

Automated Network Request Management in ServiceNow

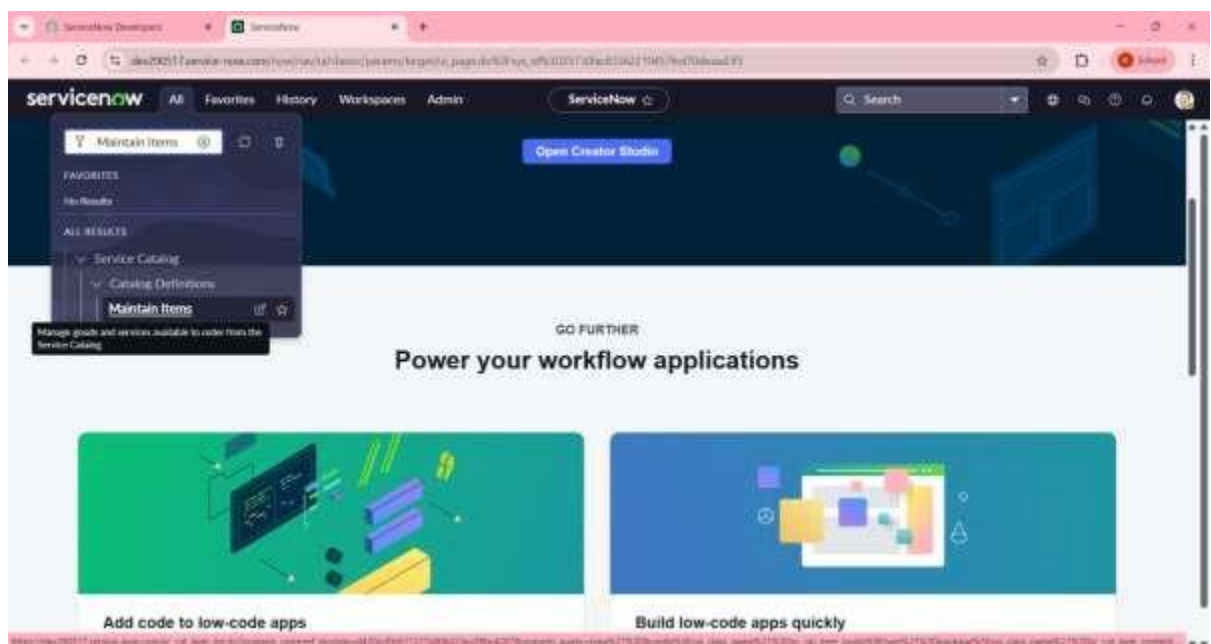
INTRODUCTION:

This project provides an automated solution in **ServiceNow** to manage network-related service requests. Through a self-service portal, users can easily submit requests, which are then validated, approved, and routed for fulfillment. Automated workflows handle approvals, notifications, and task assignments, while optional integrations with network tools reduce manual effort. The system also offers real-time updates and reporting to improve efficiency, transparency, and SLA tracking.

Process 1: Creation of Service Catalog – "Network Request"

Step 1: Navigate to Service Catalog

1. Open the **Application Navigator** in ServiceNow.
2. Go to:
All → Service Catalog → Maintain Items



Step 2: Create New Catalog Item

1. Click on **New**.
2. Fill the following details:
 - **Name:** Network Request
 - **Catalog:** Service Catalog
 - **Category:** Network and connctivity
 - **Short Description:** Network Request Management
3. Click on **Save**.

The screenshot shows the ServiceNow 'Catalog Item - Network Request' form. The form is titled 'Catalog item - Network Request' and includes a search bar and navigation links like 'Copy', 'Try It', 'Update', 'Edit in Catalog Builder', and 'Delete'. Below the title, there is a blue box with instructions: 'Catalog items are goods or services available to order from the service catalog. Items can be anything from hardware, like tablets and phones, to software applications, to furniture and office supplies. Enter a Name and Short Description to display for the item. Enter a link, approvals, variables, and other information as needed.' The form fields are as follows: 'Name' is 'Network Request', 'Catalog' is 'Service Catalog', 'Category' is 'Networks and Connectivity', 'Scale' is 'None', 'Checked out' is 'None', 'Owner' is 'System Administrator', 'Application' is 'Global', 'Active' is checked, and 'Fulfillment automation level' is 'Unspecified'. At the bottom, there is a 'Short description' field with the text 'Network request Management' and a rich text editor for the 'Description' field.

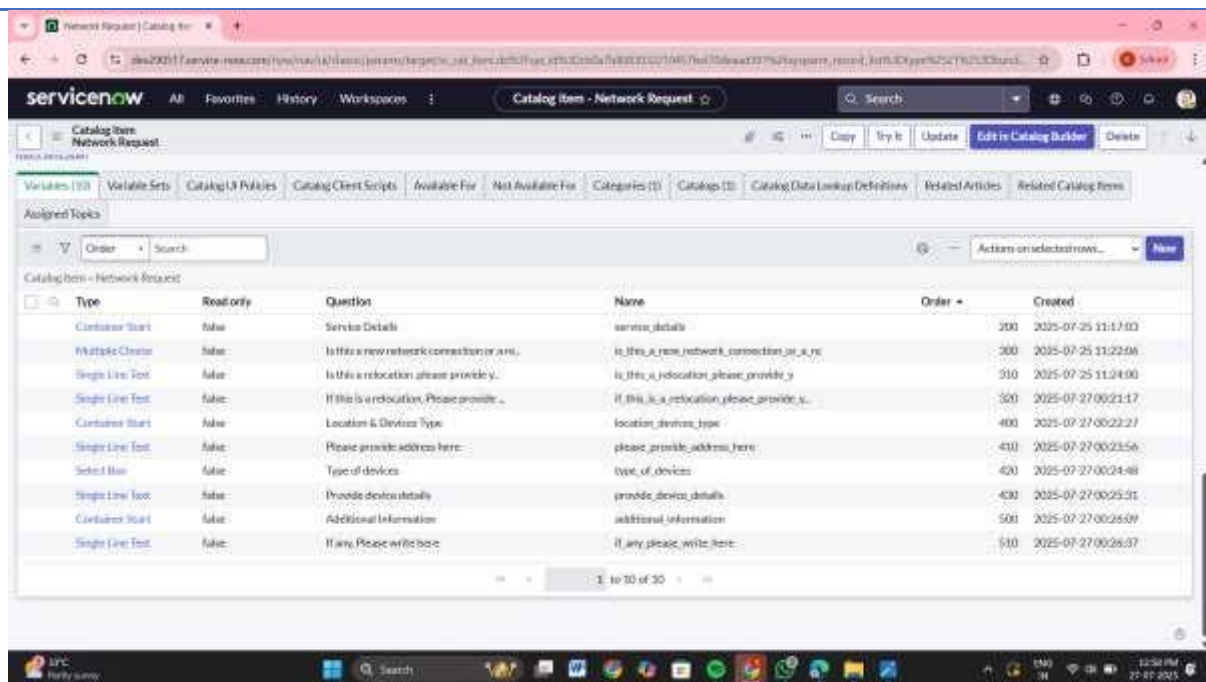
Step 3: Configure Variables

1. Open the newly created **Network Request** catalog item.
2. Scroll down to the **Variables** related list → Click **New** for each variable.
3. Fill out the following for each variable:
 - **Type:** Single line text, Choice, Reference, etc.
 - **Order:** e.g., 100, 200, 300 (controls display order)
 - **Question:** Label shown on the form
 - **Name:** Technical name (used in scripts)

- **Tooltip:** Info shown on mouse hover
- **Example Text:** Placeholder help text
- **Mandatory / Read-Only:** As required
- **Auto-populate:** Use dot-walking for dependent values

Step 4: Variable Types Configuration

| Type | Question | Order |
|------------------|---|-------|
| Container Start | Service Details | 200 |
| Multiple Choice | Is this a new network connection or a relocation? | 300 |
| Single Line Text | If this is a relocation, Please provide ... | 310 |
| Single Line Text | If this is a relocation, Please provide ... | 320 |
| Container Start | Location & Devices Type | 400 |
| Single Line Text | Please provide address here | 410 |
| Select Box | Type of devices | 420 |
| Single Line Text | Provide device details | 430 |
| Container Start | Additional Information | 500 |



Step 5: Configure Variable Set – Requester Information

5.1 Create Variable Set

1. Navigate to **Variable Sets** under Service Catalog.



2. Click on **New**.
3. Fill the following details:
 - **Title:** Requester information
 - **Internal Name:** requester_information (auto-filled)
 - **Order:** 100
 - **Type:** Single Row
 - **Layout:** 2 Columns Wide, one side, then the other

- Check the box: ☒ **Display title**

4. Click **Submit** or **Update**

The screenshot shows the 'Variable Set - Requester information' configuration page. The 'Display title' checkbox is checked. The 'Title' field is 'Requester information', 'Internal name' is 'requester_information', 'Order' is '100', and 'Type' is 'Single Row'. The 'Application' is 'Global' and 'Layout' is '2 Columns Wide, one side, then the other'. A 'Description' field is also present. At the bottom, there are tabs for 'Catalog Client Scripts', 'Included In (1)', and 'Catalog Data Lookup Definitions'.

