

Project Design Phase

Solution Architecture

Date	01.11.2025
Team ID	NM2025TMID07569
Project Name	Laptop Request Catalog Item
Maximum Marks	4 Marks

Goals of the Architecture:

- Provide an automated and user-friendly system for submitting laptop requests.
- Ensure data accuracy and consistency through dynamic Service Catalog behavior.
- Streamline approval workflows and reduce manual data entry errors.
- Maintain governance and traceability through update set management.

Key Components:

- **Service Catalog Item:** *Laptop Request* (core form for end-user submission).
- **Variables:** Fields like *Laptop Model*, *Additional Accessories*, and *Accessory Details*.

- **Catalog UI Policy:** Controls the visibility and mandatory state of fields dynamically.
- **UI Action (Reset Form):** Allows users to clear all form entries instantly.
- **Local Update Set:** Captures and manages all project-related configurations for deployment.

Development Phases:

1. Create a **Local Update Set** named *Laptop Request*.
2. Build a **Service Catalog Item** under the *Hardware* category.
3. Add variables for *Laptop Model*, *Additional Accessories*, and *Accessory Details*.
4. Implement **Catalog UI Policies** to dynamically display accessory fields.
5. Create a **UI Action (Reset Form)** using client-side scripting.
6. Export and import the **Update Set** to test cross-instance migration.
7. Validate and test the entire workflow for accuracy and usability.

Solution Architecture Description:

The solution architecture is designed to automate and simplify the **laptop request process** within the ServiceNow platform. It provides employees with a centralized, digital interface to request laptops, replacing traditional manual methods that caused inefficiencies.

At its core, the architecture integrates several ServiceNow components — **Service Catalog Items, Variables, UI Policies, UI Actions, and Update Sets** to create an intelligent and dynamic request workflow.

When a user opens the *Laptop Request* form, they can select a laptop model and optionally check *Additional Accessories*. Upon selection, a **Catalog UI Policy** triggers, revealing the *Accessory Details* field and marking it as mandatory. This ensures all necessary data is captured before submission.

Additionally, a **UI Action** adds a *Reset Form* button that clears all entered data, enhancing usability. The use of **Local Update Sets** ensures all configurations are properly tracked, exported, and imported between instances, maintaining consistency and version control.

This architecture improves operational efficiency, minimizes manual errors, and ensures a transparent, traceable request process. It supports both IT administrators and end-users by providing a seamless, interactive, and reliable system for hardware requests.

Example - Solution Architecture Diagram:

