

Uwaga:

1. Praca powinna być napisana z użyciem formy bezosobowej np. wykonano.
2. Harmonogram ten określa jednocześnie rozwój aplikacji
3. Należy rozpocząć pisanie pracy od rozdziału 3
4. W kolumnie **Termin** podano terminy spotkań – zakres materiału przedstawianego na spotkaniach podano w kolumnach: **Nazwa rozdziału** oraz **Zawartość**. Terminy przekazania rozdziałów: 1, 2, 7 nakładają na terminy przekazania pozostałych prac.
5. Liczba podanych stron w kolumnie **Liczba stron** może ulec zmianie po uzgodnieniu.

Nazwa rozdziału	Zawartość	Liczba stron	Termin
3. Opracowanie koncepcji systemu		Okolo 15	
3.1. Model biznesowy	Opis „świat rzeczywistego”: 1) Opis słowny (zasobów ludzkich, przepisów lub innych ograniczeń, warunków technicznych) – obowiązkowy (minimum) 2) <i>Opcjonalnie - diagramy typu workflow – element opisu podwyższający ocenę wykonane w terminie późniejszym</i> 3) <i>Opcjonalnie - diagramy aktywności (bez tzw. torów) przedstawiające artefakty świata rzeczywistego - element na wyższą ocenę przekazany w terminie późniejszym</i>		11.10.2021-17.10.2021 oraz prezentacja tych materiałów podczas konsultacji w dniach 14.10.2021 oraz 15.10.2021.
3.2. Wymagania oprogramowania	Wyszczególnienie wymagań funkcjonalnych i нефункциональных programu wynikające z opisu biznesowego		18.10-31.10.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w podanych terminach.
3.3. Opracowanie modelu konceptualnego	1) Diagramy przypadków użycia wraz z opisem aktorów i scenariuszami poszczególnych przypadków użycia – powinny dokładniej specyfikować wymagania tworzonego oprogramowania 2) Diagram klas typu <i>Entity</i> wraz z informacją, z jakich przypadków użycia zostały zidentyfikowane. Opisy podstawowych operacji na encjach używanych do „mapowania” do relacyjnej bazy danych. 3) Trzy diagramy aktywności modelujące przebieg scenariuszy dla wybranych przypadków użycia: pierwszy o najbardziej złożonym scenariuszu, drugi o średnio złożonym scenariuszu i trzeci o prostym scenariuszu		

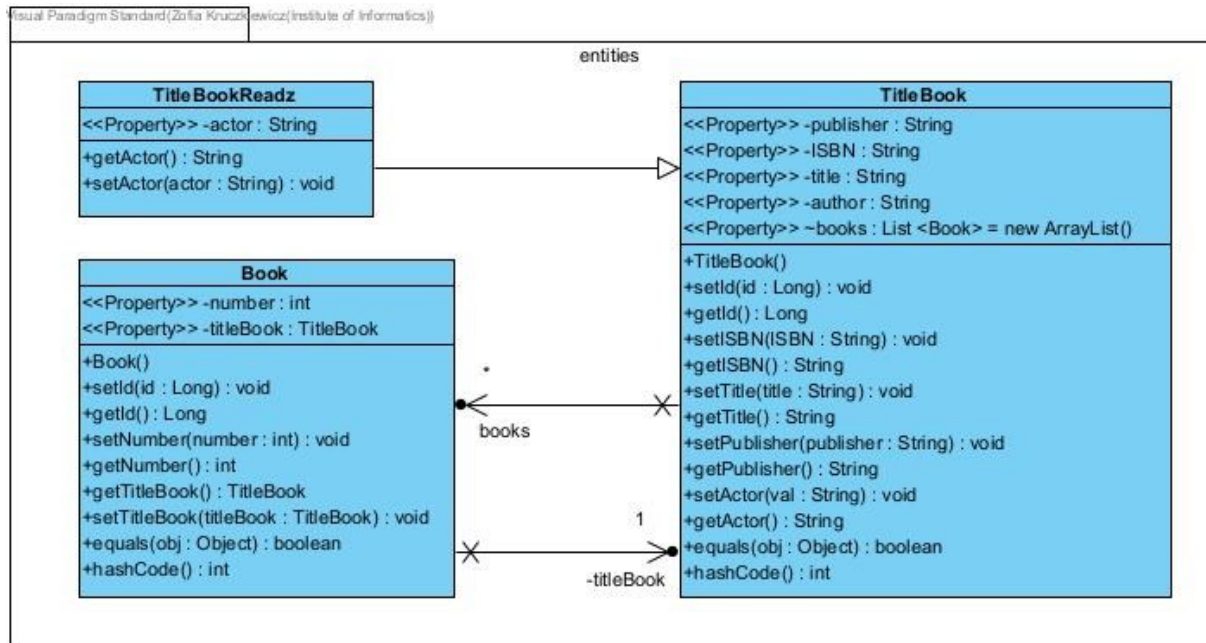
4. Projekt systemu		Okolo	
4.1. Projekt struktury przetwarzanych danych oraz struktury komponentów warstw: internetowej, biznesowej oraz integracji z bazą danych	<p>Rozdział ten powinien zawierać projekt struktury przetwarzanych danych oraz komponentów do przetwarzania danych obejmującego:</p> <ol style="list-style-type: none"> 1) Dokładny diagram klas typu <i>Entity</i> (powstały z diagramu klas zdefiniowanego w fazie analizy) - (kontynuacja p. 3.3) 2) Diagram tabel wynikający z diagramu klas typu <i>Entity</i> - (kontynuacja p. 3.3) 3) Diagram klas reprezentujący główne komponenty poszczególnych warstw: klienta (w przypadku aplikacji mobilnej), internetowej, biznesowej oraz integracji z bazą danych wynikające z technologii Spring Boot 	10 - 15	1.11-14.11.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w podanych terminach.
4.2. Projekt funkcji aplikacji	<p>Rozdział ten powinien zawierać projekt funkcji do przetwarzania danych obejmujący:</p> <ol style="list-style-type: none"> 1) Szczegółowe diagramy aktywności wybranych funkcji aplikacji zawierające tzw. tory reprezentujące warstwy aplikacji: warstwę klienta, warstwę internetową, warstwę biznesową oraz warstwę integracji z bazą danych. Należy w poszczególnych torach przedstawić procesy zachodzące w poszczególnych warstwach aplikacji (kontynuacja p.3.3) 2) <i>Opcjonalnie - szczegółowe diagramy sekwencji wybranych funkcji aplikacji zawierające tzw. linie życia reprezentujące wybrane komponenty z warstw aplikacji: klienta, internetowej, biznesowej oraz warstwy integracji z bazą danych.</i> <i>Dla każdej z 3 funkcji należy wykonać dwa składowe diagramy sekwencji: pierwszy reprezentujący scenariusz z warstwy klienta i internetowej oraz drugi (kontynuacja pierwszego scenariusza) reprezentujący scenariusz z warstwy biznesowej oraz warstwy integracji z bazą danych (kontynuacja p.3.3)</i> 		15.11-21.11.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w podanych terminach.
4.3. Projekt formularzy	Opis i rysunki prezentujące szablony wybranych formularzy		

