**Project Documentation**

**Automated Network Request Management Using ServiceNow**

**1.INTRODUCTION**

**Project Overview**  
The Automated Network Request Management system is built on the ServiceNow platform to automate and streamline the handling of network-related service requests. It enables users to submit requests such as new network connections, device relocations, and other network services via a self-service portal. Requests are routed through dynamic forms, approval workflows, and task fulfillment steps.

**Purpose**  
To eliminate manual tracking of network service requests, improve accuracy, enforce structured workflows, and deliver real-time status updates and notifications to users and IT staff.

**2.IDEATION PHASE**

**Problem Statement**

Organizations often face delays and inconsistencies in handling manual network service requests, such as relocations or device setups, due to lack of standardization, poor visibility, and no automation.

**Empathy Map Canvas**

* **Who?** End users, IT Technicians, Network Admins.
* **Think/Feel?** Uncertainty about request status, delays.
* **See?** Manual emails, poor tracking, frustrated users.
* **Say/Do?** Frequent follow-ups, repeated queries.
* **Hear?** Complaints about delays and approvals.
* **Pain?** Delayed responses, lack of transparency.
* **Gain?** Faster turnaround, email updates, traceable status

**Brainstorming**

Considered:

Traditional ITSM methods lacked self-service, standardization, and real-time status updates. ServiceNow was chosen for its workflow engine, form customization, notifications, and automation capabilities.

**3.** **REQUIREMENT ANALYSIS**

**User Journey Map**

User logs in → submits network request → approval initiated → approver approves → request routed to technician → task fulfilled → user notified

**Solution Requirements**

* Custom Catalog Item: "Network Request"
* Categories: Network Services
* Dynamic form with condition-based visibility
* Variable Set: Opened On Behalf Of, Email, Phone, etc.
* Approval Workflow: Based on type and user role
* Notifications: For submission, approval, completion
* Tables: Network Database Table

**Data Flow Diagram**

User logs in → submits network request → approval initiated → approver approves → request routed to technician → task fulfilled → user notified

**Technology Stack**

* **Platform:** ServiceNow
* **Logic:** Flow Designer, Client Scripts, UI Policies
* **Modules:** Service Catalog, Workflow, Notifications.
* **Scripting:** JavaScript,Glide APIs

**4. PROJECT DESIGN**

**Problem-Solution Fit**

The manual request process was slow and lacked traceability. This solution digitizes network requests through ServiceNow, provides dynamic forms, automates approvals, sends notifications, and enables real-time tracking.

**Proposed Solution**

To address the problem effectively, the following components were implemented:

* **Catalog Item:** Network Request
* **Variable Types:**
  + Is this a new connection or relocation? (Choice)
  + Address and device details (String)
  + Type of device (Choice)
  + Specify field visible only if 'Other' device is selected
* **Variable Set:**
  + Opened on behalf of (Reference)
  + Email, Username, Phone (Auto-populated)
  + Proof of Document (Attachment)
* **UI Policies:** Show/hide fields conditionally
* **Custom Table:** Network Database Table
* **Fields:** Status, Address, Device Type, Relocation Info
* **Approval Record:** sysapproval\_approver related list
* **Flow Designer:**
  + **Trigger:** Record Created (Network Request)
  + **Actions:**
    - Get Catalog Variables
    - Create Record in Network Table
    - Ask for Approval
    - Send Email Notification
* **Notifications:** To requester and approver

**5. PROJECT PLANNING & SCHEDULING**

During this phase, the project was broken down into manageable development steps, ensuring modular implementation and validation. Key activities included:

**Steps:**

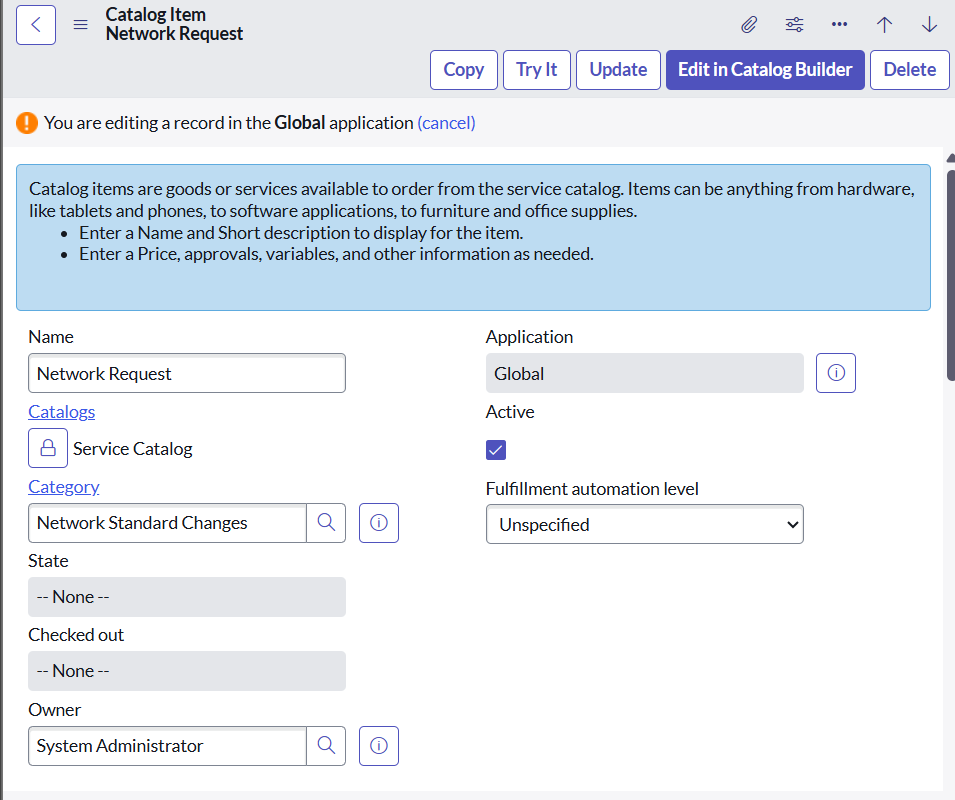
1. Set up PDI instance
2. Create Update Set: Network Request Automation
3. Catalog Item: Network Request
4. Add Variables & Variable Sets
5. Create Custom Table: Network Database
6. Add fields to the table (Status, Type, Address)
7. Create and apply UI Policies
8. Configure Flow Designer with Get Variables, Approval, Create Record, Send Email
9. Add Related Lists to form layout (Approvals)
10. Set up Notification Templates
11. Test via Service Portal

