

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

Architecture Notebook

1. Purpose

Celem dokumentu jest opis decyzji, ograniczeń oraz uzasadnień decyzji dla istotnych elementów systemu, które mają istotny wpływ na projekt oraz implementację systemu.

2. Architectural goals and constraints

W przeprowadzonej analizie biznesowej i systemowej (*pliki powstałe podczas tego procesu dostępne są w folderze abis*) zdefiniowana została zarówno dziedzina jak i wymagania, które powinna spełniać zaplanowana architektura systemu. Ustalone wymagania pomagają zrealizować ogólny cel systemu, jakim jest efektywne planowanie powierzeń zapewniające pensum wszystkim pracownikom.

Wymagania funkcjonalne:

- CRU planu powierzeń zajęć z uwzględnieniem ograniczeń wynikających z rozporządzeń wewnętrznych.
- Powiadomienie o powierzeniach i ich aktualizacjach (np. wiadomości wysyłane do opiekunów kursu o zmianie obsady zajęć towarzyszących). Zatwierdzanie/Sugestie zmian w propozycji powierzeń.
- Rekomendacje pracowników do prowadzenia zajęć na podstawie danych historycznych, preferencji, istniejących ograniczeń
- Definicja preferencji przez pracowników.
- Raportowanie: zajęć bez obsady, aktualnych powierzeń pracowników, aktualnych planów zajęć pracowników
- Wsparcie dla procesu opiniowania nauczycieli akademickich do prowadzenia zajęć przez Dziekana oraz specjalistów z zewnątrz przez Komisję Programową.

Ograniczenia systemu:

- System wymaga stałego dostępu do Internetu.
- Poprawność działania systemu jest zagwarantowana dla przeglądarki Google Chrome od wersji 78.

Wymagania niefunkcjonalne, jakie powinna zagwarantować architektura to:

- Tworzenie/edycja dokumentów tylko przez uprawnione osoby
- Kontrola zmian (rejestracja daty, rodzaju zmiany i osoby, która ją wykonała) w planach/programach/kartach przedmiotów/planach powierzeń,
- Logowanie on-site,
- Dwie wersje językowe interfejsu użytkownika: polski i angielski,

<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

W realizacji funkcjonalności systemu konieczna będzie również integracja z istniejącymi już systemami zewnętrznymi. Będą to:

- System kadrowy
 - Pobranie informacji o wszystkich pracownikach. Pobieranie i aktualizacja danych wykonywane jest na żądanie.
- System zapisów
 - Pobieranie informacji o wszystkich kursach w semestrze. Pobieranie danych wykonywane jest na żądanie.
 - Pobranie informacji o liście obszarów wiedzy.
- System uwierzytelniania - Firebase

3. Decisions and justifications

Cel	Sposób osiągnięcia
Zapewnienie ochrony przed nieuprawnionym dostępem oraz modyfikacji danych.	<ul style="list-style-type: none"> • Uwierzytelnianie - użycie Firebase jako dostawcy tożsamości w oparciu standard OAuth 2.0 • Autoryzacja <ul style="list-style-type: none"> ◦ przydzielenie ról użytkownikom w oparciu o model RBAC ◦ kontrola nad dostępem uprawnionych osób do zasobów przy pomocy Spring Security • Zwiększenie ochrony przed atakami SQL Injection dzięki wykorzystaniu Spring Data. • Zwiększenie ochrony przed atakami Template Injection dzięki mechanizmom dostępnym na platformie Angular.
Dostępność systemu w 2 wersjach językowych (angielski oraz polski) z możliwością późniejszego rozszerzenia.	<ul style="list-style-type: none"> • Tłumaczenie danych statycznych interfejsu użytkownika dzięki oddzielnym plikom z tłumaczeniami, a następnie ich wykorzystanie przy budowaniu aplikacji dla różnych języków. Zbudowane w ten sposób aplikacje będą osiągalne pod odpowiednimi ścieżkami URI. • Możliwość wprowadzania tłumaczeń dla danych dynamicznych i ich przechowywanie w bazie.
Możliwość audytu przeprowadzonych akcji w systemie	<ul style="list-style-type: none"> • Logowanie akcji prowadzonych przez użytkowników

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

4. Architectural Mechanisms

Wykorzystanie Firebase jako dostawcy tożsamości w oparciu o standard OAuth 2.0

Uwierzytelnienie użytkowników w systemie zostanie zrealizowane z wykorzystaniem serwisu Firebase, który został utworzony przez Google. Serwis ten wykorzystuje standard OAuth 2.0. Jest to otwarty standard autoryzujący, pozwalający użytkownikom udostępniać aplikacjom i stronom trzecim informacje przechowywane u innych dostawców usług. Standard ten jest szeroko wykorzystywany przez największe firmy IT m.in. przez Amazon, Google, Facebook, Microsoft oraz Twitter. Dzięki wykorzystaniu OAuth aplikacja nie ma styczności z danymi uwierzytelniającymi użytkowników oraz istnieje możliwość wykorzystania jednego serwera autoryzującego do chronienia wielu różnych zasobów. Uwierzytelnienie w serwisie Firebase jest rozwiązaniem darmowym, niezależnie od liczby użytkowników. Firebase umożliwia również łatwą skalowalność oferowanego rozwiązania.

Integracja aplikacji z serwisem Firebase zostanie przeprowadzona z wykorzystaniem FirebaseSDK poprzez wykorzystanie biblioteki AngularFire2, która przystosowuje FirebaseSDK do mechanizmów wykorzystywanych w frameworku Angular. Uwierzytelnienie będzie się odbywać poprzez email oraz hasło.

Zastosowanie mechanizmu kontroli dostępu w modelu RBAC

Jako mechanizm kontroli dostępu zostanie wykorzystany model Role-Based access control. Jest to mechanizm oparty na rolach. Role jakie będą istnieć w aplikacji to:

- Prowadzący zajęcia
- Pełnomocnik dziekana
- Komisja programowa

Role zostaną przypisane do użytkowników w momencie tworzenia konta. Dane te zostaną ustawione za pomocą mechanizmu claims, dostarczonego przez wykorzystywany system Firebase.

Kontrola nad przepływem danych z serwera z wykorzystaniem Spring Security

Do kontroli nad przepływem danych z serwera zostanie wykorzystany moduł platformy Spring dostarczający metody autentykacji i autoryzacji. Spring Security pozwala na ochronę przed atakami takimi jak session fixation, clickjacking, czy cross site request forgery. Aplikacja będzie operować na danych wrażliwych, co dodatkowo zwiększa potrzebę ochrony przed nieautoryzowanym dostępem do informacji. Każdy endpoint w aplikacji będzie zabezpieczony przed nieuprawnionym dostępem. Wszystkie żądanie kierowane do serwera będą musiały zawierać nagłówek Authorization z tokenem Firebase. Następnie token ten będzie weryfikowany poprzez Firebase Admin SDK. Dane użytkownika będą dekodowane z tokenu i na ich podstawie będą udostępniane odpowiednie zasoby. W przypadku błędów weryfikacji żądanie zakończy się błędem.

Wykorzystanie modułu Spring Data jako dostawcy JPA.

Dzięki wykorzystaniu modułu Spring data z platformy Spring zmniejszona zostanie podatność aplikacji na ataki SQL injection. Umożliwia ona pisanie bezpiecznych zapytań do bazy danych. Aplikacja będzie operować na danych wrażliwych, co dodatkowo zwiększa potrzebę ochrony przed nieautoryzowanym dostępem do informacji. Jeśli strona serwerowa nie będzie w stanie obsłużyć zapytania, do klienta zostanie wysłana tylko krótka informacja o błędzie, bez wysyłania szczegółowej informacji, co ma zapobiec poznaniu struktury projektu przez osoby nieuprawnione.

Wykorzystanie Offline Template Compiler oraz mechanizmów Angular.