Step 5.2: Add Variables to the Variable Set "Requester Information"

After creating the variable set, now it's time to add the variables one by one.

1. Opened on behalf of

- Type: **Reference**
- Reference to: **User *sys_user+**
- Name: **opened_on_behalf_of**
- Order: **100**
- This allows the requester to select a user they are raising the request for.

The screenshot shows the ServiceNow 'Variable - New Record' form. The form is for a variable named 'Requester Information' of type 'Single Line Text'. It has an order of 100 and is active. The 'Question' field is set to 'email_id', and the 'Name' field is also set to 'email_id'. The 'Example Text' field is also set to 'email_id'. The 'Submit' button is visible at the bottom left.

2. Email ID

- Type: **Single Line Text**
- Name: email_id
- Order: 200
- This will be auto-filled based on the user selected in "Opened on behalf of".
- You can use a script or dot-walking to populate the email field.

3. User Name

- Type: **Single Line Text**
- Name: user_name
- Order: 300
- This will also be auto-populated based on the user selected.
- Fetch the full name from the User table.

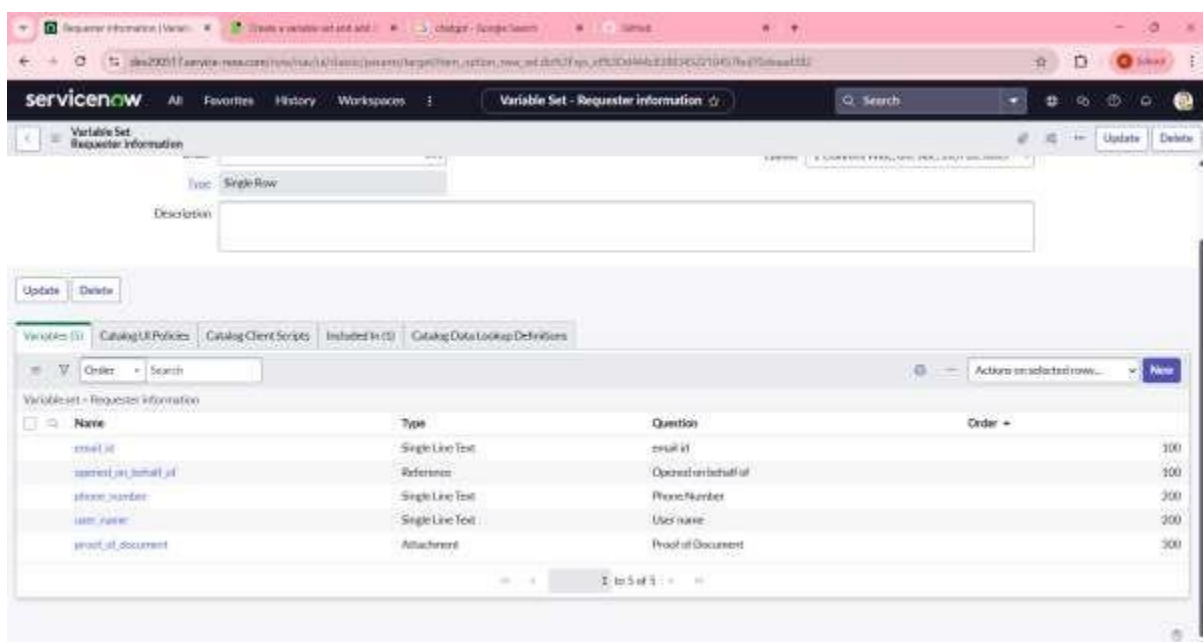
4. Phone Number

- Type: **Single Line Text**
- Name: phone_number

- Order: 400
- Same as above, it can be fetched using dot-walking or client script.

5. Proof of Document

- Type: **Attachment**
- Name: proof_of_document
- Order: 500
- This allows users to upload a file (such as proof or ID documents).



When a user is selected in the **Opened on behalf of** field, we want to automatically populate:

- Email ID
- User Name
- Phone Number

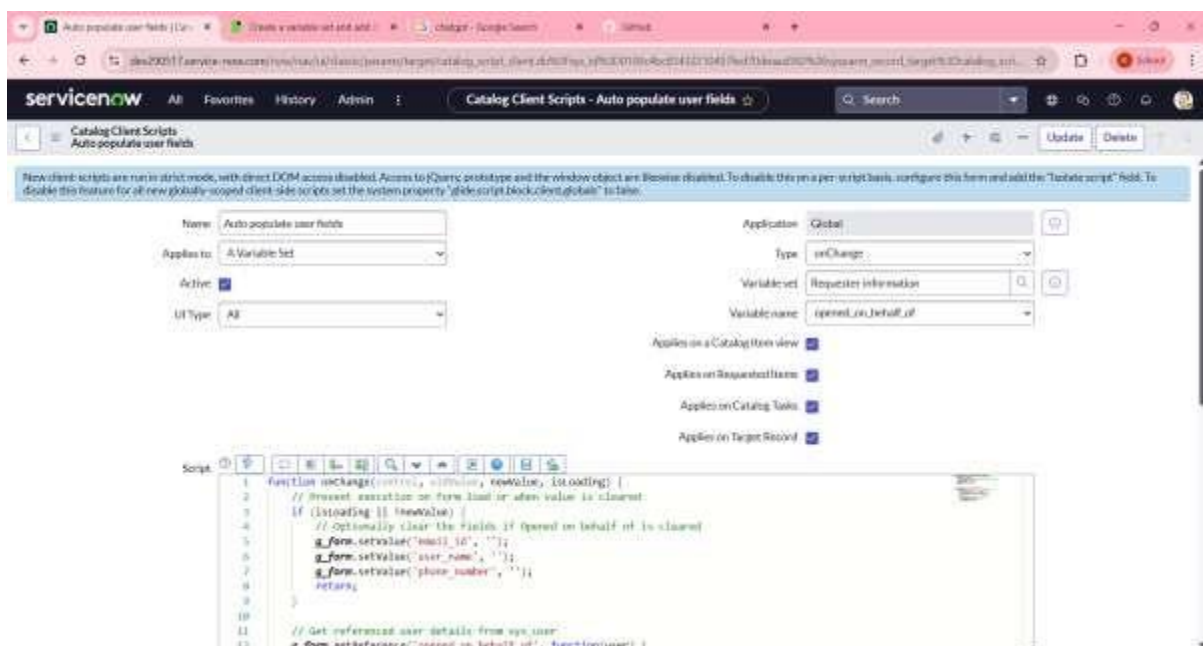
Steps to Auto-populate Fields

1. Open the Variable Set

- Navigate to: **Service Catalog > Catalog Variable Sets**
- Open your variable set: **Requester Informatio**

2. Create a Catalog Client Script

- Navigate to: **Service Catalog > Catalog Client Scripts**
- Click **New**
- Fill in details:
 - **Name:** Auto Populate User Info
 - **Applies to:** Catalog Item
 - **Variable Set:** Select *Requester Information*
 - **UI Type:** All
 - **Type:** onChange



3. Configure the Script Fields

- **Variable name:** opened_on_behalf_of
- **Script:**



Step 6: Catalog UI Policy Configuration

Goal: Show " Provide device details here " field when **Types of Devices = Others**.

1. Navigate to the **Network Request** catalog item.
2. In the related list, go to **Catalog UI Policies** → Click **New**.
3. Fill in:
 - **Applies to:** Catalog Item
 - **Catalog Item:** Network Request
 - **Condition:** Types of devices is Others
4. Click **Save**.
5. In the related list, click **New** under **UI Policy Actions**.
6. Set:
 - **Catalog Item:** Network Request
 - **Variable name:** Provide device details here
 - **Visible:** True
7. Click **Update** to save policy.
8. **Test the form** to ensure the field appears based on selection.

< Catalog UI Policy New record

Catalog UI policies are similar to standard UI policies. Catalog UI policies dynamically change variables that are part of a catalog item or change how variable sets are handled. Policies can also be applied when the variables are present in a Requested Item or Catalog Task form. [More Info](#)

Applies to
 A Catalog Item
 * Catalog item
 Network Request
 * Short description
 Display field when device is Others

Application
 Global
 Active
☒

When to Apply Script

Catalog UI policy actions are applied only if all the following conditions are met:

1. The catalog UI policy is **Active**
2. The items in the **Conditions** field evaluate to true
3. The field specified in the catalog UI policy is present on the specified catalog item

Catalog Conditions
 Add Filter Condition Add "OR" Clause
 type_of_devices is Others

Applies on a Catalog Item view

Process 2: Creation of Table and Fields in ServiceNow

>Network Database Table

Step 1: Create a New Table

1. Navigate to the Application Navigator.
2. Type: Tables under the **System Definition** module.
3. Click on **Tables**.
4. On the top-right corner, click on **New** to create a new table.