**6. IMPLEMENTATION WORKFLOW**

This phase outlines the sequence of steps performed in implementing the **Automated Network Request Management**:

**Step 1: Creation of service catalog**

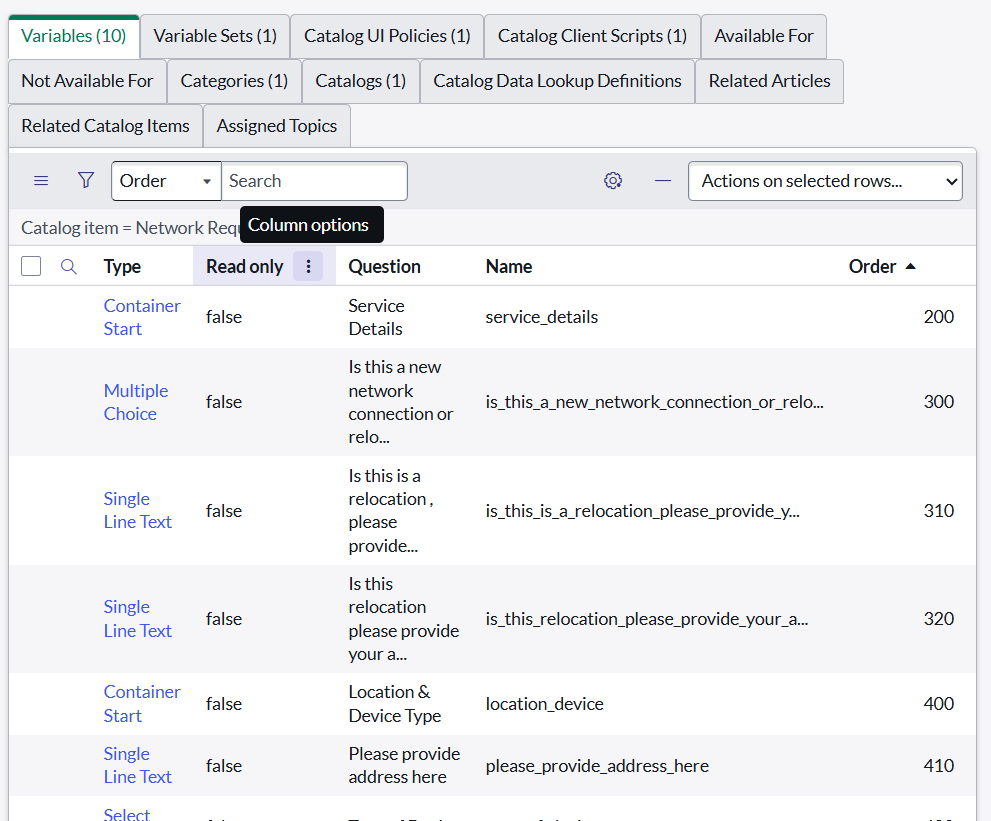
1. Navigate to Application navigator
2. Click on All >> search for Service Catalog
3. Under Service Catalog>> Maintain items
4. Click on New
5. Fill the details >> Name– Network Request
6. Select Catalog>> Service Catalog
7. Select Category>> Network
8. Fill the Short Description as Network request Management
9. Click on Save.



**Step 2: Variable Configuration**

Open the catalog item just created.  
Scroll down to the **Variables** related list and click **New** to create form fields.

