. Postawą backendu jest framework Spring Boot.

Zewnętrzne biblioteki

Wszystkie zewnętrzne zależności zostały dodane za pomocą mavena. Backend do działania wykorzystuje zagnieżdżony serwer http Tomcat (wchodzący w pakiet spring boot) oraz HikariCP do tworzenia puli połączeń z bazą danych. Spring Security 5 odpowiada za zabezpieczenie systemu.

Uproszczony schemat działania backendu

(z tyłu agenta akcje wchodzące, z przodu akcje wychodzące)



Dodatkowy opis wybranych klas

- **DataSourceConfig** - klasa odpowiadająca za konfigurację połączeń z bazą danych (źródło danych) oraz udostępniająca możliwość utworzenia połączenia. Głównymi opcjami konfiguracyjnymi jest nazwa użytkownika oraz hasło wymagane do uwierzytelnienia w bazie danych oraz maksymalna liczba jednorazowych połączeń tworzonych przez backend.
- **JDBCConnection** - jest to interfejs na metodę `getConnection` z klasy konfiguracyjnej źródła danych. Umożliwia podanie odpowiednio przygotowanego zapytania SQL podanego jako łańcuch znaków oraz parametrów w celu przetworzenia go w odpowiedni obiekt obsługiwany przez klasę łączącą się ze źródłem danych. Nie posiada automatycznego zamykania połączeń.

- Klasy z pakietu `utilty.schedule` - napisane przez Jeremiego Sobierskiego, odpowiadają za układanie planu zajęć za pomocą algorytmu Backtracking z AC3.
- `HashCalculator` - odpowiada zastosowanie kodowania mieszającego w standardzie SHA3-512 do zabezpieczenia haseł przechowywanych w bazie danych.
- `QueriesMapper` - Enum przechowujący zapytania do źródła danych (autorstwa głównie bazodanowców) odpowiednio uzupełnionymi znakami zapytania w miejsca parametrów liczbowych oraz '?' w miejsca parametrów tekstowych.
- `Studia4WebConfigurerAdapter` - klasa odpowiedzialna za uwierzytelnianie użytkowników, nie stosuje zabezpieczenia przed atakiem CSRF (brak obsługi odpowiednich tokenów).
- `Utility.exception` - w tym pakiecie znajdują się dwa własne wyjątki sygnalizujące błędy związane z parametrami zapytań.
- `ParametersValidator` - klasa sprawdzająca poprawność podanych parametrów, aktualnie sprawdza jedynie poprawność parametrów liczbowych (wszystkie parametry trafiają do backendu jako łańcuchy znaków).
- `PeselExtractor` - na podstawie nazwy użytkownika zalogowanego w danej sesji, uzyskuje jego numer pesel z bazy danych.

Modułowość Backendu

Architektura rozwiązania umożliwia prostą rozbudowę aplikacji, poprzez dodanie obsługi nowych zapytań http w klasie `Studia4Controller` oraz dodania odpowiednich im zapytań do bazy danych do `QueriesMapper'a`. W przypadku niestandardowych zapytań pod względem podawanych parametrów wystarczy oddać odpowiednią metodę do klasy `JDBCConnection`.

Testy Aplikacji Webowej

Zostały przeprowadzone testy funkcjonalne frontendu oraz backendu. Kolejno sprawdzana była każda funkcjonalność udostępniana przez interfejs poprzez jej uruchomienie, weryfikację wysłania odpowiednich zapytań do backendu oraz poprawnego przetworzenia danych, a

następnie sprawdzenie poprawności rekordów w bazie danych po potencjalnych zmianach. Po wprowadzeniu zmian testy wszystkich powiązanych części aplikacji przeprowadzane były ponownie.

Rozbudowa rozwiązania

Ograniczony czas wymógł okrojenie rozwiązania. W przypadku dalszej rozbudowy systemu zostałyby poprawione zabezpieczenie aplikacji (obsługa tokenów csrf) ewentualnie zastosowanoby model JPA zamiast JDBC w połączeniach z bazą danych. Dodatkowo rozszerzono by funkcjonalność algorytmu układającego plan zajęć o uwzględnianie preferencji czasowych studentów.