5. Fill in the table details:

- **Label:** *Network Database Table*
- **Name:** Automatically generated (or customize if needed).
- Keep **Auto-generate schema** checked.

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

The screenshot shows the ServiceNow 'Table - New Record' form. The 'Label' field is set to 'Network Database Table'. The 'Name' field is set to 'u_network_database_table'. The 'Extends table' field is empty. The 'Application' field is set to 'Global'. The 'Create module' checkbox is checked. The 'Create mobile module' checkbox is checked. The 'Add module to menu' dropdown is set to 'Create new...'. The 'New menu name' field is set to 'Network Database Table'. Below these fields, there is a section for 'Table Columns' with a search bar and a table with columns: Column label, Type, Reference, Maxlength, Default value, and Display. The table is currently empty, with a 'Insert a new row...' button at the bottom. The form has 'Submit' and 'Cancel' buttons at the bottom left.

Step 2: Add custom fields

These fields are **custom fields** that you will manually add in the Table Columns section of your custom table.

1. Name: **u_request_number**

- **Label:** Request Number
- **Type:** String
- **Reference:** —
- **Explanation:** A unique identifier for the request. Can be filled manually or auto-generated using a Business Rule.

2. Name: **u_assignment_group**

- **Label:** Assignment Group
- **Type:** Reference
- **Reference:** Group (Group table)
- **Explanation:** Defines the team or group responsible for fulfilling the request.

3. Name: u_customer_document

- **Label:** Customer Document
- **Type:** String
- **Reference:** —
- **Explanation:** Stores a document reference or identifier related to the customer, such as an ID proof or contract reference

4. Name: u_assigned_to

- **Label:** Assigned To
- **Type:** Reference
- **Reference:** User(User table)
- **Explanation:** The specific user assigned to handle the request.

5. Name: u_device_details

- **Label:** Device Details
- **Type:** String
- **Reference:** —
- **Explanation:** Captures technical details or specifications of the device involved in the request.

6. Name: u_date_of_enquiry

- **Label:** Date of Enquiry
- **Type:** Date
- **Reference:** —

- **Explanation:** The date when the enquiry was received from the customer.

7. Name: u_customer_address

- **Label:** Customer Address
- **Type:** String
- **Reference:** —
- **Explanation:** The physical or mailing address of the customer.

8. Name: u_approval_state

- **Label:** Work Status
- **Type:** String
- **Reference:** —
- **Explanation:** Indicates the current approval or work status of the request.

9. Name: u_requested_for

- **Label:** Requested For
- **Type:** String *(Normally this should be a Reference to sys_user, but in your screenshot it's String)*
- **Reference:** — *(unless you change it to a Reference type)*
- **Explanation:** Specifies the end-user for whom the request is being made.

| Column Label | Type | Reference | Max length | Default value | Display |
|-------------------|-------------------|-----------|------------|-----------------------------------|---------|
| Customer Address | String | empty | 100 | | None |
| Customer Address | String | empty | 40 | | None |
| Class | System Class Name | empty | 50 | (select tableName.getTableName()) | None |
| Assignment Group | Reference | empty | 33 | | None |
| Contact | Case Name | empty | 40 | | None |
| Business Unit | String | empty | 40 | | None |
| Sys ID | Key ID (Sys ID) | empty | 33 | | None |
| Equipment Number | String | empty | 100 | | None |
| Spacer Instance | String | empty | 40 | | None |
| Customer Address | String | empty | 40 | | None |
| Assignment ID | Reference | empty | 33 | | None |
| Updateable | String | empty | 40 | | None |
| Name | String | empty | 40 | | None |
| Work Order | String | empty | 40 | | None |
| Work Order | String | empty | 40 | | None |
| Customer Location | String | empty | 40 | | None |
| Request Number | String | empty | 40 | | None |
| Parent Display | String | empty | 40 | | None |
| Equipment Number | String | empty | 40 | | None |
| Data ID | String | empty | 40 | | None |

To Autopopulate Database Number

Using Number Maintenance

ServiceNow has a built-in feature called **Number Maintenance** to manage auto-number sequences for any table.

1. Navigate to:
System Definition > Number Maintenance.
2. Click **New**.
3. Fill in details:
 - **Table** → select your Network Database Table.
 - **Prefix** → NET.
 - **Current Value** → 1003 (or any starting number you want).
 - **Number of Digits** → 7.
4. Save.

The screenshot shows the ServiceNow interface for creating a new record in the 'Network Database Table'. The browser address bar shows the URL: `https://snc200011.service-now.com/sys/nav/4?classes=sys_rest_target%2Cnetwork_database_table%2Csys_rest_target%2Cnetwork_database_table%2C...`. The page title is 'Network Database Table - Create NET0001029'. The form contains the following fields:

- Database Number:
- Request Number:
- Created:
- Request For:
- Date of Expiry:
- Customer Address:
- Special Instructions:
- Work Status:
- Assignment Group:
- Assigned to:
- Device Details:

A 'Submit' button is located at the bottom left of the form.

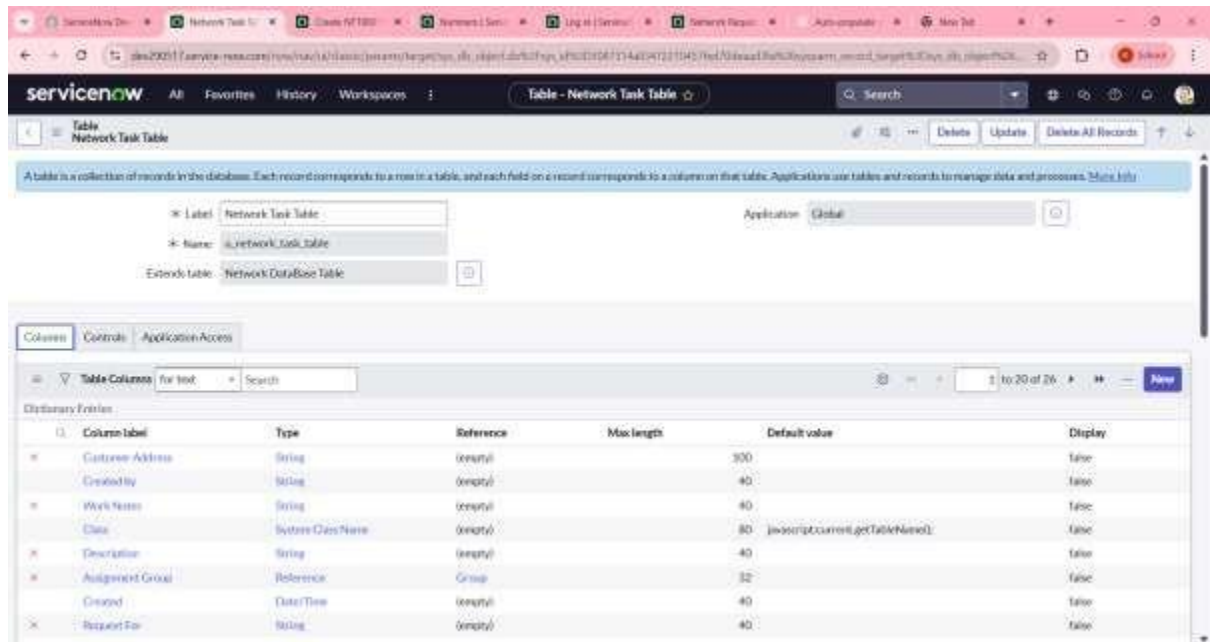
Network Task Table

Step 1: Create the Child Table (Network Task Table)

1. Navigate to:
System Definition > Tables
2. Click **New**.
3. Fill in details:
 - **Label** → Network Task Table
 - **Name** → auto-generated (u_network_task_table)
 - **Extends Table** → select **Network Database Table** (u_network_database_table)

