**Automated Network Request Management in ServiceNow**

1. Introduction

This project implements an automated system to manage network-related service requests using ServiceNow. It simplifies request submission, approval, fulfillment, and monitoring through Service Catalog, workflows, and Flow Designer.

2. Objectives

* Create a custom Service Catalog for common network requests.
* Build dynamic forms to capture relevant details from end-users.
* Implement automated approval workflows based on request type and sensitivity.
* Enable real-time notifications to requesters and technicians.
* Integrate with external infrastructure or orchestration tools (optional).
* Provide reporting and SLA monitoring for performance tracking.

3. System Design

3.1 Architecture

* Front-end: ServiceNow Service Portal (self-service request form).
* Workflow Engine: Manages request routing, approvals, and task assignment.
* Integration Layer: (Optional) APIs/scripts to connect with network automation tools.
* Database: ServiceNow tables storing request, task, and SLA data.

3.2 Workflow

1. User submits a network request via Service Catalog.
2. Request details validated using dynamic form rules.
3. Automated approval workflow triggered.
4. Fulfillment task(s) created and assigned.
5. Notifications sent at each stage.
6. SLA timers track compliance.
7. Reports generated for analysis.

4. Implementation

### 4.1 Creation of Service Catalog

The Service Catalog in ServiceNow provides a centralized location where users can request IT and business services. For this project, the catalog is customized to handle network-related requests.

* Variables Configuration
  + Define variables to capture the request details from users.
  + Example:
    - Request Type → Choice list
    - Justification → Multi-line text field.
    - Department → Reference field linked to user’s department.
* Variables Types
  + Single Line Text
  + Choice Lists
  + Reference Fields
  + Checkboxes
* Variable Set Configuration
  + Create variable sets for commonly used inputs
  + This ensures reusability across multiple catalog items.
* Catalog UI Policy Configuration
  + Add rules to control the visibility and behavior of variables.

4.2 Creation of Tables

* Navigate to Table Creation
  + Go to System Definition > Tables
  + Click New to create a new table.
* Define Table Properties
  + Label: Human-readable name (e.g., Network Device)
  + Name: System name (e.g., u\_network\_device) → auto-filled from label
  + Extends Table: Usually Task or CMDB if you want built-in functionality; else leave blank
  + Auto-number: Optional, generates unique IDs for each record

4.3 Request Approvals

Configured approval rules in the related list. Approvals were set based on sensitivity level (e.g., manager approval, IT security approval).

4.4 Flow Designer

Overview of Flows & Actions: Reviewed available triggers and actions in Flow Designer.

Creation & Implementation: Designed a flow to automate request routing, approval, task creation, and notifications.

Key Components in Flow Designer

#### 1. Triggers

• Record Trigger: Runs when a record is created, updated, or deleted in a specific table.

• Scheduled Trigger: Runs at defined time intervals or on a schedule.

• Custom Event: Triggered by custom-defined events.

#### 2. Actions

Actions define the operations performed once a flow is triggered. Common actions include:

• Create Record: Add a new record to a table.

• Update Record: Modify details of an existing record.

• Send Notification: Deliver emails, SMS, or other types of notifications.

• Run Script: Execute custom scripts to implement advanced logic.

#### 3. Data Pills

Dynamic references to data from records or earlier steps in the flow. They are used to populate inputs for subsequent actions.

#### 4. Conditions & Decisions

Enable branching logic in flows. Example: Send an approval notification only if a specific condition is met.

#### 5. Flow Logic

Provides advanced control with decision points, loops, and wait conditions. This allows the design of complex workflows for robust automation.

5.Final Testing in End User Portal & Instance

Testing in Service Portal (End User): Verified catalog submission, approval routing, and task assignment.

Testing Emails: Checked automated email notifications for request status and approvals.

6. Results

The automated workflow successfully captured and routed network service requests. End users received clear status updates via Service Portal and email. SLAs and reporting ensured compliance with defined timelines.

7.Conclusion

The creation of the Network Request Management system in ServiceNow demonstrates how IT service automation streamlines request handling. By leveraging the Service Catalog, organizations can provide users with an intuitive way to raise requests while ensuring proper categorization and approvals. This reduces manual effort, enhances transparency, and improves response times. Overall, the project highlights the efficiency and reliability gained through ServiceNow’s service catalog and workflow capabilities.