1. Select Variables type as Single, Multi line text, reference, choices etc as per requirement
2. Catalog item – Network Request
3. Order– 100, 200, 300,
4. Question– provide the variable label
5. Name–provide the variables name (used for scripting)
6. Tooltip– this will appear when cursor overed on the field
7. Example text – this will suggest what we need to enter on the field.
8. Mandatory, Read – Only – need to configure on demand
9. Auto populate– need to select dependent variable, apply dot walking to get selected value.
10. Click on Save or Submit.



**Step 3: Variable Types**

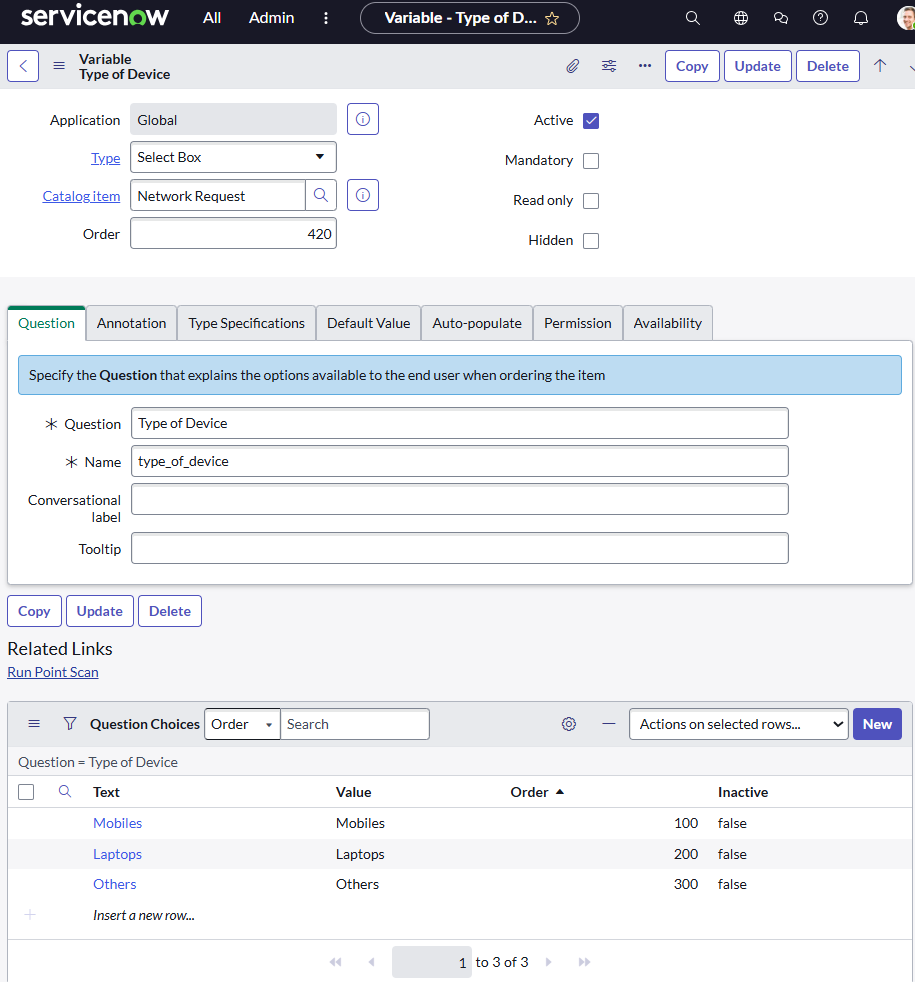
* Is this a New connection or Relocation? >> **Choice** >> **New/ Relocation/None**
* If this is a relocation, Please provide your relocated address here>>**String**
* Types of devices>> **Choice**>> **Laptop/Mobiles/Others**
* Please provide address here**>>String**
* Provide device details here>> **String**
* If anything else, please specify>> **String**

**Step 4: Variable Set Configuration**

* To enhance form usability:
  + Navigate to the **Variable Sets** (optional).
  + Follow the same procedure as we used for Variables Creation, for the variable set as well.
  + Apply variable sets to the catalog item.

**Variables Types**

1. Opened on behalf of >> Reference>> reference to user table
2. Email Id >> Single line text >> Auto populate by Opened on behalf of variable.
3. User name >>Single line text >> Auto populate by Opened on behalf of variable.
4. Phone Number >>Single line text >> Auto populate by Opened on behalf of variable.
5. Proof of Document >> Attachment