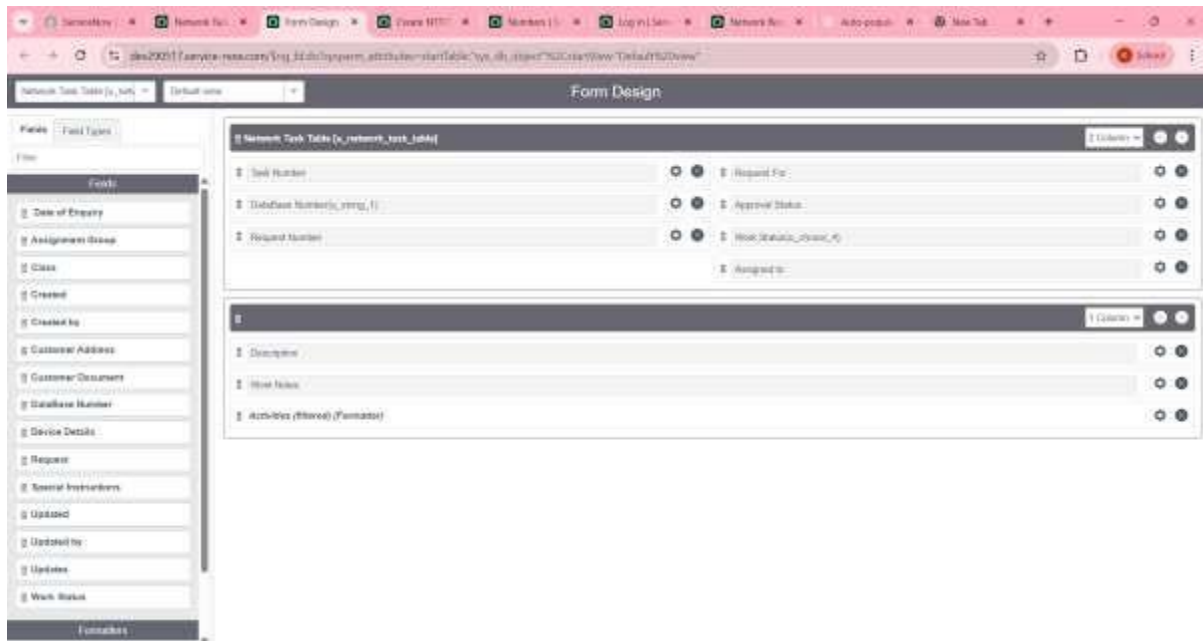
This is the important part → by choosing **Extends Table**, your Network Task Table will automatically inherit all fields from the parent.

4. Save the record.



Step 2: Verify Inherited Fields

- Open the new table (Network Task Table).
- Go to **Columns** tab.
- You'll see:
 - Fields from parent (Database Number, Request Number, Request For, etc.)
 - Plus any new fields you add specifically for tasks (Task Number, Work Status, Assigned to, etc.).



Process 3: Request Approvals Creation

The goal is to display **approval records** directly on the **Network Database table** form.

By creating a relationship between **Network Database Table** and **Approval (sysapproval_approver)**:

- We can see which approvals are associated with each record.
- We avoid searching in a separate table.
- The refineQuery ensures only relevant approvals (based on source table and document ID) are shown.

Steps to Create the Related List with Script

1. Navigate to Relationships

1. Go to **System Definition → Relationships**.
2. Click **New**.

2. Fill in the Relationship Details

- **Name** → Request Approvals

- **Applies to table** → Network Database Table *u_user_network_database+
- **Queries from table** → Approval *sysapproval_approver+
- **Active** → Checked.

3. Add the refineQuery Script

The script filters the approvals to only show records related to the current Network Database record.

```
(function refineQuery(current, parent) ,
    current.addQuery('source_table', parent.getTableName());
    current.addQuery('document_id', parent.sys_id);
-)(current, parent);
```

Script Explanation:

- source_table → Ensures only approvals linked to this specific table are fetched.
- document_id → Matches the approval record to the exact parent record.
- state filter (commented out) → Can exclude approvals not required.

4. Save and Verify

1. Click **Update**.
2. Open a **Network Database Table** record.
3. You should see the **Request Approvals** related list populated with the matching approval entries.

Steps to Add the Related List to the Form

1. Open any record from the **Network Database Table**.
2. Click the **context menu** (three dots in the top right of the form).
3. Navigate to **Configure > Related Lists**.
4. In the list of available related lists, select **Approval Request**.
5. Save the form configuration.

6. Refresh the record — you should now see the **Request Approvals** related list at the bottom of the form, displaying:

- **State**
- **Approver**
- **Comments**
- **Approval for**
- **Created**

Creation & Implementation of Flows, Actions in Flow Designer

Flow Designer in ServiceNow to automate the **Network Request** process. The flow manages the entire lifecycle of a request — from capturing catalog variables, creating a record in the Network Database, sending notifications, requesting approvals, handling logic conditions, and updating records — all without manual intervention.

This ensures:

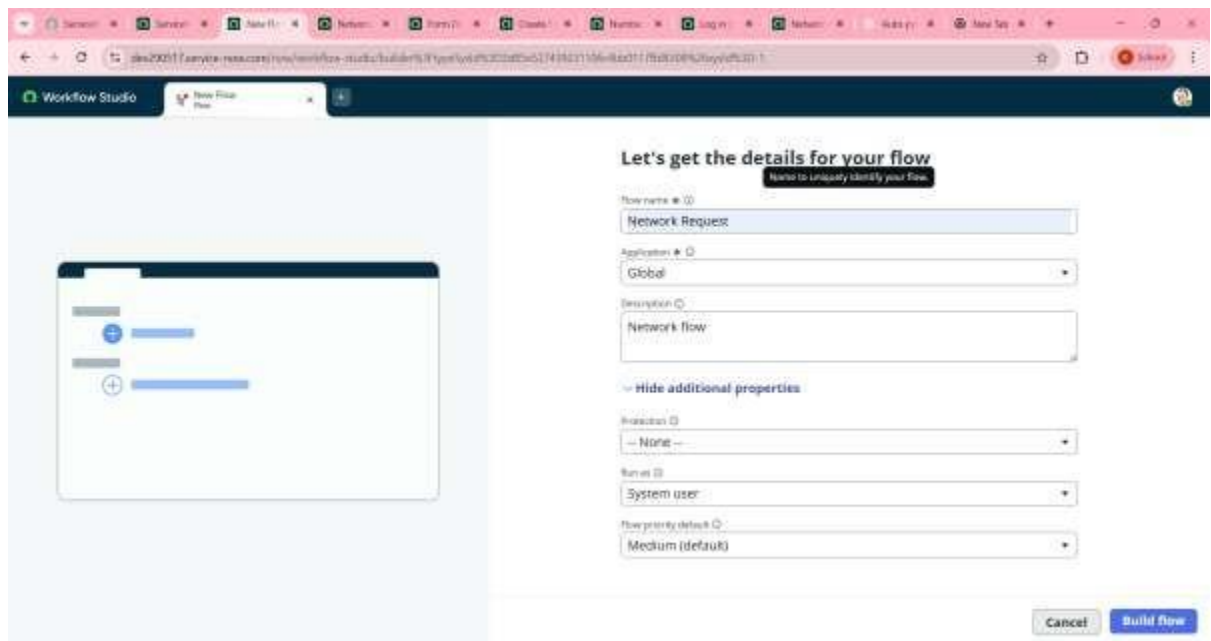
- Consistency in processing requests
- Faster execution
- Fewer manual errors
- Clear traceability of actions

Steps to Create the Flow

1. Creating the Flow

1. Navigate to **Flow Designer** home page.
2. Click **New** to create a new flow.
3. Enter:
 - **Flow Name:** Network Request
 - **Description:** *(e.g., Automates network request creation, approvals, and updates.)*

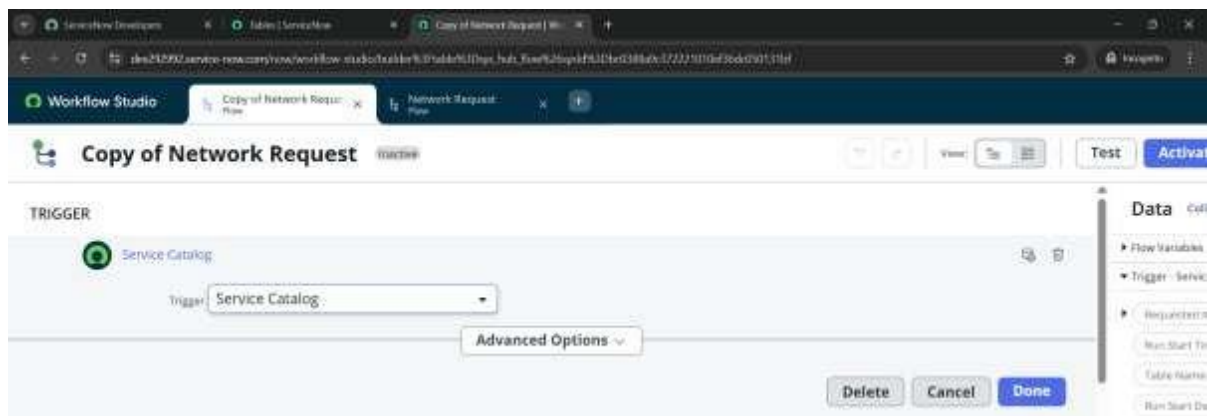
4. Click **Build Flow**.



