

Projektowanie systemu

Laboratorium 2

Wymagania

1. Oprogramowanie umożliwiające tworzenie diagramów sekwencji (PlantUML – rekomendowane np. w postaci wtyczki do programu Visual Studio Code, Microsoft Visio, Draw.io lub inne)
2. Oprogramowanie umożliwiające tworzenie schematów blokowych.

Wstęp

Projektowanie systemu to proces przygotowania elementów systemu takich jak:

- architektura,
- moduły
- komponenty i ich interfejsy
- dane przechodzące przez system.

Każdy z projektów wymaga osadzenia docelowego rozwiązania w odpowiednim środowisku, dopasowanym do wymagań projektowych. W związku z tym nie można wskazać uniwersalnych rozwiązań pasujących do każdego tworzonego systemu. Dobrze zaprojektowany system powinien spełniać następujące postulaty:

- Zapewnienie poufności przetwarzanych i przechowywanych danych
- Weryfikacja integralności przekazywanych komunikatów
- Gwarancja dostępności usług
- Autoryzacja dostępu do zasobów udostępnianych przez system
- Zapewnienie wydajności systemu

Projektowanie systemu jest fazą wstępną, która pozwala zrozumieć zależności pomiędzy poszczególnymi modułami tworzonego rozwiązania. Odpowiednio przygotowany projekt stanowi ważny element dokumentacji oraz pozwala na przejście do sprawnego procesu implementacji systemu oraz konfiguracji środowiska docelowego, na którym zostaną uruchomione przygotowane usługi. Projektowany system składa się z następujących elementów:

1. **Architektura** – to model koncepcyjny, który odnosi się do struktury i sposobu działania. Architektura może być przedstawiona np. za pomocą schematu blokowego.
2. **Moduły** - są to komponenty, które obsługują jedno określone zadanie w systemie. System składa się z kombinacji wielu modułów.

3. **Komponenty** - zapewniają określoną funkcję lub grupę powiązanych funkcji. Składają się z modułów.
4. **Interfejsy** - połączenie przez które elementy systemu wymieniają się informacjami i odnoszą się do siebie.
5. **Dane** - zarządzanie przepływem informacji i danych.

Zadanie 1

Zadanie polega na utworzeniu diagramów sekwencji dla określonych procesów z uwzględnieniem wymagań określonych w treści.

Projektowany system to popularna aplikacja, której zasoby wystawione są za pośrednictwem interfejsu REST API, a klientem aplikacji jest inna aplikacja serwerowa. Dla uproszczenia przyjmijmy usługę pozwalającą na realizację prostego interfejsu CRUD (Create, Read, Update, Delete) przetwarzającego dowolne dane np. dane o użytkowniku. Diagramy powinny prezentować proces tworzenia użytkownika (lub dowolnie wybranego, innego modelu), pobierania informacji na podstawie identyfikatora obiektu, aktualizacji danych obiektu, pobierania informacji o wszystkich użytkownikach przechowywanych w systemie. Przy tworzeniu diagramu należy uwzględnić, że dostęp do zasobów systemu powinien uwzględniać autoryzację w oparciu o framework OAuth2.0 Client Credentials ([RFC 6749](#))*. W związku z tym interakcje z serwerem zasobów powinno poprzedzać odpowiednie wywołanie serwera autoryzacji a następnie przekazanie tokena do serwera zasobów i jego weryfikacja. Na diagramie należy uwzględnić odpowiednie walidacje (np. czy użytkownik o podanym identyfikatorze istnieje). Diagramy powinny być jak najbardziej szczegółowe – osoba czytająca nie powinna mieć wątpliwości w jaki sposób działa system oraz jakie interakcje zachodzą pomiędzy jego poszczególnymi modułami.

** Alternatywnie metoda uwierzytelniania aplikacji klienckiej oraz autoryzacji i kontroli dostępu do zasobów może zostać zastąpiona dowolnie wybranym mechanizmem, pasującym do opisu systemu.*

Zadanie 2

Poprzednie zadanie pozwoliło na określenie wymagań dotyczących interfejsów i interakcji pomiędzy modułami projektowanego systemu.

1. Na podstawie określonych wymagań za pomocą schematu blokowego przedstawić projekt architektury systemu. Projekt powinien uwzględniać zależności pomiędzy wymienionymi modułami (serwer autoryzacji, serwer zasobów) oraz sposób przechowywania i dostępu do danych (baza danych). Utworzony schemat powinien przedstawiać mechanizmy gwarantujące wysoką dostępność i niezawodność systemu, a więc np. uwzględnienie w architekturze mechanizmu loadbalancing, skalowania horyzontalnego, cache'owania, replikacji bazy danych.
2. Każdy z wymienionych modułów powinien zostać krótko opisany z wyszczególnieniem jego roli w projektowanym systemie. Przykładowe diagramy: [diagram 1](#), [diagram 2](#)

Zadanie 3

Przeanalizować obiekty występujące podczas przetwarzania danych w systemie i przedstawić je za pomocą dowolnego diagramu (np. diagram encji, diagram klas). Zgodnie z wymaganiami najprostsza wersja systemu powinna zawierać przynajmniej dwie encje – użytkownik (lub dowolnie wybrany model) oraz token dostępu. Zadanie to ma na celu przemyślenie struktury przetwarzanych danych, wymagań dotyczących formatu pól opisujących encje/klasy, które z właściwości obiektów powinny być unikalne (np. pesel), jakie walidacje przy przetwarzaniu danych należy uwzględnić. Oczywiście projektowany system można dowolnie rozbudowywać np. poprzez wprowadzenie typów enumerycznych lub dodatkowych klas. W odniesieniu do serwera autoryzacji należy zapoznać się ze standardową strukturą tokena JWT. Utworzone diagramy załączyć do sprawozdania i krótko opisać – encje oraz przechowywane właściwości.