Dzięki wykorzystaniu Offline Template Compiler zmniejszona zostanie podatność aplikacji na ataki template injection. Zabezpieczenie to zostanie zrealizowane dzięki udostępnionym mechanizmom wbudowanym w platformę Angular.

Dostępność aplikacji frontendowej dla różnych języków

Aplikacja będzie wykorzystywać moduł dostępny w platformie Angular - x18n. Wydobywa on do pliku xlf, osobnego dla każdego języka, oznaczone frazy do przetłumaczenia. Format xlf jest stosowany jako standardowy sposób wymiany danych możliwych wykorzystywany podczas tłumaczeń. Następnie, budowana będzie wersja aplikacji zawierająca przetłumaczone frazy w odpowiednim języku.

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

Przechowywanie w bazie danych tłumaczeń dla danych ładowanych dynamicznie

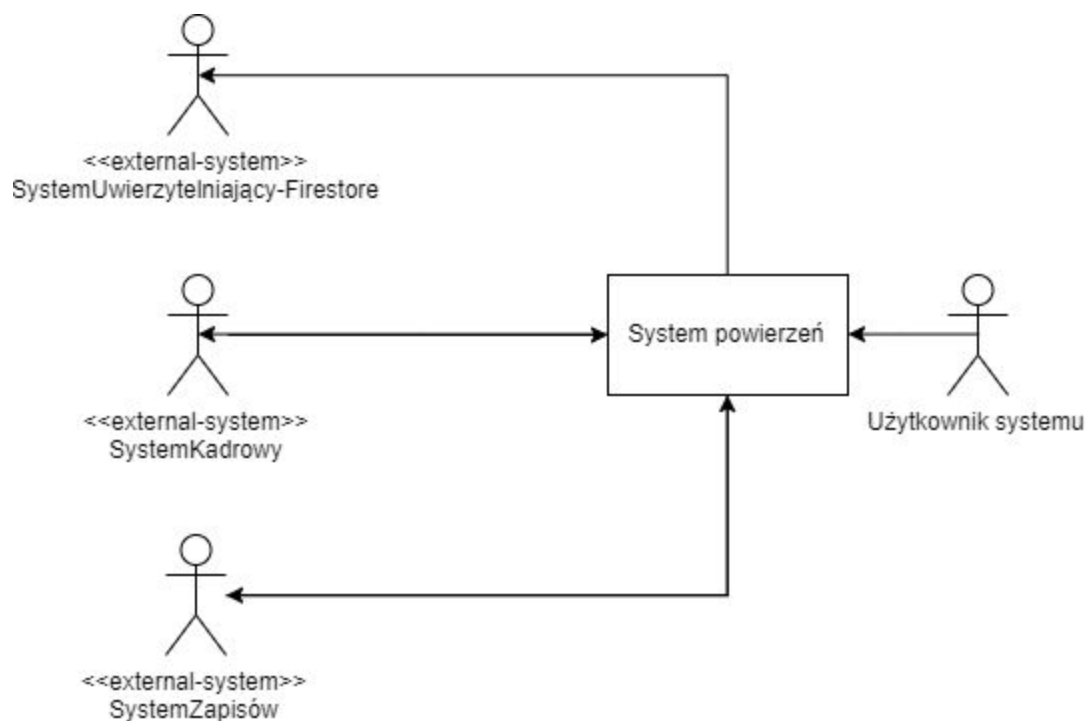
Aplikacja w bazie danych będzie posiadała tłumaczenia danych, których wyświetlanie będzie możliwe w kilku językach. Dane te będą wyświetlane użytkownikowi w wybranym przez niego języku. Baza danych będzie zawierać odpowiednie tabele zawierające tłumaczenia danych ładowanych dynamicznie. Aby uzyskać tłumaczenia do części interfejsu użytkownika, aplikacja kliencka będzie przekazywać parametr specyfikujący aktualnie wybrany język.

Logowanie akcji przeprowadzanych w systemie

System będzie przeprowadzał logowanie akcji przeprowadzanych w systemie do pliku. Do tworzenia logów zostanie wykorzystany logger dostarczony przez bibliotekę SLF4J. Aby rozróżnić powód logowania informacji, zostaną użyte poziomy logów takie jak: debug, error, czy info. Logowanie akcji systemu umożliwi m.in. przeprowadzenie audytu zachowań użytkowników, identyfikację podejrzanych zachowań oraz logowanie błędów, które wystąpiły w aplikacji