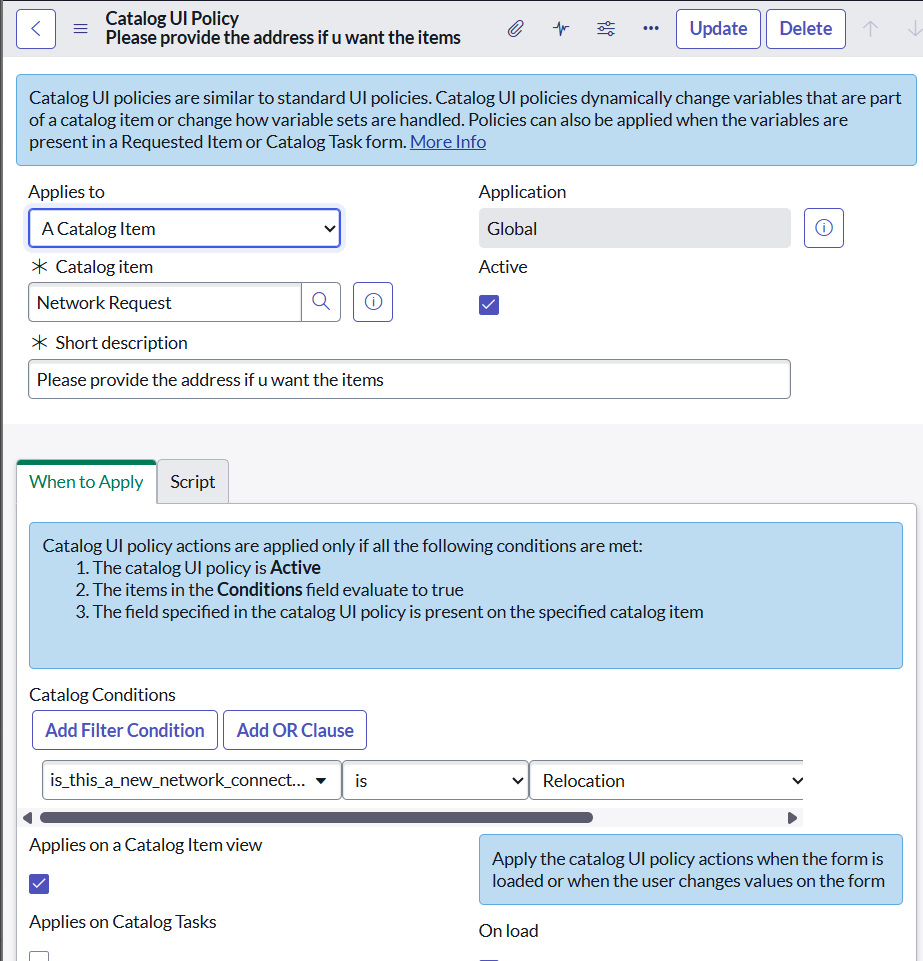


**Step 5: Catalog UI Policy Configuration**

Scenario: If user selects types of devices is **Others,** then Please specify field should populate.

**Procedure:**

1. Navigate to catalog items
2. Open Network Request item
3. In related list, we have Catalog UI policy
4. Click on New button to configure New UI policy
5. Select Applies to as Catalog item
6. Select catalog item as Network Request
7. Provide short description, if required
8. Apply condition>> **types of devices** is **others**
9. Clickon save, after saving the form will get UI policy actions in the related list
10. Click on New button to configure new UI Policy action, and Select the variable which we want to display on condition
11. Make Visible True as per our requirement
12. Update the UI Policy and Test the same on Catalog form.



**Step 6: Creation of Table**

**Navigate to**: System Definition > **Tables**.

· Click **New** to create a new table.

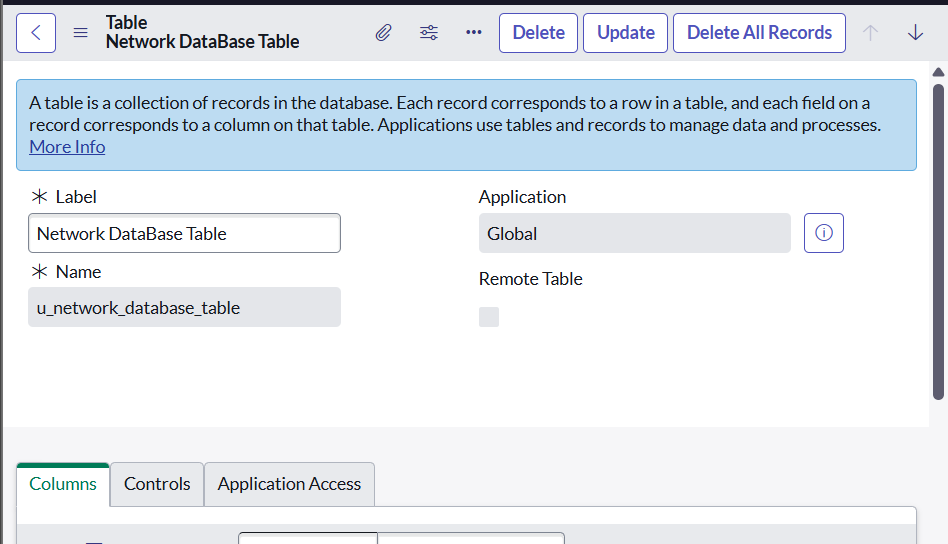
· Fill **in Table Information**:

·   **Name**: Name of the table ------

·   **Label**: Backend name of the table------

·   **Auto-generate schema**: Leave it checked if you'd like ServiceNow to auto-generate schema fields.

· Click **Submit** to create the table.



**Step 7: Creation Of Fields**

In ServiceNow, fields are created at the **table** level. To create a field, you first need to identify the table where the field will reside.

