**Project Documentation**

**Network Request Management Using ServiceNow**

**1. INTRODUCTION**

**Project Overview**  
The Network Request Management system is designed to streamline the handling of employee network-related requests within an organization. Built on the ServiceNow platform, it enables structured intake, automated approvals, and seamless communication, ensuring efficiency and transparency in network operations.

**Purpose**  
The main purpose of this project is to eliminate manual processing delays and errors in network request handling. By creating a dedicated Service Catalog item with dynamic variables, custom tables, and Flow Designer automation, the system reduces dependency on physical paperwork or ad-hoc communication. It provides real-time tracking, role-based approvals, and timely email notifications.

**2. IDEATION PHASE**

**Problem Statement**  
Organizations often face inefficiencies in managing network-related requests such as new connections, device allocations, or relocation requirements. Manual handling results in delays, lack of transparency, and incomplete record-keeping.

**Empathy Map Canvas**

* **Who?** Employees, IT Staff, Network Administrators
* **Think/Feel?** Employees are uncertain about request status; IT staff struggle with high request volume.
* **See?** Long email chains, scattered information, and delayed responses.
* **Say/Do?** Employees repeatedly follow up; IT staff manually track requests.
* **Hear?** Complaints about delayed network access or unresolved issues.
* **Pain?** No central system for request visibility or approval tracking.
* **Gain?** Automated workflow, timely notifications, accurate records, and faster resolution.

**Brainstorming**  
Different approaches such as email-based request tracking and Excel sheets were considered. ServiceNow was selected due to its scalable catalog system, workflow automation, and ability to integrate with approval and notification mechanisms.

**3. REQUIREMENT ANALYSIS**

**User Journey Map**  
Employee submits Network Request → System captures details via catalog form → Approval workflow triggered → IT staff fulfill request → Notifications sent → Request closed.

**Solution Requirements**

* Catalog Item: Network Request
* Variables: connection type, device type, address, device details, attachments
* Variable Set: user details (name, email, phone, proof documents)
* UI Policies: conditional display of fields (e.g., if device = others, prompt for details)
* Custom Table: Network Database for tracking requests
* Related List: approval linkage for tracking request states
* Flow Designer Automations: request creation, approval, record updates, notifications

**Data Flow Diagram**  
User → Service Portal → Catalog Item → Workflow/Flow Designer → Approval/Task Table → Notifications → User

**Technology Stack**

* Platform: ServiceNow
* Modules: Service Catalog, Flow Designer, Tables, Notifications, Access Control
* Logic: Catalog UI Policies, Flow Logic, Approvals
* Scripts: JavaScript (Glide APIs), HTML for templates

**4. PROJECT DESIGN**

**Problem-Solution Fit**  
Manual handling of network requests causes bottlenecks and miscommunication. The ServiceNow-based automated process provides structured forms, conditional logic, and approval workflows, addressing these issues directly.

**Proposed Solution**

* Catalog Item: Network Request
* Variables: device details, relocation address, attachments
* Variable Set: requester details auto-populated from user records
* UI Policies: dynamic visibility for conditional fields
* Custom Table: Network Database
* Flow Designer: automation of record creation, email notifications, approvals, and updates
* Notifications: automatic updates for requestors and approvers

**Solution Architecture**

* **Frontend:** Service Portal with catalog forms
* **Logic Layer:** Flow Designer (triggers, actions, approvals, conditions)
* **Backend:** Custom tables and related lists for tracking
* **Notification Layer:** Automated emails for status updates
* **Access Control:** Role-based visibility and permissions

**5. PROJECT PLANNING & SCHEDULING**

Key steps undertaken:

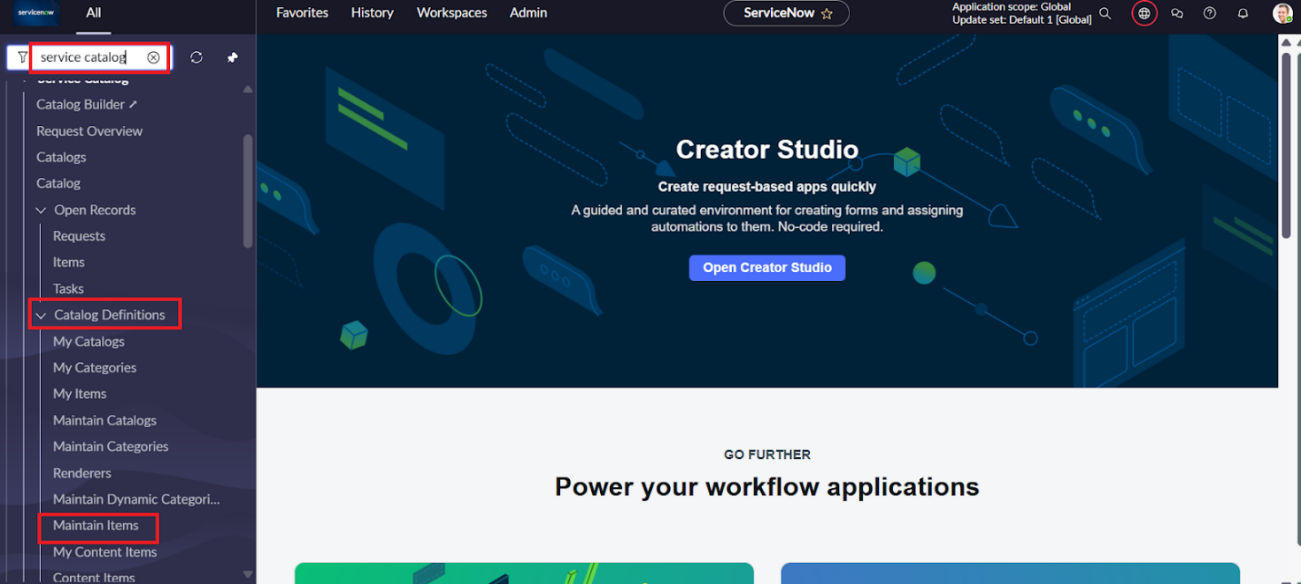
1. Catalog Item Creation – Added Network Request with categories and descriptions.
2. Variable & Variable Set Design – Captured request details and requester information.
3. UI Policy Setup – Configured conditional field visibility.
4. Table Creation – Built Network Database table with required fields.
5. Field Configuration – Added custom fields such as device type, address, status.
6. Related List Setup – Linked approvals to the request table.
7. Flow Designer Development – Configured actions like Get Variables, Create Record, Send Email, Ask for Approval, Update Record.
8. Portal Integration – Enabled employees to submit requests through the Service Portal.
9. Email Notifications – Configured HTML-based email templates.
10. Testing – Validated form submission, approvals, and notifications.

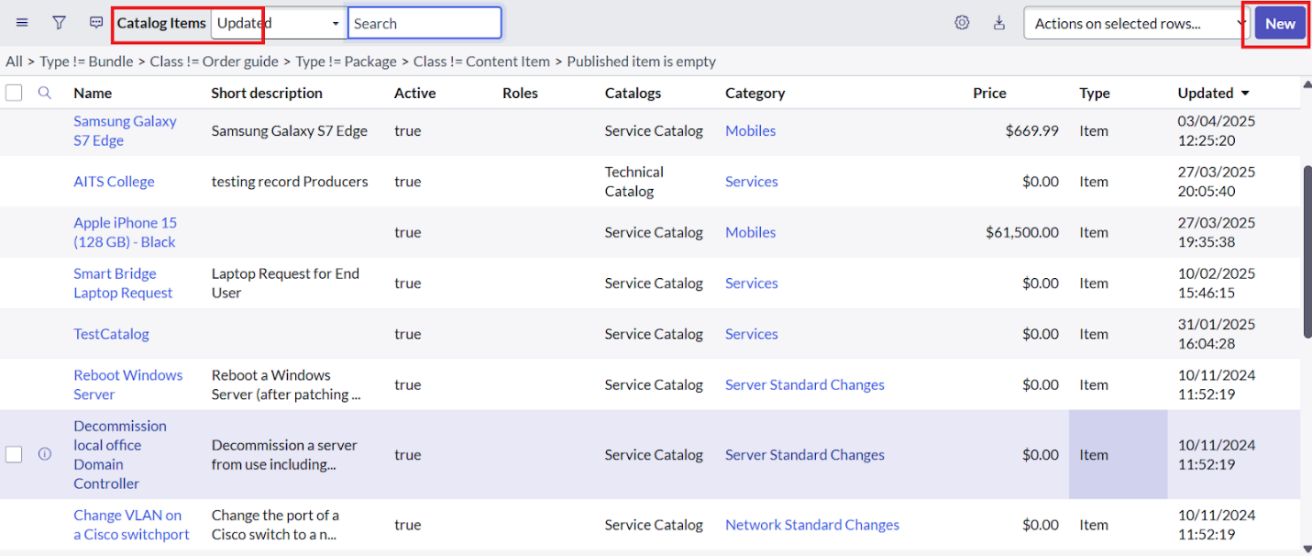
**6. IMPLEMENTATION WORKFLOW**

This phase outlines the sequence of steps followed to implement the Network Request Management system in ServiceNow. Each step was carried out in a structured manner to ensure that catalog items, workflows, and notifications function as intended.

Step 1: Catalog Item Creation

* Navigate to Application Navigator → Service Catalog → Maintain Items.
* Click on New and create a catalog item named Network Request.
* Select Catalog → Service Catalog and set Category → Network.
* Provide a meaningful short description such as Network Request Management.
* Save the catalog item.



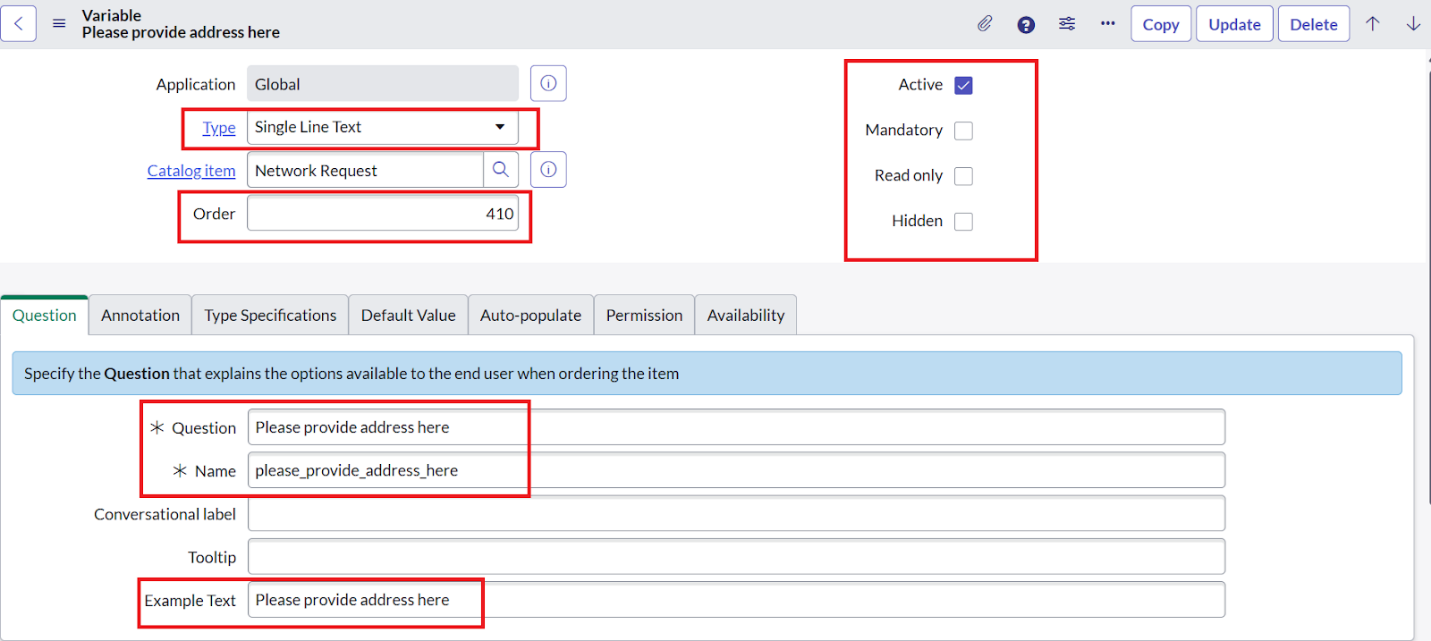




Step 2: Variable and Variable Set Configuration

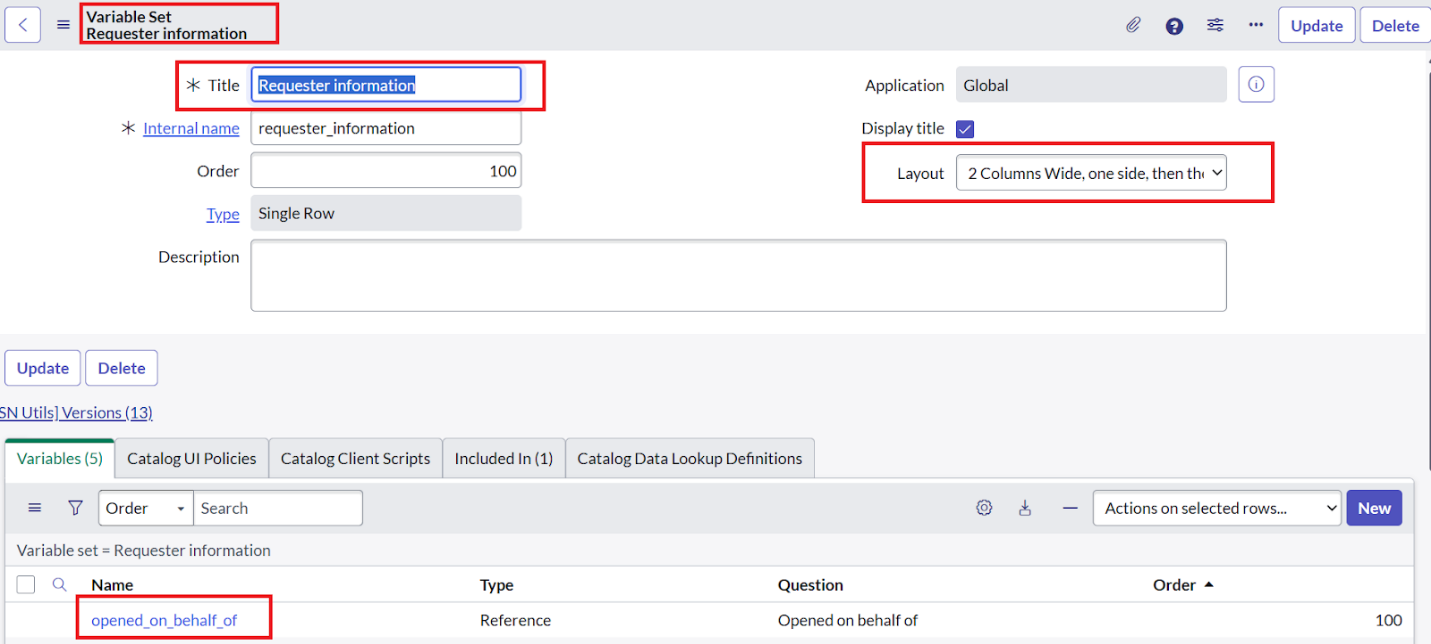
* Open the Network Request catalog item and scroll down to the Variables related list.
* Create variables such as:
  + Is this a new connection or relocation? (Choice)
  + If relocation, provide new address (String)
  + Type of devices (Choice: Laptop, Mobile, Others)
  + Device details (String)
  + Proof of document (Attachment)
* Configure variable properties: order, mandatory, tooltip, read-only, and auto-populate.