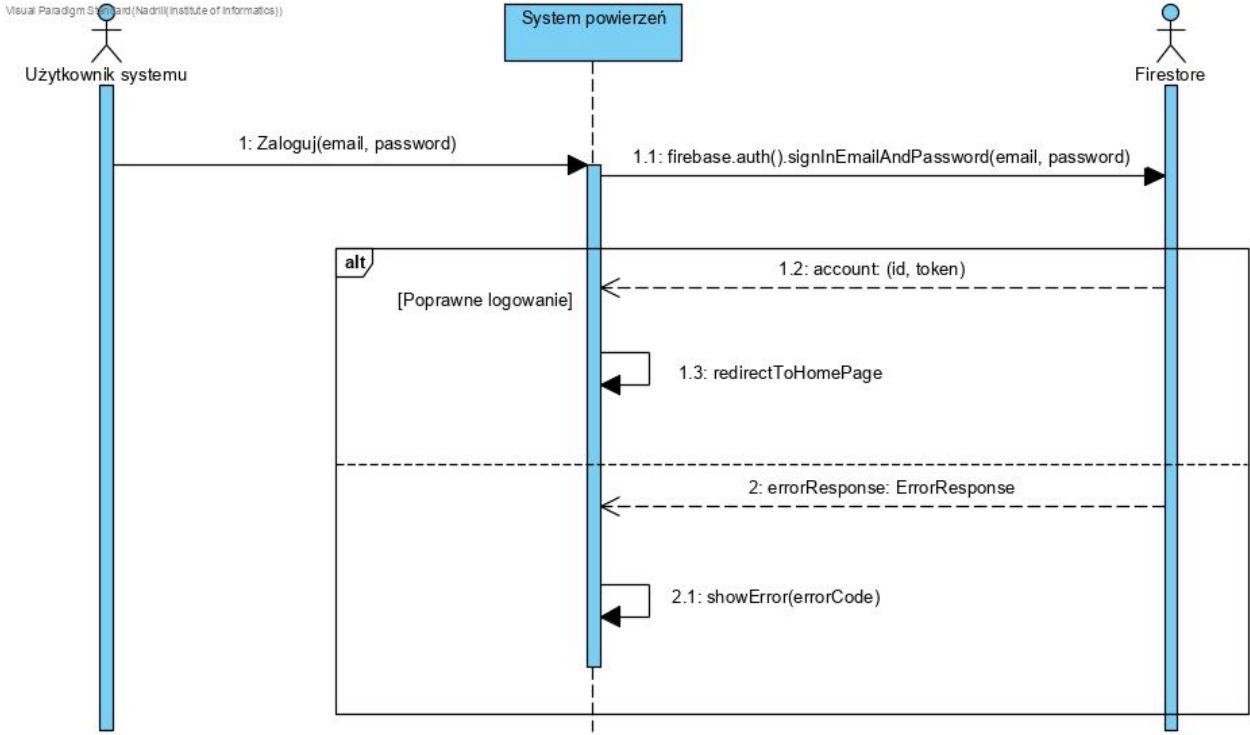
5. Architectural views

5.1 Context view

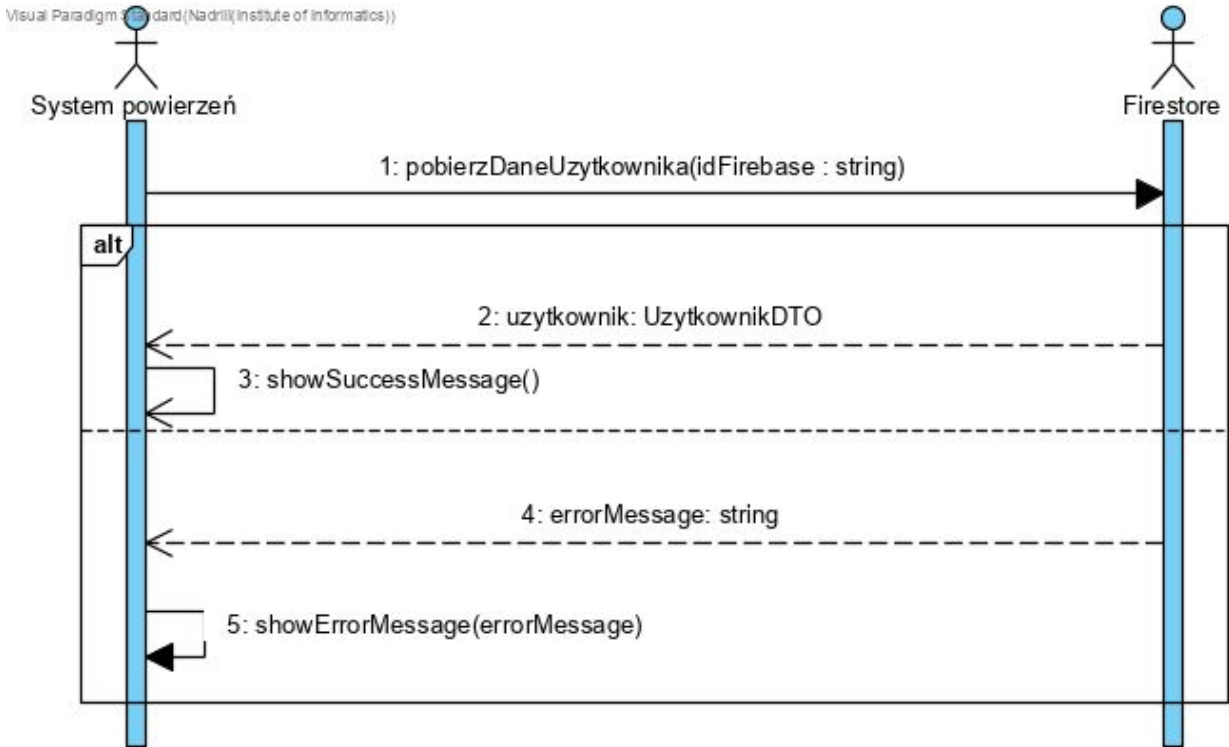


5.2 Interaction scenarios

Uwierzytelnianie użytkownika



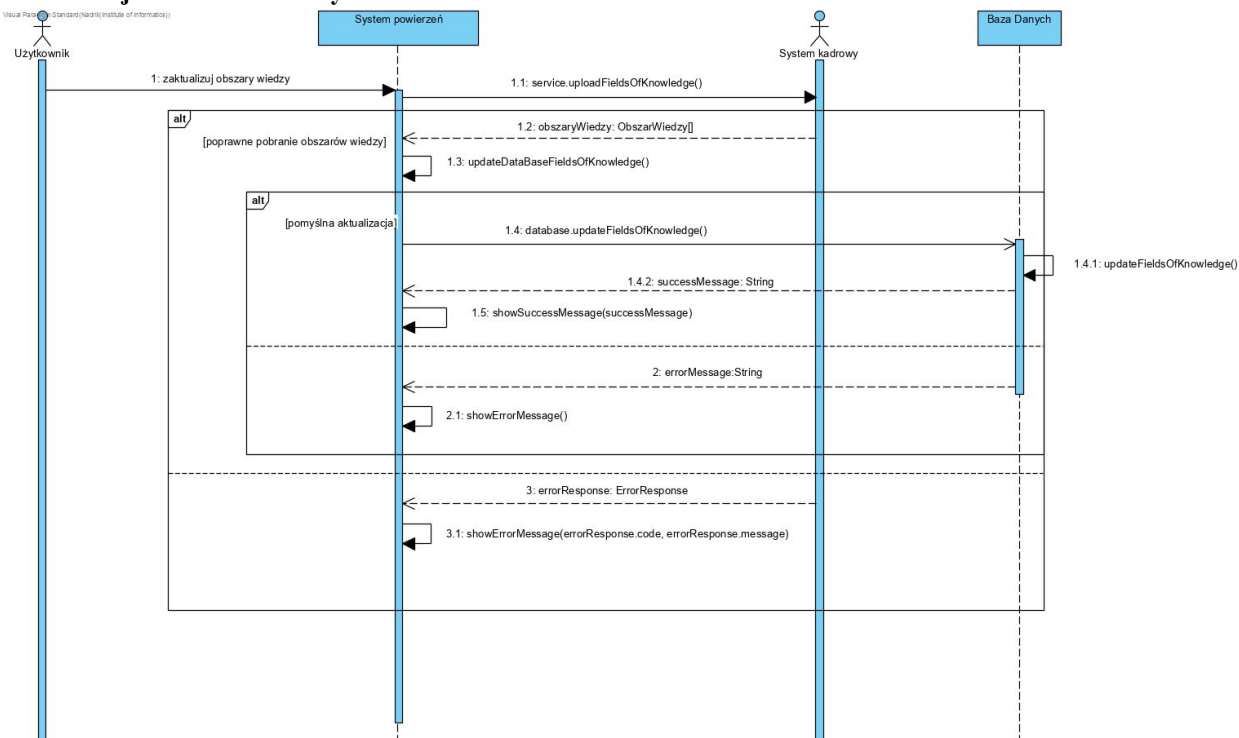
Pobieranie danych użytkownika



<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

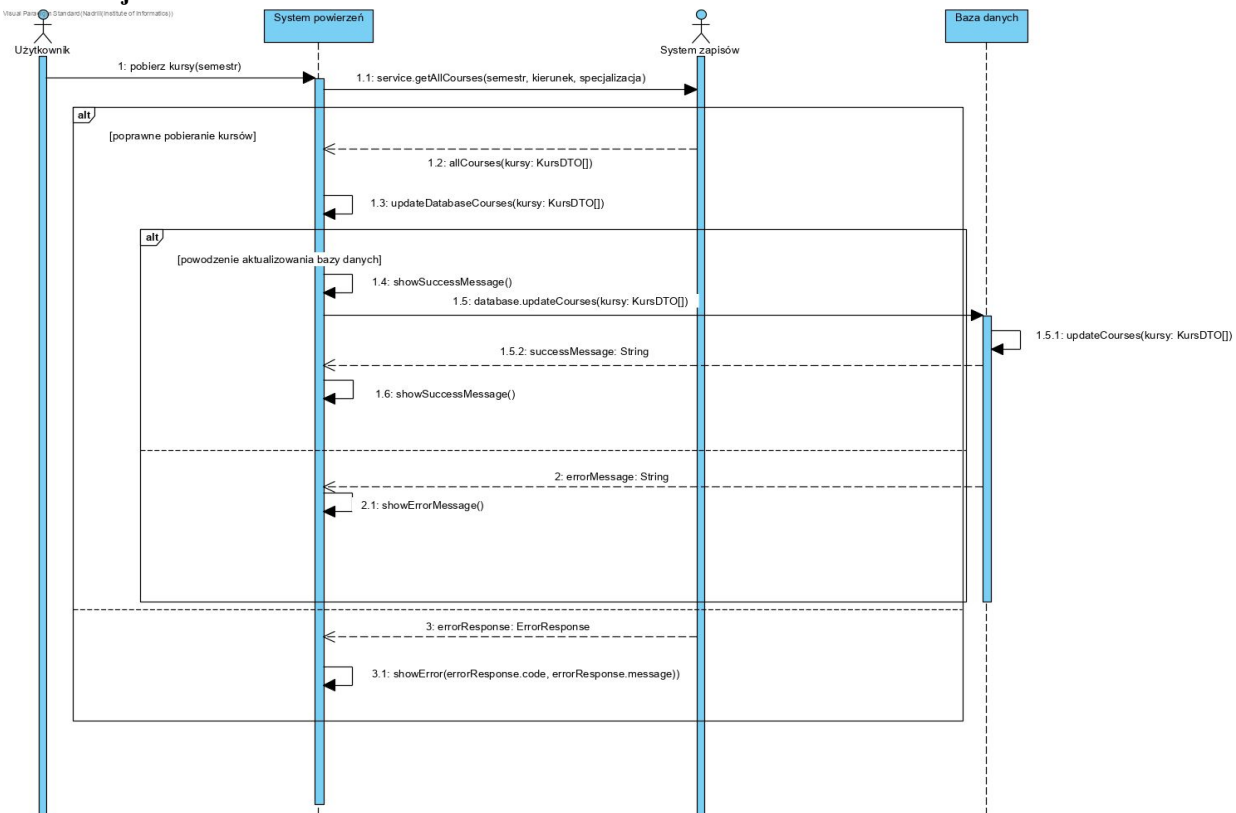
Aktualizacja obszarów wiedzy

Visual Paradigm (Standard/Institute of Information)



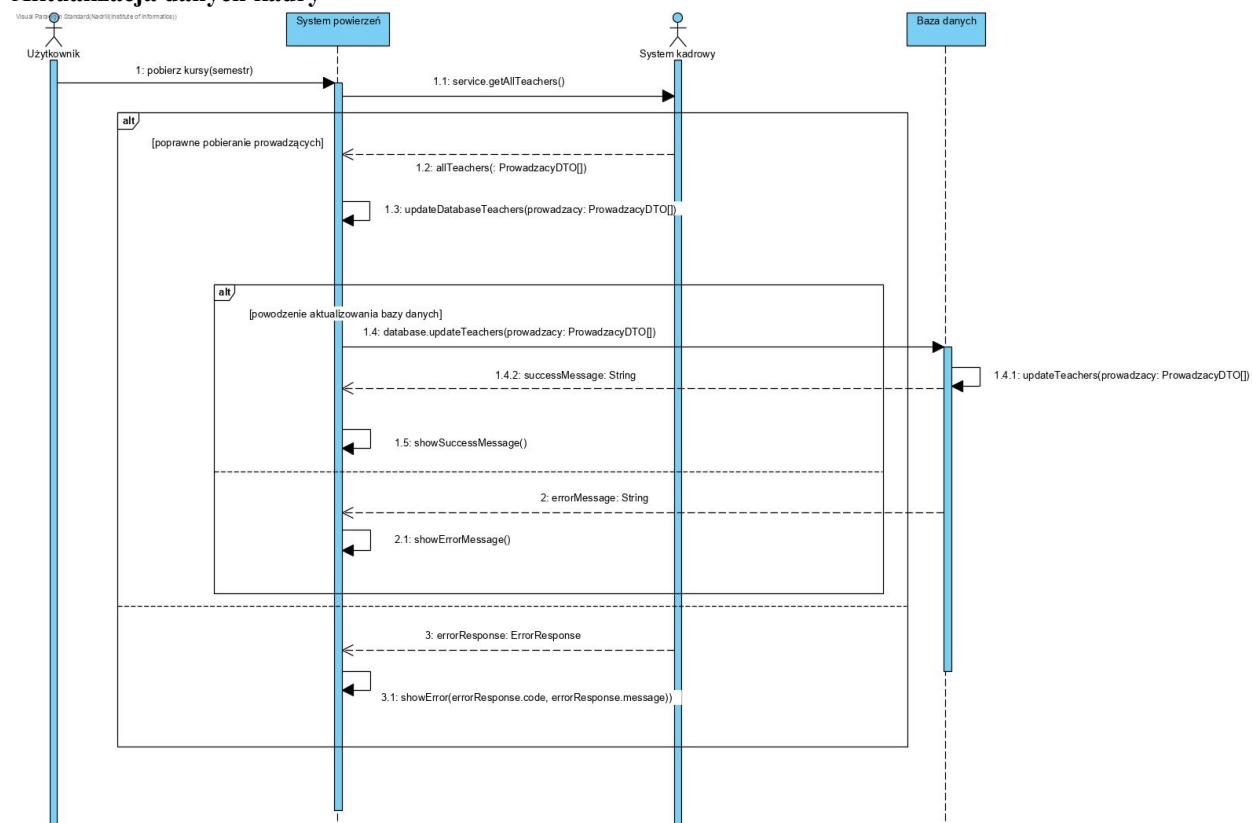
Aktualizacja kursów

Visual Paradigm (Standard/Institute of Information)



<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

Aktualizacja danych kadry



5.3 Integration interfaces – logical level

Interface 1 - Uwierzytelnianie użytkownika

Status	Planowany	
	Source application	Target application
Application name	System powierzeń	Firestore Auth
Integration technology	Firebase Auth SDK	HTTPS
Authentication mechanism	Google Services Api Key	
Data contract	Wejście : email i hasło Wyjście : token Firestore Format: Obiekt	
Does the interface manipulate on the sensitive data (RODO)?	Tak	
Middleware used	Brak	
Initiating side	System powierzeń	

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