5. Implementacja		Okolo	
5.1. Architektura systemu	Opis warstw systemu, rozmieszczenie geograficzne oprogramowania – wykaz pakietów oraz ich zawartości. Należy opisać konfigurację części serwerowej oraz warstwy klienta np. w przypadku aplikacji mobilnych.	10	22.11-12.12.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w podanych terminach.
5.2. Oprogramowanie	Rozdział ten powinien zawierać informację o kodzie wybranych trzech funkcji, które szczegółowo przedstawiono w rozdziałach 3 i 4: <ul style="list-style-type: none"> informacje o podstawowych kodach źródłowych należących do poszczególnych warstw aplikacji wykazanie, że zostały zachowane zasady budowy oprogramowania dotyczące budowy aplikacji wielowarstwowej np. separacji kodu w poszczególnych warstwach np. oddzielenie logiki biznesowej od warstwy prezentacji, ograniczenie interakcji między warstwą klienta i warstwą prezentacji itd. 		
5.3. Realizacja	Przykładowe realizacje projektów formularzy (nawiązanie do projektów z p. 4.3.)		22.11-12.12.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w podanych terminach.
5.4. Zabezpieczenia programu	Uwierzytelnianie, autoryzacja, szyfrowanie (np. SSL), przydział ról użytkowników, eliminacja niebezpiecznego kodu		
6. Testowanie systemu		Okolo	
6.1. Opracowanie koncepcji testów	Podrozdział ten powinien zawierać opis koncepcji testów (cel testu i jego opis), danych wejściowych i spodziewanych wyników. Do testowania należy wybrać szczegółowiej przedstawiane funkcje programu - czyli wybrane przypadki użycia modelowane za pomocą diagramów aktywności w rozdziale 3, następnie projektowane za pomocą szczegółowych	9	13.12-27.12.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w

