

# Projekt zespołowy - przykład

Plan:

1. Opis projektu – założenia, funkcje.
2. Analiza systemowa:
  - a. przypadki użycia,
  - b. diagram przypadków użycia, diagram sekwencji,
  - c. podział prac w projekcie,
  - d. etapy wykonania projektu.
3. Projekt architektury:
  - a. wybór technologii,
  - b. projekt architektury aplikacji,
  - c. projekt bazy danych,
  - d. przypadki użycia w modułach,
  - e. projekt API backend'u,
  - f. diagram klas.
4. Implementacja i testowanie aplikacji.

---

## 1. Opis projektu – założenia, cele projektowe

*W tym przykładzie zostanie przedstawiony projekt aplikacji służącej do obsługi wypożyczalni DVD, niektóre elementy pochodzą z innych projektów i mają charakter czysto poglądowy.*

Założenia:

Aplikacja ma za zadanie realizację funkcjonalności pozwalającej na: obsługę użytkowników, zarządzanie użytkownikami, nadawanie praw użytkownikom, wyświetlanie treści serwisu wypożyczalni, przeglądaniem szczegółowych informacji o filmach, rezerwowaniem płyt do wypożyczenia, możliwością wypożyczenia więcej niż jednej płyty (koszyk), wysyłanie maili z przypomnieniem o terminie oddania płyt lub przekroczeniu tego terminu, wyświetlaniem statystyk wypożyczeń, tworzenia promocji, wysyłania maili newsletterów, wysyłania maili reklamowych (za zgodą) .... *i inne – tu należy wypisać wszelkie możliwe funkcjonalności.*

Funkcje:

- tworzenie konta administratora,
- zarządzanie kontem administratora,
- tworzenie użytkownika,
- zarządzanie kontem użytkownika,
- tworzenie/edycja pozycji filmu,
- dodanie/usunięcie filmu do/z rezerwacji,
- dodanie filmu do koszyka,
- usunięcie filmu z koszyka,
- ... itp., *funkcje powinny pokrywać założenia*

## 2. Analiza systemowa

### a. przypadki użycia

**Przypadek UC-1**

**Identyfikator: UC-1**

**Nazwa:** Założenie konta przez klienta

**Opis:** Zapewnienie klientowi możliwości założenia konta

**Stan początkowy:** Klient nie posiada swojego konta, nie może wypożyczać filmów

**Stan końcowy:** Klient ma własne konto, dzięki czemu może wypożyczać filmy

**Aktorzy pierwszoplanowi:** Klient

**Aktorzy drugoplanowi:** -

**Podstawowy przebieg zdarzeń:**

1. Przypadek użycia zaczyna się, kiedy klient chce zarejestrować się do serwisu
2. Klient wyświetla stronę główną
3. Klient klika w przycisk "zarejestruj"
4. System wyświetla kartę do rejestracji
5. Klient wpisuje dane
6. System sprawdza poprawność danych [Alternatywa A]
7. Klient zatwierdza rejestrację klikając w przycisk "zarejestruj"
8. Klient loguje się na swoją pocztę, żeby potwierdzić rejestrację
9. Przypadek użycia kończy się, kiedy klient może się zalogować na swoje konto

**Alternatywny przebieg zdarzeń:** Dane do rejestracji nie były poprawne

A6. twierdza błąd w danych

A7. System wyświetla informację o błędnie wypisanych danych i prośbę o ich poprawienie

A8. Przypadek użycia jest kontynuowany w punkcie 5 podstawowego przebiegu zdarzeń

**Wyjątki:**

5a. Klient naciska przycisk "Anuluj". Wyświetlona zostaje strona główna serwu. Przypadek użycia kończy się

**Przypadek UC-2**

**Nazwa:** Klient podejmuje próbę wypożyczenia filmu

**Opis:** Zapewnienie klientowi możliwości wypożyczenia filmu

**Stan początkowy:** Klient postanawia wypożyczyć film

**Stan końcowy:** Klient wypożycza wybrany film

**Aktorzy pierwszoplanowi:** Klient

**Aktorzy drugoplanowi:** -

**Podstawowy przebieg zdarzeń:**

1. Przypadek użycia zaczyna się, kiedy klient chce wypożyczyć film
2. Klient loguje się za pomocą swoich danych
3. System wyświetla konto użytkownika
4. Klient klika w zakładkę "filmy"
5. Klient wybiera interesujący go film
6. Klient klika przycisk "wypożycz"[Alternatywa A]
7. System wyświetla komunikat o kwocie do zapłaty
8. Klient dokonuje płatności
9. Przypadek użycia kończy się, kiedy wybrany film pojawia się w sekcji "wypożyczone filmy"