Communication model	Synchroniczny na żądanie użytkownika
Performance	100 / godzinę (50 użytkowników, 2 logowania na godzinę)
Volumetry	Liczba wywołań: 1000 / dzień Rozmiar danych: jedno wywołanie to jeden token
Required accessibility	HIGH

Interface 2 - Pobieranie danych użytkownika

Status	Planowany	
	Source application	Target application
Application name	System powierzeń	Firestore Auth
Integration technology	Firebase Auth SDK	HTTPS
Authentication mechanism		Google Services Api Key
Data contract	Wejście : idFirebase Wyjście : <div data-bbox="531 816 920 1020" data-label="Diagram"> <pre> classDiagram class UzytkownikDTO { -idFirebase : string {id} -rola : Role -idZewnetrzne : int } </pre> </div> Format: Obiekt	
Does the interface manipulate on the sensitive data (RODO)?	Nie	
Middleware used	Brak	
Initiating side	System powierzeń	
Communication model	Synchroniczny na żądanie użytkownika	
Performance	30 / godzinę	
Volumetry	Liczba wywołań: 500 / dzień	
Required accessibility	HIGH	

<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

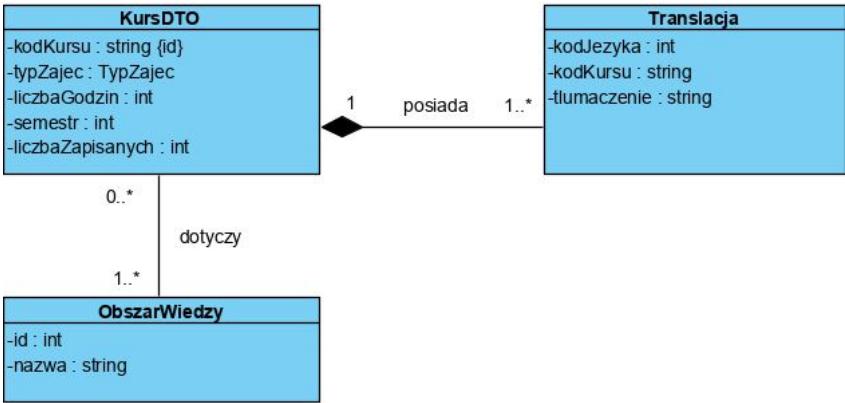
Interface 3 - Aktualizowanie obszarów wiedzy