	diagramów aktywności z rozdziału 4 i których kod przedstawiono w rozdziale 5.		podanych terminach.
6.2.Realizacja testów	Wyniki testów np. w formie tabel i/lub wykresów		
6.3.Analiza wyników testów	Sformułowanie wniosków		
1. Wstęp	Rozdział ten powinien zawierać zarys najważniejszych elementów pracy <ul style="list-style-type: none"> • Krótki opis dziedziny pracy • Cel pracy • Wyszczególnienie zadań zaplanowanych do wykonania • Opis struktury pracy 	2	2.11-5.12.2021
2. Zagadnienia związane z realizacją aplikacji		Okolo 10	
2.1. Zagadnienia dotyczące dziedziny obsługiwanej przez tworzoną aplikację	Przegląd zagadnień wybranych zagadnień z dziedziny pracy np. prezentacja wybranych metod rehabilitacji obsługiwanych przez tworzoną aplikację.		2.11-5.12.2021 oraz prezentacja tych materiałów podczas konsultacji w odbywających się w podanych terminach.
2.2. Zagadnienia dotyczące wybranych technologii	Przegląd technologii do budowy aplikacji internetowych zawierające uzasadnienie ich wyboru do realizacji pracy. Podrozdziały te powinny zawierać informacje istotne z punktu widzenia przyjętego celu pracy		
2.3. itd			
7. Podsumowanie pracy	Rozdział ten powinien zawierać podsumowanie osiągniętych wyników pracy oraz perspektywy rozwoju wykonanego systemu	2	13.12-27.12.2021
Literatura (minimum 5 pozycji przeczytanych w całości lub częściowo), linki do ważnych stron internetowych	Należy literaturę posortować alfabetycznie wg nazwiska pierwszego autora oraz tytułu strony – i powoływać się na nią w tekście, podając numer pozycji np. [2], w tych miejscach, gdzie przytaczane są pewne fakty przedstawiane w cytowanej literaturze.		13.12-27.12.2021

Przykład struktury rozdziału 4

To jest diagram klas jako obiektowy model danych z rozdziału 3, zidentyfikowany na podstawie scenariuszy przypadków użycia.

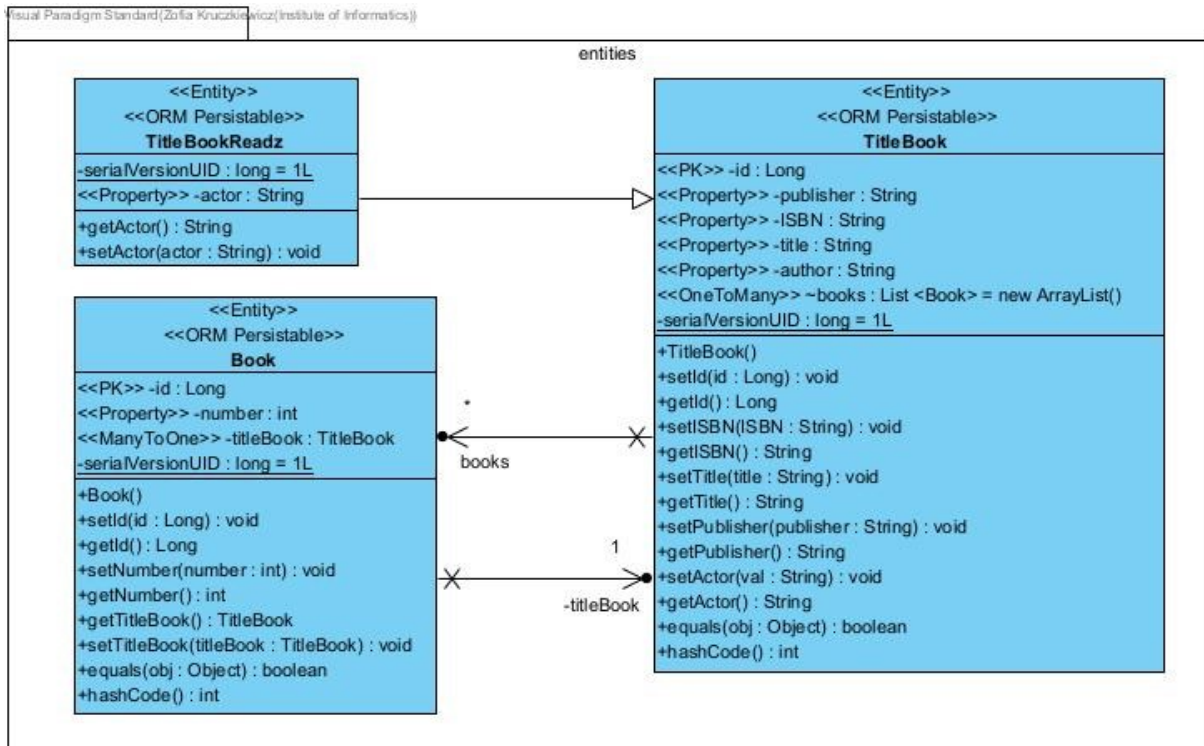


Zawartość rozdziału 3 powinna być zgodna z informacją podaną w tabeli, zamieszczonej powyżej. Powinna również zawierać diagramy aktywności scenariuszy wybranych przypadków użycia zawierające 2 tor: jeden reprezentujący *Warstwę klienta* lub aplikację *Front-end* i drugi tor reprezentujący scenariusz przypadku użycia, czyli *Logikę biznesową aplikacji* czyli *Back-end*.