1. In the **Application Navigator** (left-side panel), type **Tables** in the search bar.

2. Under **System Definition**, click **Tables**. This will take you to a list of all tables in the system.

**Select the Table to Add the Field**

·   From the list of tables, search for and select the **table** you want to add a field to. For example, if you want to add a field to the **Network database** table:

1. Type "**Network database**" in the search box or scroll through the list.

2. Click on the **Network database** table name. You’ll now see a list of all fields (columns) associated with the **Network database** table.

**Open the Table's Columns**

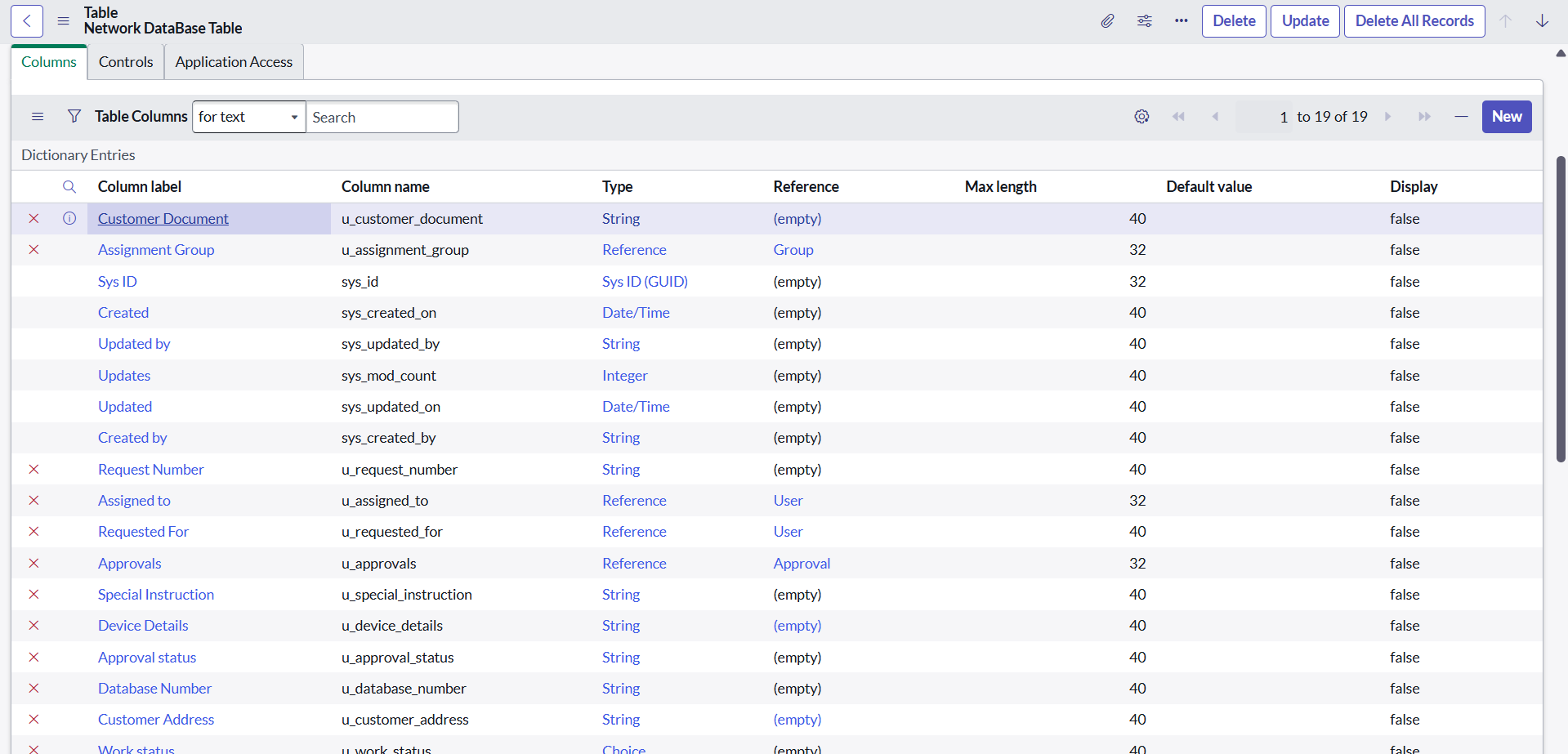
·   After selecting the table, you'll be brought to a view that lists all the columns (fields) that currently exist on that table.

·   To create a new field (column), go to the **Columns** tab (this is where all fields for the selected table are listed).

**Create a New Field**

1. In the **Columns** tab, click the **New** button located at the top-right corner of the page to create a new field.

2. You’ll now be prompted with a form where you need to define the new field. The following fields need to be filled out:



**Task: Defined Field Properties**

Fill in the following details for your new field:

**1. Column Label (Field Label)**

·   **Description**: This is the name that will be displayed on the forms, lists, and records.

·   **Example**: Customer Name

**2. Column Name**

·   **Description**: This is the internal name of the field and is auto-generated based on the column label. It should be unique for each field. Do not manually edit this unless necessary.