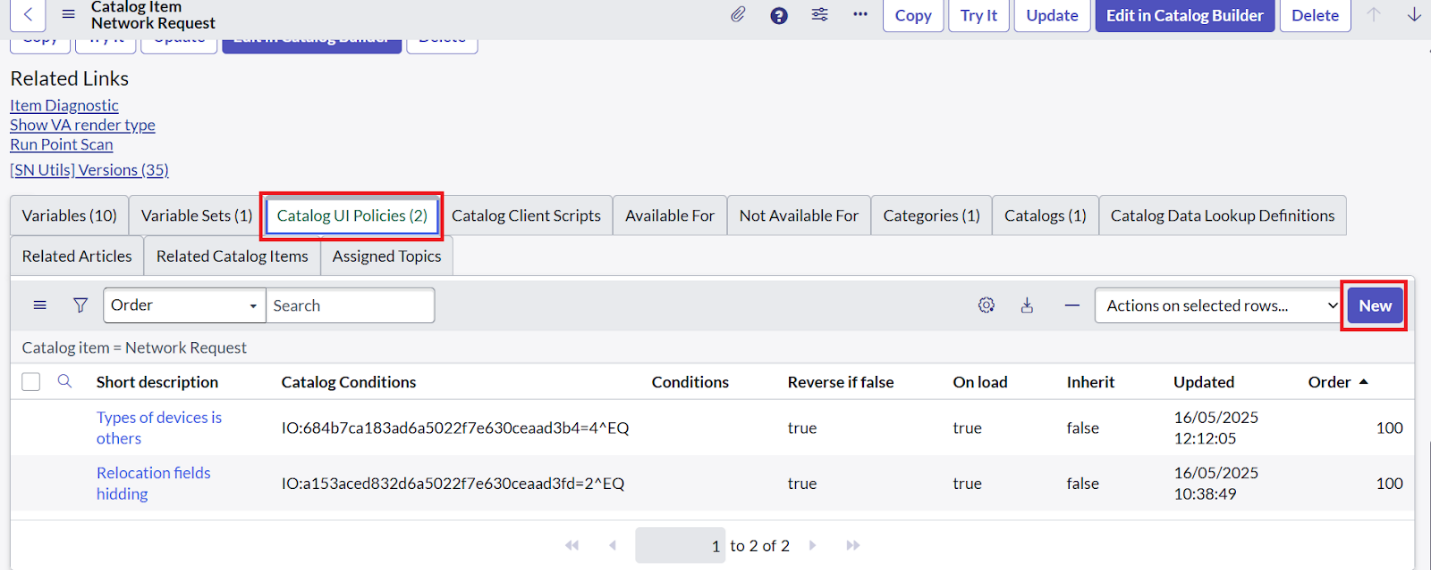
* Create a Variable Set for user details:
  + Opened on behalf of (Reference to User table)
  + Email ID, Phone Number, User Name auto-populated using dot-walking.
  + Proof of Document for supporting files.
* Apply the variable set to the catalog item.

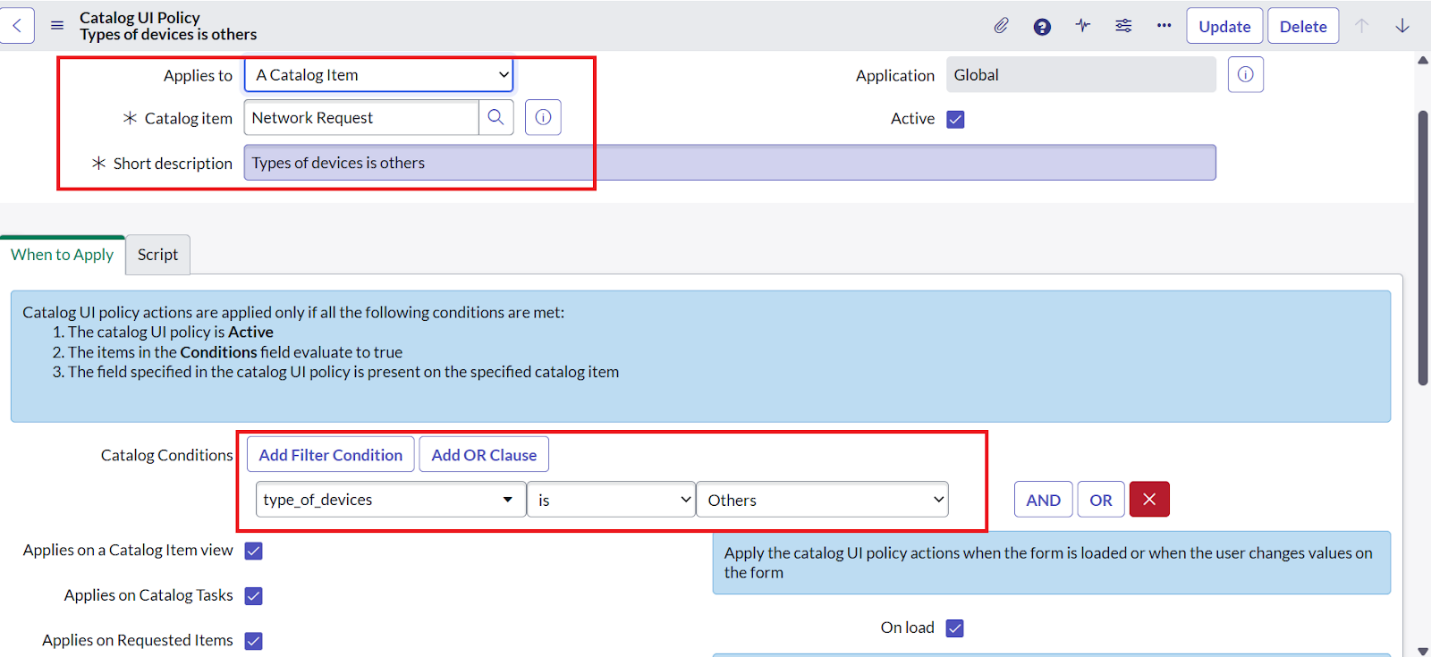


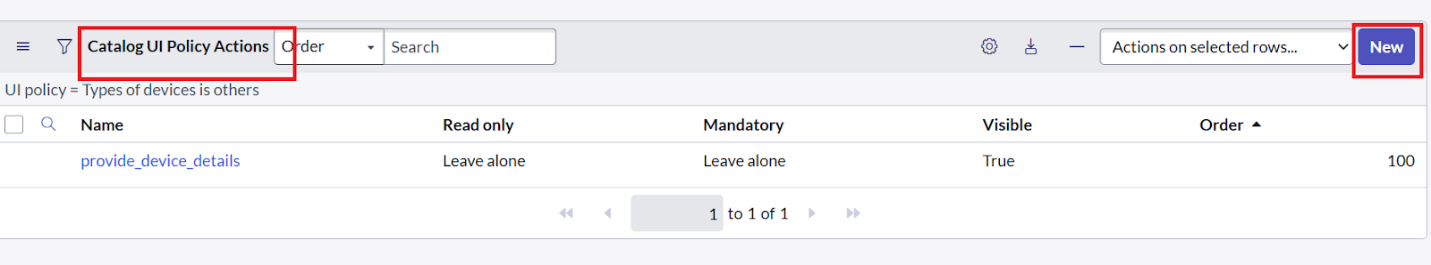


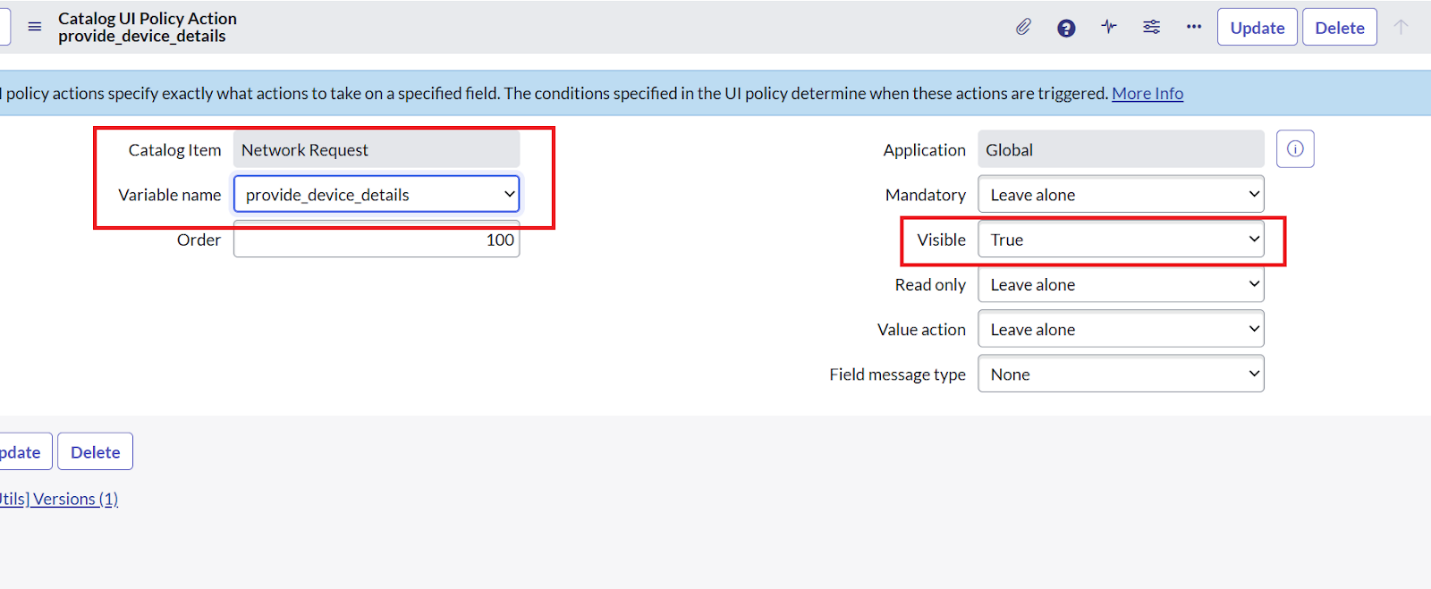
Step 3: Catalog UI Policy Configuration

* Create a UI Policy to display fields dynamically.
* Example: If Device Type = Others, then show the field Please specify device details.
* Define UI policy actions to set field visibility, read-only, or mandatory status based on conditions.



