

# Automated Network Request Management in ServiceNow

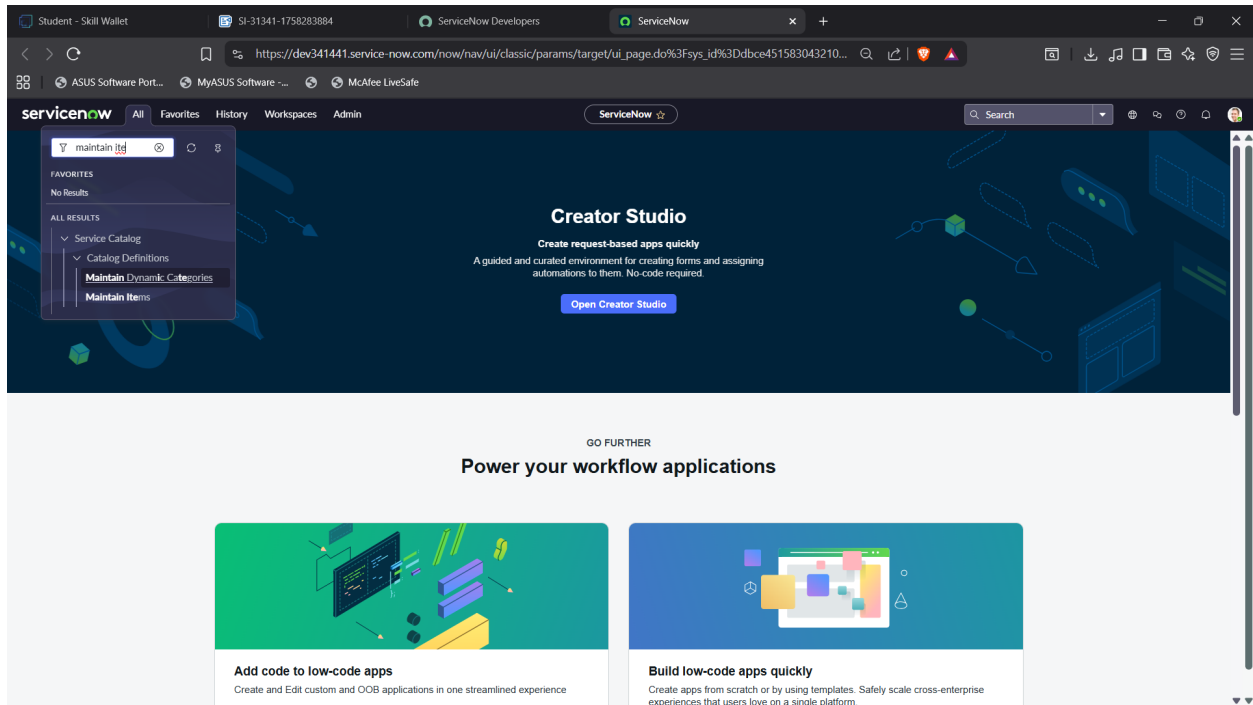
## INTRODUCTION:

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:
  - a. **Name:** NetworkRequest
  - b. **Catalog:** ServiceCatalog
  - c. **Category:** Networkand connctivity
  - d. **Short Description:** Network Request Management
3. Click on **Save**.



Variable - provide the variable label

Application: Global

Type: Single Line Text

Catalog Item: Network Request

Order: 300

Active: ☒

Mandatory: ☐

Read only: ☐

Hidden: ☐

Question

Specify the Question that explains the options available to the end user when ordering the item

\* Question: provide the variable label

\* Name: provide\_the\_variable\_label

Conversational label:

Tooltip: this will appear when cursor overed on t

Example Text: this will suggest what we need to enter on the field.

Copy Update Delete

Related Links

[Run Point Scan](#)

- e. **Tooltip:** Info shown on mouse hover
- f. **Example Text:** Placeholder help text
- g. **Mandatory / Read-Only:** As required
- h. **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 LineText	If this is a relocation, Please provide...	310

Single LineText	If this is a relocation, Please provide...	320
Container Start	Location & Devices Type	400
Single LineText	Please provide address here	410
Select Box	Type of devices	420
Single LineText	Provide device details	430
Container Start	Additional Information	500

Single Line Text      If any, Please write here      510

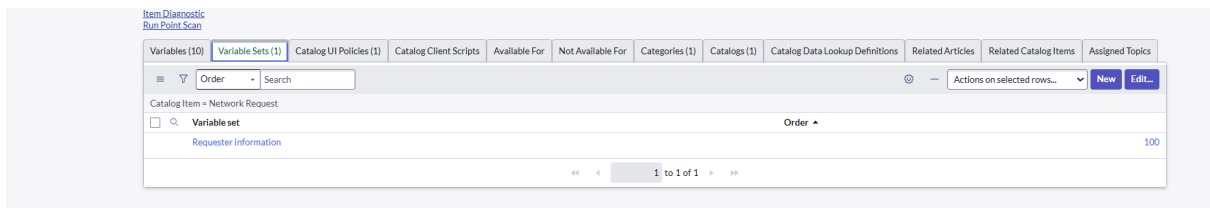
The screenshot shows the ServiceNow Catalog Builder interface for a 'Network Request' item. The top navigation bar includes 'Catalog Item - Network Request' and a search bar. Below the navigation bar, there is a section for 'Related Links' with links to 'Item Diagnostics' and 'Run Point Scan'. The main content area displays a table of variables for the 'Network Request' item. The table has columns for 'Type', 'Question', and 'Order'. The variables are listed in ascending order of their line numbers.