·   **Example**: customer\_name

·   **Description**: The type of field determines the kind of data it will store. You need to choose the correct type based on the data you want to store (e.g., text, number, date, etc.). Some of the most common types include:

o   **String**: For short text values (e.g., name, description).

o   **Integer**: For numbers without decimals (e.g., age, number of items).

o   **Choice**: A dropdown list of options.

o   **Reference**: A field that links to another table (e.g., linking to a User table).

o   **Boolean**: A true/false checkbox.

o   **Date**: For a date picker field.

o   **Date/Time**: For both date and time.

·   **Example**: String, Choice, Reference

**3.  Max Length (Optional)**

·   **Description**: If you are creating a string-type field, you can specify the maximum length of the text allowed.

·   **Example**: 255 characters (default length for a string field).

**4. Mandatory**

·   **Description**: Check this box if the field should be required when creating or updating records.

·   **Example**: For a "Customer Name" field, this might be required.

**5. Default Value (Optional)**

·   **Description**: You can set a default value for the field if desired. This value will appear automatically when creating a new record.

·   **Example**: Set the default value to "New Customer" for a "Customer Name" field.

**6. Read-Only**

·   **Description**: Check this box if the field should be read-only (users cannot modify its value). This is commonly used for calculated or system-generated fields.

·   **Example**: "Created Date" or "Record Number".

**7: Save the Field**

·   Once you’ve configured all the necessary field properties, click **Submit** or **Save** to create the field.

·   After saving, ServiceNow will create the new field and add it to the list of columns for the selected table.



**Task: Add The Field To A Form**

After creating the field, you may want to add it to a form so that users can view or update it.

1. To do this, navigate to **System UI > Forms** in the application navigator.

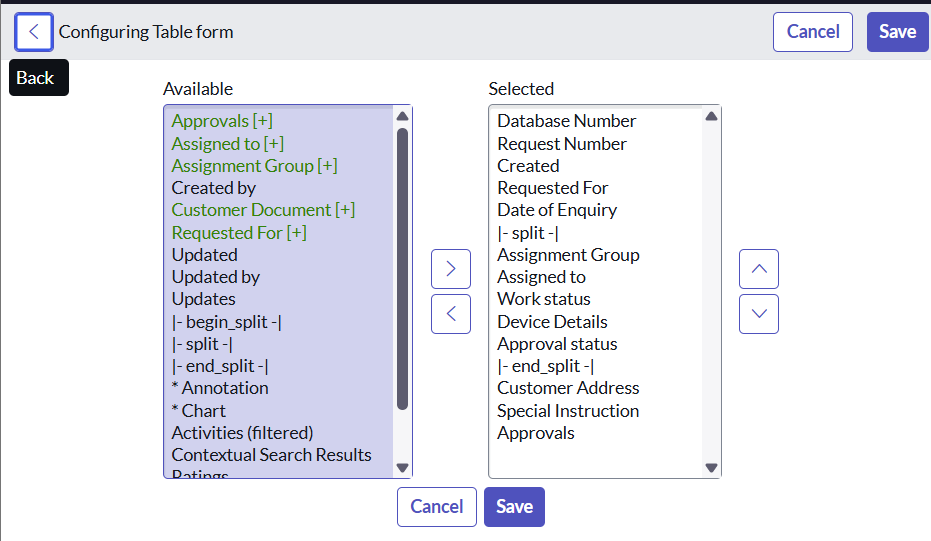
2. Select the **form** you want to modify (e.g., Incident form).

3. Open the **Form Designer** (click on the "Design" icon).

4. From the **Field Navigator** on the left side, search for the new field you created.

5. Drag the field onto the form layout where you want it to appear.

6. Click **Save** or **Publish** to apply the changes.



**Test the New Field**

**·** Go to a record in the table where the field was added (e.g., create a new incident or record).

·   Check if the new field appears on the form.

·   Verify the field behaves as expected (e.g., required, read-only, etc.).

Key Field Types in ServiceNow:

·   String: Short text input (e.g., a name, description).

·   Integer: Whole numbers.

·   Choice: Dropdown list with predefined options.

·   Reference: A reference field to another table (e.g., referencing an User table).

·   Date: A date picker.

**·** Date/Time: A combination of date and time.

**·** Boolean: Checkbox (True/False).

·   Currency: Currency field with monetary values.