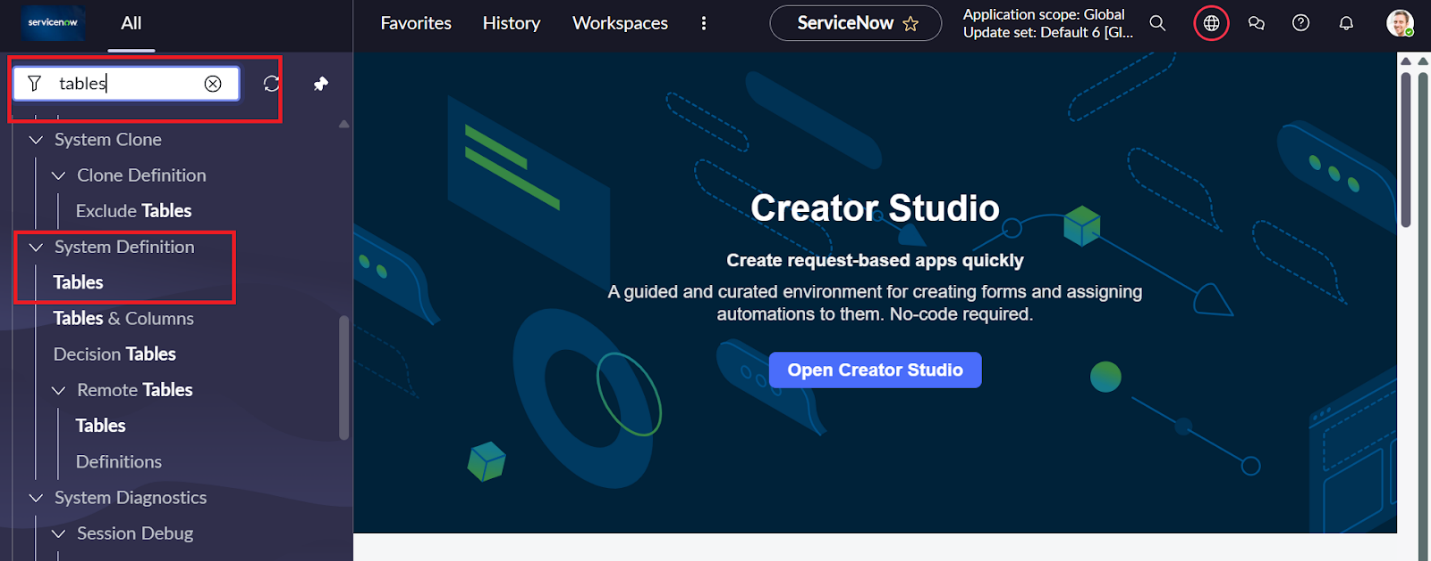


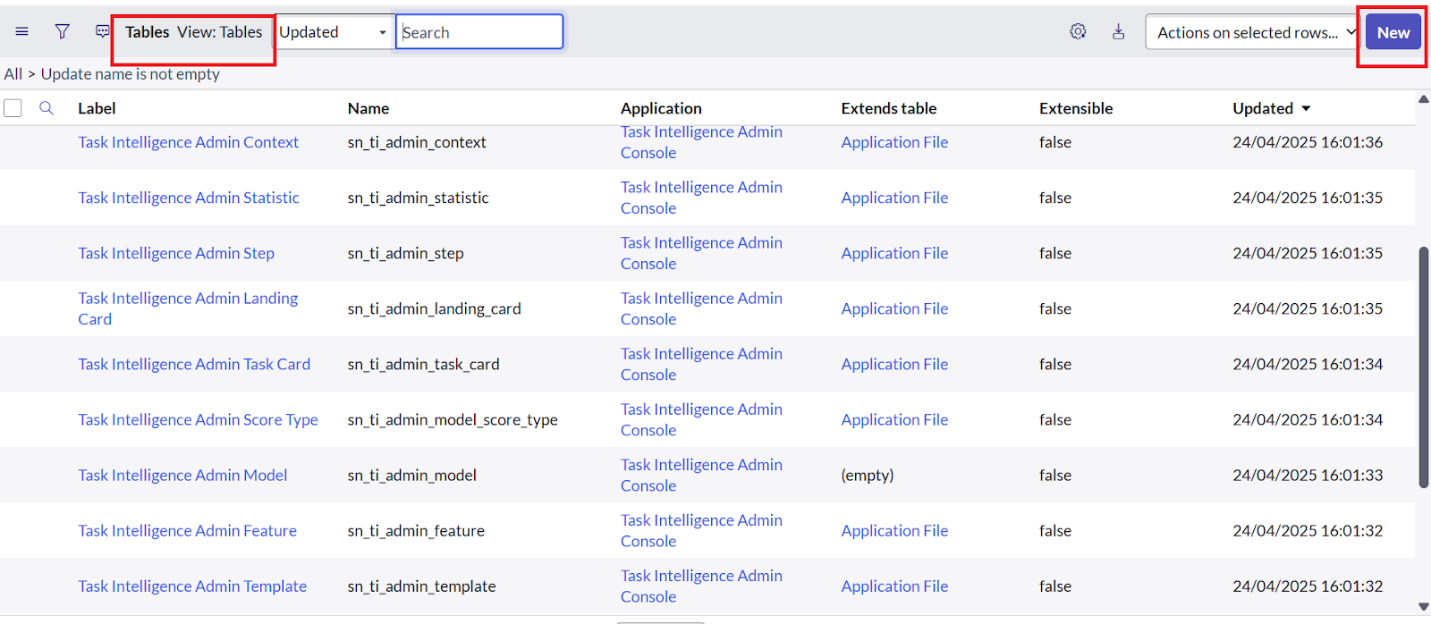


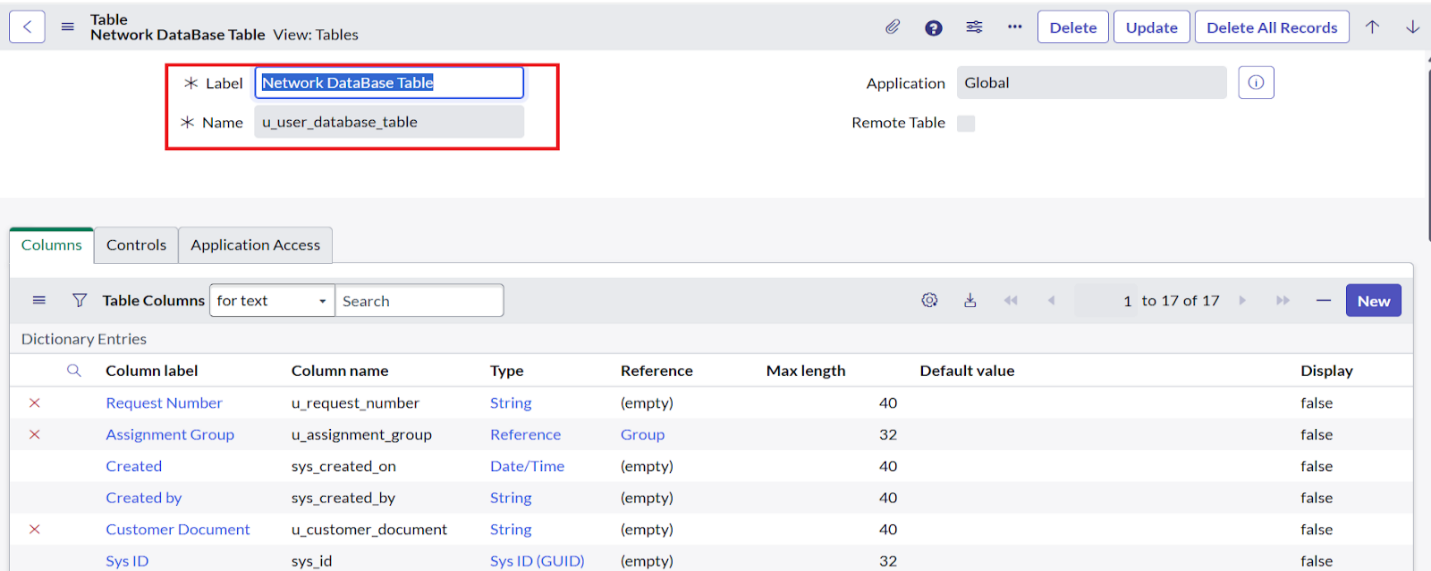


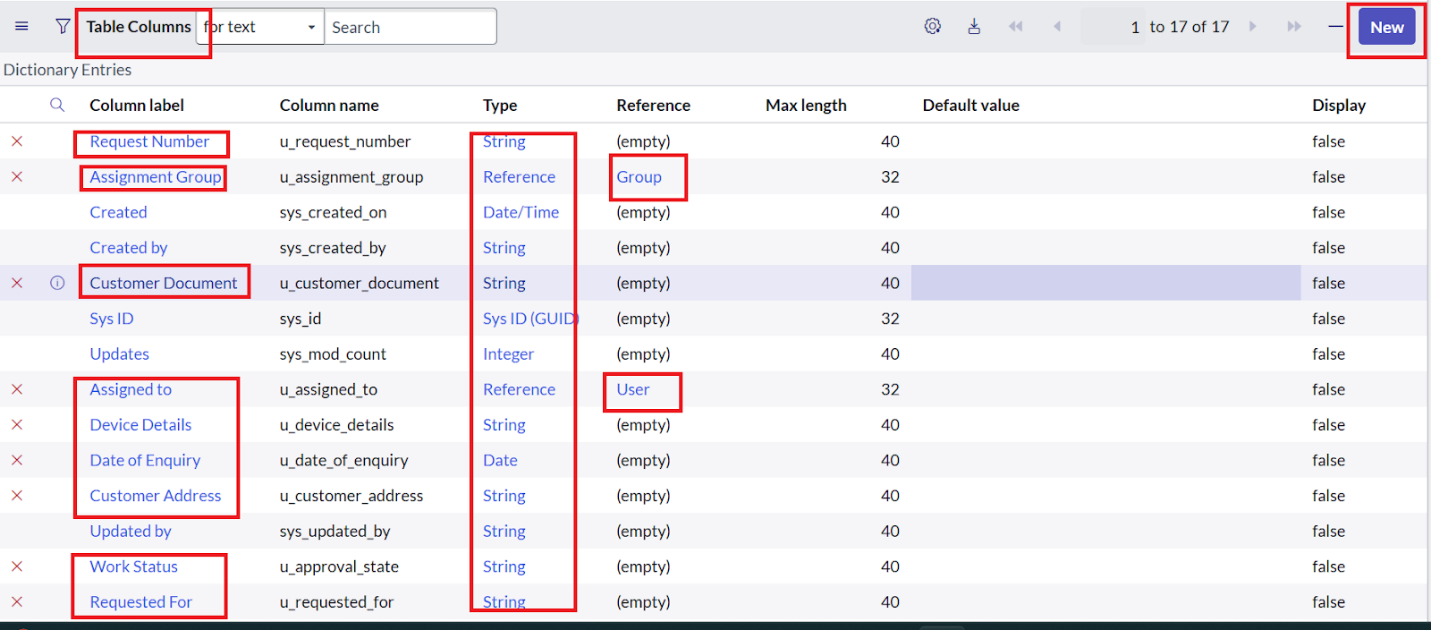
Step 4: Creation of Custom Table

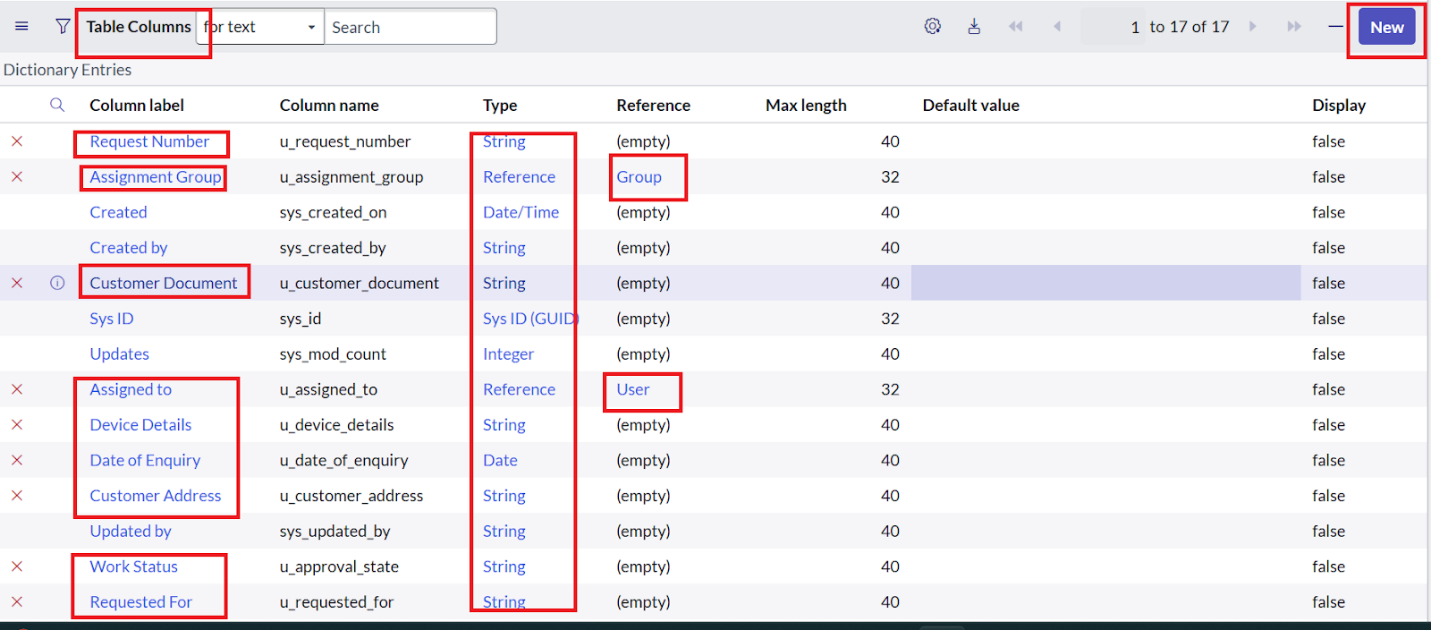
* Navigate to System Definition → Tables → New.
* Create a custom table named Network Database for storing requests.
* Add fields such as Request Number, Requestor Name, Device Type, Address, Status, and Approver Comments.
* Ensure proper column labels, field types (string, choice, reference), and mandatory constraints are set.





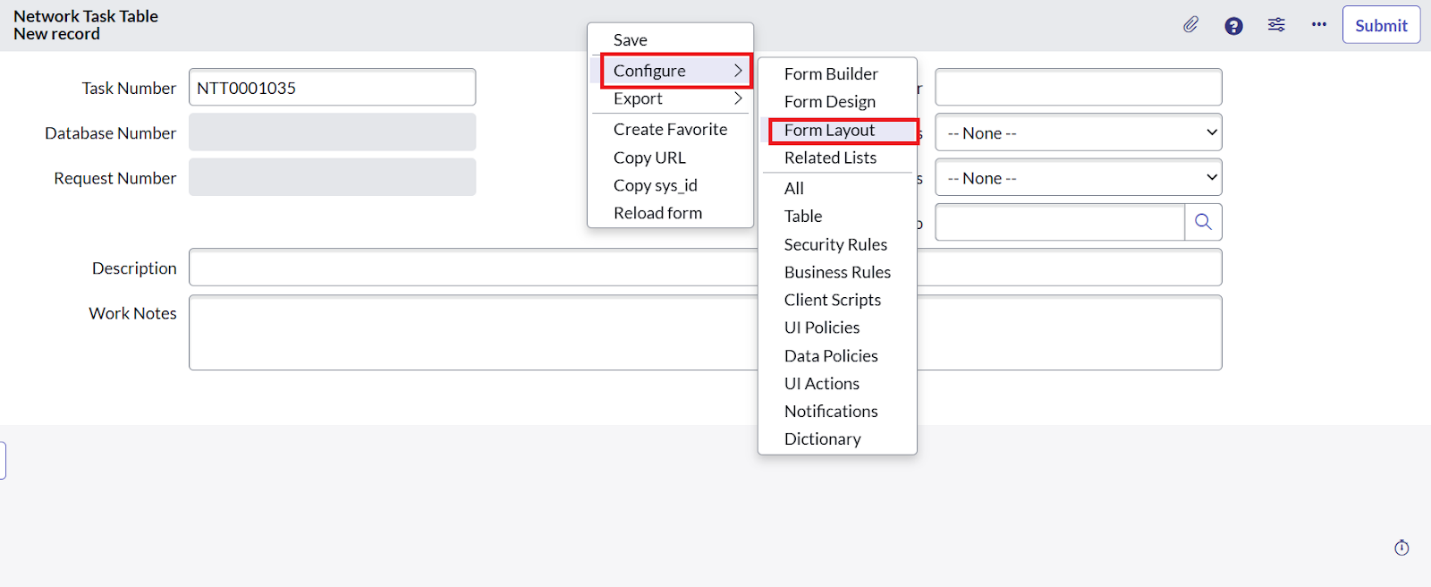


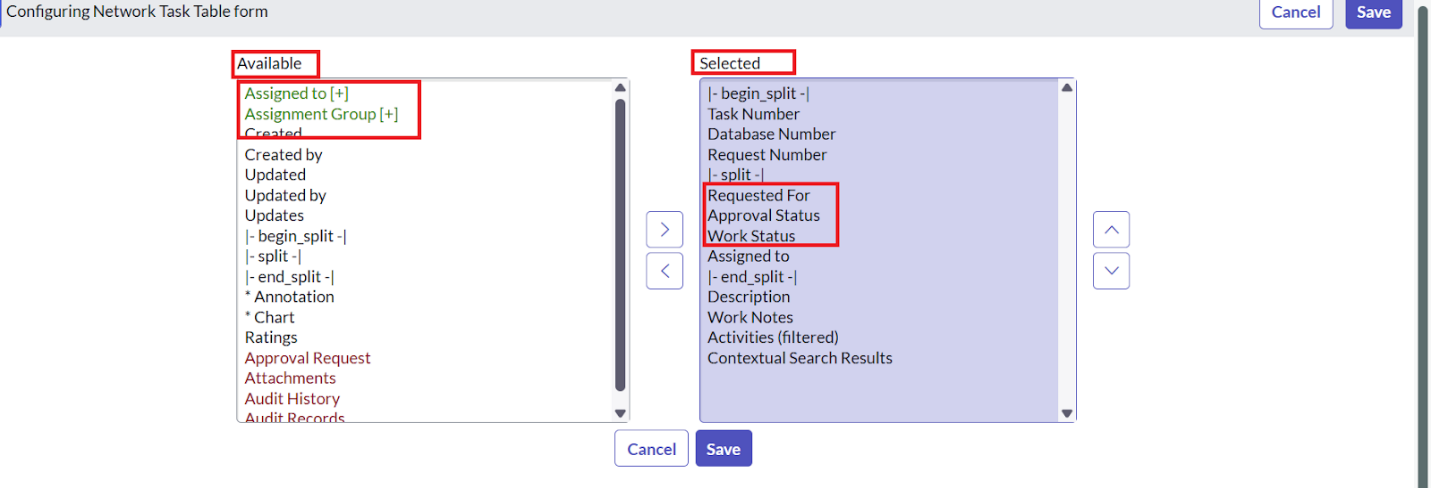




Step 5: Field Configuration

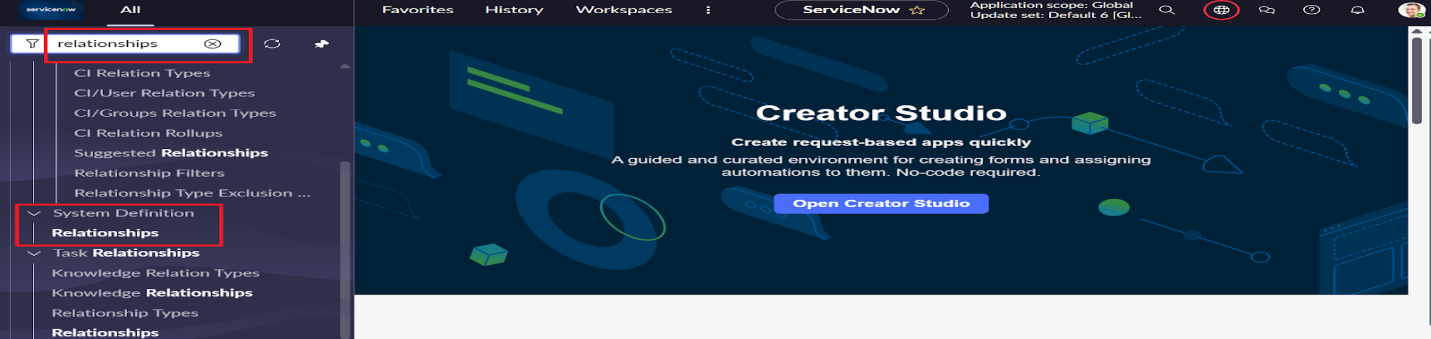
* Add fields to the Network Database table via the Columns tab.
* Example fields: Customer Name, Device Type, Relocation Address, Request Status.
* Configure field properties (max length, default values, mandatory flags).
* Add new fields to forms using Form Designer.