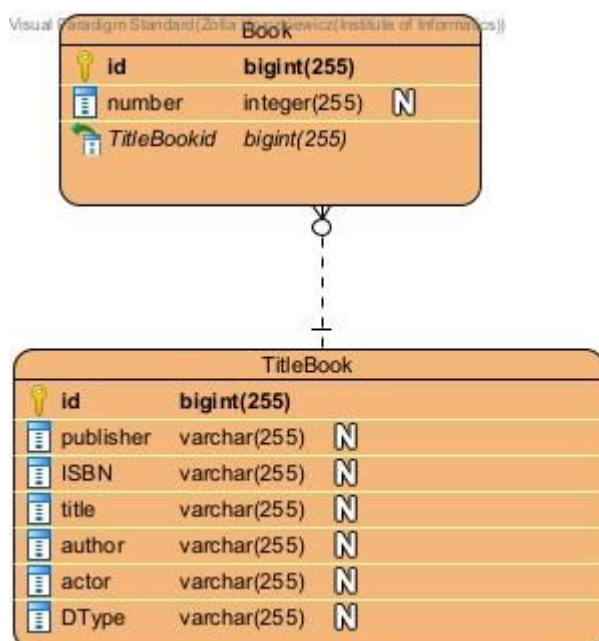
Realizacja projektu w rozdziale 4.

4.1. Projekt struktury przetwarzanych danych oraz struktury komponentów warstw: internetowej, biznesowej oraz integracji z bazą danych

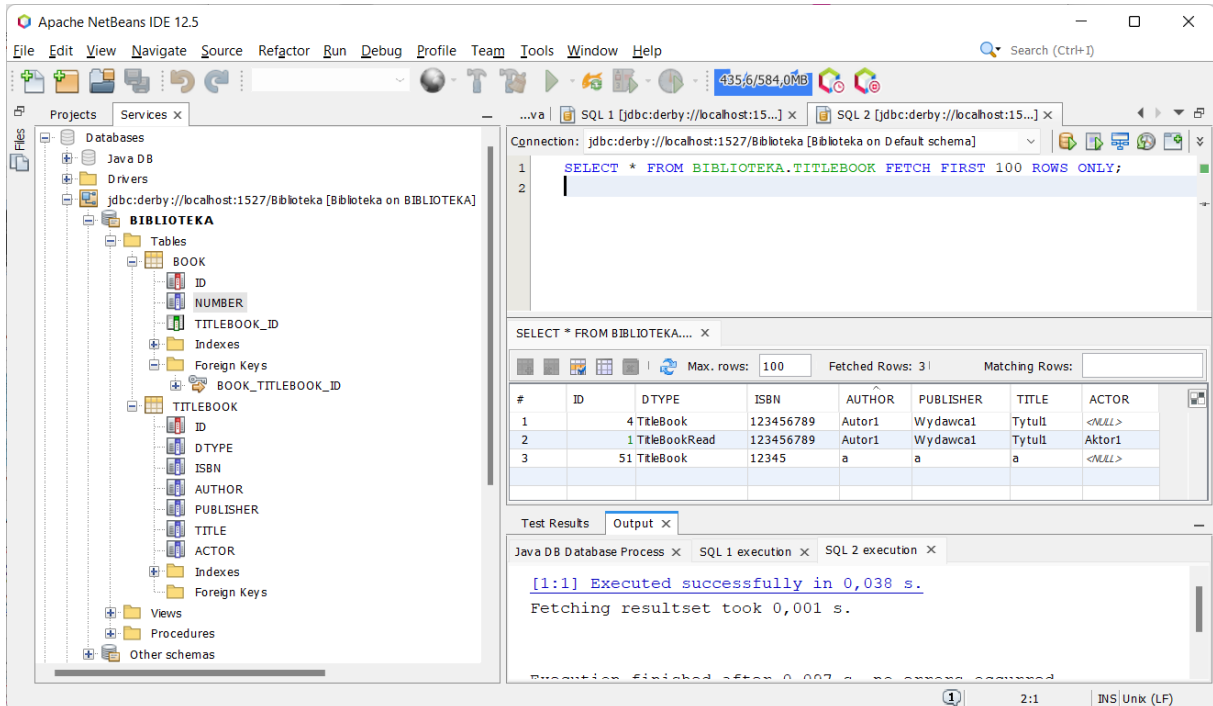
4.1.1. Dokładny diagram klas typu *Entity* (powstały z diagramu klas zdefiniowanego w fazie analizy) - (kontynuacja p. 3.3)