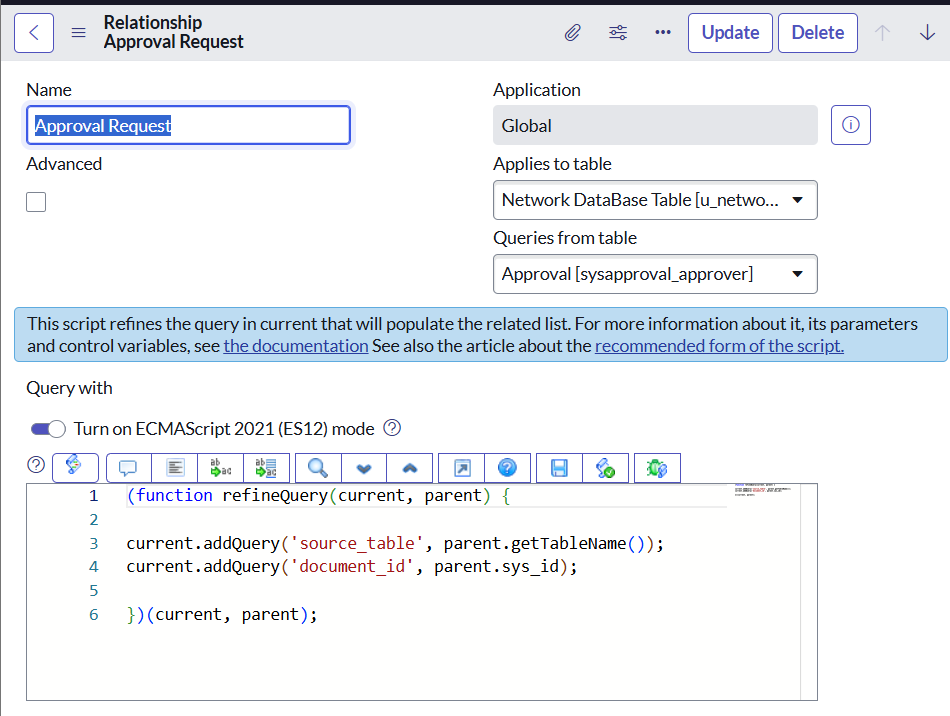
**Additional Tips:**

**·** Field Data Types: Make sure you choose the correct field type based on the type of data you want to store (e.g., Text, Integer, Date).

·   UI Policies/Client Scripts: These can be used to make fields visible, read-only, or mandatory based on certain conditions.

·   Naming Conventions: Follow proper naming conventions for field labels and column names to maintain consistency.

* **Creation Of Related List**
* Navigate to **System Definition > Relationships**.
* Click **New** to create a new relationship.
* Fill in the following details:
* **Name**: Approval Request
* **Applies to Table** : Network Database table.
* **Queries from Table** : Sysapprovals table.
* **Active**: Make sure it's set to **True**.
* Save the relationship.



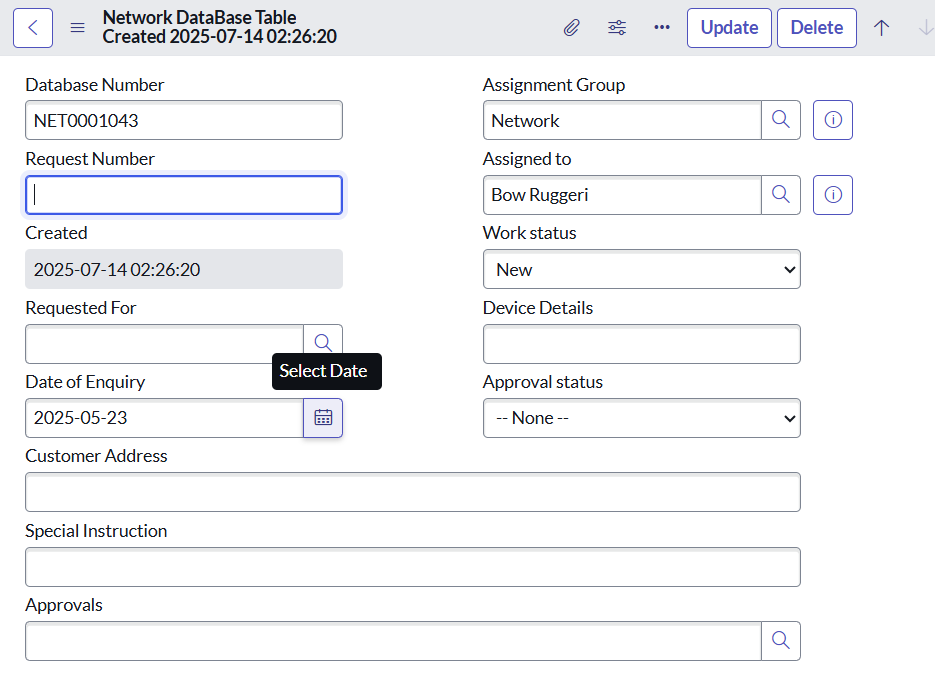
**Task: Adding Related List To The Table**

You can create a **Related List** on a form to display the related records. This helps in easily viewing the relationships between records.

·   Navigate to **Form Designer** for the table where you want to show related records.

·   Add a **Related List** widget to the form.