**Alternatywny przypadek zdarzeń:** Nie ma filmów na stanie

A7. Brak wybranego filmu

A8. Komunikat o niedostępności filmu

A9. Możliwość wybrania innego filmu

A10. Przypadek użycia będzie kontynuowany w punkcie 6 podstawowego przebiegu zdarzeń

**Wyjątki:**

5a. Klient naciska przycisk "Anuluj". Wyświetlona zostaje strona główna serwu. Przypadek użycia kończy się

### **Przypadek UC-3**

**Nazwa:** Administrator dodaje nową pozycję filmową do biblioteki

**Opis:** Administrator może dodać film do bazy danych za pomocą konta administratora

**Stan początkowy:** Administrator jest zalogowany do swojego konta, film nie jest wprowadzony do bazy

**Stan końcowy:** Film został dodany do bazy

**Aktorzy pierwszoplanowi:** Administrator

**Aktorzy drugoplanowi:** -

**Podstawowy przebieg zdarzeń:**

1. Przypadek użycia zaczyna się, gdy zalogowany administrator widzi ekran główny panelu administracyjnego
2. Administrator przechodzi do zakładki "dodaj film"
3. System pokazuje skrypt dodawania filmów
4. Administrator wprowadza informacje o filmie
5. Administrator zatwierdza dane[Alternatywa A]
6. System aktualizuje swoją bazę
7. Wyświetla się komunikat o poprawnym dodaniu filmu
8. Przypadek użycia kończy się, gdy film zostanie dodany do bazy

**Alternatywny przypadek zdarzeń:** Błędne informacje o filmie

A5. Źle wprowadzone dane

A6. Wyskakuje komunikat o błędzie

A7. Administrator decyduje, czy chce wprowadzić raz jeszcze dane

A8. Przypadek użycia będzie kontynuowany w punkcie 4 podstawowego przebiegu zdarzeń

### **Przypadek UC-4**

**Nazwa:** Klient wystawia opinię o filmie

**Opis:** Klient ma możliwość wystawienia opinii o filmie

**Stan początkowy:** Klient jest zalogowany na swoje konto i chce dodać opinię o filmie

**Stan końcowy:** Klient dodał opinię o wybranym filmie

**Aktorzy pierwszoplanowi:** Klient

**Aktorzy drugoplanowi:** -

**Podstawowy przebieg zdarzeń:**

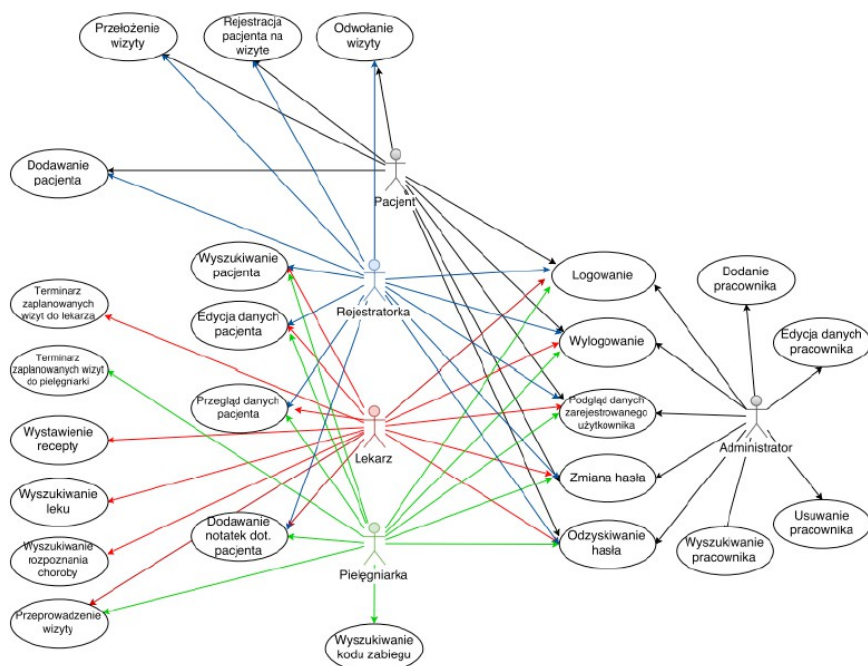
1. Przypadek użycia zaczyna się, gdy klient widzi ekran główny panelu klienta
2. Klient przechodzi do zakładki "ocena"
3. System pokazuje skrypt dodawania opinii
4. Klient wprowadza opinię o filmie
5. Klient zatwierdza opinię
6. Przypadek użycia kończy się, kiedy administrator dostaje opinię do zatwierdzenia

