Automated Network Request Management in ServiceNow

Final Report

1. INTRODUCTION

1.1 Project Overview

The Automated Network Request Management System is a ServiceNow-based solution designed to modernize the lifecycle of network service requests by replacing manual workflows with digital automation. By leveraging the Service Catalog for centralized intake and Flow Designer for orchestration, the project streamlines approval and fulfilment processes, significantly reducing human error and turnaround time. This system enhances operational efficiency and governance by providing real-time visibility, automated SLA tracking, and a transparent audit trail for all network operations.

1.2 Purpose

- Automate request intake, routing, and fulfilment.
- Standardize approval workflows.
- Improve visibility, reporting, and end-user experience.
- Provide a scalable framework for future IT service automations.

2. IDEATION PHASE

2.1 Problem Statement

Manual network request handling causes:

- Delays in request fulfilment
- Errors due to incomplete or inconsistent information
- Lack of standardized approvals and tracking
- Poor visibility for requesters and approvers

2.2 Empathy Map Canvas

Stakeholder	Think & feel	See & hear	Say & do	Pain Points
End Users	Frustrated with delays	System inconsistencies	Submits network requests	Requests take a long time, no updates
IT Admins	Overloaded with tickets	Complex configurations	Maintains workflows	Manual interventions, errors
Network Team	Wants clarity in the request for data	Requests missing info	Executes tasks	Human errors, inconsistent data
Approvers	Need compliance & fast approvals	Multiple approval chains	Approves/rejects requests	Lack of structured workflow

2.3 Brainstorming

- Use ServiceNow Flow Designer for automation
- Dynamic approval logic based on department & role
- Auto-populated fields for efficiency
- UI/UX improvements in Service Portal forms
- Role-based access control

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

- Requester logs in via Service Portal
- Selects Network Request catalog item
- Fills dynamic form with auto-populated fields
- Flow Designer triggers workflow
- Approval routed to manager/network group
- Task created in Network Database
- Notifications sent to requester & approver
- Request completed and closed

3.2 Solution Requirement

- Service Catalog creation with dynamic forms
- Approval workflow automation
- Email notifications
- Custom table Network Database for data storage
- Role-based access control & ACLs

3.3 Data Flow Diagram

Service Portal Form → Flow Designer Trigger → Get Catalog Variables → Create Record → Ask for Approval → Send Email → Update Record → Request Closed

3.4 Technology Stack

- **Platform:** ServiceNow PDI
- **Tools:** Flow Designer, Catalog UI Policies, UI Actions
- **Database:** ServiceNow Custom Table (u_network_database)
- **Languages:** Minimal scripting (client scripts for auto-populate)

4. PROJECT DESIGN

4.1 Problem Solution Fit

Manual request handling was replaced by an automated workflow, ensuring streamlined approvals, timely notifications, and uncompromised data integrity.

4.2 Proposed Solution

- Custom table Network Database to store requests
- Flow Designer automates task creation, approvals, and notifications
- Dynamic Service Portal forms with auto-populated fields
- UI policies for field visibility and mandatory enforcement

4.3 Solution Architecture

Components:

- **Frontend:** Service Portal Form
- **Backend:** Flow Designer, Network Database Table
- **Automation:** Approval routing, record updates, email notifications
- **Security:** ACLs for role-based access

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase	Key Activities	Duration
Phase 1 – Requirement Analysis	Stakeholder mapping, catalog scoping	1 week
Phase 2 – Backend Development	Table creation, fields, flow designer logic	2 weeks
Phase 3 – UI/UX Customization	Service Portal, dynamic forms, variables	1 week
Phase 4 – Testing & Security	QA testing, data integrity checks, ACLs	1 week
Phase 5 – Deployment & Documentation	Deploy, create manuals, project demo	1 week

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- Simulated multiple network request submissions
- Verified flow execution under concurrent requests
- Approval routing tested for department-specific conditions
- Email notifications validated for all request types
- System logs confirmed no errors or failures

7. RESULTS

7.1 Output Screenshots

- Dynamic form with auto-populated fields
- Approval task assignment in Flow Designer
- Email notification received by requester and approver
- Record created in Network Database table

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Faster request processing
- Reduced human errors
- Automated approvals and notifications
- Improved tracking and visibility
- Scalable for other IT services

Disadvantages:

- Initial setup requires ServiceNow knowledge
- Dependent on Flow Designer and Service Portal availability
- Minor scripting may be needed for complex auto-populate logic

9. CONCLUSION

The system successfully automates network request management, covering catalog creation, backend development, dynamic forms, flow automation, approvals, and notifications. It ensures efficiency, data integrity, and improved user experience while remaining scalable for future IT services.

10. FUTURE SCOPE

- Integration with Cisco DNA Centre or Ansible for auto-provisioning
- Dashboard reporting for request volumes and fulfilment metrics
- Expansion to other IT service domains (device provisioning, access requests)
- Advanced analytics for performance tracking and process optimization