Description	Na żądanie użytkownika zaktualizowane zostaną w bazie danych obszary wiedzy	
Status	Planowan	
	Source application	Target application
Application name	System powierzeń	System kadrowy
Integration technology		HTTP/HTTPS
Authentication mechanism		Klucz API
Data contract	<p>Wejście: Brak Wyjście: lista obszarów wiedzy</p> <div data-bbox="527 842 1003 1003" data-label="Diagram"> <pre> classDiagram class ObszarWiedzy { -id : int -nazwa : string } </pre> </div> <p>Format: Json Przykład:</p> <pre> "result": ["obszar_wiedzy": { "id": 1, "name": "projektowanie oprogramowania" }, "obszar_wiedzy": { "id": 2, "name": "bazy danych" }, "obszar_wiedzy": { "id": 3, "name": "sztuczna inteligencja" }, ...] </pre>	
Does the interface manipulate on the sensitive data (RODO)?	Nie	
Middleware used	Brak	
Initiating side	System powierzeń	
Communication model	Synchroniczny na żądanie użytkownika	
Performance	1 / miesiąc	

<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

Volumetry	Liczba wywołań: 10 / rok Pamięć: Dane aktualizowane w bazie danych
Required accessibility	MEDIUM

Interface 4 - Aktualizacja kursów

Description	Na żądanie użytkownika aktualizowane są w bazie danych informacje o kursach	
Status	Planowany	
	Source application	Target application
Application name	System powierzeń	System zapisów
Integration technology		HTTP/HTTPS
Authentication mechanism		Klucz API
Data contract	<p>Wejście: semestr, kierunek, (opcjonalnie) specjalizacja Wyjście: lista kursów odpowiadająca zadany kryteriom</p>  <pre> classDiagram class KursDTO { -kodKursu : string {id} -typZajec : TypZajec -liczbaGodzin : int -semestr : int -liczbaZapisanych : int } class Translacja { -kodJezyka : int -kodKursu : string -tlumaczenie : string } class ObszarWiedzy { -id : int -nazwa : string } KursDTO "1" -- "1..*" Translacja : posiada KursDTO "0..*" -- "1..*" ObszarWiedzy : dotyczy </pre> <p>Format: JSON Przykład:</p> <pre> "result": ["kurs":{ "kod_kursu": "CC2133", "typ_zajec": "Projekt", "liczba_godzin": 30, "semestr": 7, "liczbaZapisanych": 240 "translacje": ["translacja": { "kod_jezyka": 0, "tlumaczenie" : "Podstawy baz danych" }, "translacja": { "kod_jezyka": 1, "tlumaczenie" : "Database basics" }] }, "obszary_wiedzy": ["obszar_wiedzy": { "id": 2, </pre>	

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

	<pre> "name": "bazy danych" }] }, ...] </pre>
Does the interface manipulate on the sensitive data (RODO)?	Nie
Middleware used	Brak
Initiating side	System powierzeń
Communication model	Synchroniczny na żądanie użytkownika
Performance	1 / pół roku
Volumetry	Liczba wywołań: 2 / rok Pamięć: Dane aktualizowane w bazie danych
Required accessibility	MEDIUM

Interface 5 - Aktualizacja kadry

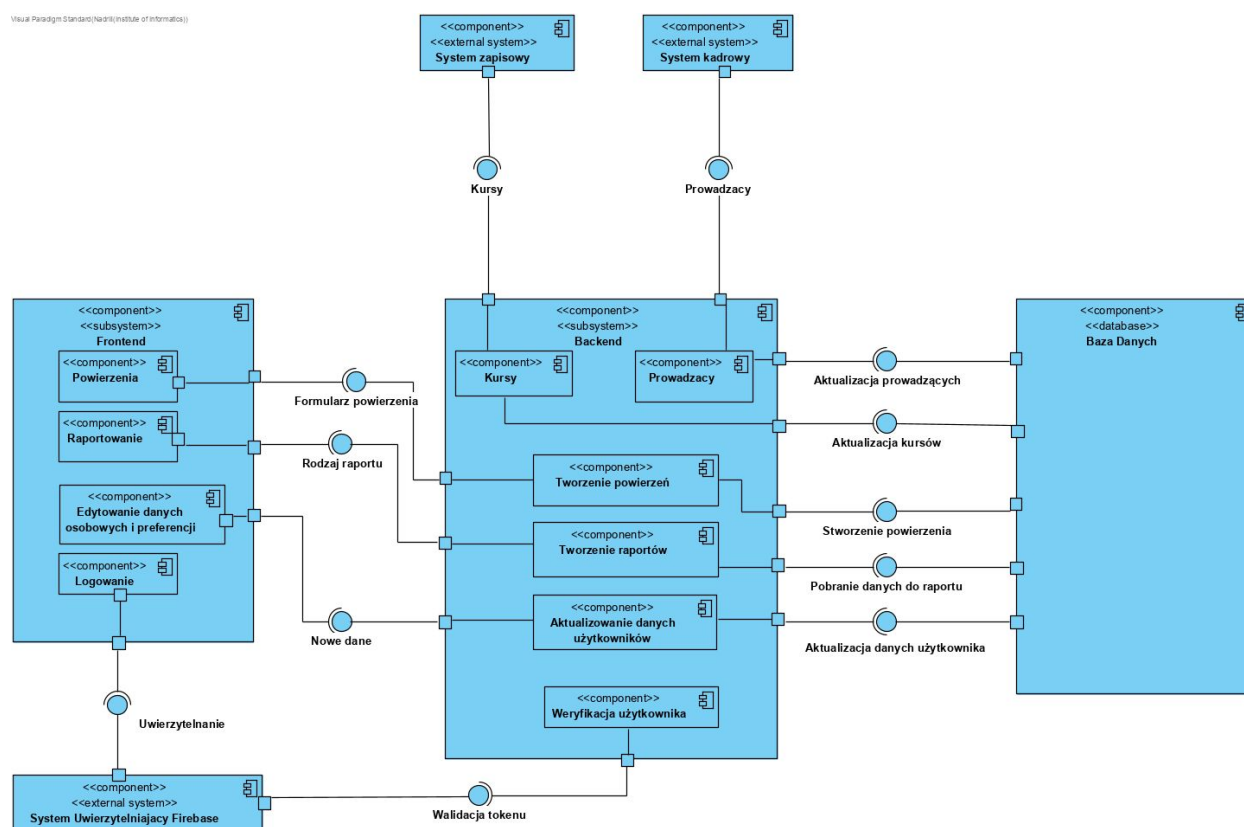
Description	Na żądanie użytkownik może zaktualizować bazę danych o informacje o prowadzących	
Status	Planowany	
	Source application	Target application
Application name	System powierzeń	System kadrowy
Integration technology		HTTP/HTTPS
Authentication mechanism		Klucz API
Data contract	<p>Wejście: Brak Wyjście: Lista dostępnych w systemie prowadzących</p> <div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff; margin: 10px 0;"> <p style="text-align: center; margin: 0;">ProwadzacyDTO</p> <pre> -id : int {id} -imie : string -nazwisko : string -PESEL : string -tytul : Tytul -typPracownika : TypPracownika </pre> </div> <p>Format: JSON Przykład:</p> <pre> "result": ["prowadzacy": { "id": 1, "imie": "Adam", "nazwisko": "Abacki", "PESEL": "123456789", "tytul": "Doktor", "typPracownika": "nauczyciel akademicki" }, ... </pre>	

<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

]
Does the interface manipulate on the sensitive data (RODO)?	Tak
Middleware used	Brak
Initiating side	System powierzeń
Communication model	Synchroniczny na żądanie użytkownika
Performance	1 / pół roku
Volumetry	Liczba wywołań: 2 / rok Pamięć: Dane aktualizowane w bazie danych
Required accessibility	MEDIUM

6. Functional view

Visual Paradigm Standard (Institute of Informatics)



7. Perspektywa wdrożeniowa (Deployment view)

7.1. Wprowadzenie

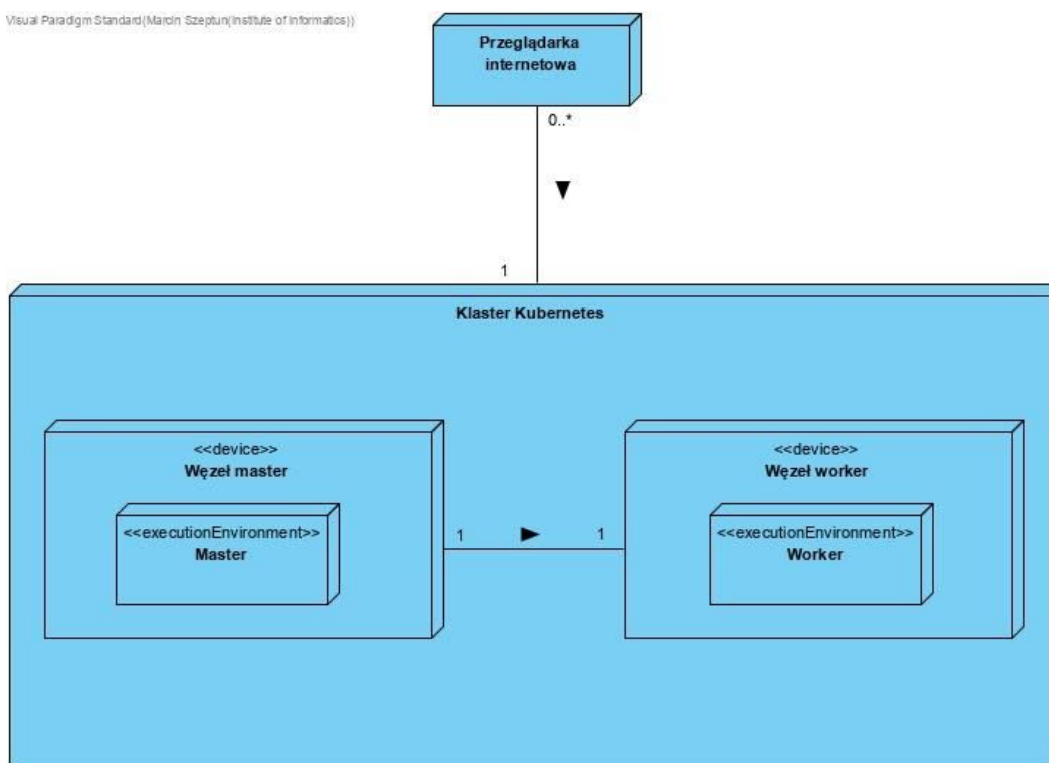
Do wdrożenia systemu, zarządzania i skalowania zostanie wykorzystany system Kubernetes. Na klastery będą

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

składać się 1 węzeł roboczy i 1 węzeł główny. Natomiast aplikacja uruchamiana będzie za pomocą platformy Docker, gdzie na osobnych kontenerach znajdować się będzie baza danych, aplikacja webowa oraz aplikacja serwerowa. Cała architektura uruchamiana będzie na jednej maszynie z użyciem wirtualnych maszyn.

7.2. Diagram fizyczny

Visual Paradigm Standard (Marek Szeptun/institute of informatics)



7.2.1. Kubernetes Master

General information	
Name	Kubernetes Master
Virtual	Tak
Data center?	Nie
OS	Ubuntu 16.0.4
Description	Kubernetes Master odpowiedzialny jest za utrzymywanie odpowiedniego stanu klastra.

Hardware configuration	
Vendor	Dell
Processor	2x 2.9 GHZ
RAM	8 GB
HDD	256 GB
RAID i HDD Netto	RAID 0
RAID connected?	Brak

<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

Net cards bonding	Nie
-------------------	-----

Software configuration	
Users and groups	Użytkownik root, który odpowiedzialny jest za działanie maszyny.
Poziom pracy systemu, czy jest wymagane środowisko graficzne	System będzie działał za pomocą trybu konsolowego. Środowisko graficzne nie jest wymagane
Dodatkowe pakiety z dystrybucji systemu	Brak
Dodatkowe pakiety spoza dystrybucji systemu	<ul style="list-style-type: none"> • docker • kubectl

7.2.2. Kubernetes Worker

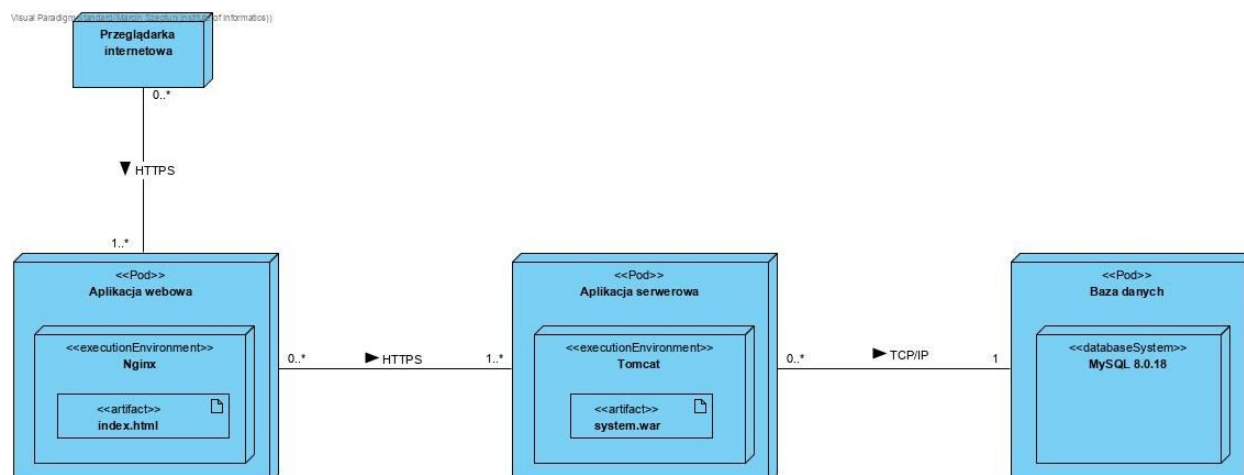
General information	
Name	Kubernetes Worker
Virtual	Tak
Data center?	No
OS	Ubuntu 16.0.4
Description	Kubernetes Worker, który odpowiedzialny jest za wykonywanie zadań zleconych przez Mastera

Hardware configuration	
Vendor	Dell
Processor	4x 2.9 GHZ
RAM	8 GB
HDD	512 GB
RAID i HDD Netto	RAID 0
RAID connected?	Brak
Net cards bonding	Nie

Software configuration	
Users and groups	Użytkownik root, który odpowiedzialny jest za działanie maszyny.
Poziom pracy systemu, czy jest wymagane środowisko graficzne	System będzie działał za pomocą trybu konsolowego. Środowisko graficzne nie jest wymagane
Dodatkowe pakiety z dystrybucji systemu	Brak
Dodatkowe pakiety spoza dystrybucji systemu	<ul style="list-style-type: none"> • docker • kubectl

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

7.3. Diagram logiczny



7.3.1 Aplikacja webowa

General information	
Name	Aplikacja webowa
Virtual	Tak
Obraz docker	nginx 1.16.1
Description	Kontener na którym uruchomiona zostanie aplikacja frontendowa napisana za pomocą frameworka Angular. Może być uruchamiana w wielu instancjach.