### **Przypadek UC-5 ...**

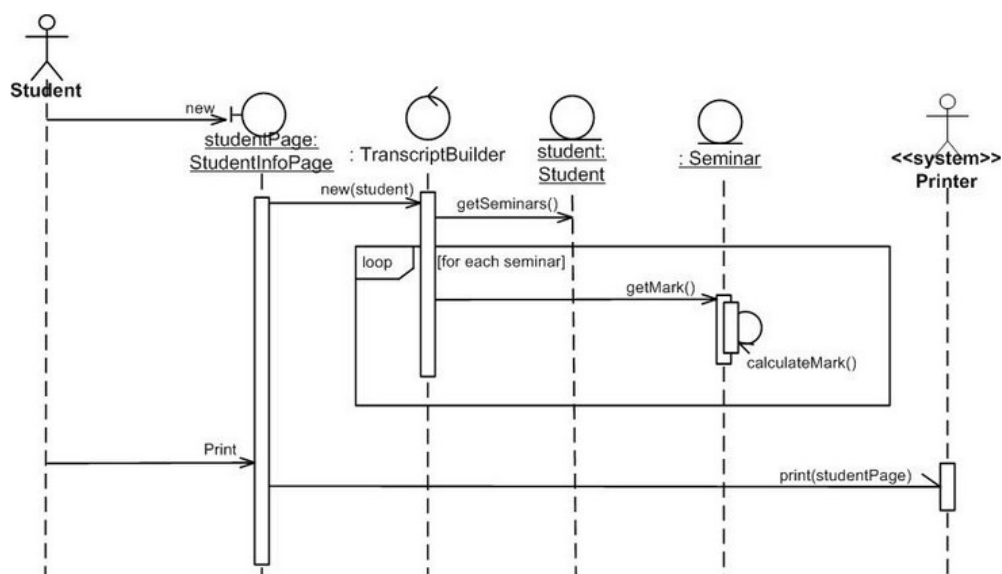
*Aplikacja powinna mieć co najmniej 30 przypadków użycia, pozostałe przypadki należy opisywać podobnie jak zostało to przedstawione wyżej. .*

### **b. diagram przypadków użycia, diagram sekwencji**

*Zazwyczaj zamieszcza się diagram przypadków użycia – diagram sekwencji jest opcjonalny i dotyczy głównie bardziej złożonych funkcji backend'owych. Poniżej został przedstawiony przykładowy diagram przypadków użycia.*



Poniżej został przedstawiony przykładowy diagram sekwencji.



### c. podział prac w projekcie

Jan Kowalski : design interfejsu użytkownika, testy funkcjonalne, opiekun dokumentacji,  
 Karol Nowak : implementacja frontendu (HTML, CSS),  
 Cezary Żak : implementacja frontendu (AngularJS),  
 Anna Potocka : implementacja backendu (Java EE).

### d. etapy wykonania projektu

Etap	Data wykonania	Nazwa elementu projektu	Odpowiedzialna(y)
I	30.10.2016	Analiza systemowa (w 90%) i dokumentacja techniczna (w 30%)	Jan Kowalski
II	20.11.2016	Analiza systemowa (w 100%) i dokumentacja	Jan Kowalski

		techniczna (w 100%)	
III	20.11.2016	Implementacja backendu (40%)	Anna Potocka
IV	11.12.2016	Implementacja frontendu (AngularJS) (60%)	Cezary Żak
...	...	...	...