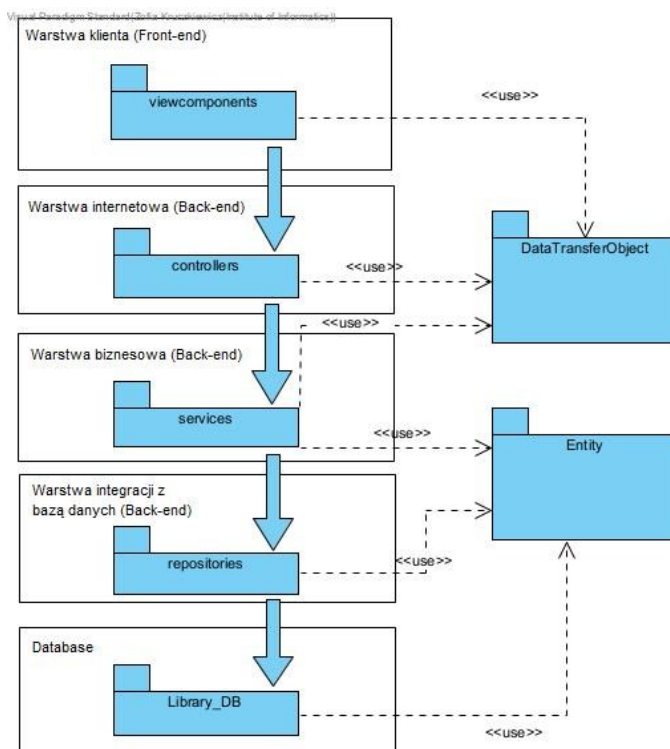
4.1.2. Diagram tabel wynikający z diagramu klas typu *Entity*



Poniżej zrzut z ekranu jedynie jako wyjaśnienie kolumny DType, która implementuje dziedziczenie:

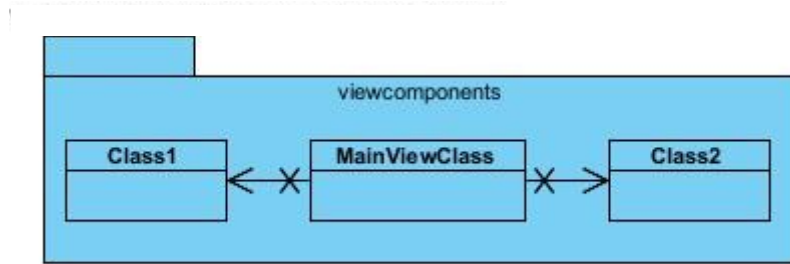


4.1.3. Diagram warstw reprezentujący zbiory głównych komponentów poszczególnych warstw: klienta (w przypadku aplikacji mobilnej), internetowej, biznesowej oraz integracji z bazą danych wynikające z technologii Spring Boot – reprezentowane przez pakiety poszczególnych warstw aplikacji. Warstwa klienta dotyczy aplikacji mobilnej. W przypadku aplikacji internetowej pakiet powinien być zastąpiony folderami stron internetowych.

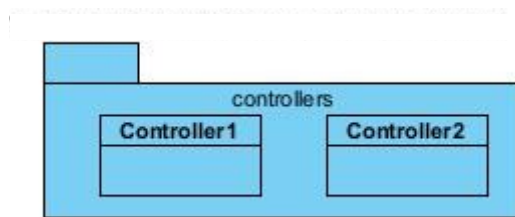


a) Należy wykonać **diagramy klas poszczególnych pakietów** reprezentujących komponenty poszczególnych warstw aplikacji (4 diagramy oraz 5-y reprezentujący obiekty transferowe aplikacji)

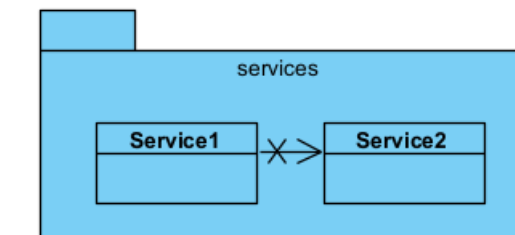
Struktura warstwy klienta (w przypadku aplikacji mobilnej, React)



