**4+1 Architecture View Model for Full Stack Application (Spring Boot Backend and React Frontend)**

Contents

[Introduction 1](#_Toc182690652)

[Purpose 1](#_Toc182690653)

[Scope 1](#_Toc182690654)

[Overview 2](#_Toc182690655)

[1. Logical View 2](#_Toc182690656)

[Description: 2](#_Toc182690657)

[2. Development View 3](#_Toc182690658)

[Description: 3](#_Toc182690659)

[3. Process View 3](#_Toc182690660)

[Description: 3](#_Toc182690661)

[4. Physical View 4](#_Toc182690662)

[Description: 4](#_Toc182690663)

[5. Scenarios (Use Case View) 4](#_Toc182690664)

[Description: 4](#_Toc182690665)

[Conclusion 5](#_Toc182690666)

# Introduction

## Purpose

* + This section explains the reason for creating the document. It outlines the objectives of the architecture document and what it aims to achieve.
  + Example: "The purpose of this document is to provide a comprehensive architectural overview of the Online Bookstore Application, focusing on its structure, components, interactions, and deployment."

## Scope

* + This section defines the boundaries of the document. It specifies what aspects of the system architecture are covered and any limitations.
  + Example: "This document covers the architectural design of the Online Bookstore Application, including its logical, development, process, physical, and scenario views. It does not cover implementation details or specific coding standards."

## Overview

* + This section provides a high-level summary of the document and the 4+1 architectural views. It serves as an introduction to the subsequent detailed sections.
  + Example: "This document uses the 4+1 architectural view model to describe the architecture of the Online Bookstore Application. The views include:
    - Logical View: Describes the system’s functional requirements.
    - Development View: Describes the system’s static organization in the development environment.
    - Process View: Describes the system’s dynamic aspects and processes.
    - Physical View: Describes the system’s physical deployment.
    - Scenarios (Use Case View): Describes the system’s functionality from the end-user’s perspective."

# 1. Logical View

***Focus: Describes the functionality of the system that is visible to the end-user.***

## Description:

* + This view focuses on the functionality that the system provides to end-users. It includes the various classes, their attributes, methods, and relationships among them.
* **Example:**
  + **Class Diagram:**
    - Shows the main classes like User, Book, Order, and OrderItem along with their attributes and methods.
    - **Sample:**
    - - Book: title (String), author (String), isbn (String), price (double), quantity (int)
    - - User: id (Long), username (String), password (String), email (String), role (String)
    - - Order: id (Long), user (User), totalAmount (double), orderDate (Date)
    - - OrderItem: id (Long), book (Book), order (Order), quantity (int), price (double)
  + **State Diagram:**
    - Shows the different states of an object and transitions between these states.
    - For example, the lifecycle of an Order (Created, Processing, Shipped, Delivered).

# 2. Development View

***Focus: Describes the static organization of the software modules in the development environment.***

## Description:

* This view describes the static organization of the software in its development environment. It includes the main components, their responsibilities, and dependencies.
* **Example**:
  + **Component Diagram**:
    - Illustrates the main components of the application such as frontend components (React components), backend components (Spring Boot controllers, services, repositories), and how they interact.
  + **Package Diagram:**
    - Shows how the various classes and interfaces are grouped into packages.

# 3. Process View

***Focus: Describes the dynamic aspects of the system, system processes, and interactions between components.***

## Description:

* This view describes the dynamic aspects of the system, including system processes and interactions between components.
* **Example**:
  + **Sequence Diagram**:
    - Shows the interaction between the frontend (React components) and backend (Spring Boot REST APIs) during a specific use case, such as placing an order.
  + **Activity Diagram:**
    - Describes the flow of activities in a specific process, such as user authentication or the checkout process.

# 4. Physical View

***Focus: Describes the physical deployment of the system's hardware and software.***

## Description:

* This view describes the physical deployment of the system’s hardware and software.
* **Example**:
  + **Deployment Diagram**:
    - Illustrates how software components are deployed on hardware nodes.
    - For instance: -

- Frontend (React) deployed on web server (e.g., Nginx).

- Backend (Spring Boot) deployed on application server (e.g., Tomcat).

- Database (e.g., MySQL) deployed on a database server.

* + **Node Diagram:**
    - Describes the physical nodes and their interactions.

# 5. Scenarios (Use Case View)

***Focus: Describes the functionality of the system from the end-user's perspective.***

## Description:

* This view describes the functionality of the system from the end-user's perspective, capturing the main functional requirements and interactions between users and the system.
* ***Example:***
  + ***Use Case Diagram:***
    - Captures main use cases like "Browse Books", "Place Order", "Manage Inventory", and "User Authentication".
    - ***Sample:***
    - - Actor: Customer
      * - Use Cases: Browse Books, Place Order
    - - Actor: Admin
      * - Use Cases: Manage Inventory, Process Orders

# Conclusion

In this section, you provide a brief summary of the key architectural decisions and highlights discussed in the document. It may include:

* Recap of the system architecture using the 4+1 view model.
* Final thoughts on the architecture's suitability for meeting the system's requirements.
* Any recommendations for future improvements or considerations.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*