### 3. Projekt architektury

#### a. wybór technologii

Aplikacja będzie korzystała z:

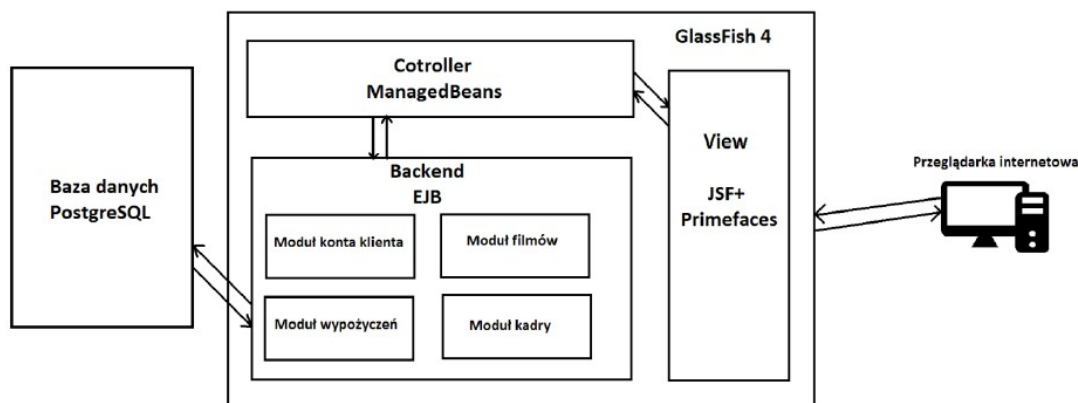
- bazy danych - osobny serwer z baza danych w naszym przypadku PostgreSQL, który przechowuje dane,
- technologii EJB - moduł z metodami serwisowymi (dostępu do bazy) które operują na encjach. Metody serwisowe realizowane poprzez bezstanowe ziarna EJB. Encje - obiektowy odpowiednik tego co znajduje się w bazie danych,
- ManagedBeans - moduł komunikacji pomiędzy frontendem a backendem, każdy widok ma swój kontroler (MB)
- Frontend - JSF + Primefaces(rozszerzenie JSF'ow), czyli warstwa widoku, to co widzi USER (pliki .xhtml w których zawierają się odwołania do MB).

W realizacji aplikacji zostaną wykorzystane:

- AngularJS - framework, bazujący który upraszcza tworzenie interfejsu użytkownika, mogący być klientem usług REST aplikacji Java EE,
- PrimeFaces -zawiera ponad 90 gotowych komponentów *JSF (Java Server Faces)*. Zawiera ona między innymi takie komponenty, jak: LightBox, Edytor HTML, Kalendarz, Terminarz, Grid, Menu, Wykresy, Dashboard.
- EJB (Enterprise Java Beans) -technologia działająca po stronie serwera, która jest jednym z elementów specyfikacji JEE.
- JPA (Persistence API) - Z punktu widzenia programisty jest to możliwość operowania na obiektach - zwanych **encjami** - oraz zapisywania wyników operacji do relacyjnej bazy danych.

#### b. projekt architektury aplikacji

Na podstawie przypadków użycia i wymagań aplikacji pogładowa architektura wygląda jak niżej:



#### c. projekt bazy danych

Baza danych ma za zadanie przechowywać informacje. Do projektu aplikacji została wykorzystana relacyjna baza danych, w której tabele są powiązane ze sobą przy pomocy relacji. Jest to jeden z najprostszych i najczęściej stosowanych modeli bazy danych.

Poniżej zostaną zaprezentowane tabele bazy danych oraz opis poszczególnych kolumn i relacji.

Tabela **item** - tabela zawierająca przedmioty wystawiane na aukcje

Kolumny:

- id item - identyfikator przedmiotu
- id category - identyfikator kategorii
- item name - nazwa przedmiotu
- description - opis przedmiotu
- market price - wartość przedmiotu
- delivery price - cena dostawy
- image - zdjęcie przedmiotu