Type	Question	Order
Container Start	ServiceDetails	200
Multiple Choice	Is this a New connection or 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
Single Line Text	If any, Please write here	510

## Step 5: Configure VariableSet – Requester Information

### 5.1 Create Variable Set

## 1. Navigate to **Variable Sets** under ServiceCatalog.

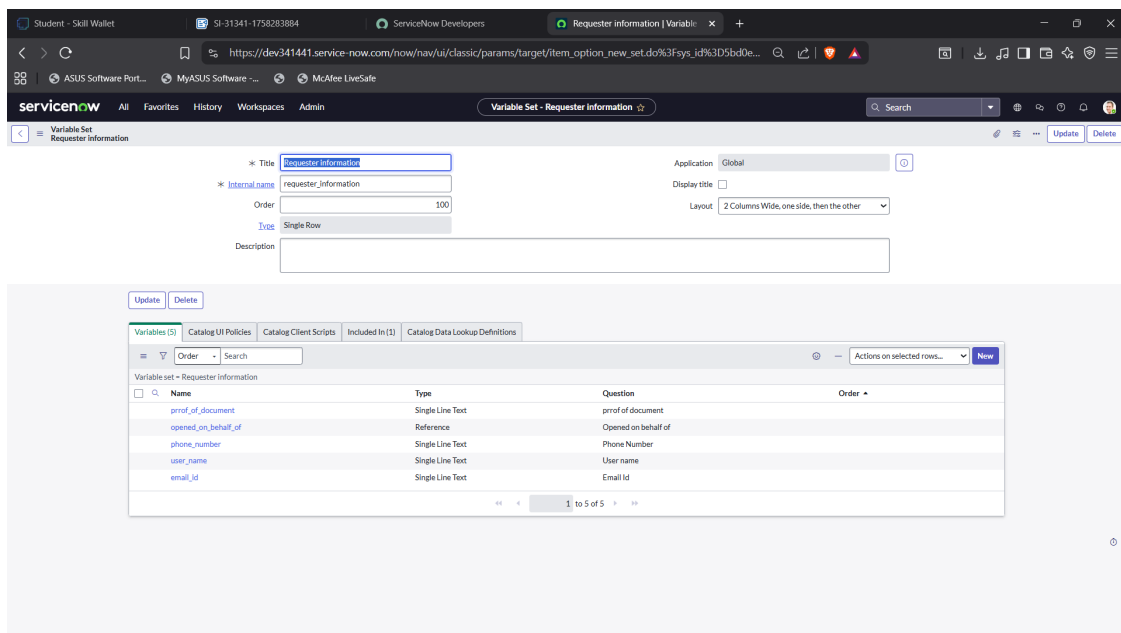


## 2. Click on **New**.

## 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**



## 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

- a. Type: **Reference**
- b. Reference to: **User \*sys\_user\***
- c. Name: **opened\_on\_behalf\_of**
- d. Order: 100
- e. This allows the requester to select a user they are raising the request for.

The screenshot shows the ServiceNow interface for configuring a variable. The browser address bar indicates the URL: [https://dev341441.service-now.com/now/nav/ui/classic/params/target/item\\_option\\_new.do%3Fsys\\_id%3D0162e68a8...](https://dev341441.service-now.com/now/nav/ui/classic/params/target/item_option_new.do%3Fsys_id%3D0162e68a8...). The ServiceNow header shows the user 'Student - Skill Wallet' and the application 'ServiceNow Developers'. The variable being configured is 'prrof of document'.

Configuration details for the variable 'prrof of document':

- Application: Global
- Type: Single Line Text
- Order: (empty field)
- Variable set: Requester Information
- Active: ☒
- Mandatory: ☐
- Read only: ☐
- Hidden: ☐

The 'Question' tab is selected, showing the following fields:

- \* Question: prrof of document
- \* Name: prrof\_of\_document
- Conversational label: (empty field)
- Tooltip: (empty field)
- Example Text: (empty field)

Buttons for 'Copy', 'Update', and 'Delete' are visible. Below the form, there is a 'Related Links' section with a link to 'Run Point Scan'.

### 2. Email ID

- a. Type: **Single Line Text**

- b. Name: email\_id
- c. Order: 200
- d. This will be auto-filled based on the user selected in "Opened on behalf of".
- e. You can use a script or dot-walking to populate the email field.

### **3. User Name**

- a. Type: **Single Line Text**
- b. Name: user\_name
- c. Order: 300
- d. This will also be auto-populated based on the user selected.
- e. Fetch the full name from the User table.

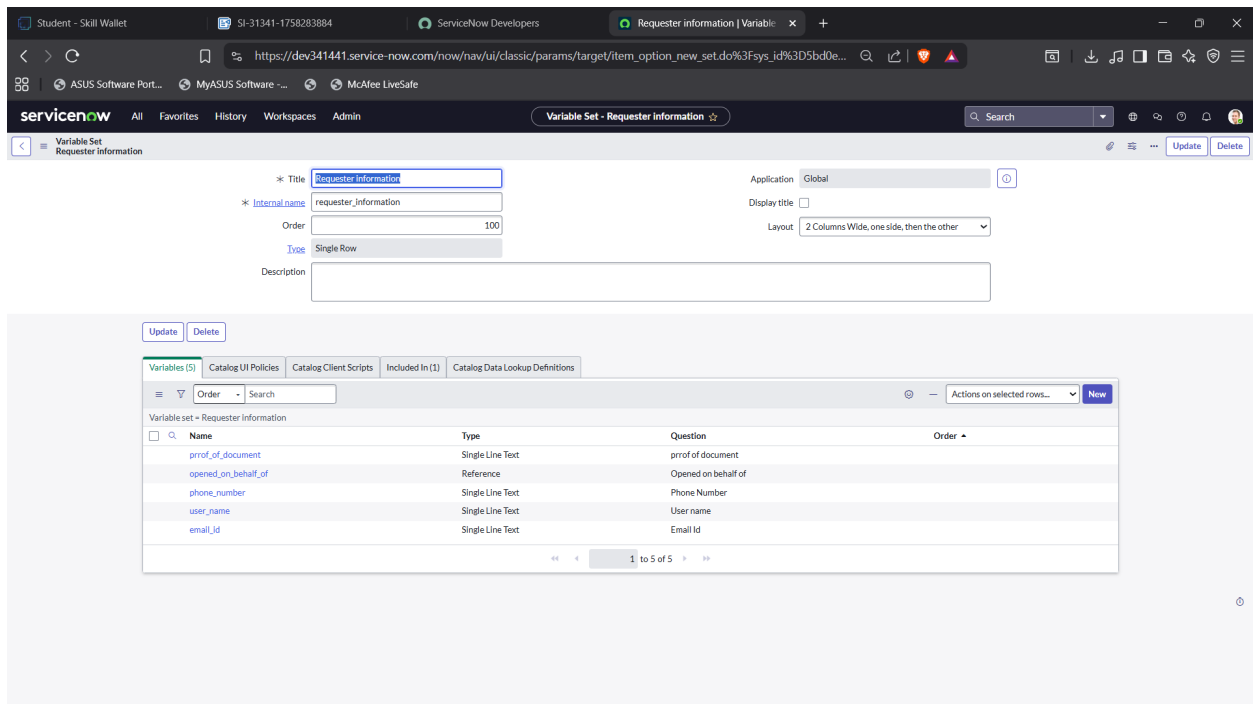
### **4. Phone Number**

- a. Type: **Single Line Text**
- b. Name: phone\_number
- c. Order: 400
- d. Same as above, it can be fetched using dot-walking or client script.

