**Project Design Phase - Solution Architecture**

| Field | Value |
|---|---|
| **Date** | 02 November 2025 |
| **Team ID** | NM2025TMID03898 |
| **Project Name** | Laptop Request Catalog Item |
| **Maximum Marks** | 4 Marks |

**Solution Architecture:**

**Goals of the Architecture:**

- Provide a self-service portal for laptop requests using Service Catalog
- Maintain data integrity through dynamic form validation
- Enable deployment across instances through update sets

**Key Components:**

- Service Catalog Item (sc_cat_item table)
- Four custom variables (laptop_model, justification, additional_accessories, accessories_details)
- Catalog UI Policy (conditional field visibility)
- UI Action on Shopping Cart (sc_cart) for form reset
- Local Update Set for change management

**Development Phases:**

1. Create local update set named "Laptop Request"
2. Create service catalog item under Hardware category
3. Add four custom variables to the catalog item
4. Implement UI policy for conditional field visibility
5. Create UI action for form reset functionality
6. Export update set as XML
7. Import and deploy to target instance
8. Test catalog item functionality (variable visibility, form reset)

**Solution Architecture Description:**

The solution architecture is designed to provide a streamlined laptop request process within the ServiceNow platform by implementing a Service Catalog item with dynamic form behavior. The architecture focuses on ensuring data

completeness and user experience across the service catalog and variable tables, using a catalog UI policy to control field visibility.

This UI policy checks the state of the "additional_accessories" checkbox through a condition and automatically shows/hides the "accessories_details" field accordingly, making it mandatory when visible. The development process involves creating an update set, building the catalog item with variables, applying UI policies and UI actions, and verifying the functionality through testing. Additionally, a UI action on the sc_cart table provides form reset capability using client-side scripting.

This architecture reduces manual processing, enhances user self-service capabilities, and promotes operational efficiency in IT service management environments.

**Solution Architecture Diagram:**