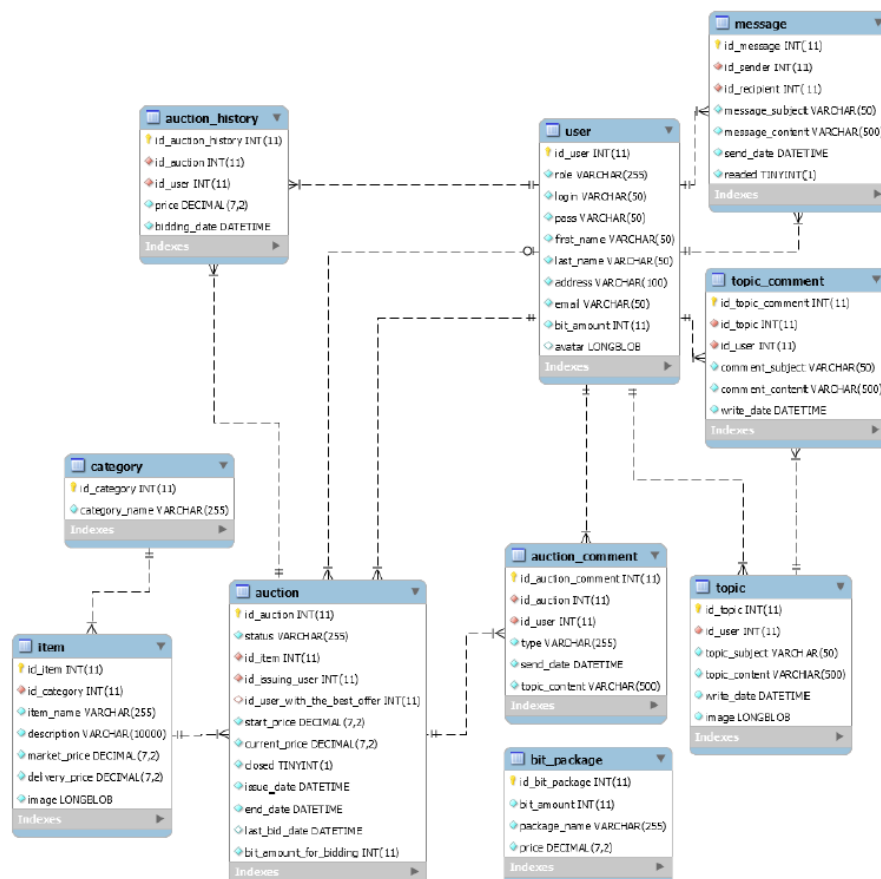
Tabela **user** - tabela zawierająca dane użytkowników zarejestrowanych w aplikacji

Kolumny:

- id user - identyfikator użytkownika
- role - typ użytkownika (administrator, moderator, klient)
- login - unikalna nazwa użytkownika
- password - hasło logowania
- first name, last name, address, email - imię, nazwisko, adres zamieszkania, adres email
- avatar - obrazek zdefiniowany przez użytkownika

*Tabele należy opisać jak wyżej, dobrze jest także podać ich typ SQLowy, jeśli takowy jest już znany. Elementem projektu bazy danych jest jej diagram i taki też należy zamieścić, przykład jest poniżej.*

Diagram bazy danych:



#### **d. przypadki użycia w modułach**

*W zależności od modułu należy wskazać jaki będzie jego interface oraz jakie funkcje będzie udostępniał.*

##### **Moduł konta klienta.**

- UC-1 Założenie konta przez klienta
- UC-6 Administrator usuwa konto klienta
- UC-13 Logowanie do konta klienta
- UC-15 Klient zapisuje się do newsletter

##### **Moduł wypożyczeń.**

- UC-2 Klient podejmuje próbę wypożyczenia filmu
- UC-8 Zwrot filmu
- UC-9 Sprawdzanie terminu zwrotu
- UC-16 Naliczanie kary za niedotrzymanie terminu
- UC-19 Email do klienta z przypomnieniem o oddaniu filmu

*Pozostałym modułom również należy opisać jak wyżej ...*

#### **e. projekt API backend'u**

*W zależności od modułu należy wskazać jaki będzie jego interface oraz jakie funkcje będzie udostępniał.*

##### **Moduł klienta – sposób komunikacji REST:**

- +dodajKlienta
- +edytujKlienta
- +usunKlienta
- +sprawdzDaneLogowaniaKlienta
- +dodajKlientaDoSubskrypcji
- +usunKlientaZSubskrypcji

##### **Moduł wypożyczeni – sposób komunikacji EJB (remote/local):**

- +dodajWypożyczenie
- +sprawdzPoprawnośćWypożyczenia
- +zakonczWypożyczenie
- +sprawdzTerminZwrotu
- +naliczKare
- +wyslijPrzypomnienie