Resources limit	
Processor	2x 2.0 GHz
RAM	8 GB
HDD	30 GB

7.3.2 Aplikacja serwerowa

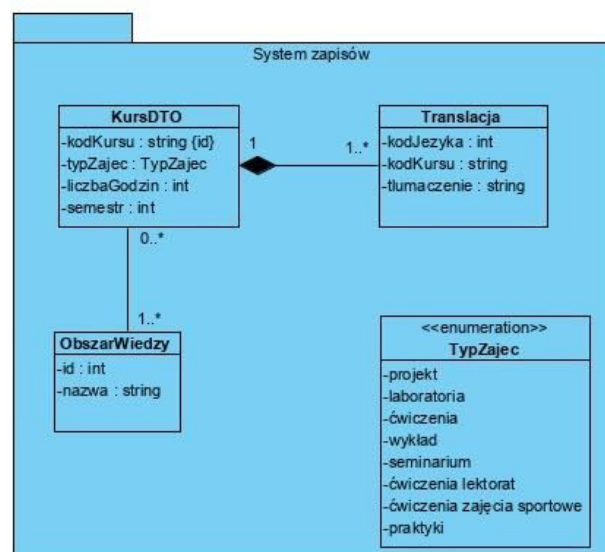
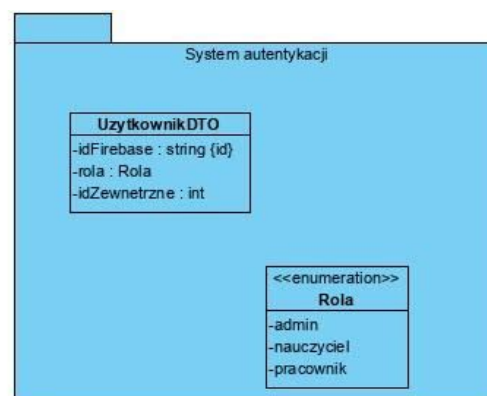
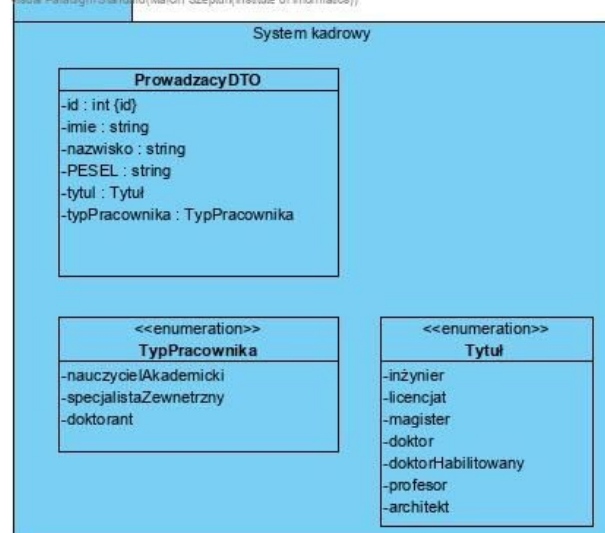
General information	
Name	Aplikacja serwerowa
Virtual	Tak
Obraz docker	tomcat 9.0.29
Description	Kontener na którym uruchomiona zostanie aplikacja serwerowa napisana z wykorzystaniem frameworka Spring Boot. Aplikacja będzie łączyć się z bazą danych i dostarczać z niej informacje. Może być uruchamiana w wielu instancjach.

Resources limit	
Processor	2x 2.0 GHz
RAM	8 GB

<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

8.2. Model DTO

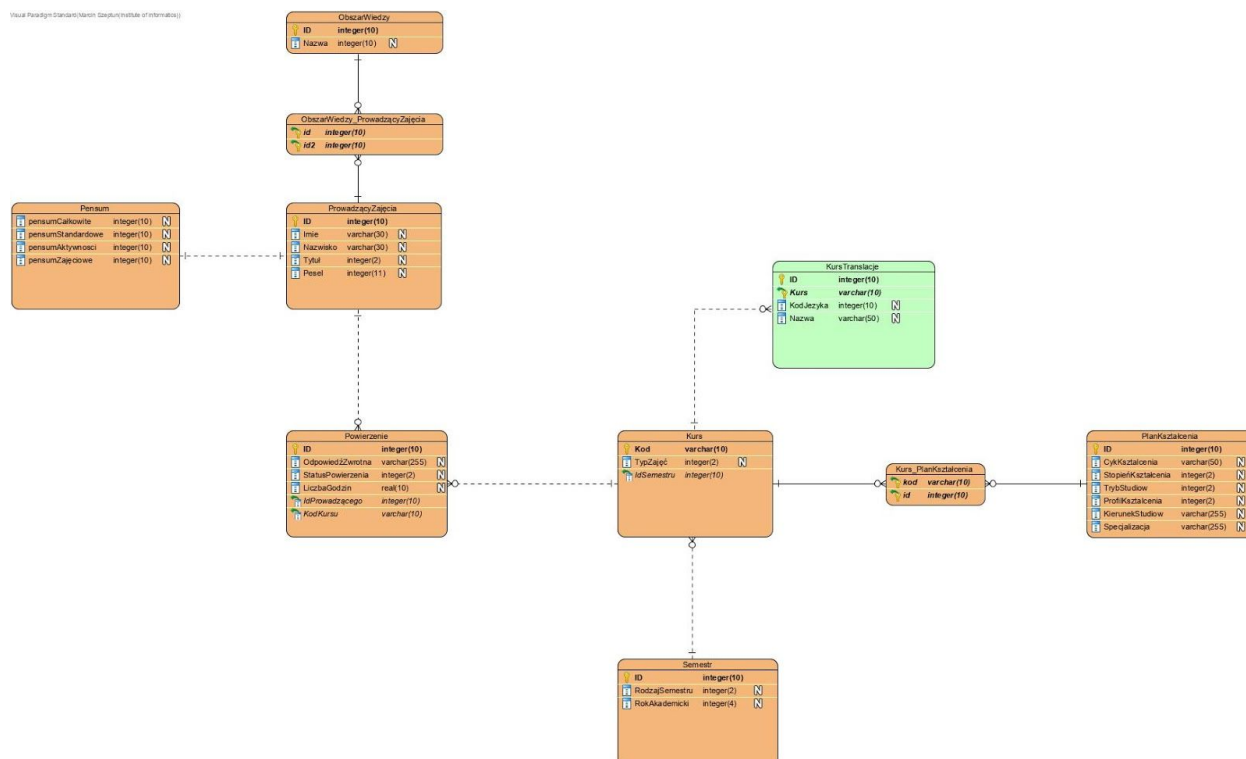
Visual Paradigm Standard (Marcin Szepturny/institute of informatics)



<Project Name>	
Architecture Notebook	Date: <dd/mm/yy>

8.2. Projekt bazy danych

Visual Paradigm Standard Edition (Copyright © 2014)



Database General information (one table per server)	
SID/Service Name	Database
Server name	db-assignments
Port	3306
Type	MySQL 8.0.18
Character coddng	Standardowe
Description	Baza danych relacyjna, przechowująca informacje dotyczące kursów, pracowników naukowych oraz powierzeń.
Technologies	Brak

Backup	
wolumen	- 10 000 ostatnich rekordów
częstotliwość	- raz w miesiącu
tryb	- inkrementacyjny
okres protekcji	- rok od daty backupu

8.2.1. Data schemas

Schema information	
Schema name	AssigmentsScheme
Initial capacity	100 mb
Capacity increment (year)	20mb
Necessary rights	brak
Others	brak