The screenshot shows the 'Let's get the details for your flow' form in the Workflow Studio interface. The form is titled 'Let's get the details for your flow' with a subtitle 'Name to uniquely identify your flow'. It contains several input fields: 'Flow name' (Network Request), 'Application' (Global), 'Description' (Network flow), 'Environment' (None), 'Run as' (System user), and 'Flow priority default' (Medium (default)). There are 'Cancel' and 'Build flow' buttons at the bottom right.

2. Configuring the Trigger

1. Click the **(+)** icon to add a trigger.
2. Select:
 - **Trigger Type:** Application → Service Catalog
3. Click **Done**.



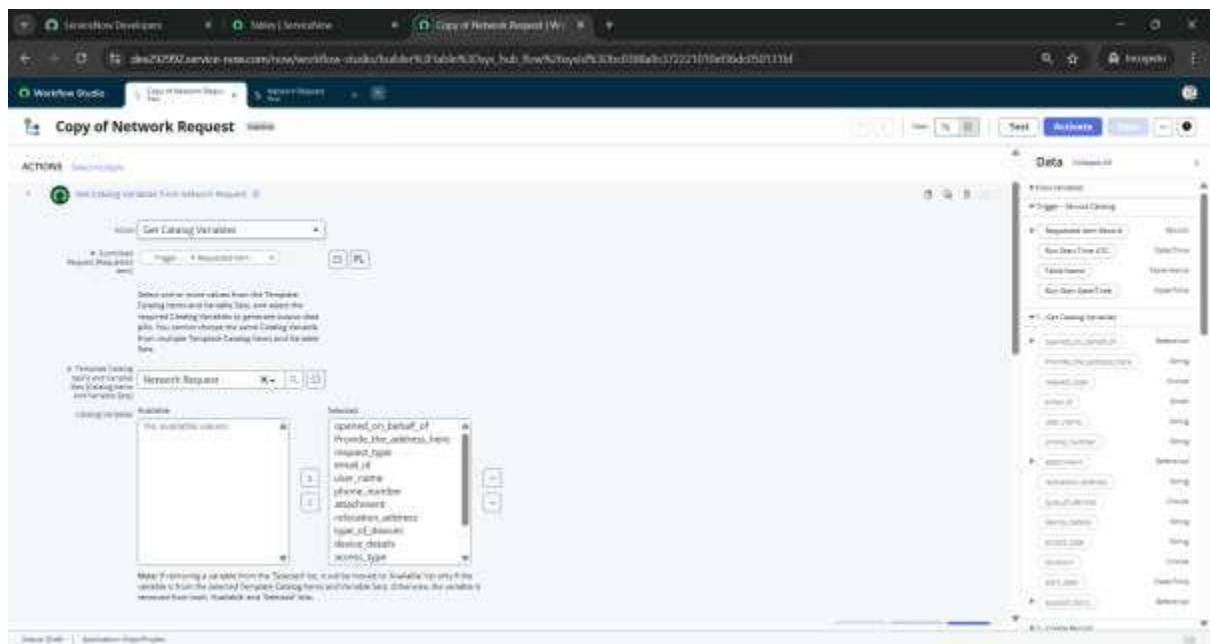
The screenshot shows the 'Copy of Network Request' flow configuration in the Workflow Studio interface. The 'TRIGGER' section is expanded, showing 'Service Catalog' as the selected trigger. There is an 'Advanced Options' dropdown button. At the bottom right, there are 'Delete', 'Cancel', and 'Done' buttons. The right sidebar shows a 'Data' section with 'Flow Variables' and 'Trigger - Service Catalog'.

3. Adding Actions

A. Get Catalog Variables

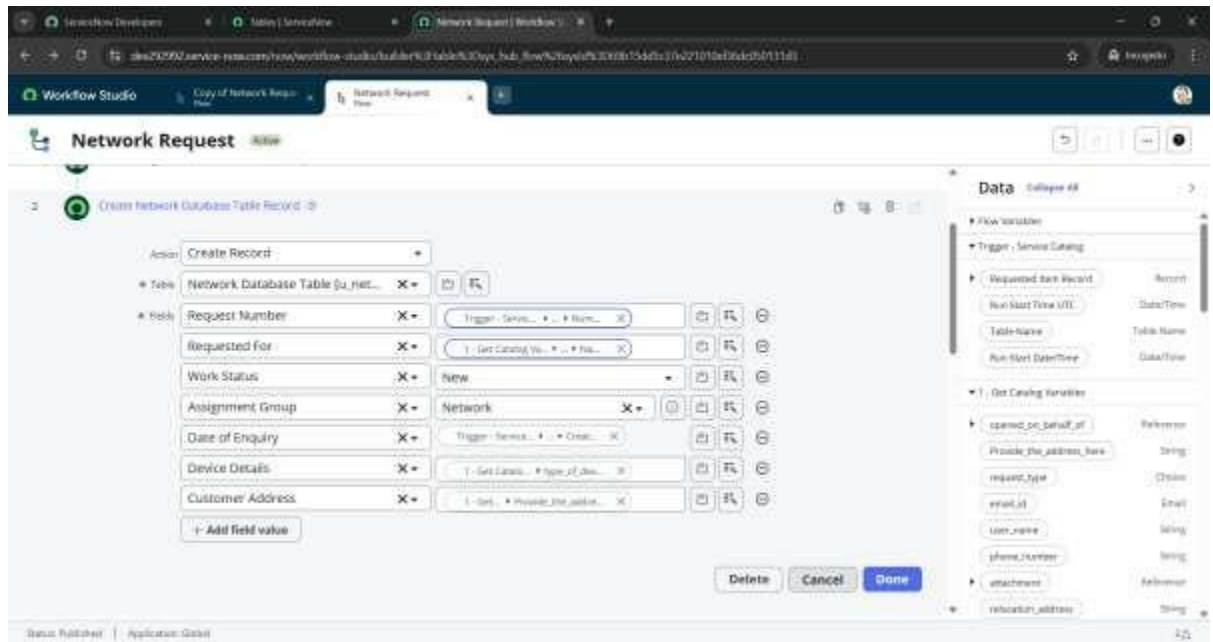
1. Click **Actions**.
2. Search for **Get Catalog Variables**.

3. Select **Get Catalog Variables**.
4. Configure **Action Inputs**:
 - **Trigger** → **Service Catalog** → **Requested Item**
5. In **Template catalog items**:
 - **Select Table**: Network Request
 - Move required variables to the **Selected** area.
6. Click **Done**.



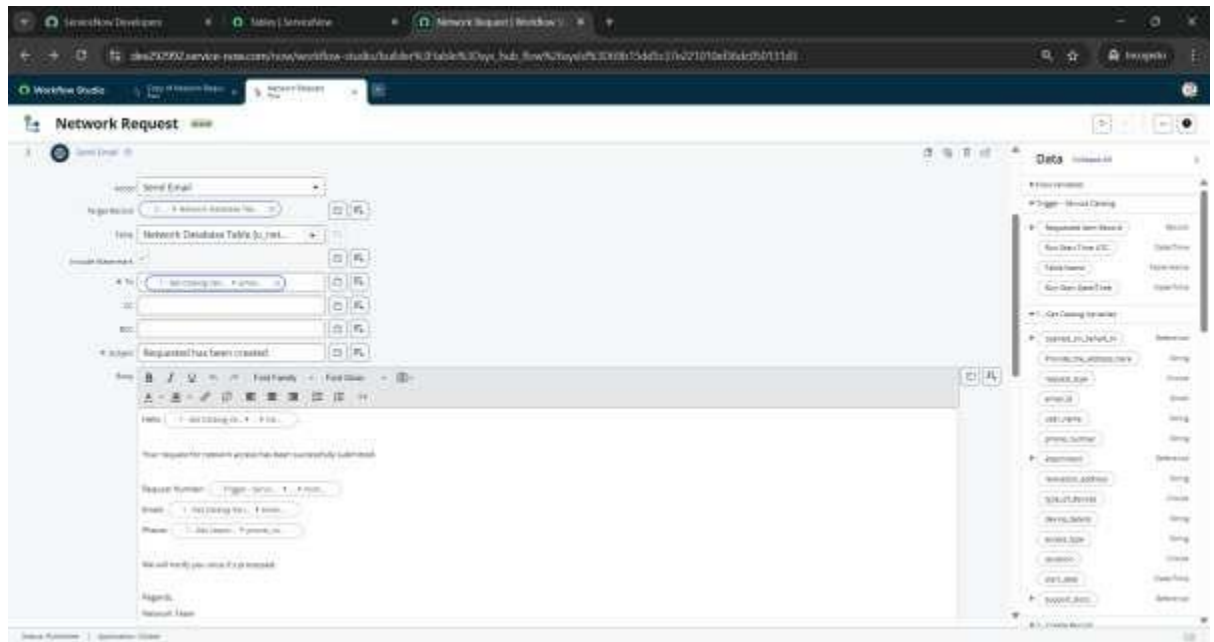
B. Create Record

1. Add a new action → **Create Record**.
2. Select **Table:** Network Database.
3. Click **Add Fields** and configure:
 - Map catalog variables to the respective table fields as per your requirements .
4. Click **Done**.