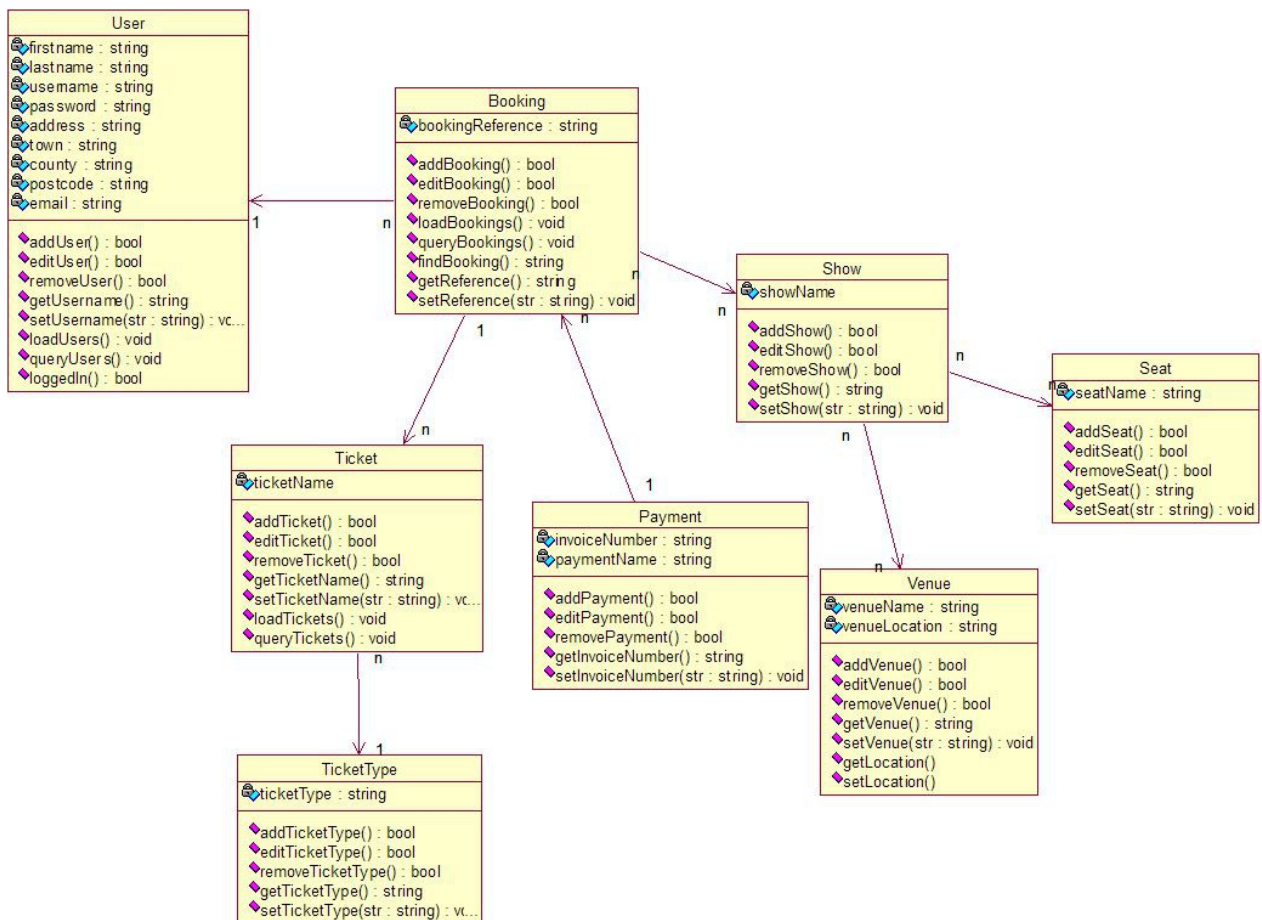
##### **Moduł filmów - – sposób komunikacji: remote EJB i REST:**

- +dodajFilm
- +edytujFilm
- +usunFilme
- +pobierzNiepotwierdzoneOpinie
- +akceptujOpinie
- +odrzucOpinie
- +dodajOceneFilmu

*Opis pozostałych modułów i ich API...*

## f. diagram klas

Poniżej został przedstawiony przykładowy diagram klas.



## 4. Implementacja i testowanie aplikacji

Ten etap polega na opracowaniu i przedstawieniu w formie pisemnej sposobu testowania aplikacji (np. w formie scenariuszy testowych, wynikających z przypadków użycia oraz diagramów sekwencji). Ponadto należy przedstawić działającą aplikację – co najmniej 10 zaimplementowanych przypadków użycia.