<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

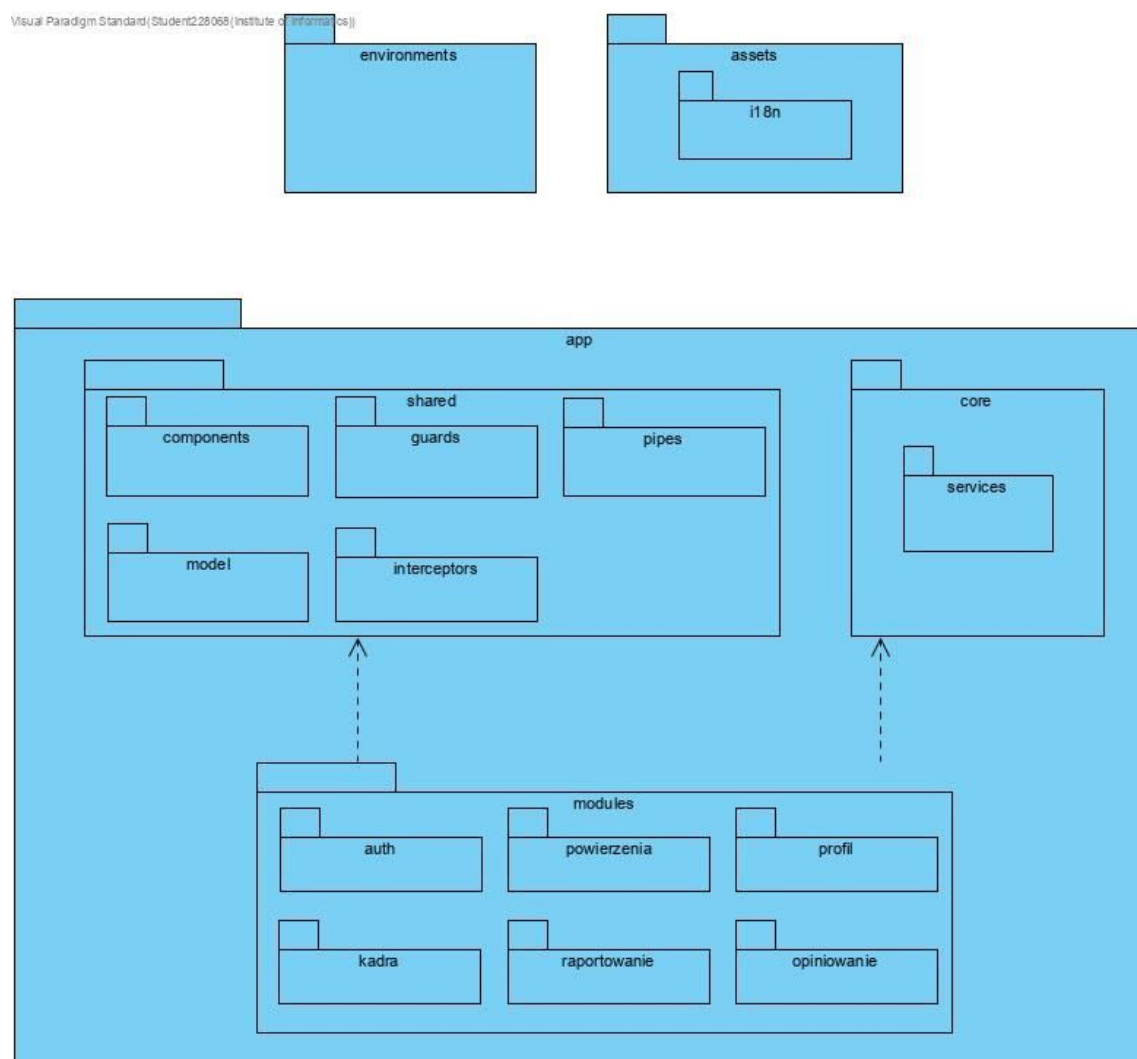
9. Development view

Aplikacja webowa (Frontend)

Aplikacja zostanie napisana z wykorzystaniem TypeScript oraz frameworku Angular w wersji 8. Przy planowaniu rozmieszczenia pakietów korzystano z wytycznych oficjalnej dokumentacji frameworku Angular.

Aplikacja będzie posiadać pakiet modułów, który będzie zawierać pakiety domenowe. Pakiet ten będzie korzystać z pakietu core zawierający serwisy wykorzystywane m.in. do żądań http. Dodatkowo, pakiet modules korzystać będzie z pakietu shared, który będzie zawierać elementy współdzielone. Zgodnie ze standardami aplikacji Angular, różnicą pomiędzy pakietem core a pakietem shared jest to, że pakiet core będzie ładowany do aplikacji jednorazowo, przy jej starcie, zaś pakiet shared ładowany jest przy ładowaniu każdego modułu.

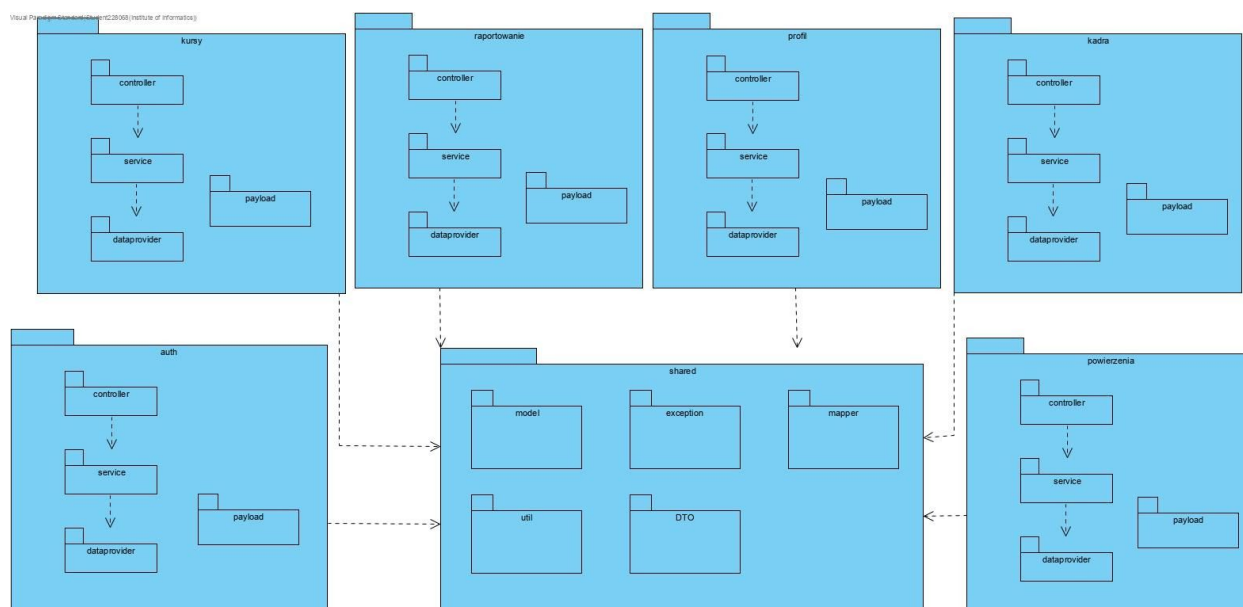
Zastosowania zostanie następująca struktura pakietów.



<Project Name>	
Architecture Notebook	Date: <dd/mmm/yy>

Aplikacja serwerowa (Backend)

Aplikacja zostanie napisana z wykorzystaniem Kotlin oraz frameworku Spring Boot w wersji 2.2.2. Przy ustalaniu architektury aplikacji serwerowej zastosowano podejście warstwowe w formie top-down z pionowym podziałem. Zaletami takiego podejścia są zarządzalność, możliwość ponownego użycia warstw lub ich elementów i łatwość testowania. Wadą zaś jest obniżona wydajność spowodowana komunikacją między warstwami. Zastosowana zostanie następująca struktura pakietów: Pakiet shared będzie zawierał części współdzielone, które będą mogły być wykorzystywane przez inne pakiety.



10. Use-case realizations (for selected use-cases)

Przebieg procesu przedstawiono w formie diagramu sekwencji. Zrezygnowano z pokazania szczegółów implementacji po stronie Frontendowej.