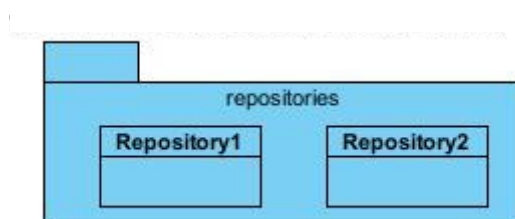
Struktura warstwy internetowej



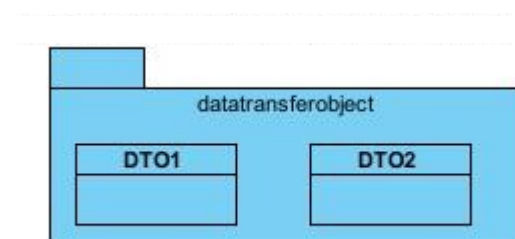
Struktura warstwy biznesowej



Struktura warstwy integracji z bazą danych

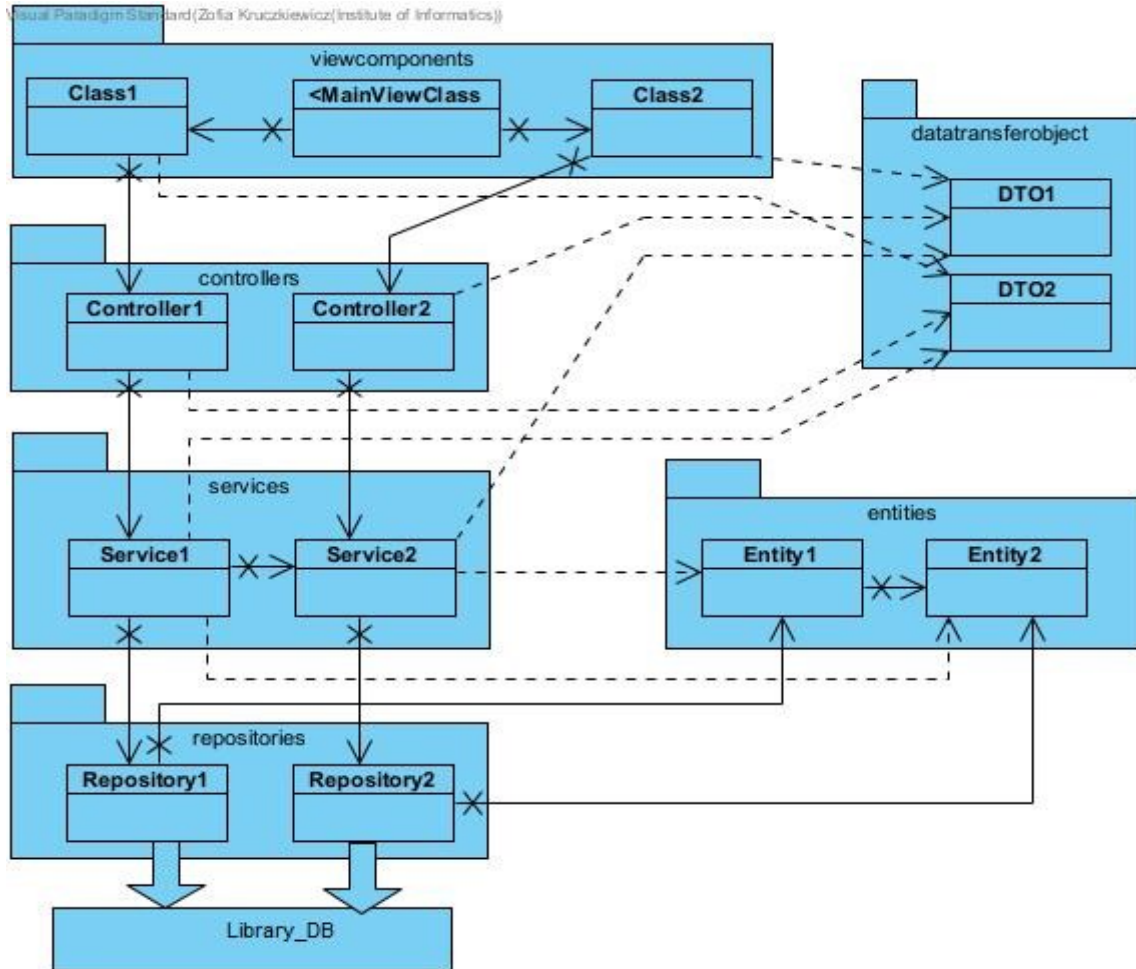


Struktura klas obiektów transferowych



b) Należy wykonać diagramy klas poszczególnych warstw pokazujące jedynie te klasy, które są wykorzystane do obsługi kodu reprezentującego implementację wybranych przypadków użycia (modelowanych za pomocą diagramów aktywności w rozdziale 3)

np. Model struktury klas przypadku użycia PU Nazwa1



Model struktury klas przypadku użycia PU Nazwa2 (diagram wg dla PU Nazwa1)



Model struktury klas przypadku użycia PU Nazwa3 (diagram wg dla PU Nazwa1)



4.2. Projekt funkcji aplikacji

Rozdział ten powinien zawierać projekt funkcji do przetwarzania danych obejmujący:

- 3) Szczegółowe diagramy aktywności wybranych funkcji aplikacji zawierające tzw. tory reprezentujące warstwy aplikacji: warstwę klienta, warstwę internetową, warstwę biznesową oraz warstwę integracji z bazą danych. Należy w poszczególnych torach przedstawić procesy zachodzące w poszczególnych warstwach aplikacji (kontynuacja p.3.3)