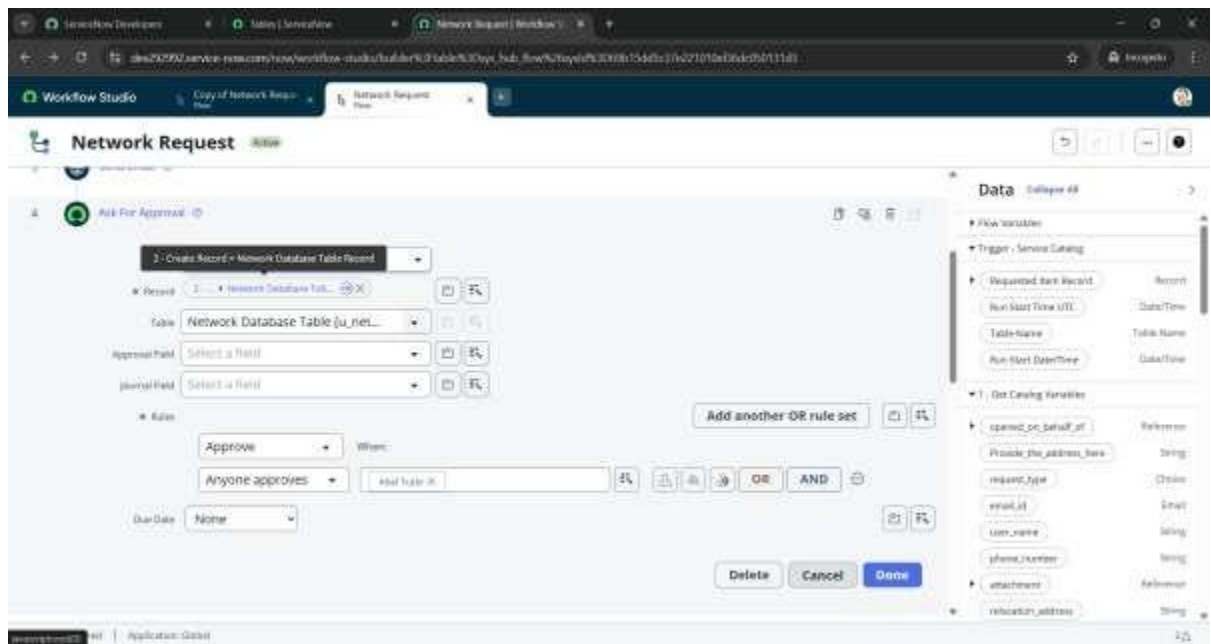
C. Send Email

1. Add a new action → **Send Email**.
2. **Target Record:** Select → **Create Record** → **Network Database Table** (auto-selected).
3. Configure:
 - **To / CC / BCC:** Static or dynamic recipients.
 - **Subject & Body:** Use variables and static text as shown in the design screenshot.
4. Click **Done**



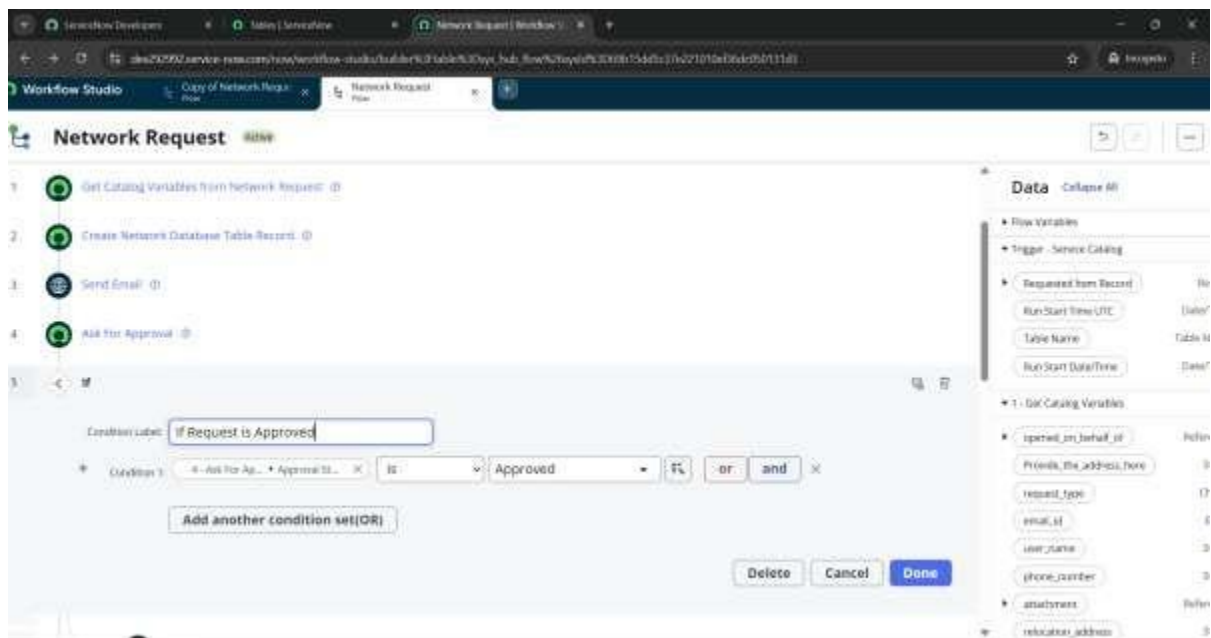
D. Ask for Approvals

1. Add a new action → **Ask for Approval**.
2. **Target Record:** Create Record → Network Database Table.
3. Configure:
 - **Approval Reason:** "Waiting for Approval".
 - **Approval Rules:** Approve, Reject, Approve/Reject.
 - **Approval Type:** Anyone approves, Everyone approves, etc. (static/dynamic assignment).
 - Here we chose **abel** tutor
4. Click **Done**



E. Flow Logic (If Condition)

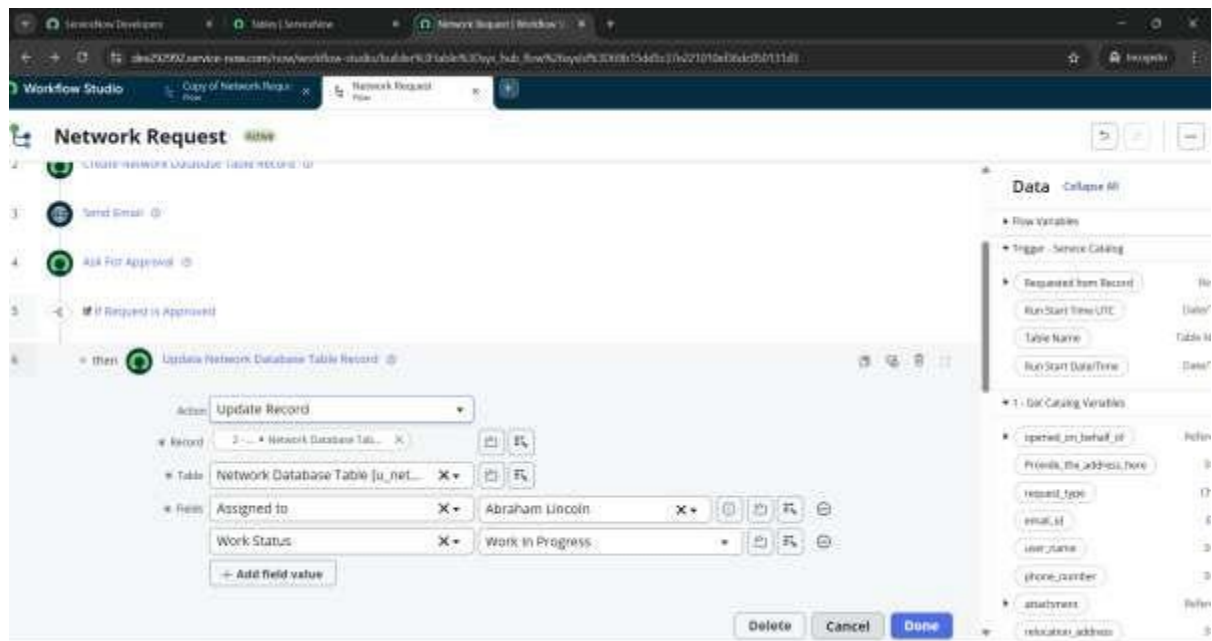
1. Add a new action → **Flow Logic** → **If Condition**.
2. Configure:
 - Condition: "Ask for approvals" state is Approved .
3. Click **Done**.