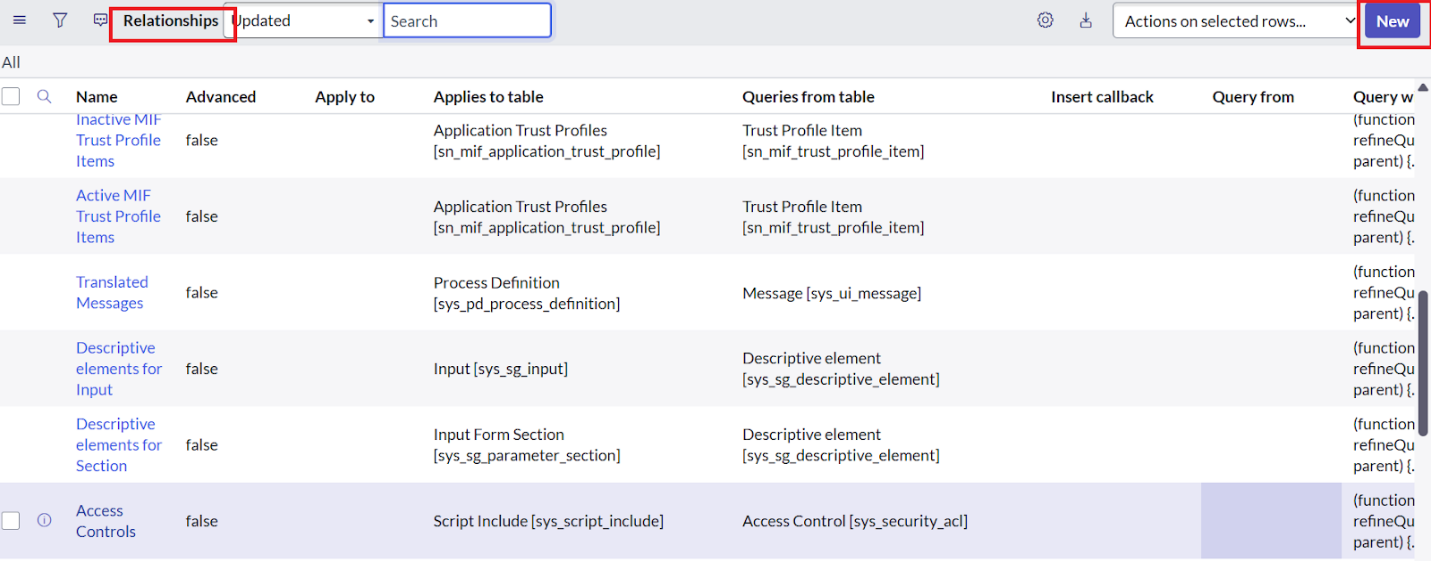
****

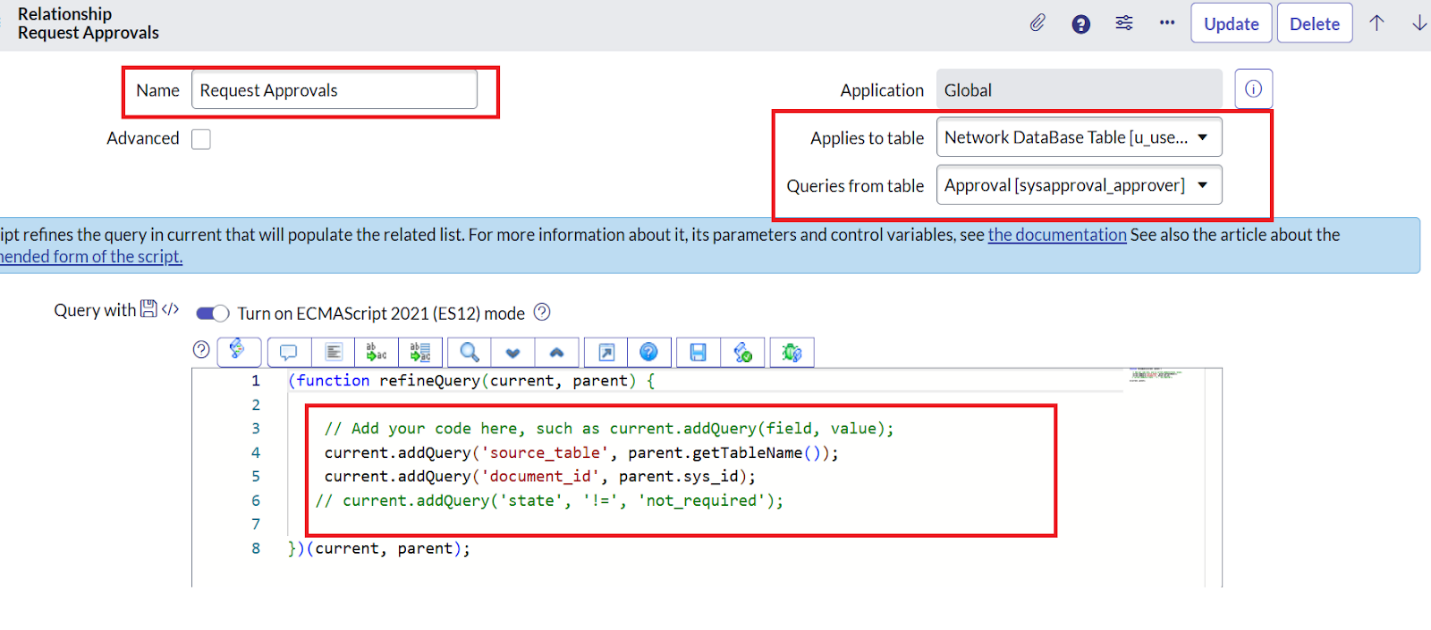


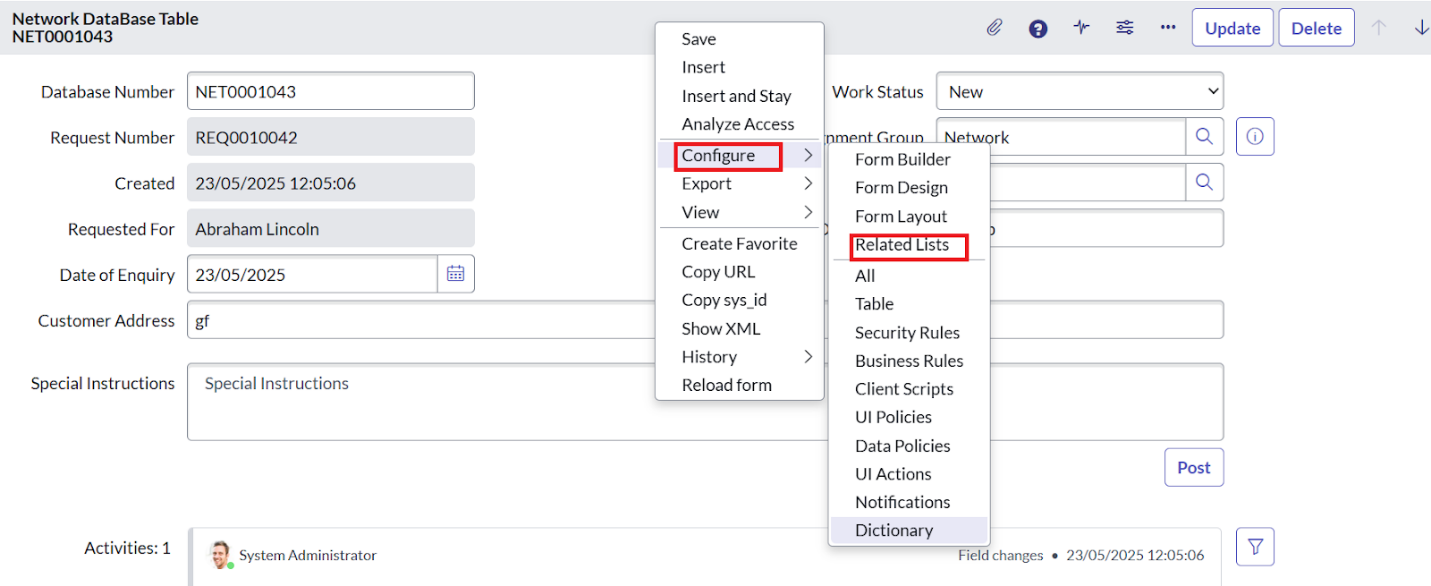
Step 6: Related List Creation

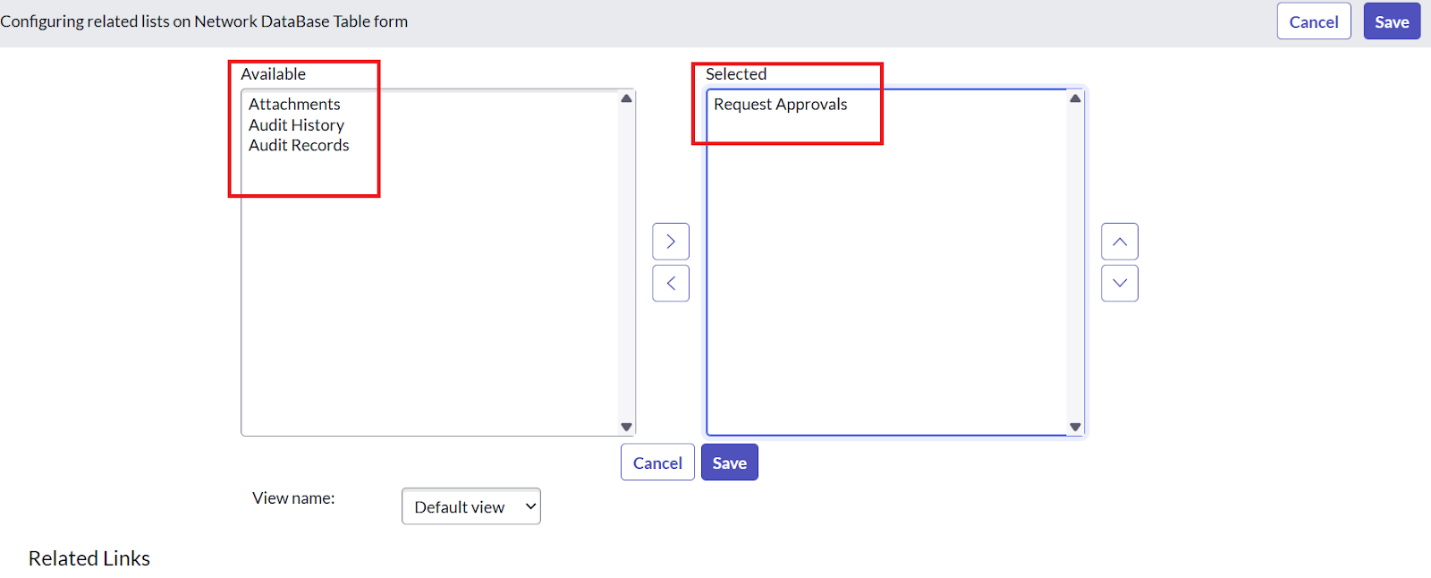
* Navigate to System Definition → Relationships.
* Create a relationship named Approval Request linking the Network Database table with Sysapproval table.
* Add the related list widget in Form Designer for easy tracking of approvals linked to each request.