Warstwa klienta	Warstwa internetowa	Warstwa biznesowa	Warstwa integracji z bazą danych	Silnik bazy danych

Projekt przypadku użycia PU Nazwa1

Warstwa klienta	Warstwa internetowa	Warstwa biznesowa	Warstwa integracji z bazą danych	Silnik bazy danych

Projekt przypadku użycia PU Nazwa2

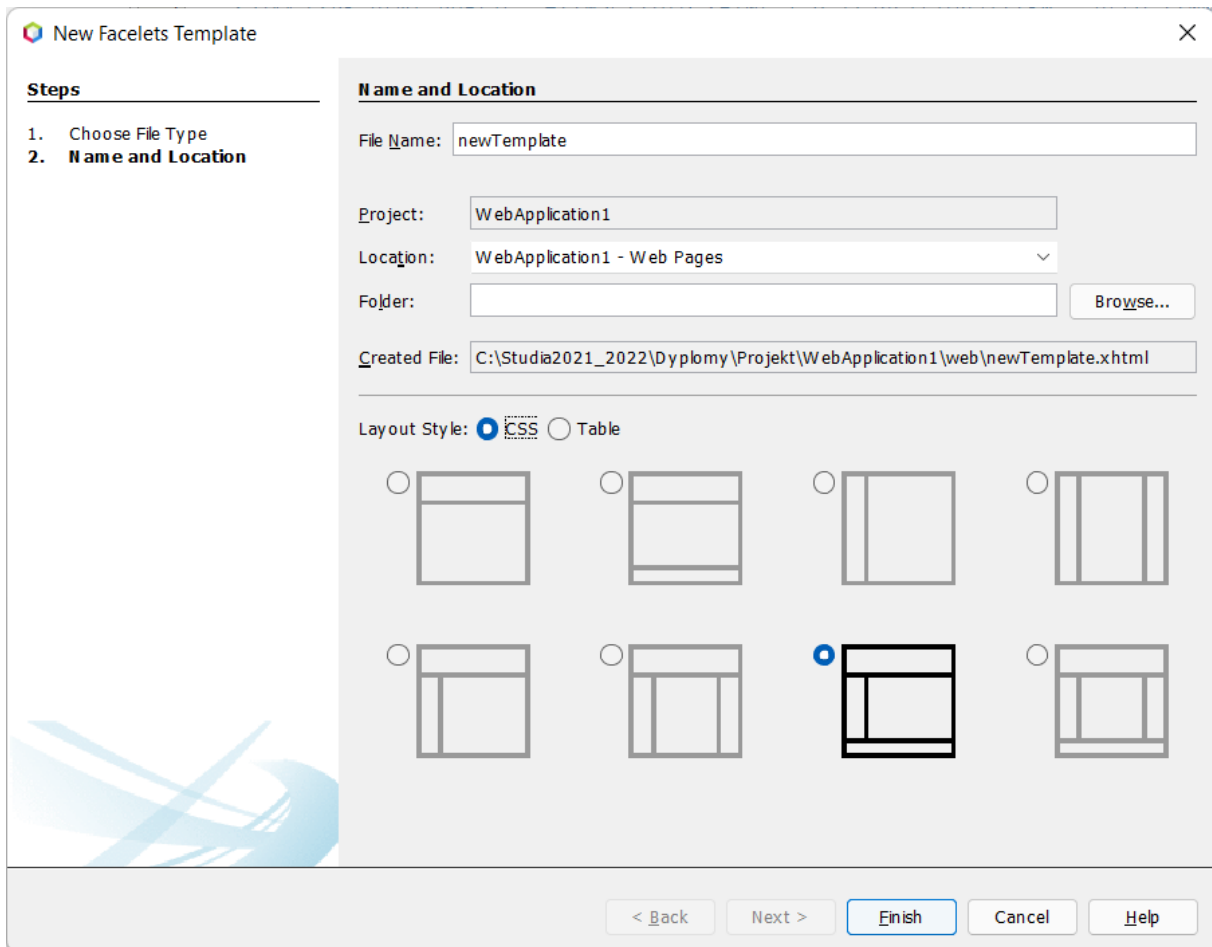
Warstwa klienta	Warstwa internetowa	Warstwa biznesowa	Warstwa integracji z bazą danych	Silnik bazy danych

Model struktury klas przypadku użycia PU Nazwa3

4.3. Projekt formularzy

Opis i rysunki prezentujące szablony wybranych formularzy

Przykłady szablonów formularzy internetowych i desktopowych



1) Szablony formularzy dla urządzeń mobilnych:

[Foundation - Material Design](https://material.io/design/foundation-overview) czyli <https://material.io/design/foundation-overview>