### **5. Proof of Document**

- a. Type: **Attachment**
- b. Name: proof\_of\_document
- c. Order: 500
- d. 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 VariableSet

- Navigate to: **Service Catalog> Catalog VariableSets**
- Open your variableset: **Requester Information**

### 2. Create a Catalog ClientScript

- Navigate to: **Service Catalog> Catalog ClientScripts**

- b. Click **New**
- c. Fill in details:
  - i. **Name:** Auto PopulateUser Info
  - ii. **Applies to:** CatalogItem
  - iii. **Variable Set:** *SelectRequester Information*
  - iv. **UI Type:** All
  - v. **Type:** onChange

The screenshot shows the ServiceNow interface for configuring a Catalog Client Script. The title bar indicates the script is named 'Auto populate user fields'. The configuration section includes:

- Name:** Auto populate user fields
- Applies to:** A Variable Set
- Active:** ☒
- UI Type:** All
- Application:** Global
- Type:** onChange
- Variable set:** Requester information
- Variable name:** opened\_on\_behalf\_of
- Applies on:** Catalog Item view, Requested Item, Catalog Task, Target Record (all checked)

The script editor contains the following JavaScript code:

```

1 function onChange(control, oldValue, newValue, isLoading) {
2     // Prevent execution on form load or when value is cleared
3     if (!isLoading) {
4         // Optionally clear the field if opened on behalf of is cleared
5         g_form.setValue('email_id', '');
6         g_form.setValue('user_name', '');
7         g_form.setValue('phone_number', '');
8         return;
9     }
10
11     // Get referenced user details from open user
12     g_form.setValue('opened_on_behalf_of', function(user) {

```

### 3. Configure the ScriptFields

- a. **Variable name:** opened\_on\_behalf\_of
- b. Script:**

Applies on Catalog Tests ☒  
 Applies on Target Record ☒

```

1 function onChange(event, oldValue, newValue, isLoading) {
2     // Prevent execution on form load or when value is cleared
3     if (!isLoading || !oldValue) {
4         // Optionally clear the fields if opened on behalf of is cleared
5         g_form.setValue('email_id', '');
6         g_form.setValue('user_name', '');
7         g_form.setValue('phone_number', '');
8         return;
9     }
10
11     // Get referenced user details from nps_user
12     g_form.getReference('opened_on_behalf_of', function(user) {
13         if (user) {
14             g_form.setValue('email_id', user.email || '');
15             g_form.setValue('user_name', user.name || '');
16             g_form.setValue('phone_number', user.mobile_phone || ''); // updated line
17         } else {
18             // fallback in case user couldn't be retrieved
19             g_form.setValue('email_id', '');
20             g_form.setValue('user_name', '');
21             g_form.setValue('phone_number', '');
22         }
23     });
24 }
  
```

## Step 6: Catalog UI Policy Configuration

**Goal:** Show " Provideddevice details here " fieldwhen **Types of Devices= Others**.

1. Navigate to the **Network Request** catalog item.
2. In the relatedlist, go to **Catalog UI Policies** → Click **New**.
3. Fill in:
  - a. **Applies to:** Catalog Item
  - b. **Catalog Item:** Network Request
  - c. **Condition:** Types of devices is Others
4. Click **Save**.
5. In the relatedlist, click **New** under **UI Policy Actions**.
6. Set:
  - a. **Catalog Item:** Network Request

- b. **Variable name:** Provide device details here
  - c. **Visible:** True
7. Click **Update** to save policy.
8. **Test the form** to ensure the field appears based on selection.

The screenshot shows the ServiceNow interface for configuring a Catalog UI Policy. The browser address bar shows the URL: `https://dev341441.service-now.com/now/nav/ui/classic/params/target/sc_cat_item.do%3Fsys_id%3D45239d8283447...`. The page title is "Catalog UI Policy - types of device is other".

**Policy Configuration:**

- Applies to:** A Catalog Item
- Application:** Global
- Active:** ☒
- \* Catalog Item:** Network Request
- \* Short description:** types of device is other

**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](#)

**Applies on:**

- ☒ Applies on a Catalog Item view
- ☐ Applies on Catalog Tasks
- ☐ Applies on Requested Items

**Actions:**

- ☒ Apply the catalog UI policy actions when the form is loaded or when the user changes values on the form
- ☒ On load
- ☒ Reverse the effects of the catalog UI policy actions when the Conditions evaluate to false
- ☒ Reverse if false

**Buttons:** [Update](#) [Delete](#)

**Related Links:** [Run Point Scan](#)

**Catalog UI Policy Actions Table:**

UI policy = types of device is other	Name	Read only	Mandatory	Visible	Order
<input type="checkbox"/>	<a href="#">provide device details</a>	Leave alone	Leave alone	True	100

## 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:
  - a. **Label:** *Network Database Table*
  - b. **Name:** Automatically generated (or customize if needed).
  - c. Keep **Auto-generate schema** checked.
6. Click **Submit** to create the table.

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)

\* Label:  Application:  Remote Table: ☐

\* Name:

Columns Controls Application Access

Table Columns for text Search

Column label	Type	Reference	Max length	Default value	Display
Date of Enquiry	Date	(empty)	40	false	false
Work Status	String	(empty)	40	false	false
Created	Date/Time	(empty)	40	false	false
Assignment Group	Reference	Group	32	false	false
Device Details	String	(empty)	40	false	false
Updated by	String	(empty)	40	false	false
Updates	Integer	(empty)	40	false	false
Customer Address	String	(empty)	40	false	false
Sys ID	Sys ID (GUID)	(empty)	32	false	false
Updated	Date/Time	(empty)	40	false	false
Request Number	String	(empty)	40	false	false
assigned to	Reference	User	32	false	false
Customer Document	String	(empty)	40	false	false
Created by	String	(empty)	40	false	false
Requested For	String	(empty)	40	false	false

Insert a new row...

## 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**
  - a. **Label:** Request Number
  - b. **Type:** String

- c. **Reference:** —
- d. **Explanation:** A unique identifier for the request. Can be filled manually or auto-generated using a Business Rule.