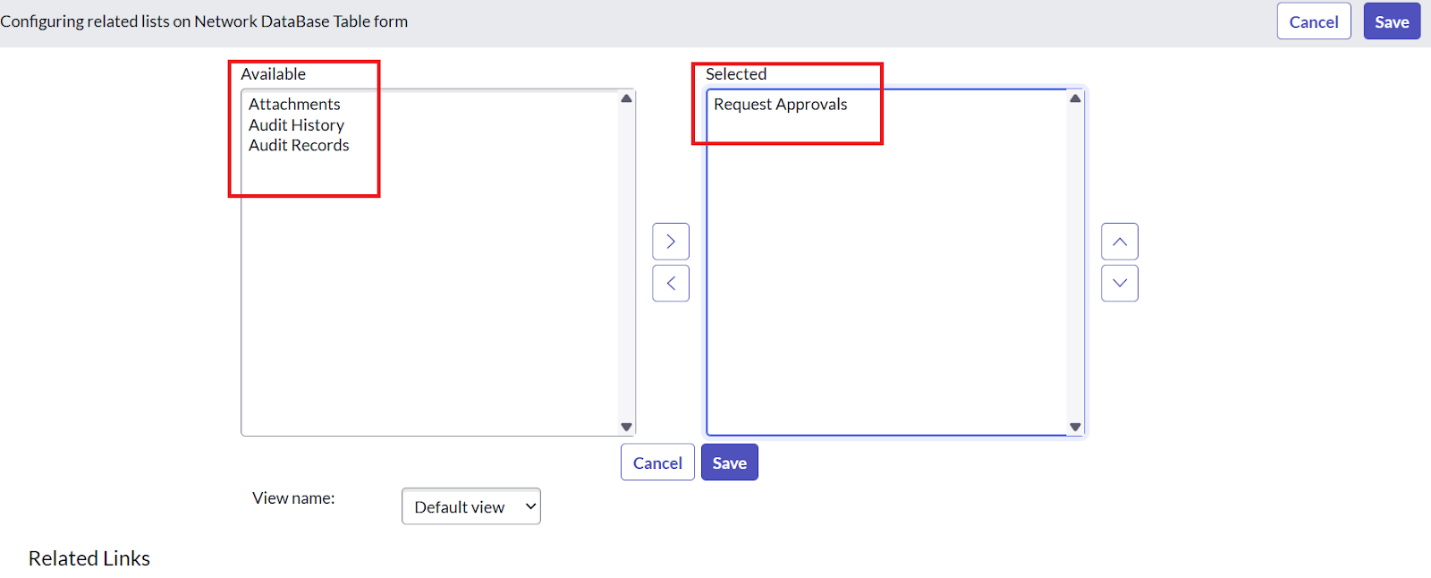


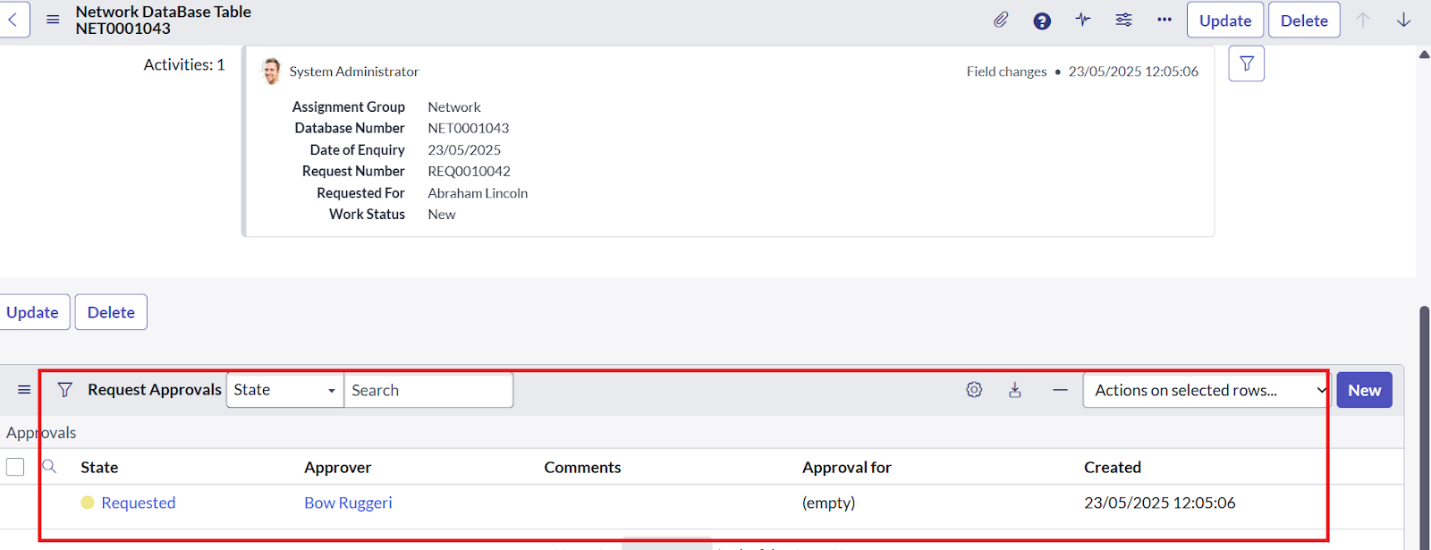






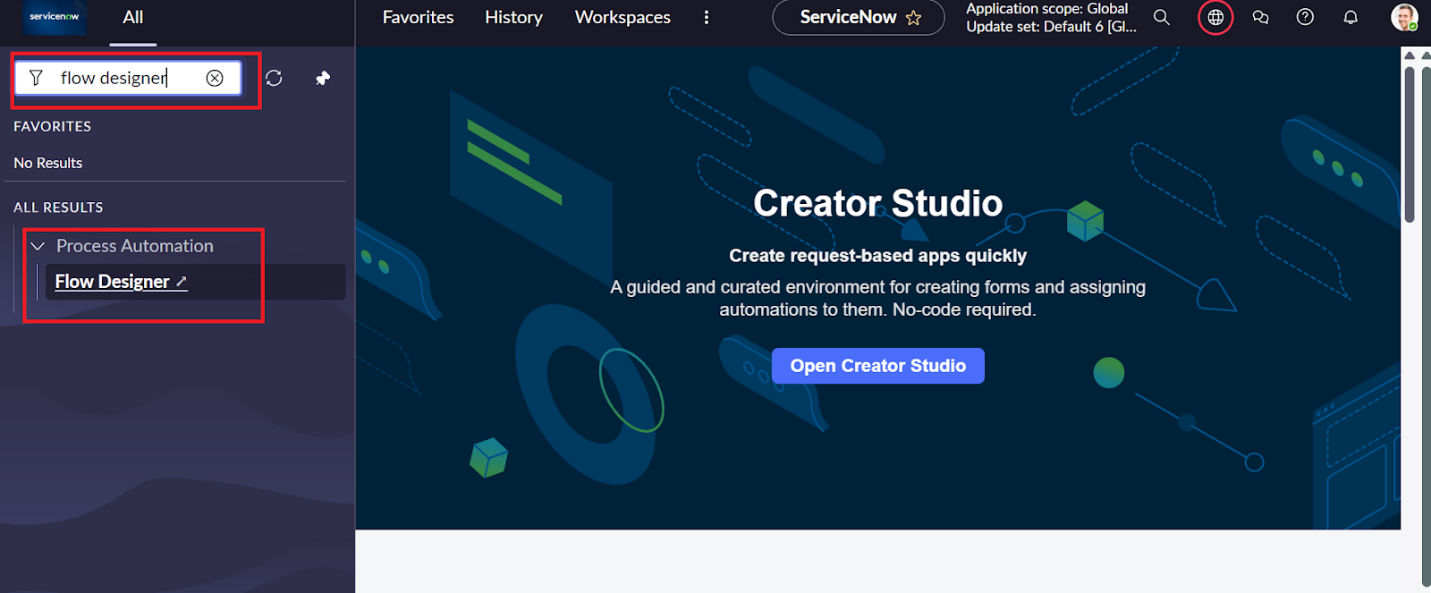


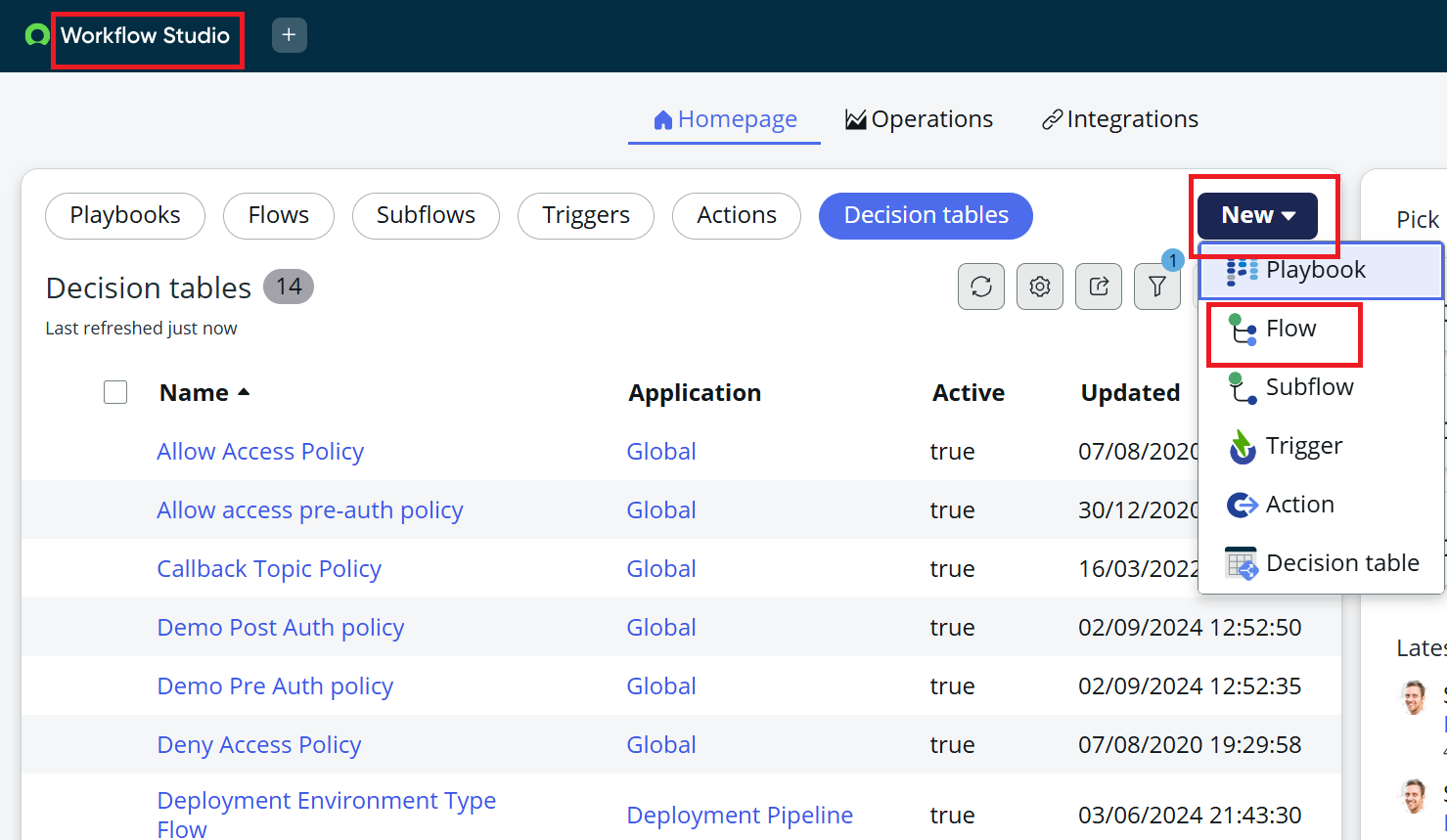


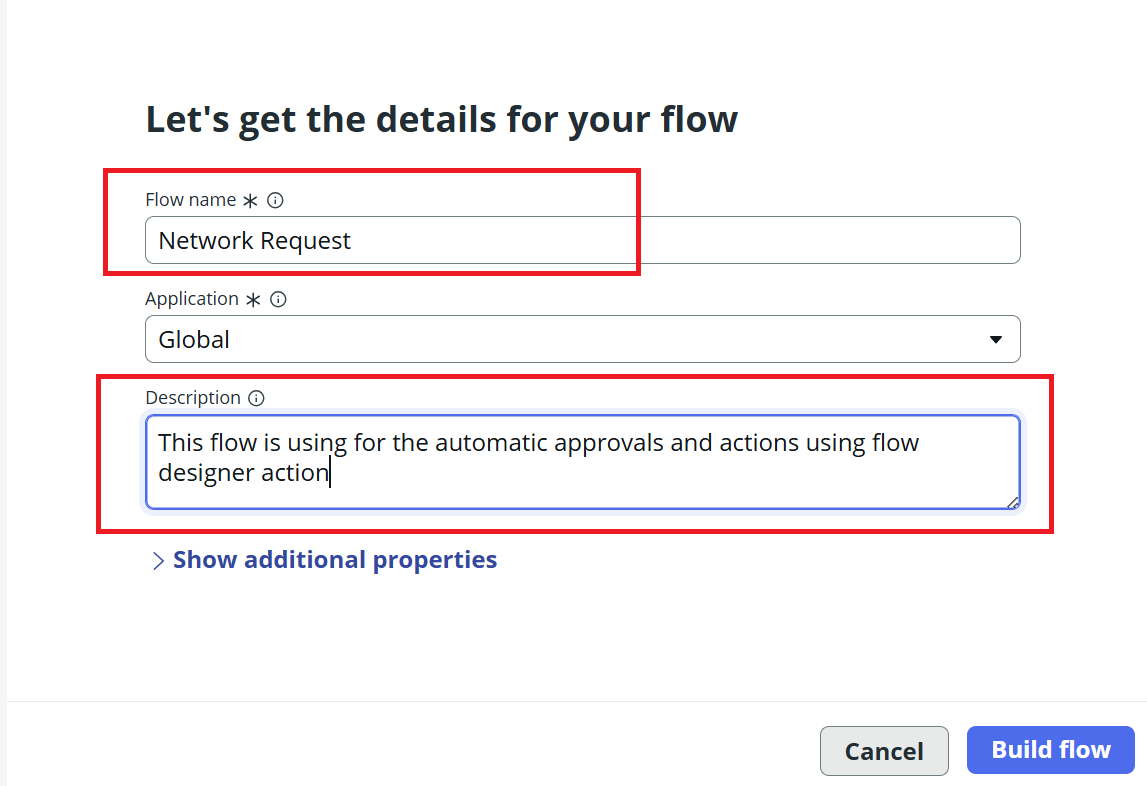


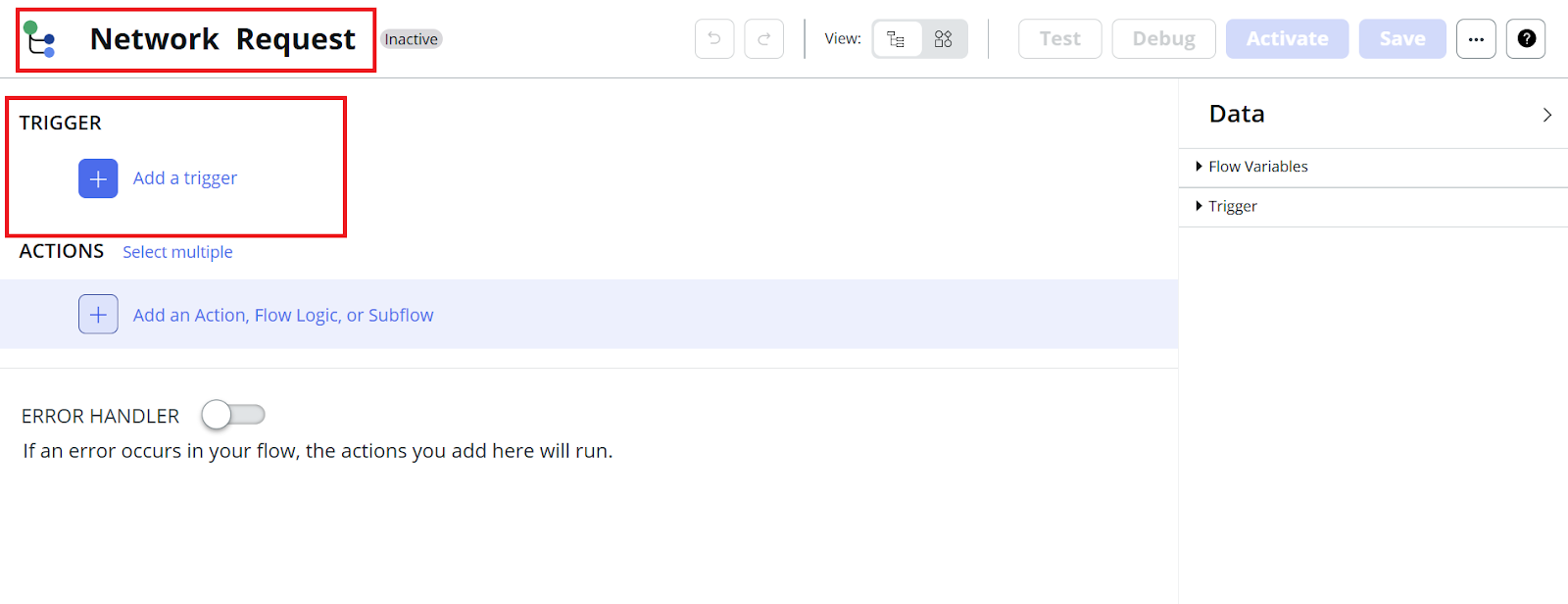
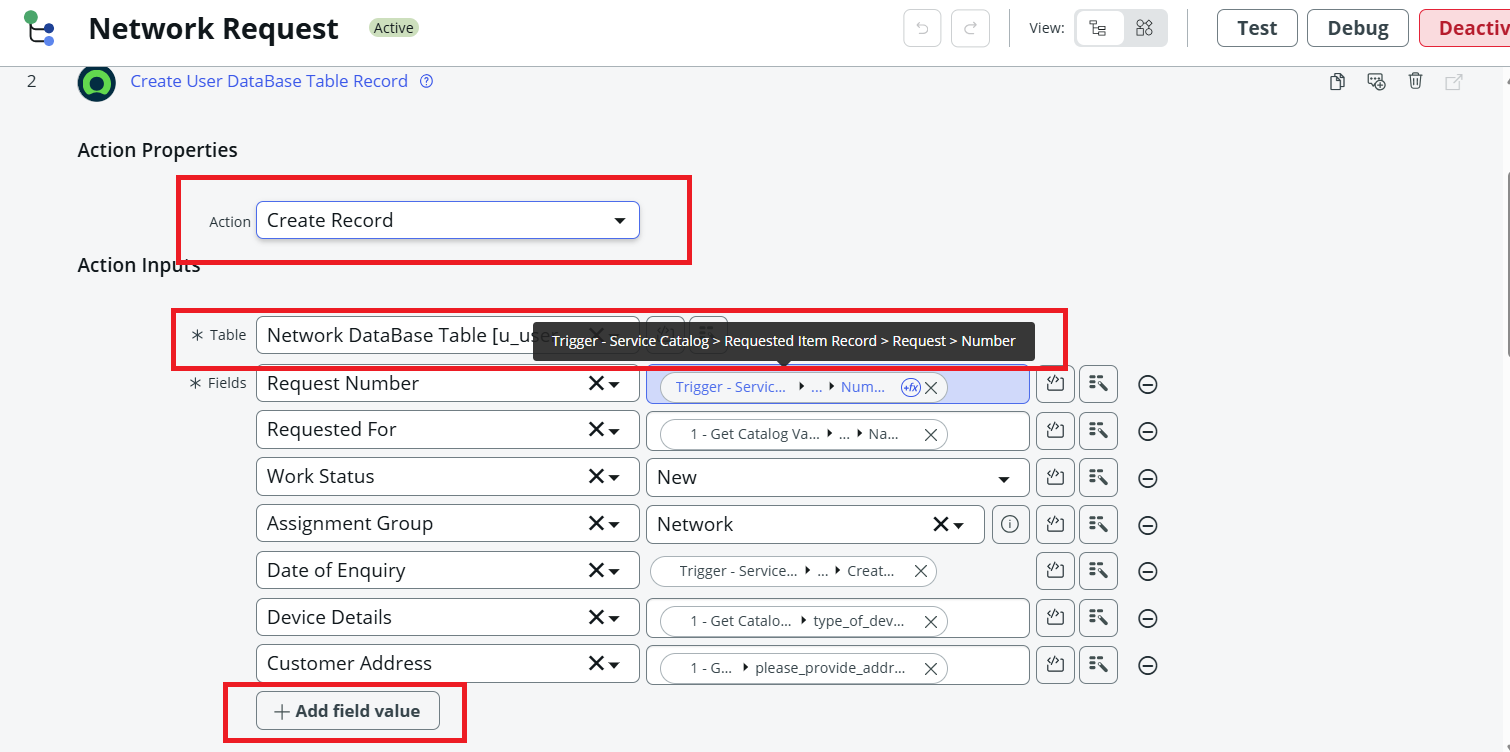
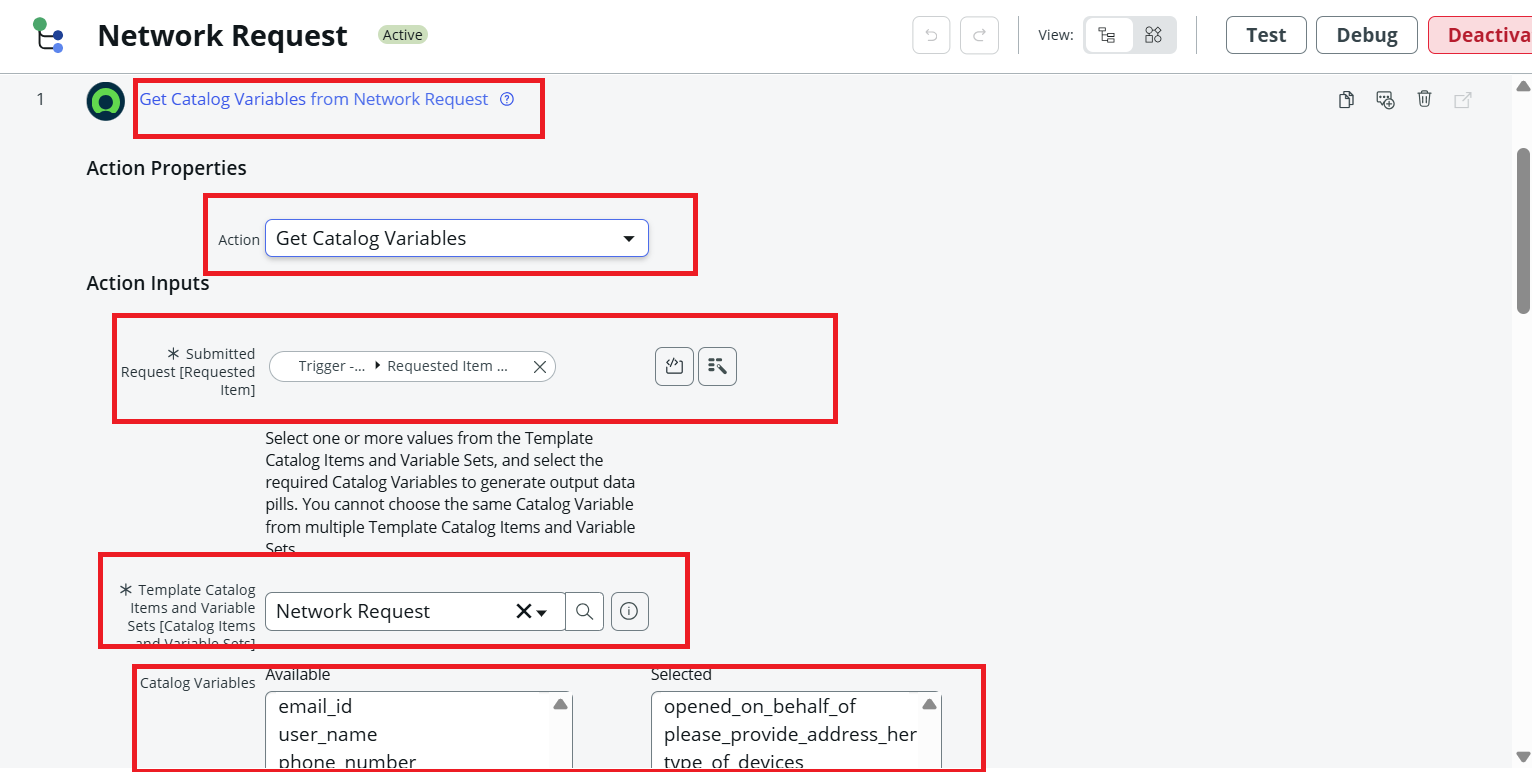
Step 7: Flow Designer Automation  
Using ServiceNow Flow Designer, the workflow was implemented as follows:

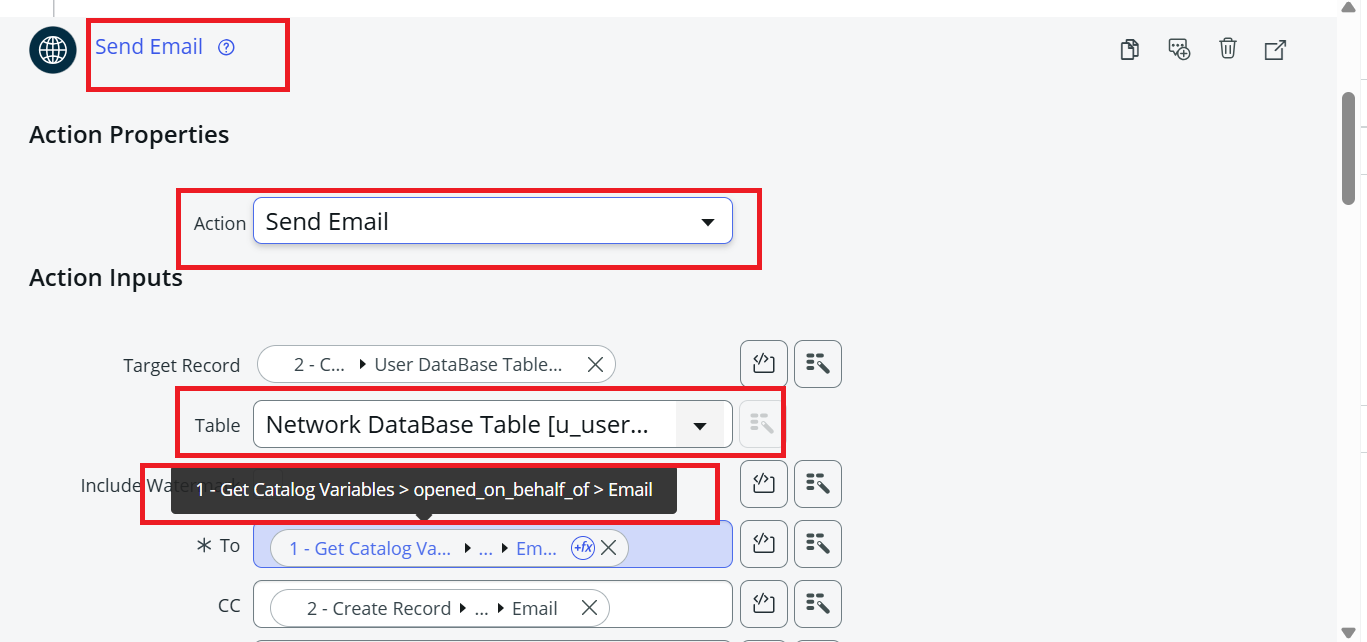
1. Trigger
   * Defined as Catalog Item Requested for Network Request.
   * The flow is triggered whenever a request record is created.
2. Actions
   * Get Catalog Variables: Retrieve request details (device type, address, attachments, user details).
   * Create Record: Automatically create a record in the Network Database table.
   * Send Email: Notify the requester with a confirmation email. Include request number and details.
   * Ask for Approval: Route the request to approvers (e.g., Network Admin group). Options include Approve, Reject, Rework.
   * Flow Logic: Add If conditions to check approval state. If Approved, update the request status to In Progress. If Rejected, update status to Rejected and notify the requester.
   * Update Record: Modify the Network Database record with the final outcome and comments.
3. Flow Testing
   * Use Flow Execution Logs to verify each step.
   * Validate triggers, record creation, and notifications.