·   Select the **Related List** you want to show



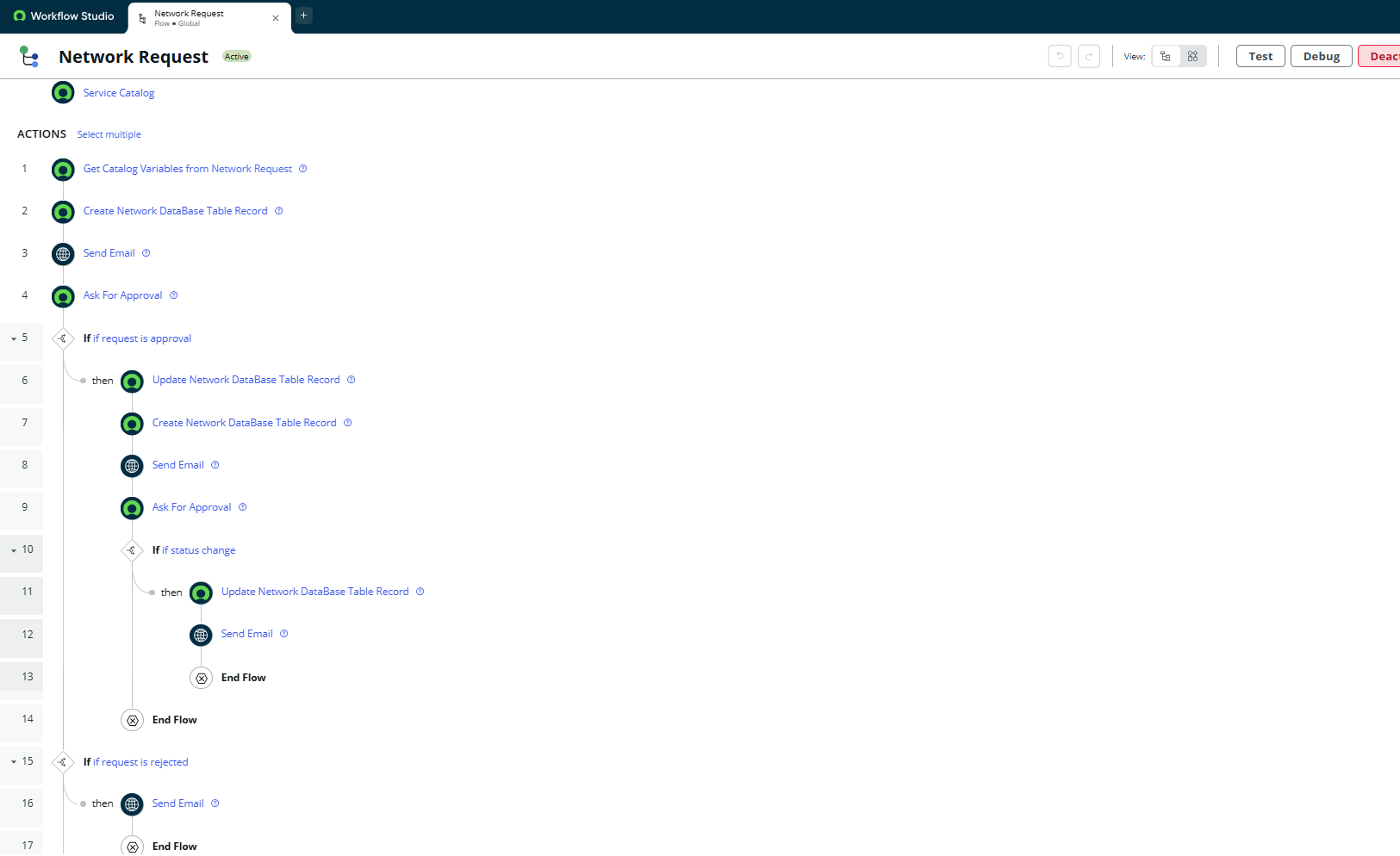
**Task :Navigating To Flow Designer**

To access Flow Designer:

1.Go to Flow Designer by typing Flow Designer in the left-hand application navigator, or navigate through All > Flow Designer.

**7. Creation & Implementation of Flows, Actions In Flow Designer**

* **Trigger:** Catalog Item Requested
* **Actions:** Get Variables, Create Record, Ask for Approval, Send Email



**8. Functional And Performance Testing**

**Tests Performed:**

* Catalog Form: All fields displayed correctly
* UI Behavior: Dynamic fields worked as expected
* Flow Execution: Approval triggered and logged
* Email Notification: Sent to requester and approver
* Data Accuracy: Record created in the custom table
* Related List: Approvals appeared after adding correct relationship

**9. ADVANTAGE & LIMITATIONS**

**Advantages**

* Reduces manual network request errors
* Ensures proper routing via approvals
* Tracks SLA and resolution time
* Easy integration with network automation tools

**Limitations**

* Requires admin-level access
* Initial setup is configuration-heavy

**9. CONCLUSION**

This project successfully implemented an automated, approval-based system for managing network service requests within ServiceNow. Using Service Catalog, dynamic forms, Flow Designer, and notifications, the system improves efficiency, accuracy, and visibility for both users and IT staff.

**10. Testing In Service Portal (End User)**

1. Login to PDI: https://devXXXXXX.service-now.com
2. Open Service Portal: https://devXXXXXX.service-now.com/sp
3. Search: "Network Request"
4. Fill in request form and click Submit
5. Receive confirmation with REQ number
6. Check for email confirmation

**11. FUTURE SCOPE**

* Integration with network orchestration tools (e.g., Ansible, Cisco APIs)
* Auto-assignments based on device type/location
* Dashboard for request tracking and analytics
* Enhanced mobile compatibility via mobile studio
* SLA breach alerts and escalations