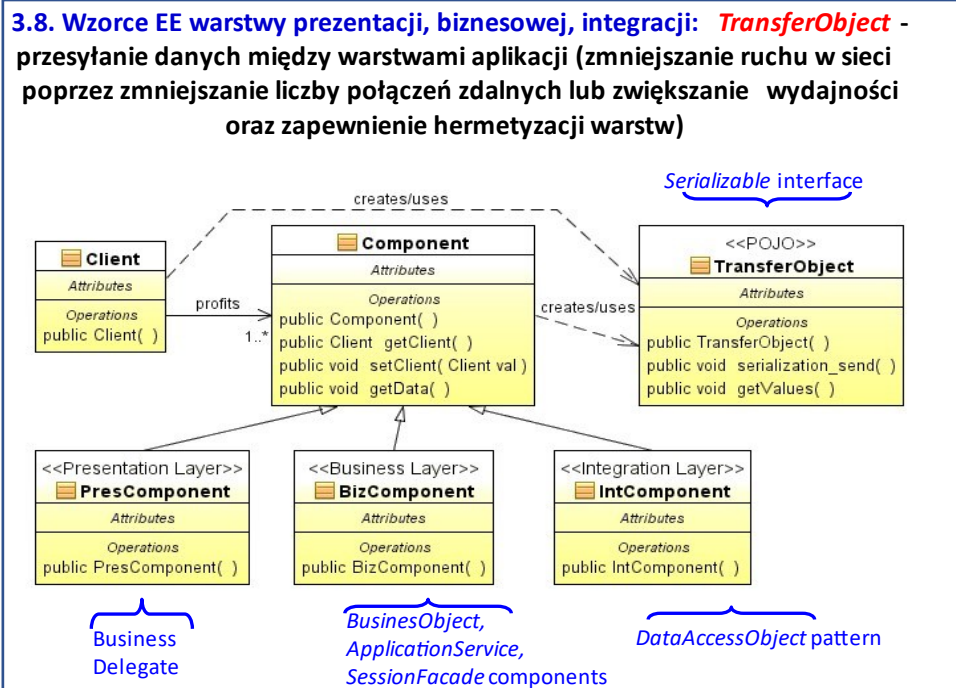
2) Należy przynajmniej dla każdego przypadku użycia zaprezentować propozycję formularzy

5. Rozdział 5

Opis implementacji (czyli kodu klas wykorzystanych do obsługi zaprojektowanych funkcji wynikającej z wybranych przypadków użycia) w poszczególnych warstwach oprogramowania wg podanej tabeli ze str. 1-4.

Wyjaśnienie, dlaczego należy zastosować obiekty transferowe (str.7, 8, 9)

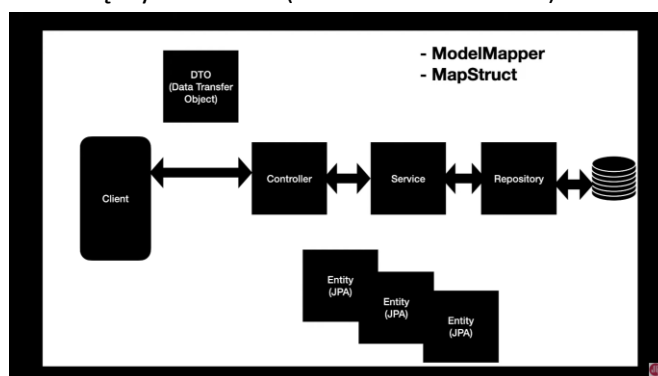
Poniżej ogólna definicja wzorca projektowego *TransferObject* ze strony



Na stronie: [Spring Boot DTO Example - Entity To DTO Conversion \(javaguides.net\)](https://www.javaguides.net/2021/02/spring-boot-dto-example-entity-to-dto.html) czyli

<https://www.javaguides.net/2021/02/spring-boot-dto-example-entity-to-dto.html>

przedstawiono zastosowanie obiektu transferowego między warstwą klienta i warstwą internetową. Ten sposób należy koniecznie zastosować również pomiędzy warstwą internetową i biznesową. Poniżej pokazuję architekturę tej propozycji, która musi być zastosowana między **Controller** (warstwa internetowa) i **Service** (warstwa biznesowa). Obiekty



transferowe należy wykonać bez użycia biblioteki lombok. Należy definicję klasy obiektu transferowego dostosować do potrzeb tworzonych modeli widoków przez **Controller** i jednocześnie do potrzeb przetwarzania tych danych przez **Service**. Np. dane osoby przy zapisie różnią od danych podczas jedynie jej wyszukiwania – i to powinno być uwzględnione przez projektowaniu klas obiektów transferowych. Wskazany tutorial pokazuje jedynie sposób używania obiektów transferowych i nie wyczerpuje różnych przypadków ich zastosowania.

6. Testy funkcji przedstawionych w rozdziale 4 i 5.

Konieczne jest przedstawienie testów funkcji: koncepcji, wyników oraz podsumowanie rezultatów.