## 2. Name: u\_assignment\_group

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

## 3. Name: u\_customer\_document

- a. **Label:** CustomerDocument
- b. **Type:** String
- c. **Reference:** —
- d. **Explanation:** Stores a document reference or identifier related to the customer, such as an ID proof or contract reference

## 4. Name: u\_assigned\_to

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

## 5. Name: u\_device\_details

- a. **Label:** DeviceDetails

- b. **Type:** String
- c. **Reference:** —
- d. **Explanation:** Capture technical details or specifications of the device involved in the request.

**6. Name: u\_date\_of\_enquiry**

- a. **Label:** Date of Enquiry
- b. **Type:** Date
- c. **Reference:** —
- d. **Explanation:** The date when the enquiry was received from the customer.

**7. Name: u\_customer\_address**

- a. **Label:** CustomerAddress
- b. **Type:** String
- c. **Reference:** —
- d. **Explanation:** The physical or mailing address of the customer.

**8. Name: u\_approval\_state**

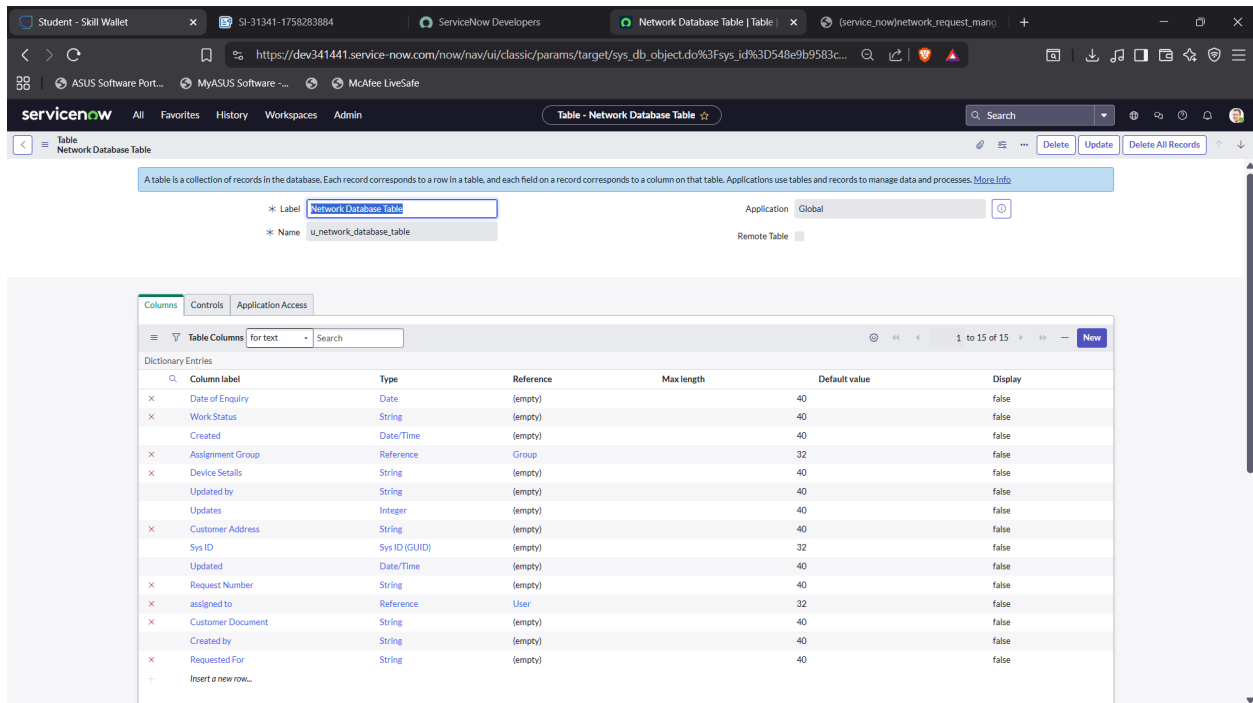
- a. **Label:** Work Status
- b. **Type:** String
- c. **Reference:** —
- d. **Explanation:** Indicates the current approval or work status of the request.

**9. Name: u\_requested\_for**

- a. **Label:** RequestedFor
- b. **Type:** String (*Normally this should be a Reference to sys\_user,*

*but in your screenshot it's String)*

- d. **Explanation:** Specifies the end-user for whom the request is being made.



## 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:
  - a. **Table** → select your Network Database Table.
  - b. **Prefix** → NET.



- c. **Current Value** → 1003 (or any starting number you want).
  - d. **Number of Digits** → 7.
4. Save.

The screenshot shows the ServiceNow interface for creating a new record in the 'Network Database Table'. The browser tabs at the top include 'ServiceNow Dev', 'NET1 Number', 'Create NET100', 'Network Database Table', 'Log in (ServiceNow)', 'Network Request', 'Auto-configure', and 'New Tab'. The URL bar shows a path to the 'Network Database Table' form. The form itself has a header 'Network Database Table - Create NET0001029' and a 'Submit' button on the right. The form fields are organized into two columns. The left column contains: 'Database Number' (filled with 'NET0001029'), 'Request Number', 'Created', 'Request For', 'Date of Expiry', 'Customer Address', and 'Special Instructions'. The right column contains: 'Work Status' (filled with 'New'), 'Assignment Group', 'Assigned to', and 'Device Details'. A 'Submit' button is located at the bottom left of the form.