F. Update Record

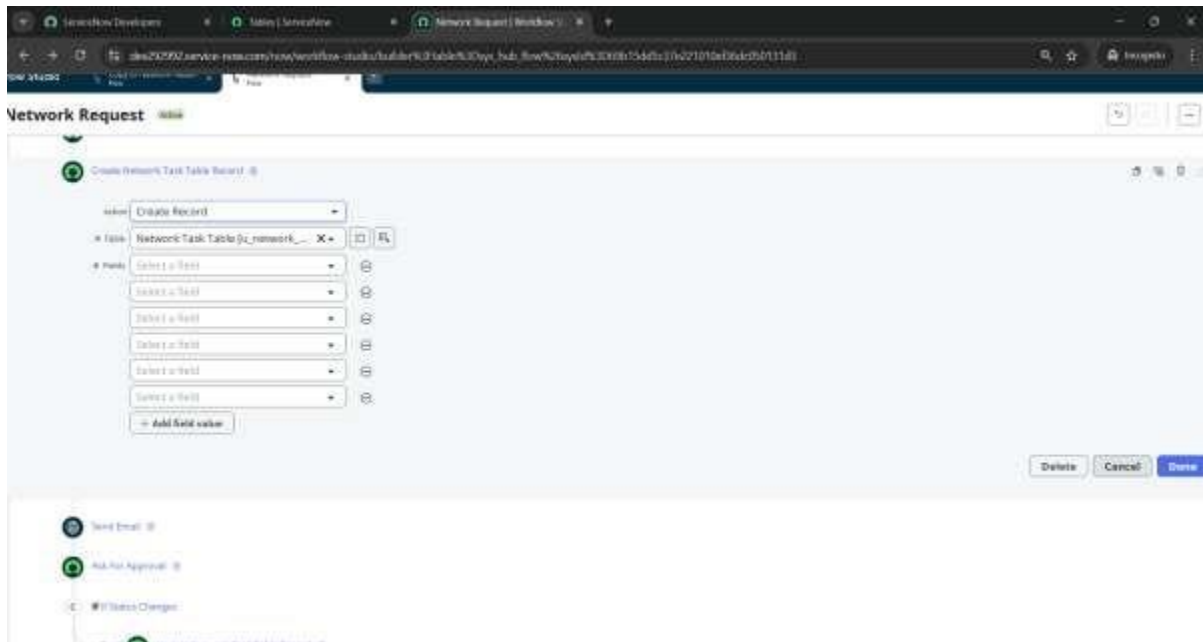
1. Add a new action → **Update Record**.

2. **Target Record:** Create Record → Network Database Table (auto-selected).
3. Configure required fields (like Assigned to -> Abraham Lincoln
Work Status -> Work in Progress).
4. Click **Done**.



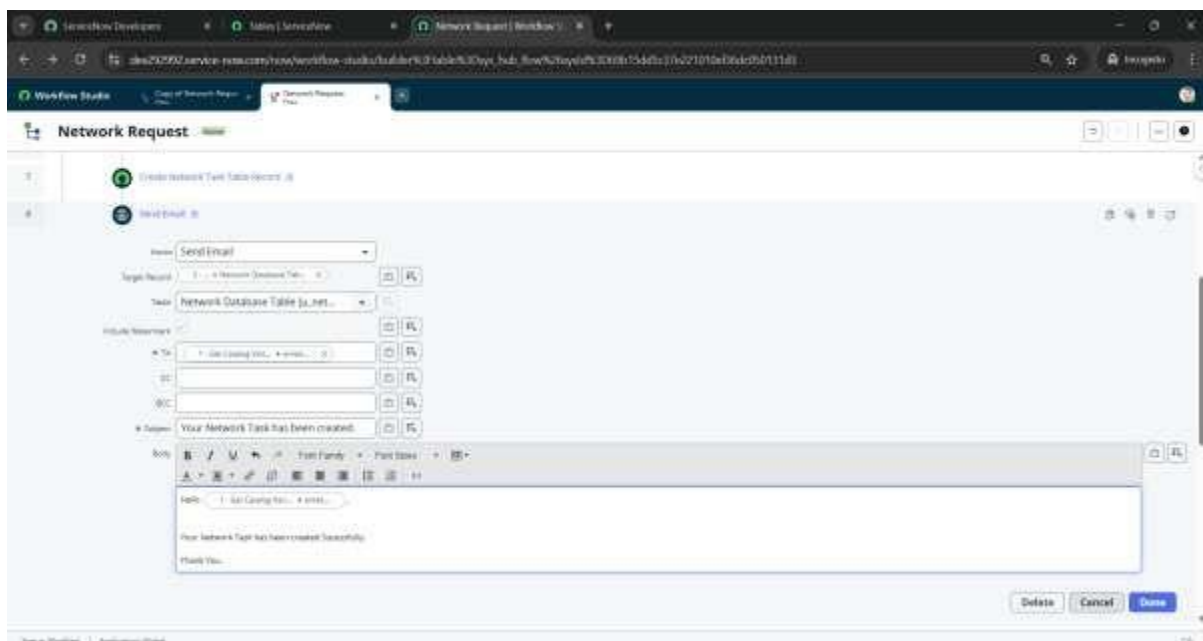
G: Create Network Task Table Record

1. Add a new action → **Create Record**.
2. Select **Table** → *Network Task Table [u_network_task]*.
3. Under **Fields**, map Service Catalog variables to the table fields:
 - **Database Number** → Auto-populated (Number Maintenance / Business Rule).
 - **Request Number** → Map from Catalog Variable (e.g., *Request Number*).
 - **Requested For** → Map from Catalog Variable (Requested For).
 - **Description** → Map from Catalog Variable (Description of request).
 - **Priority** → Map from Catalog Variable (Priority).
 - **Assignment Group** → Network Assignment Group (static or from variable).
 - **Assigned To** → Leave blank initially (will be set later after approval).
4. Click **Done**.



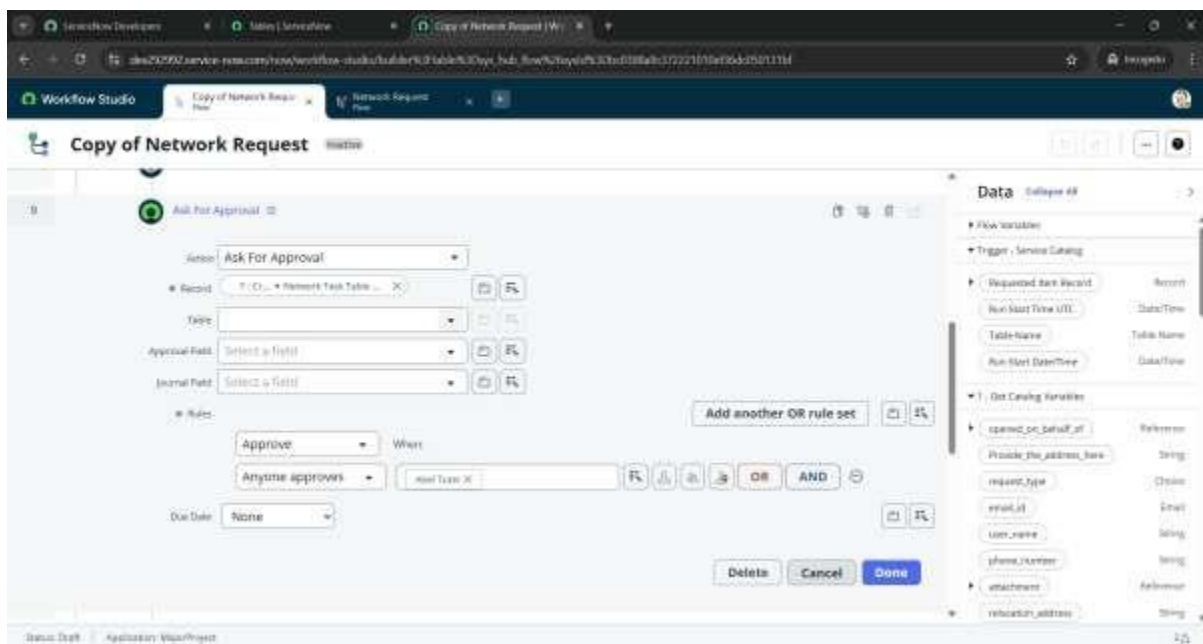
H. Send Email (Request Created)

1. Add a new action → **Send Email**.
2. Target Record → *Create Network Task Table Record*.
3. Configure:
 - **To:** Requestor / Requested For.
 - **Subject:** "Your Network Task has been created."
 - **Body:** Include Task Number, Database Number, Request Number.
4. Click **Done**.