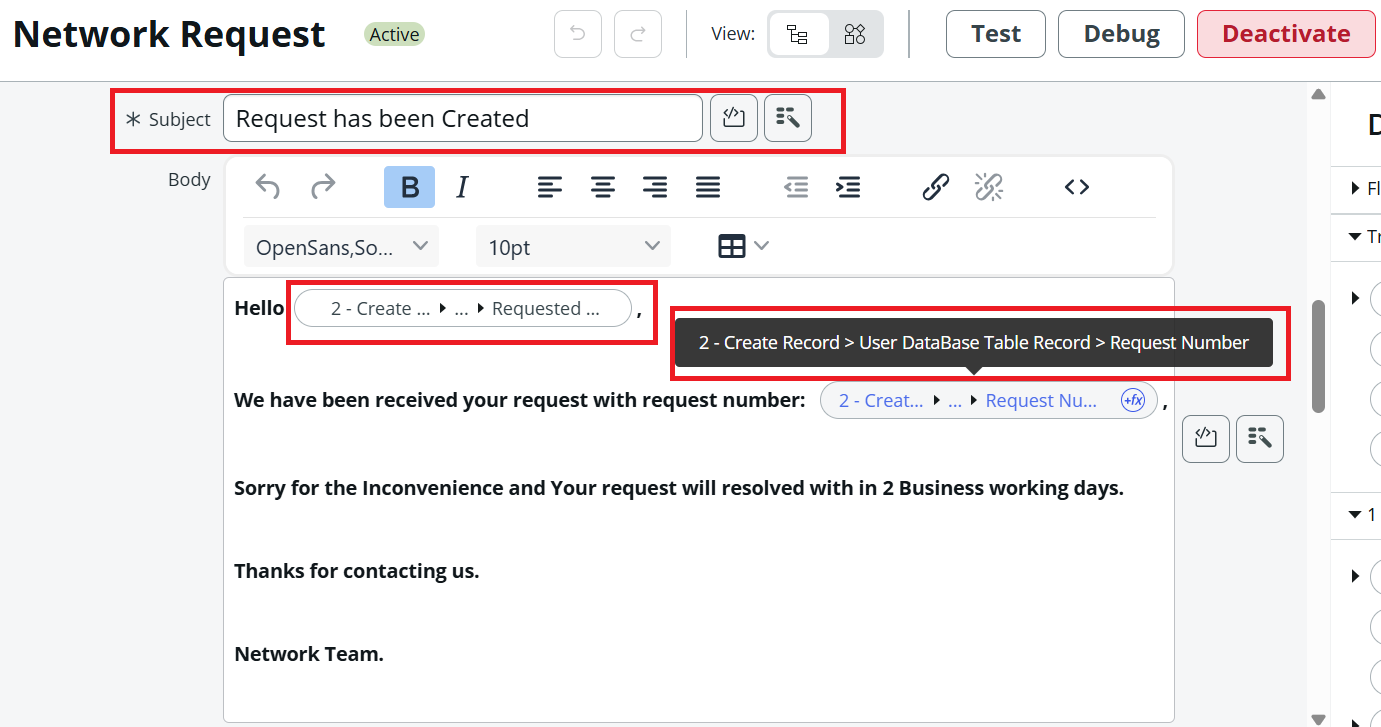
****

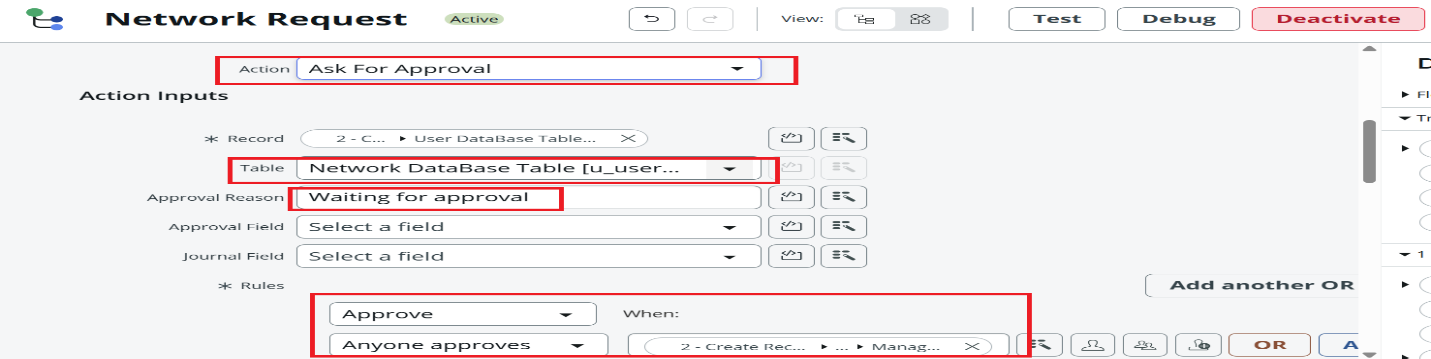




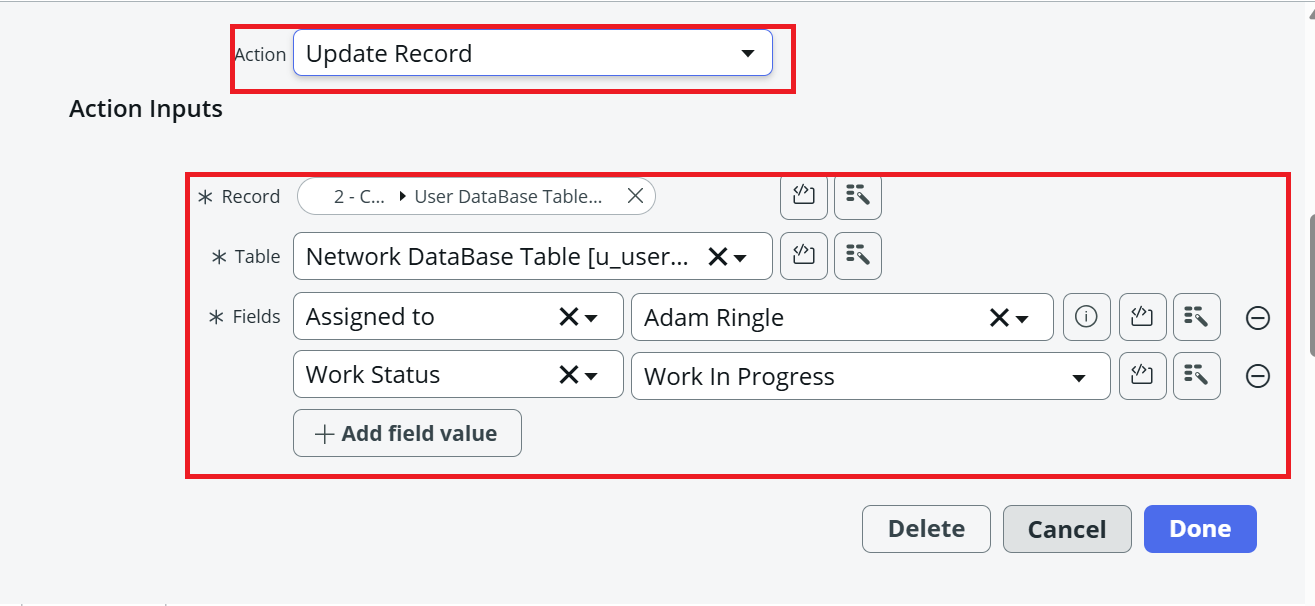
****

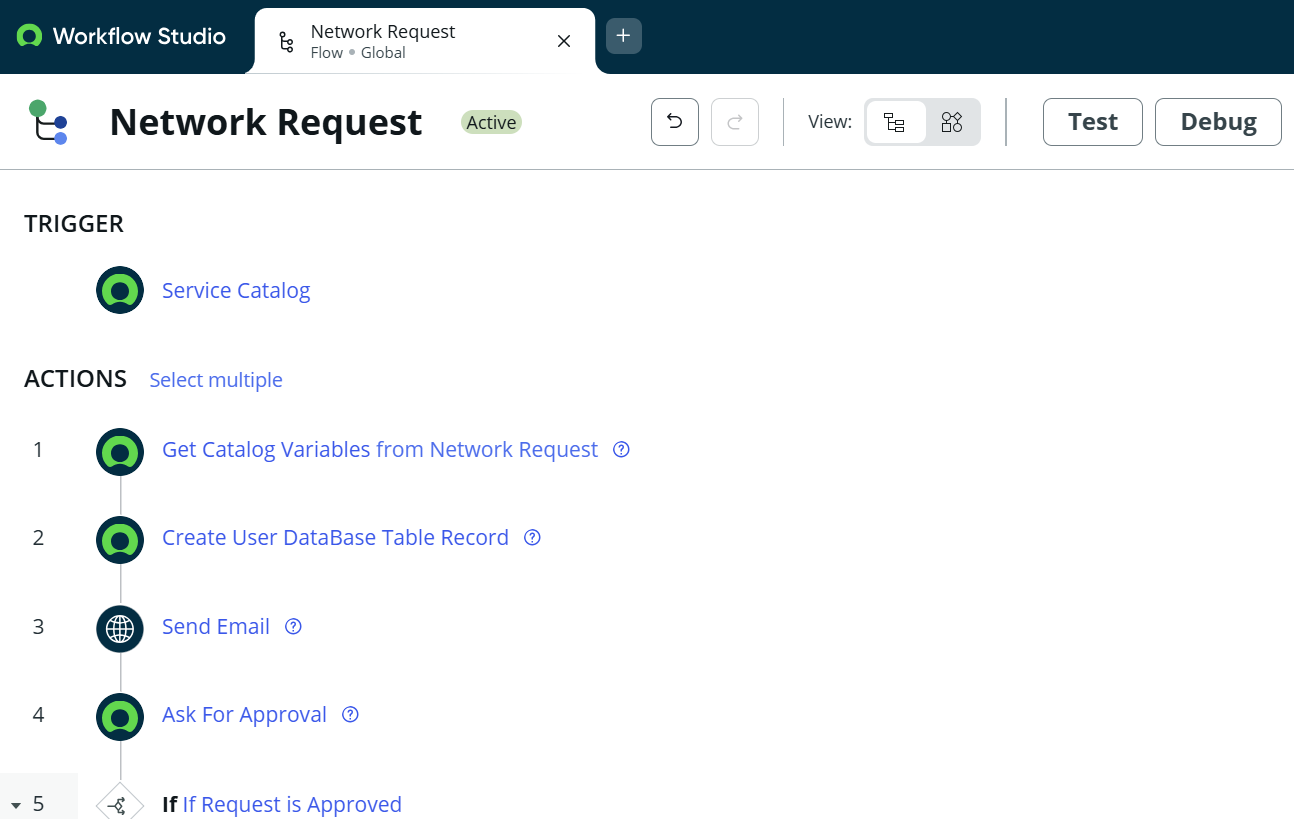




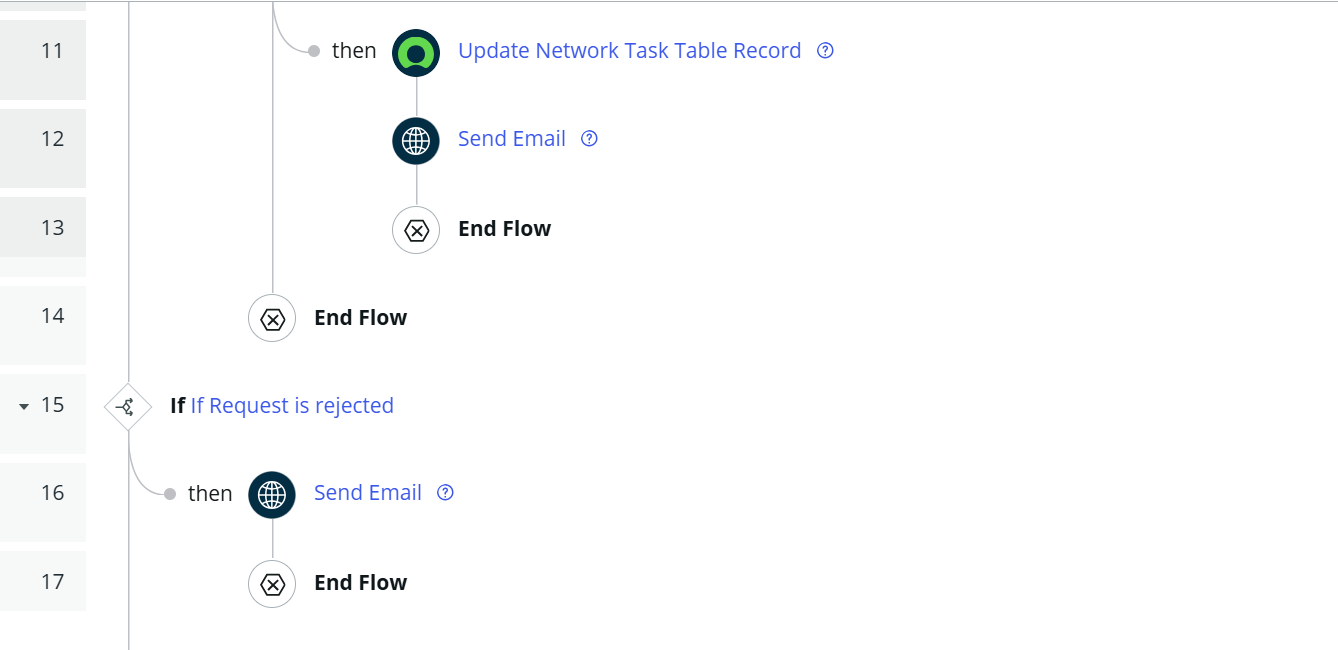






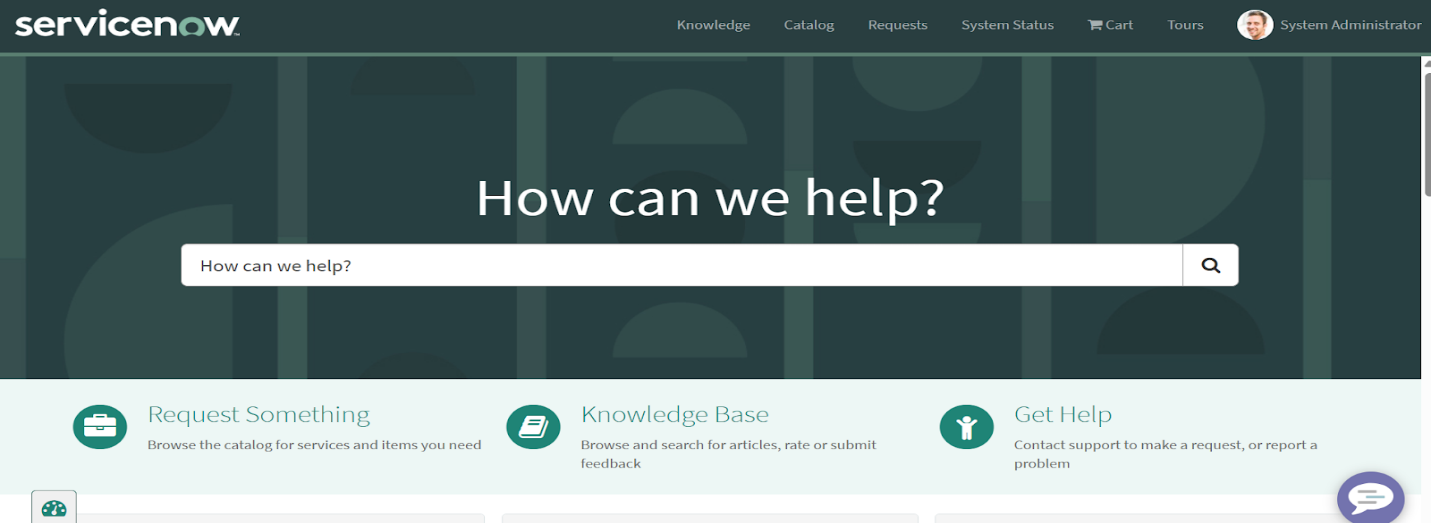
****

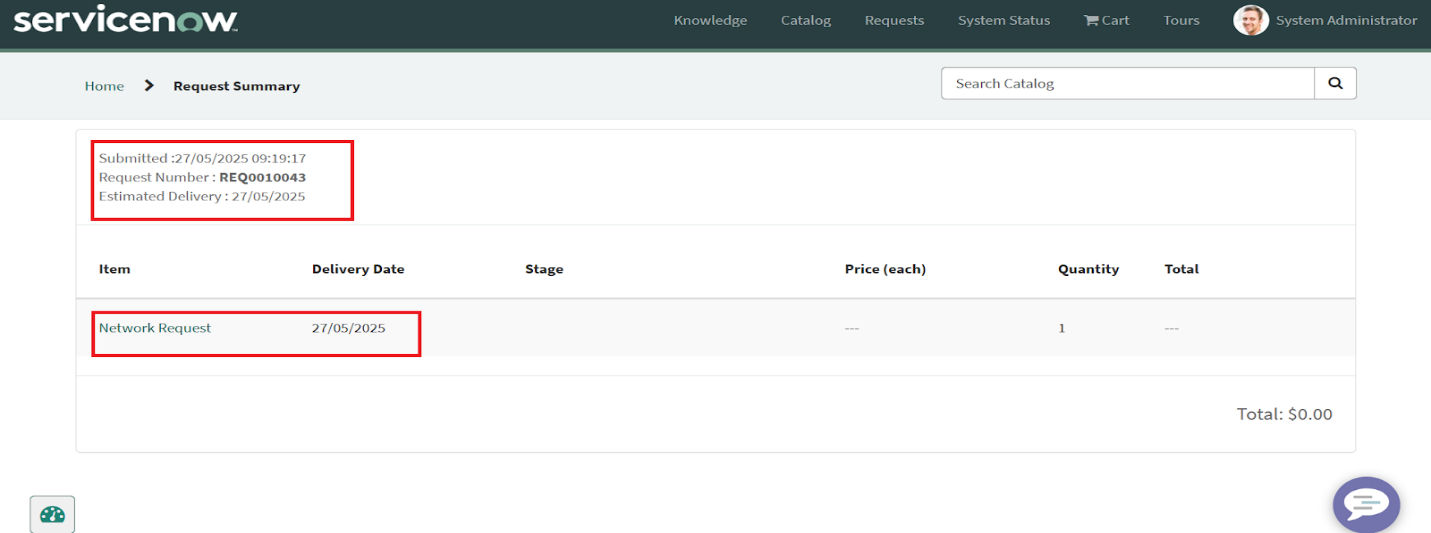
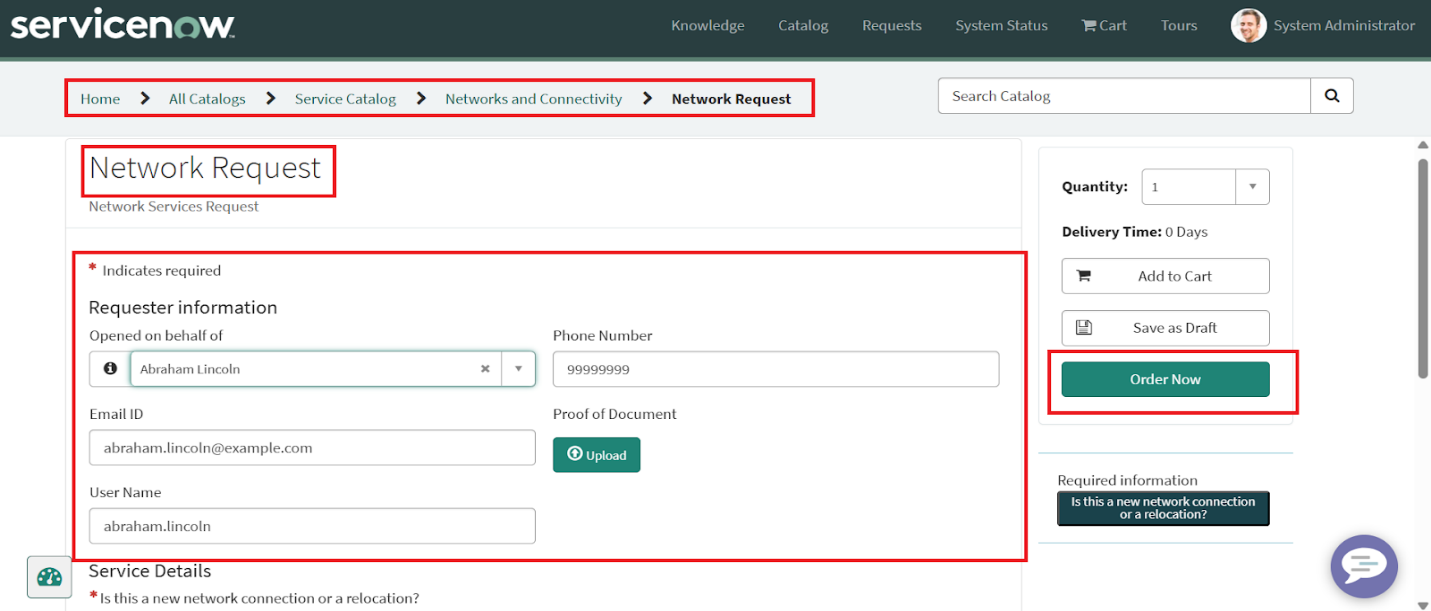
****

****

Step 8: Testing in Service Portal (End User View)

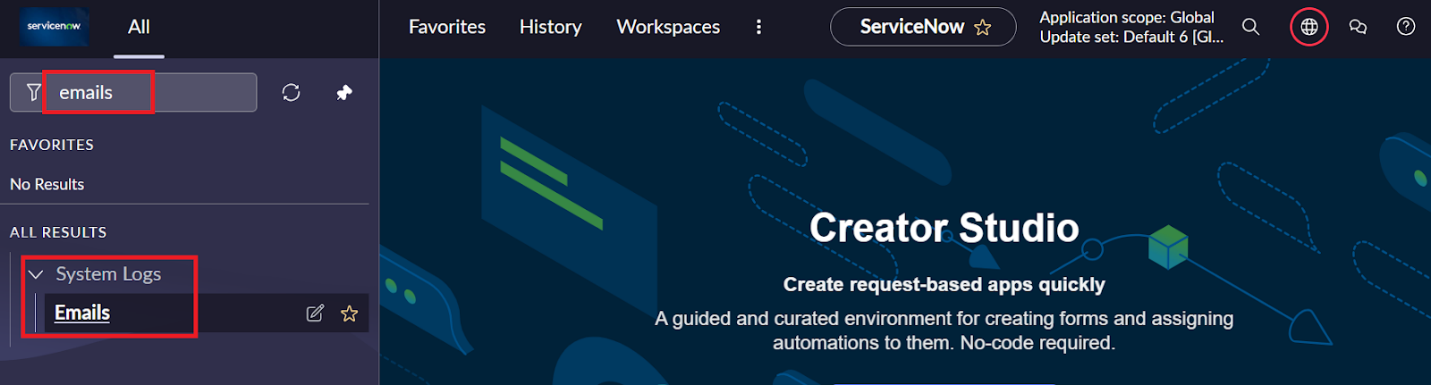
* Access the Service Portal (instance domain + /sp).
* Search for Network Requests.
* Submit a new request with required details and proof documents.
* Verify that a request number is generated and emails are triggered.

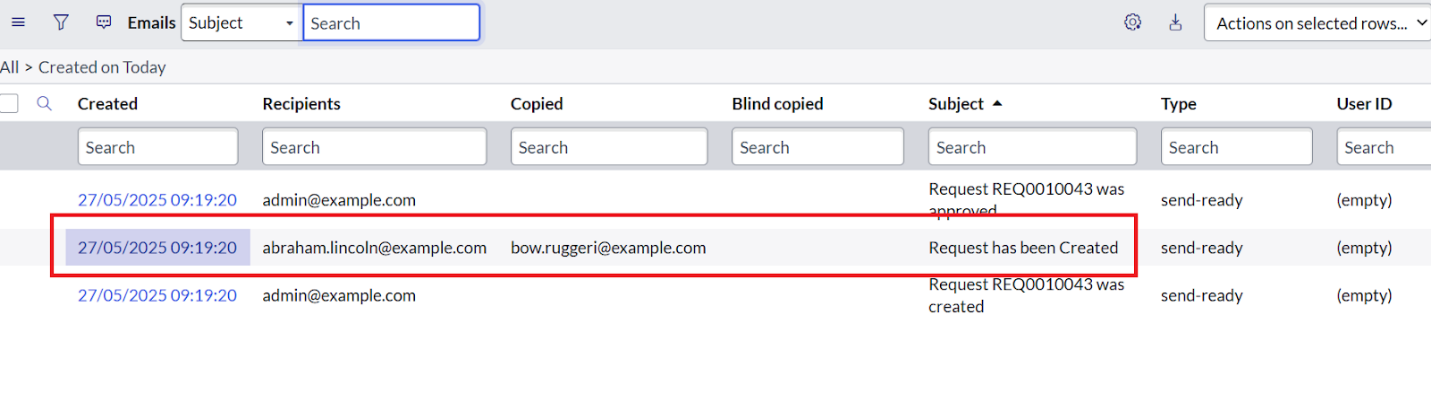
****

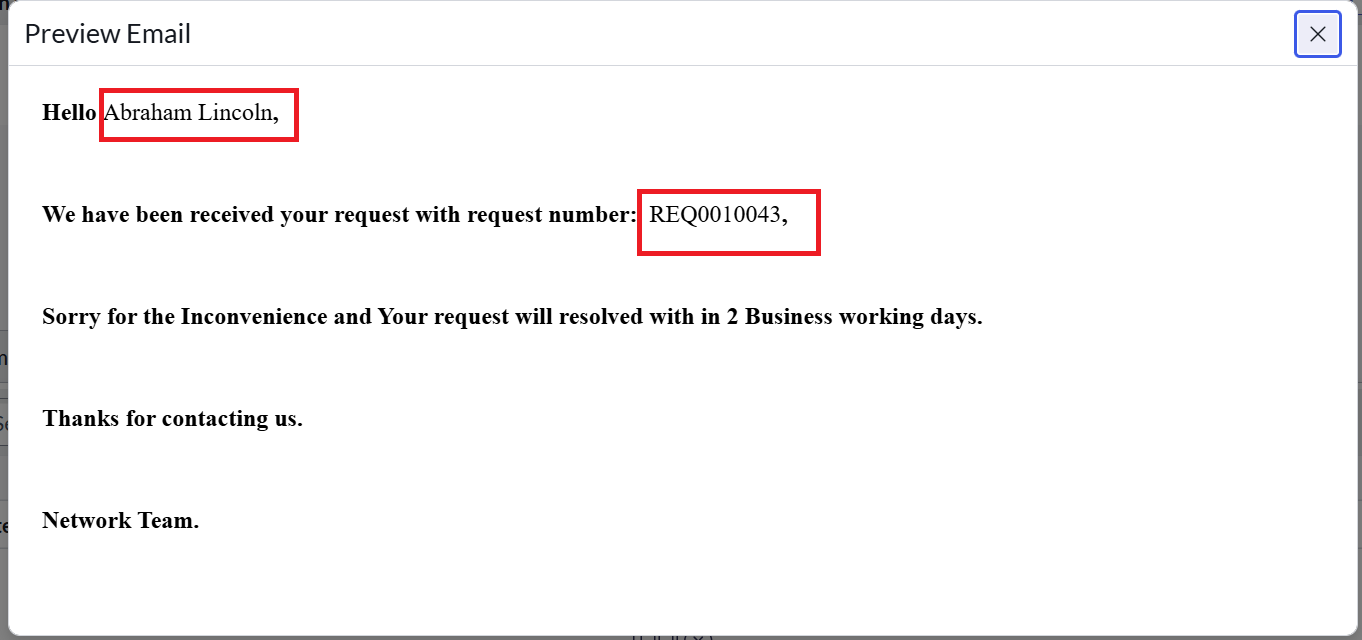
****

Step 9: Testing Email Notifications

* Navigate to System Logs → Emails.
* Apply filters by Created Date or Subject.
* Validate that the requester, approver, and IT staff received the correct email notifications.