## Network Task Table

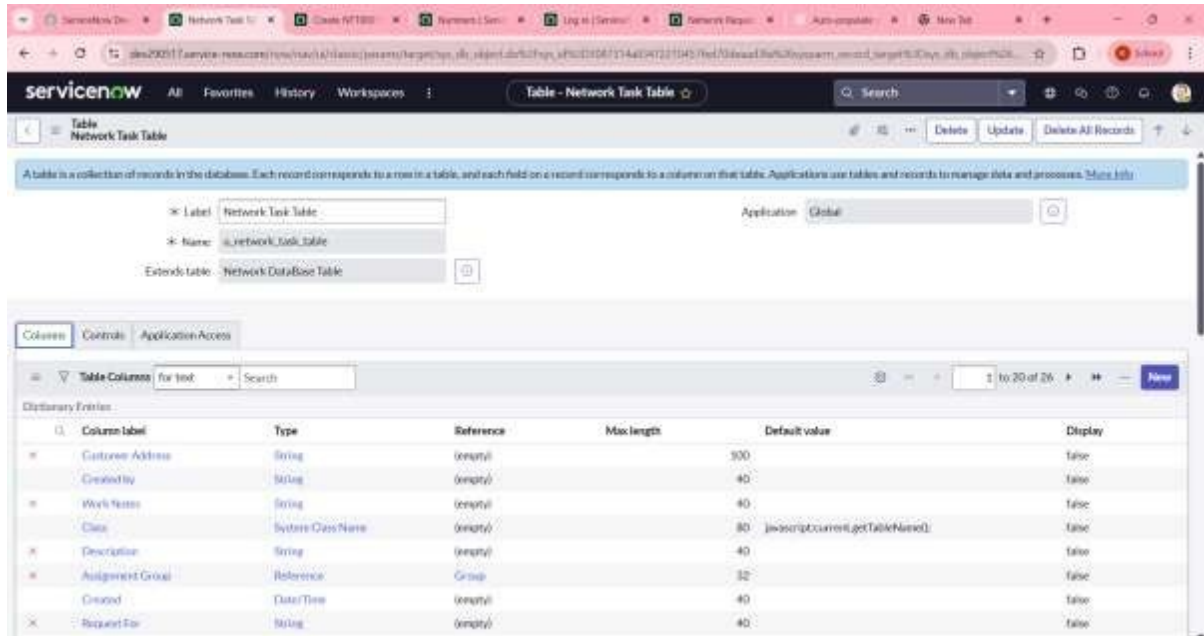
### Step 1: Create the Child Table (NetworkTask Table)

1. Navigate to:  
**System Definition > Tables**
2. Click **New**.
3. Fill in details:
  - a. **Label** → NetworkTask Table
  - b. **Name** → auto-generated (u\_network\_task\_table)
  - c. **Extends Table** → select **Network DatabaseTable**  
(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 (NetworkTask 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.).

Field Name	Type	Reference	Max Length	Default Value	Display
Customer Address	String	jenkins	320		false
Created At	String	jenkins	40		false
Work Notes	String	jenkins	40		false
Date	DateTime	jenkins	30	jenkins@jenkins.com	false
Description	String	jenkins	40		false
Assigned To	Reference	jenkins	30		false
Created	DateTime	jenkins	40		false
Request For	String	jenkins	40		false
Sys ID	String	jenkins	30		false
Database Number	String	jenkins	30		false
Special Service Area	String	jenkins	40		false
Customer Address	String	jenkins	40		false
Assigned To	Reference	jenkins	30		false
Requested By	String	jenkins	40		false
Update	String	jenkins	40		false
Work Status	String	jenkins	40		false
Work Status	String	jenkins	40		false
Request Status	String	jenkins	40		false
Customer Classification	String	jenkins	40		false
Request Number	String	jenkins	40		false

### Step 3: Configure Auto Numbering for Task Table

If you want separate auto numbering for **NetworkTasks** (like NTT0001001):

1. Navigate to **System Definition > Number Maintenance**.
2. Click **New**.
3. Fill details:
  - a. **Table** → NetworkTask Table
  - b. **Prefix** → NTT
  - c. **Current Value** → 1001
  - d. **Number of Digits** → 7
4. Save.

Now each task will have a unique Task Number (NTT0001001, NTT0001002 ...).

### Step 4: Adjust the Form Layout

1. Open a record in **Network Task Table**.
2. Right-click the header → **Configure > Form Layout**.
3. Add inherited fields (Database Number, Request Number, etc.) and new fields (Task Number, Work Notes, etc.).
4. Arrange as you like.

The screenshot shows the 'Form Design' window for the 'Network Task Table'. On the left, there is a 'Fields' pane with a list of available fields including 'Date of Expiry', 'Assignment Group', 'Class', 'Created', 'Created by', 'Customer Address', 'Customer Document', 'Database Number', 'Device Details', 'Request', 'Special Instructions', 'Updated', 'Updated by', 'Updates', and 'Work Status'. The main design area on the right contains two sections. The first section, titled 'Network Task Table (sys\_network\_task\_table)', contains fields: 'Task Number', 'Database Number (sys\_req\_id)', 'Request Number', 'Request For', 'Approval Status', 'Work Status (sys\_req\_id)', and 'Assigned to'. The second section, titled 'sysapproval\_approver', contains fields: 'Description', 'Work Notes', and 'Activities (History) (Parameters)'.

### 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)**:

- a. We can see which approvals are associated with each record.
- b. We avoid searching in a separate table.
- c. 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

- d. **Name** → Request Approvals
- e. **Applies to table** → Network Database Table  
\*u\_user\_network\_database+
- f. **Queries from table** → Approval \*sysapproval\_approver+
- g. **Active** → Checked.

### 3. Add the refineQuery Script

The script filtersthe approvals to only show records relatedto 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:

- h. **source\_table** → Ensures only approvals linkedto this specifictable are fetched.
- i. **document\_id** → Matchesthe approval record to the exact parentrecord.
- j. **state filter (commented out)** → Can exclude approvalsnot 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:
  - a. **State**
  - b. **Approver**
  - c. **Comments**
  - d. **Approval for**
  - e. **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.