I. Ask for Approval

1. Add a new action → **Ask For Approval**.
2. Target Record → *Network Task Table Record*.
3. Configure:
 - **Approval Reason:** "Waiting for Network Task approval".
 - **Approval Rules:** Approve / Reject.
 - **Approval Type:** Choose (e.g., *Anyone Approves*).
4. Click **Done**.



J. If Condition – Approval Status Changes

1. Add action → **If Condition**.
2. Condition → *Approval State is Approved*.
3. In the **Then branch**:

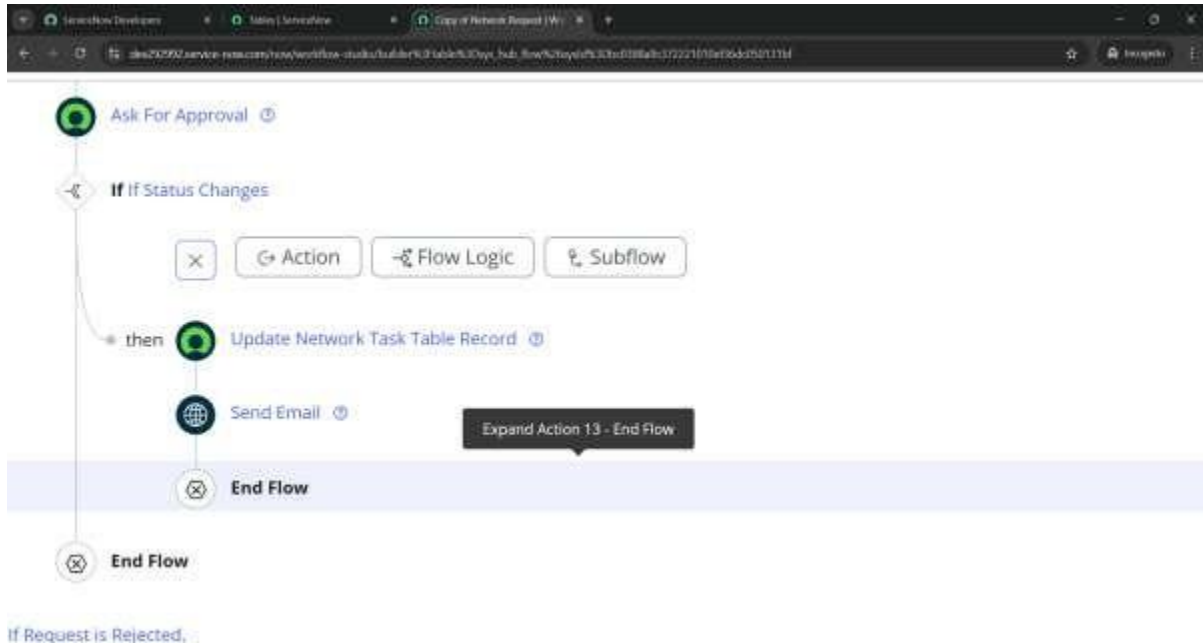
Update Record

- Target Record → *Network Task Table Record*.
- Update fields:
 - Assigned To → *Adam Ringle*.
 - Work Status → *Work in Progress*.
- Click **Done**.

Send Email (Approved)

- Add action → Send Email.
- Notify requestor that the task is approved and in progress.

(same as above)

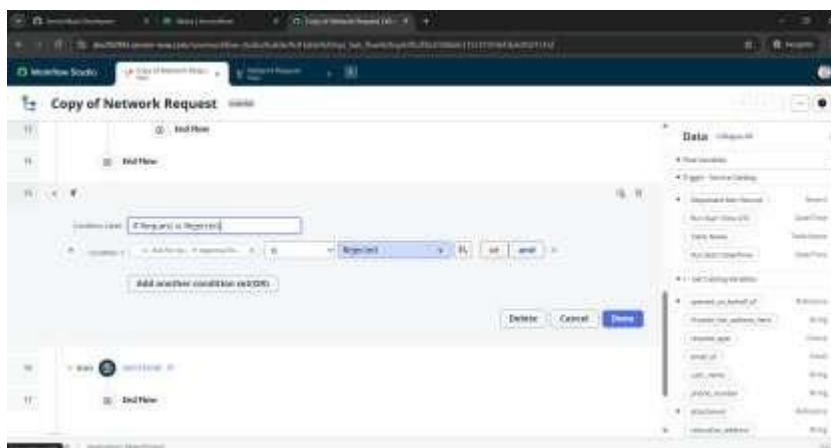


K. If Condition – Request Rejected

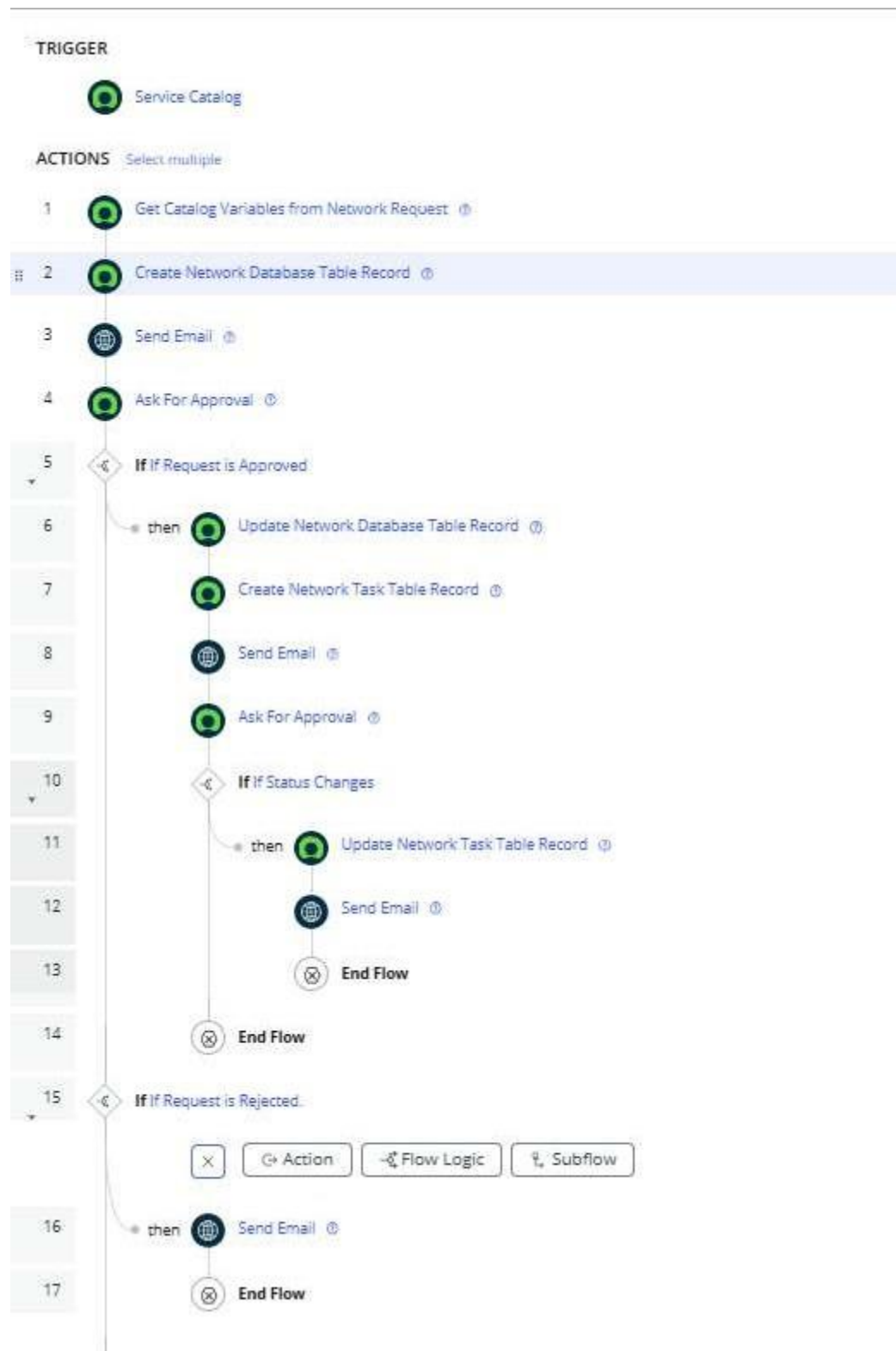
1. Add another If Condition for *Approval State is Rejected*.
2. In the Then branch:

Send Email (Rejected)

- Notify requestor that their request was rejected.
- Optionally include rejection comments.



OVERALL FLOW:



Summary

This project delivers an efficient ServiceNow-based solution for handling network service requests. By using a dedicated service catalog, automated approval workflows, and real-time notifications, it streamlines the request process for both users and technicians. The system ensures accurate request capture, faster resolution through automation, and better visibility with reporting and SLA tracking.

- **Prathipati Deepthisri**