Ajaya
DATE / /
PAGE NO.:

Name: Manu Sharma

Roll No: 2023BCEX1250

Reg No: 2023PUFCEBCEX135563

Subject: BCECCES102

Section: G

Submitted to: Mr. Aaditya Nautiyal Sir

Q2 Dependency Injection & Bean lifecycle ⇒

A dependency team is building a modular banking app. Using the spring framework. The project manager insists that are components should be loosely coupled so that ~~all components should~~ they can be replaced or modified without changing other parts of the system. The team members decide to Dependency Injection (DI) and Bean scopes in spring.

Question

Explain how spring Dependency Injection helps in achieving loose coupling in this system. Also, describe how Bean scopes could affect the behaviour of the Notification service bean when multiple users access the application

Teacher's Signature.....

Manu Sharma, 2023113568, 7623BCEX1250

DATE / /
PAGE NO.:

In large-scale application like a Banking System, maintaing loose coupling between various modules (eg:- Account service Notification Service and Transaction service) is crucial for flexibility and maintainability. The spring framework achieve the using Dependency Injection (DI) and Bean scopes

Dependency Injection (DI):-

Dependency Injection is a design pattern used in spring to achieve loose coupling between different components of an application, in DI, objects are not created manually inside a class, instead they are injected by the spring container.

For example

```
public class AccountService {
    private NotificationService NotificationService;

    public void setNotificationService(NotificationService
                                       notificationService)
    {
        this.notificationService = notificationService;
    }
}
```

Teacher's Signature.....

Mamu Sharma , 2023113568 , 2023BCEX1250

AJAYA
DATE / /
PAGE NO.:

How it helps in loose coupling:-

→ Each class depends on interfaces rather than concrete implementations.

→ Any component (like NotificationService) can be easily replaced or modified without affecting other parts of the system

→ This makes the application more flexible, testable and maintainable

Bean Scopes in Spring

Spring beans have different scopes, which define how and when the bean instances are created and shared.

(i) Singleton Scope:-

→ Default scope in Spring

→ Only one instance of the bean is created for the entire spring container.

→ Example:- A NotificationService bean configured

(ii) Prototype Scope:-

→ A new instance of the bean is created

Teacher's Signature.....

Name Manu Sharma Reg No: 2023/13566 Roll No 2023BCEX125O

every time it is created every time it is requested

→ Useful for components that maintain user-specific or temporary data.

Effect of Bean Scope in Notification service:-

- Singleton: One shared instance serves all users. It is memory efficient but may cause data conflicts if the bean holds user-specific information

- Prototype: A new instance is created for each user request.
  It isolates user data and ensures thread safety but consumes more memory.