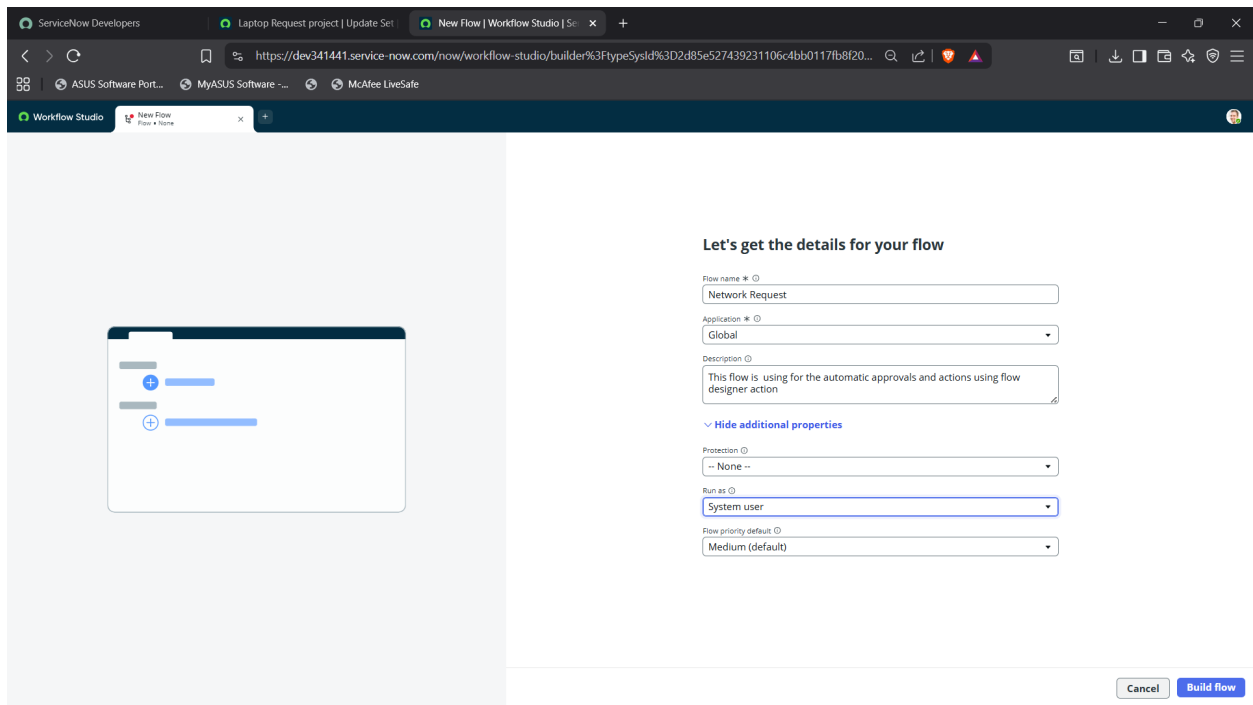
This ensures:

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

### **Steps to Create the Flow**

#### **1. Creating the Flow**

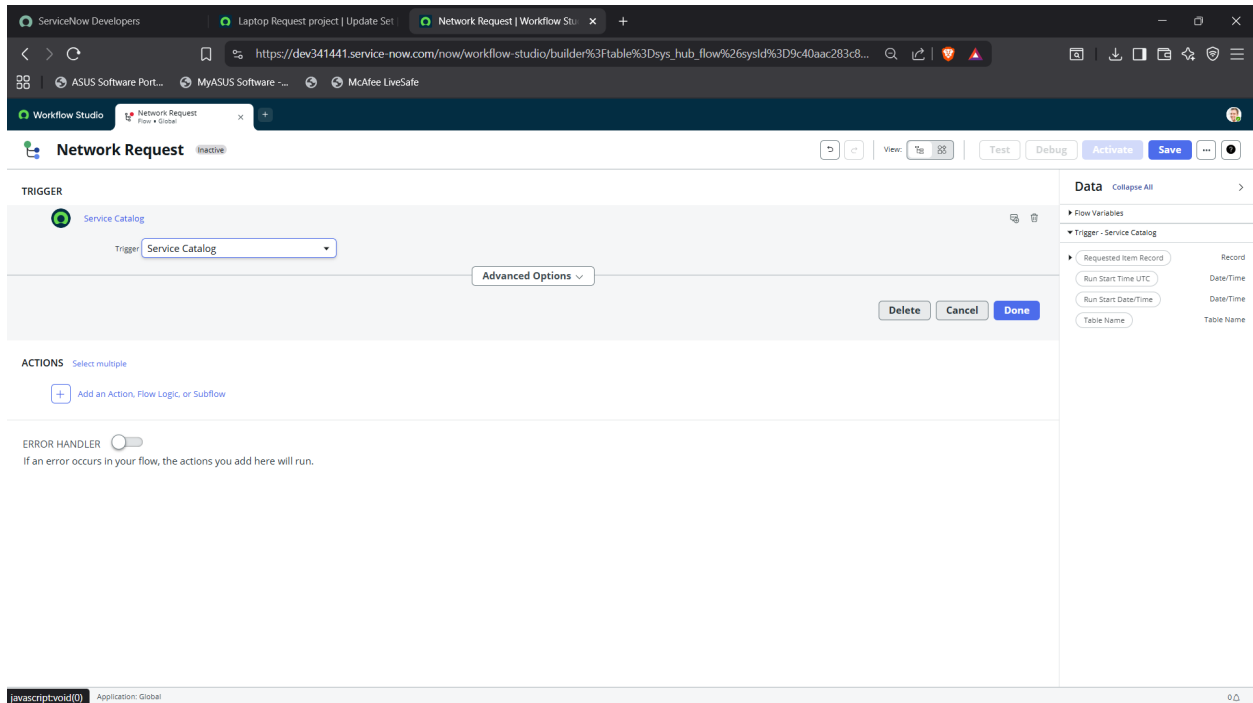
1. Navigate to **Flow Designer** home page.
2. Click **New** to create a new flow.
3. Enter:
  - a. **Flow Name:** Network Request
  - b. **Description:** *(e.g., Automates network request creation, approvals, and updates.)*
4. Click **Build Flow**.



## 2. Configuring the Trigger

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





### 3. Adding Actions

#### a. Get Catalog Variables

- i. Click **Actions**.
- ii. Search for **Get Catalog Variables**.
- iii. Select **Get Catalog Variables**.
- iv. Configure **Action Inputs**:

#### 1. Trigger → Service Catalog → Requested Item

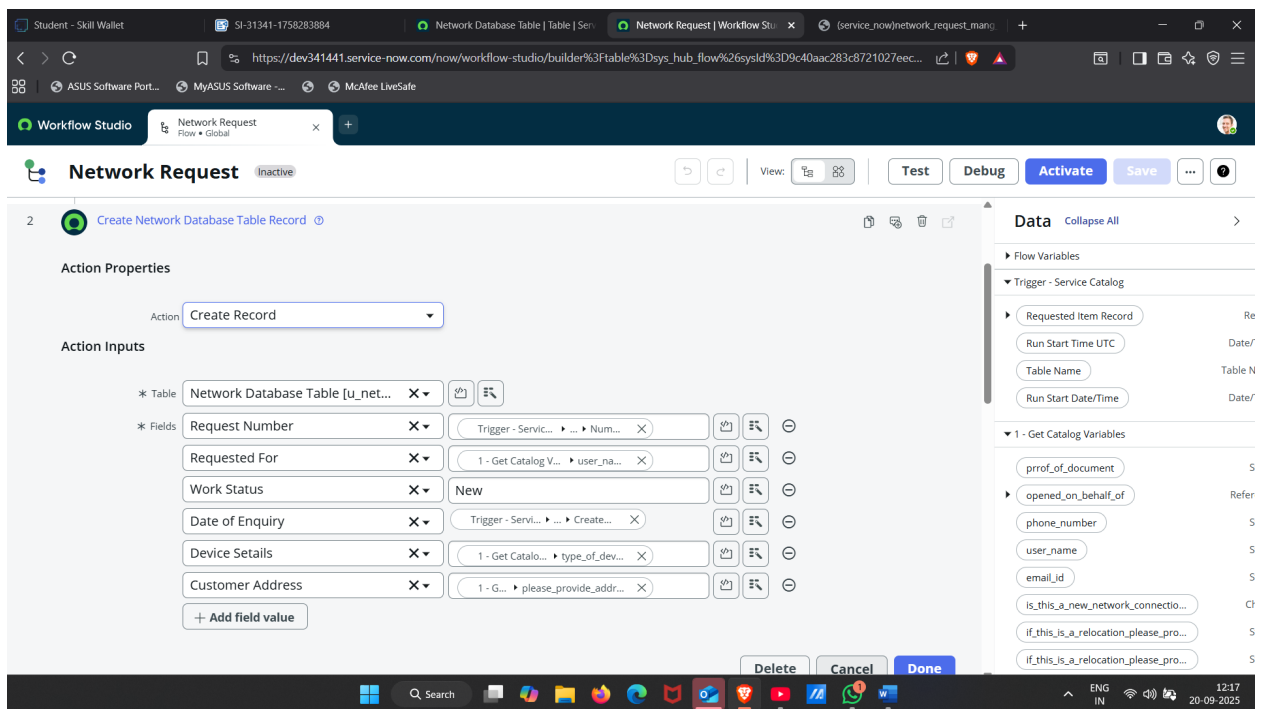
#### v. In **Template catalog items**:

1. **Select Table**: Network Request
2. Move required variables to the **Selected** area.

#### vi. Click **Done**.

## b. Create Record

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

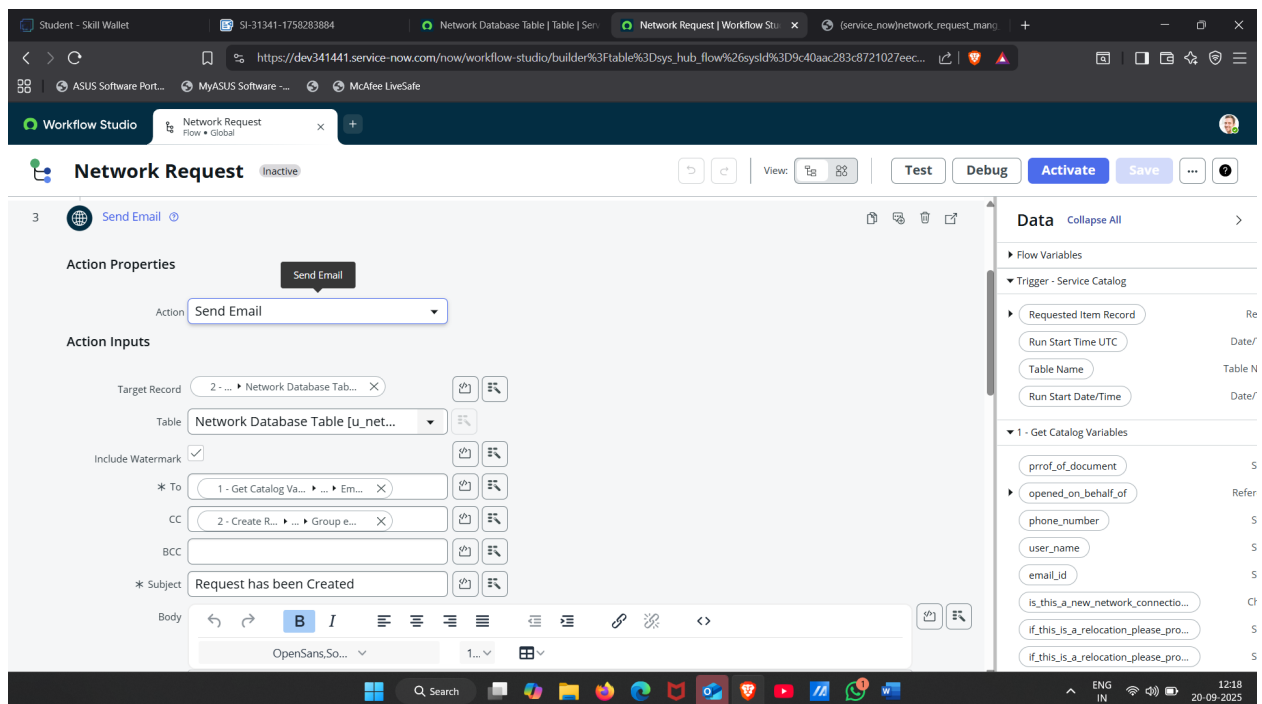


## c. Send Email

- i. Add a new action → **Send Email**.
- ii. **Target Record: Select → Create Record → Network Database Table**  
(auto-selected).
- iii. Configure:

1. **To / CC / BCC:** Static or dynamic recipients.
2. **Subject & Body:** Use variables and static text as shown in the design screenshot.