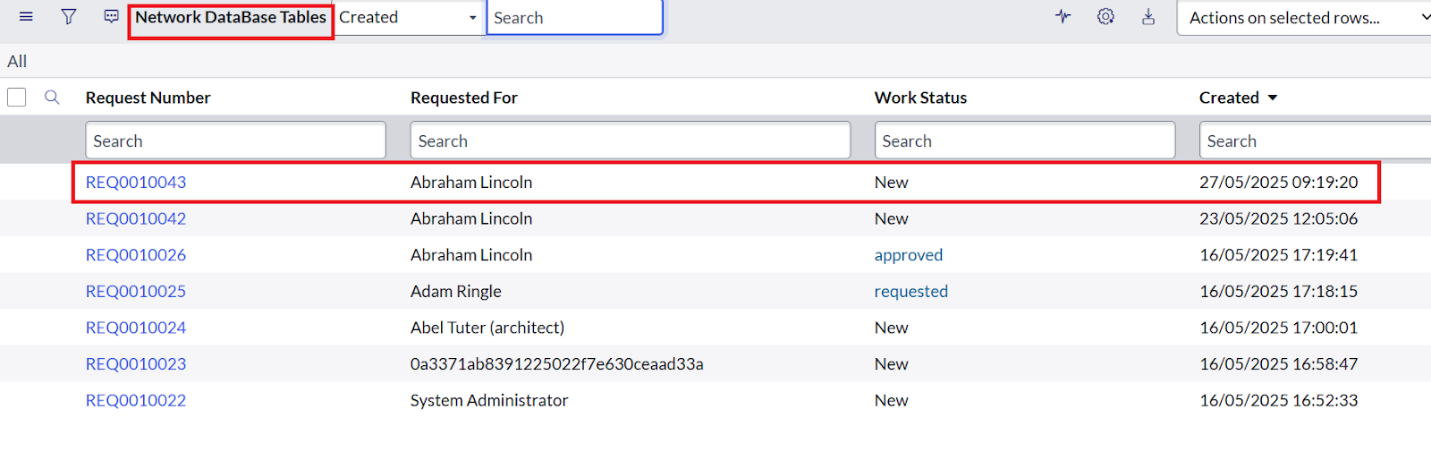


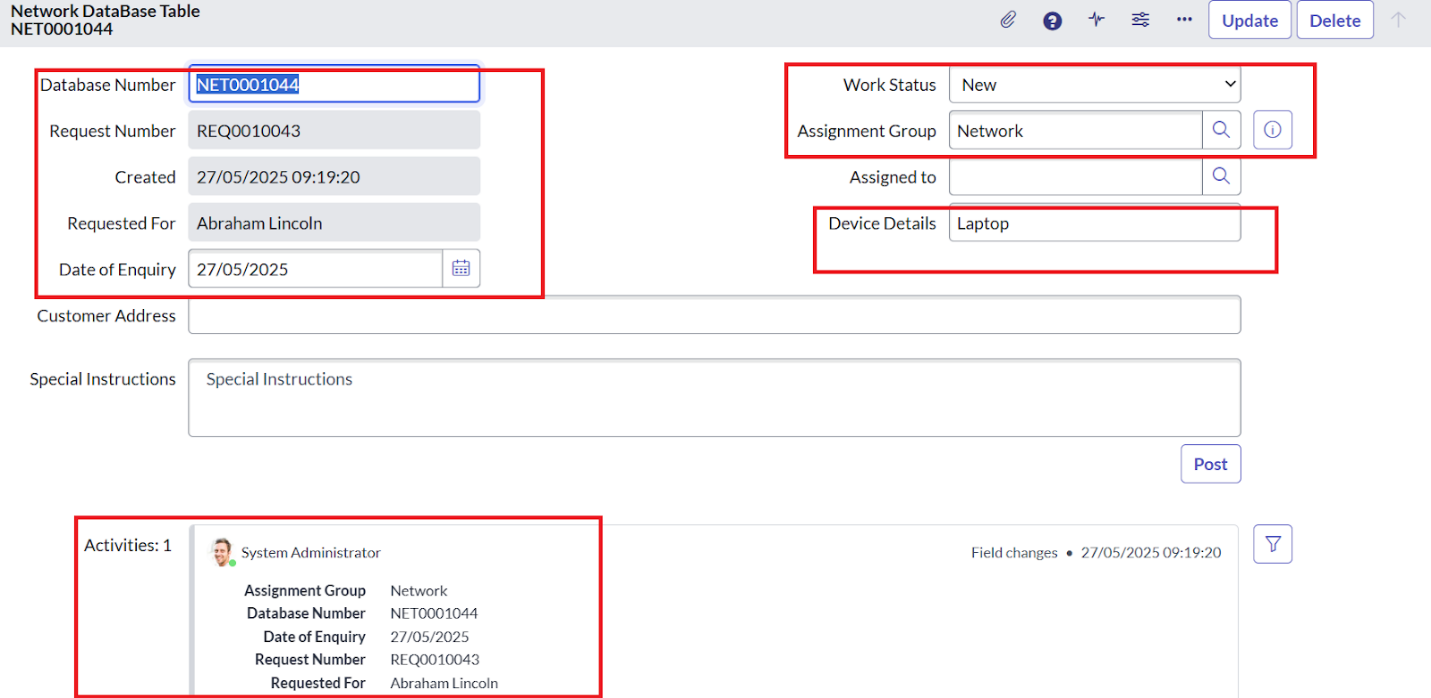


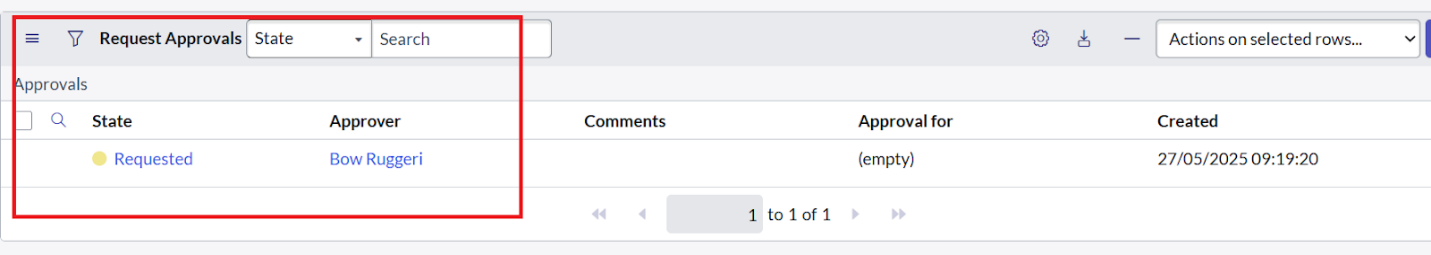
****

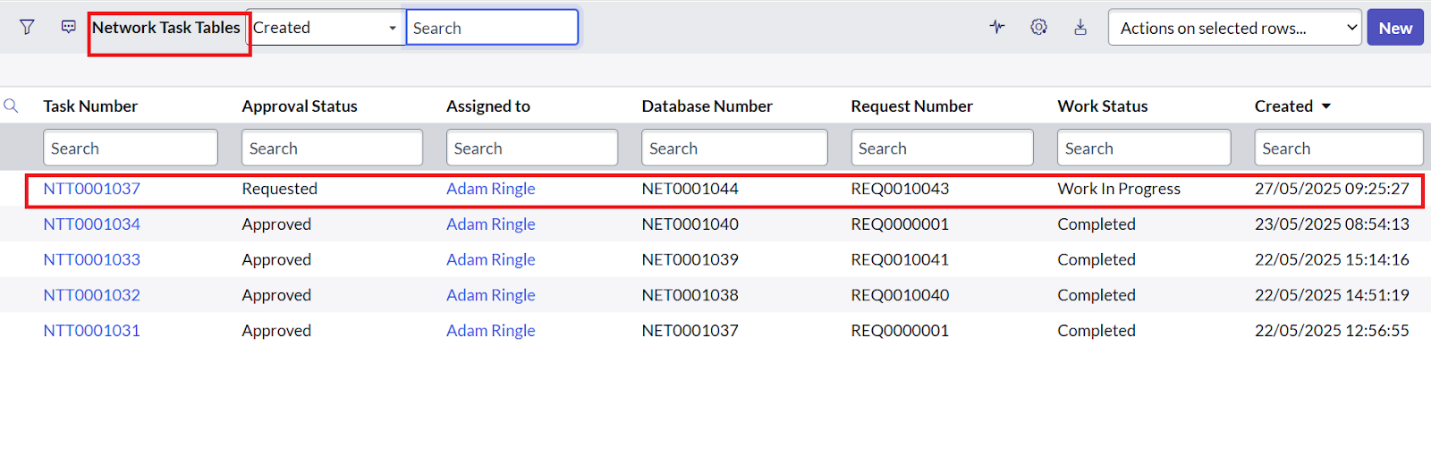
Step 10: Testing with Custom Tables

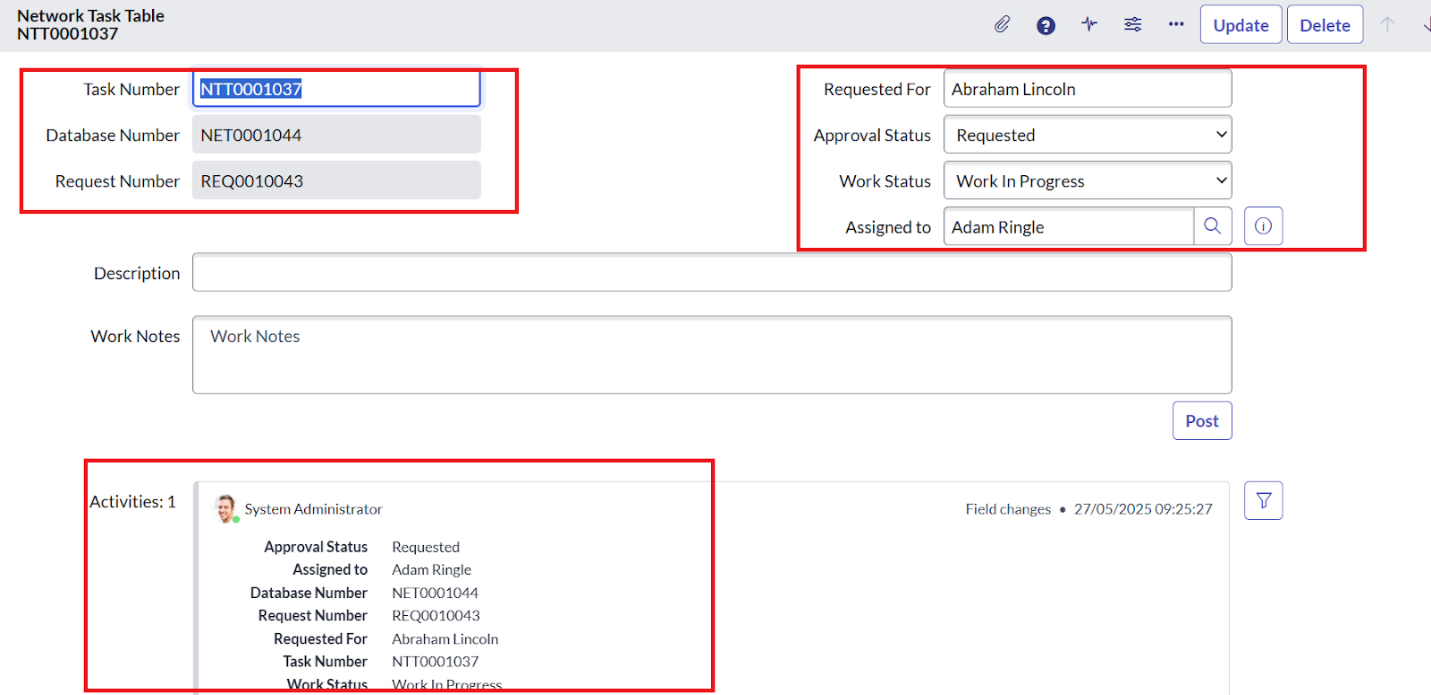
* Open the Network Database table after submission.
* Verify that fields are correctly populated with request details.
* Check related approval records and confirm state changes after approvals.
* Validate updates when approvers approve, reject, or reassign requests.

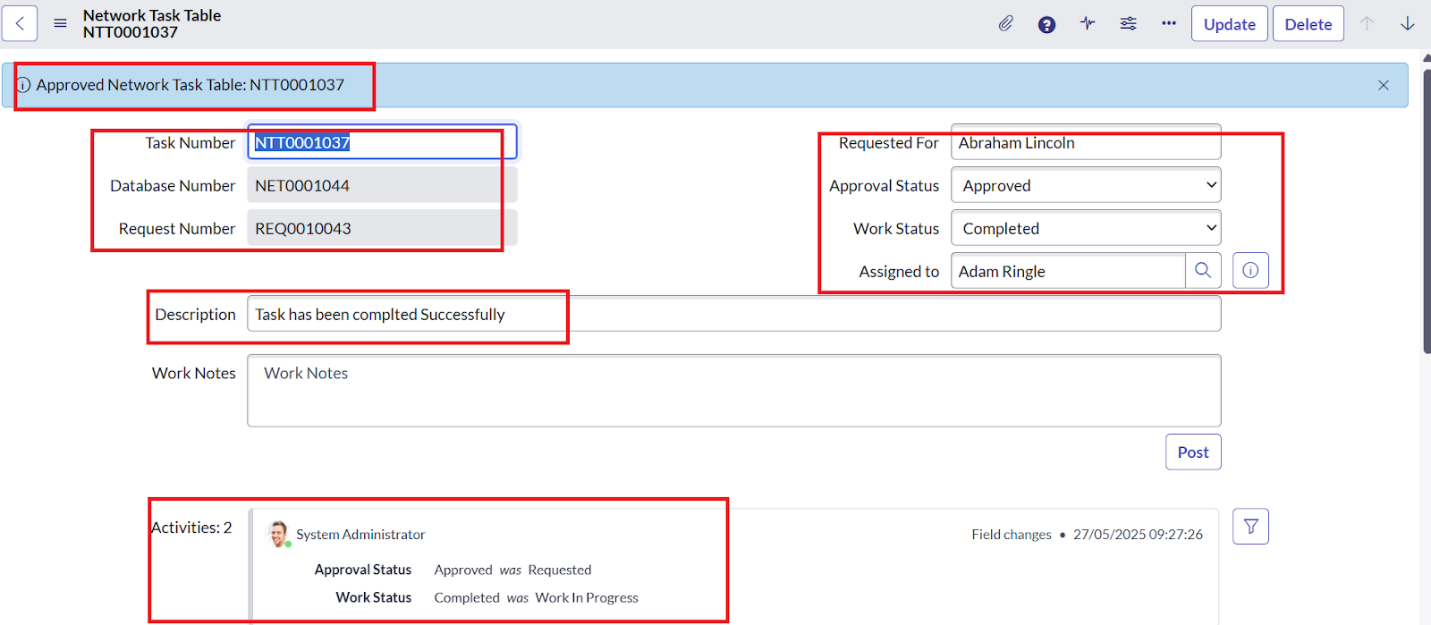


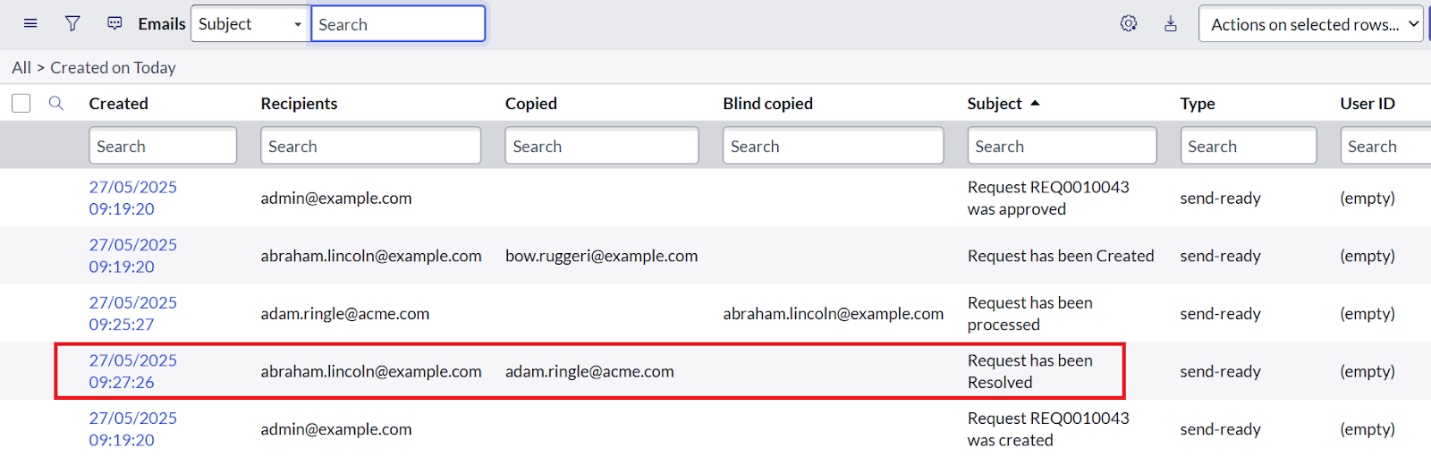


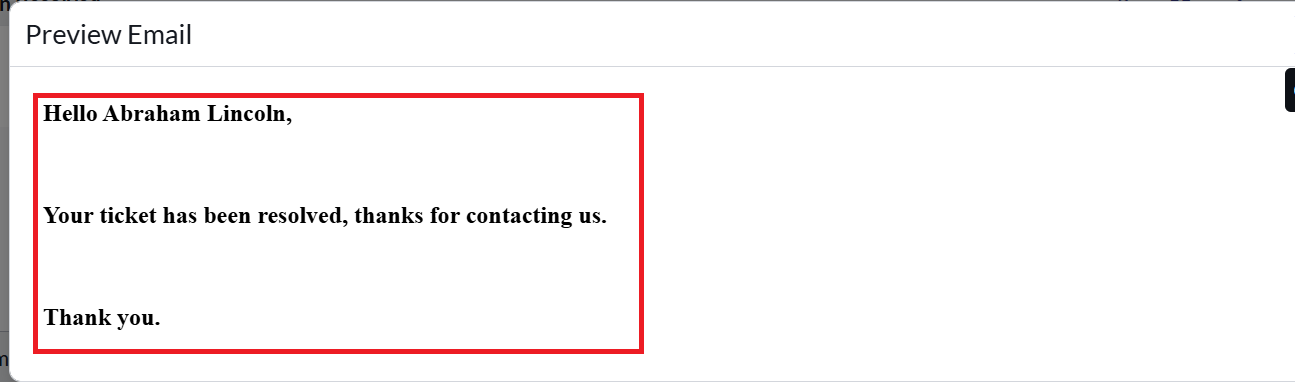












Result  
At the end of implementation, a fully functional Network Request Catalog was created. Requests flow through dynamic forms, approval chains, and notifications, ensuring real-time visibility and efficient resolution.

**7. FUNCTIONAL AND PERFORMANCE TESTING**

* Form Validation: Verified mandatory fields and conditional display.
* Workflow Execution: Confirmed approval routing and state changes.
* Email Testing: Checked dynamic recipient details and triggered messages.
* Table Updates: Ensured records stored accurately in Network Database.
* Performance: Tested with multiple simultaneous requests to confirm system responsiveness.

**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* Eliminates manual tracking and reduces delays
* Provides transparency with role-based approvals
* Ensures real-time email notifications
* Easy to scale for future network services

**Limitations**

* Requires ServiceNow expertise for configuration
* Initial setup is time-intensive
* Relies on proper access controls to prevent misuse

**9. CONCLUSION**

The Network Request Management system in ServiceNow automates request submission, approval, and fulfillment. By leveraging Service Catalog, Flow Designer, and custom tables, it ensures streamlined workflows, transparency, and improved efficiency. This solution reduces manual intervention while enhancing user satisfaction and organizational productivity.

**10. FUTURE SCOPE**

* Dashboard Integration: Role-based dashboards for employees and IT staff.
* Chatbot Interface: Allow employees to raise requests via virtual assistants.
* SMS/WhatsApp Notifications: Expand beyond emails for faster updates.
* Analytics & Reporting: Monitor request trends, SLA compliance, and workload distribution.
* Integration with Monitoring Tools: Auto-generate network requests from system alerts.