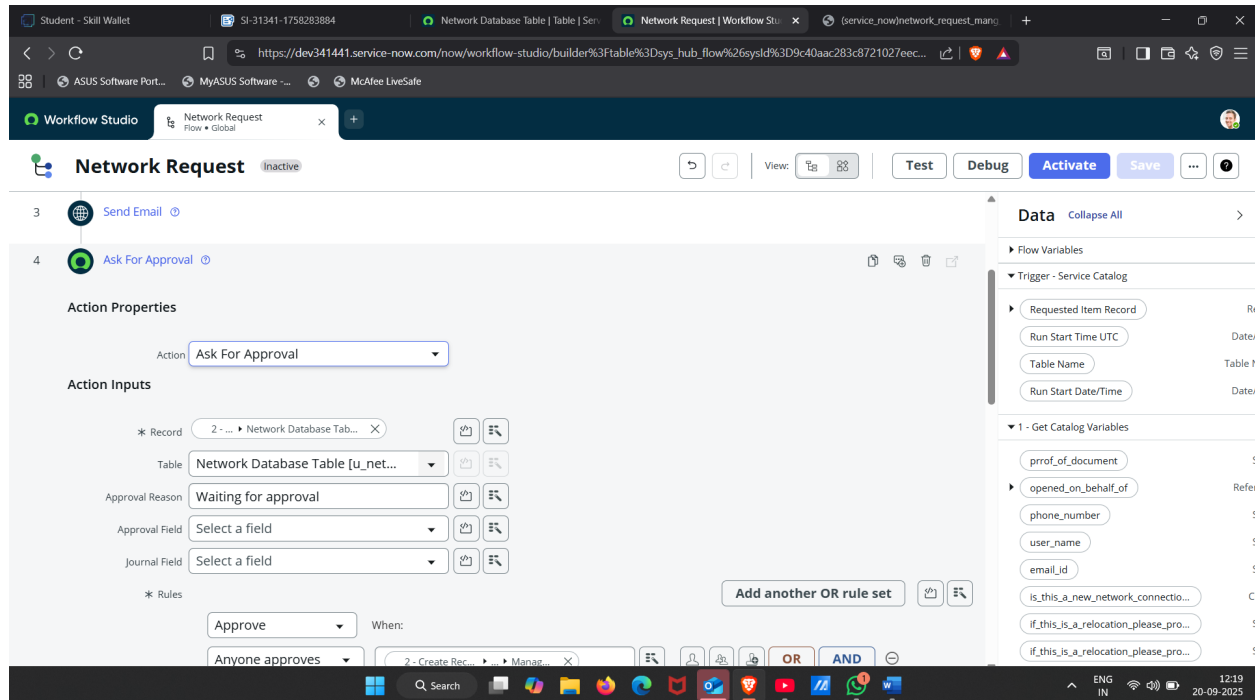
iv. Click **Done**



#### d. Ask for Approvals

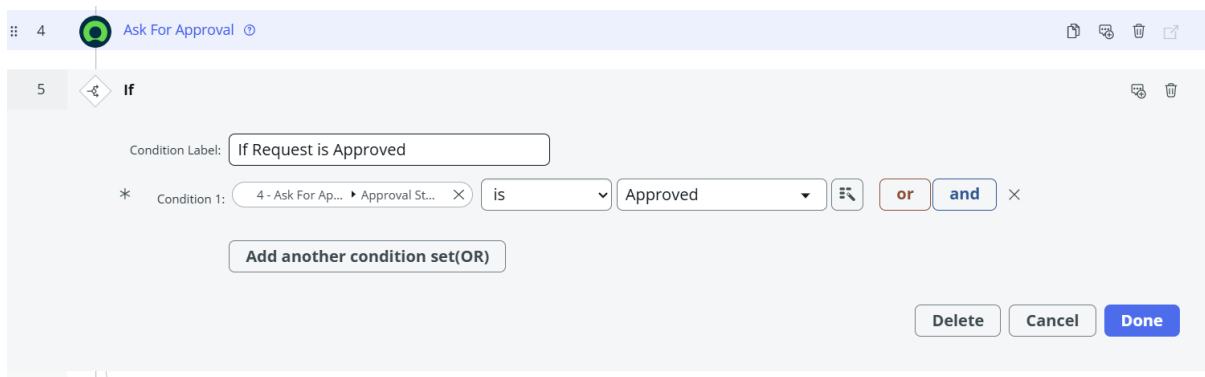
- i. Add a new action → **Ask for Approval**.
- ii. **Target Record:** Create Record → Network Database Table.
- iii. Configure:
  1. **Approval Reason:** "Waiting for Approval".
  2. **Approval Rules:** Approve, Reject, Approve/Reject.
  3. **Approval Type:** Anyone approves, Everyone approves, etc.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

#### iv. ClickDone



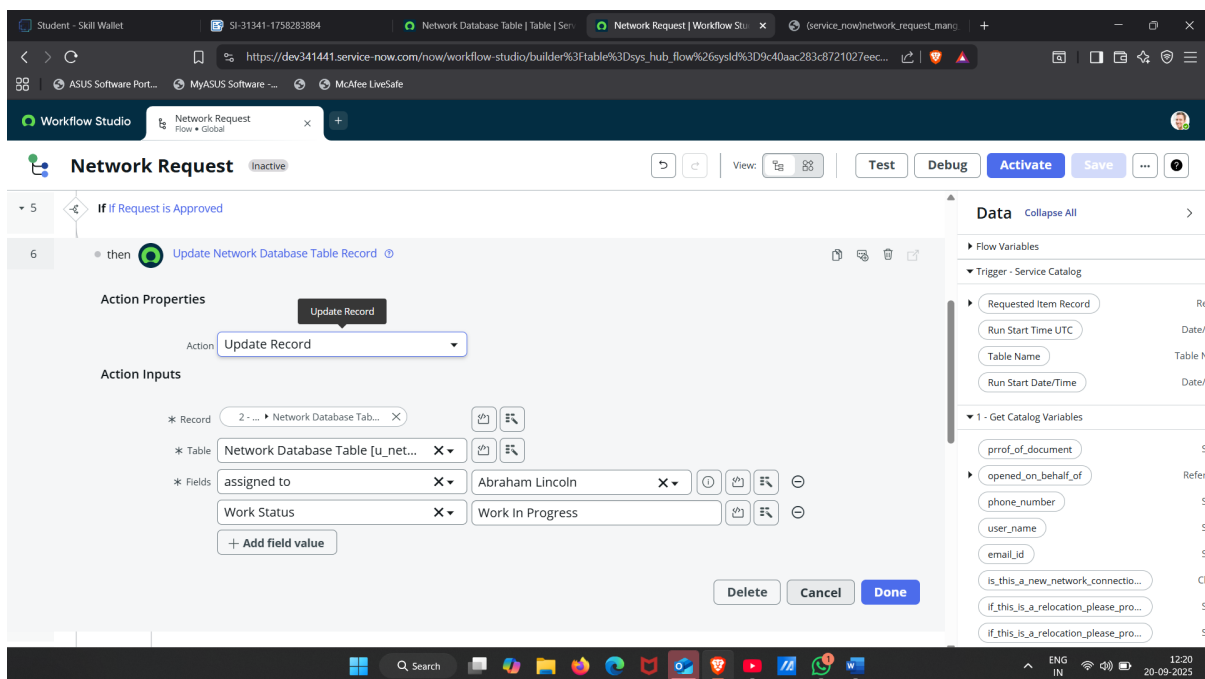
#### e. Flow Logic (If Condition)

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



## f. Update Record

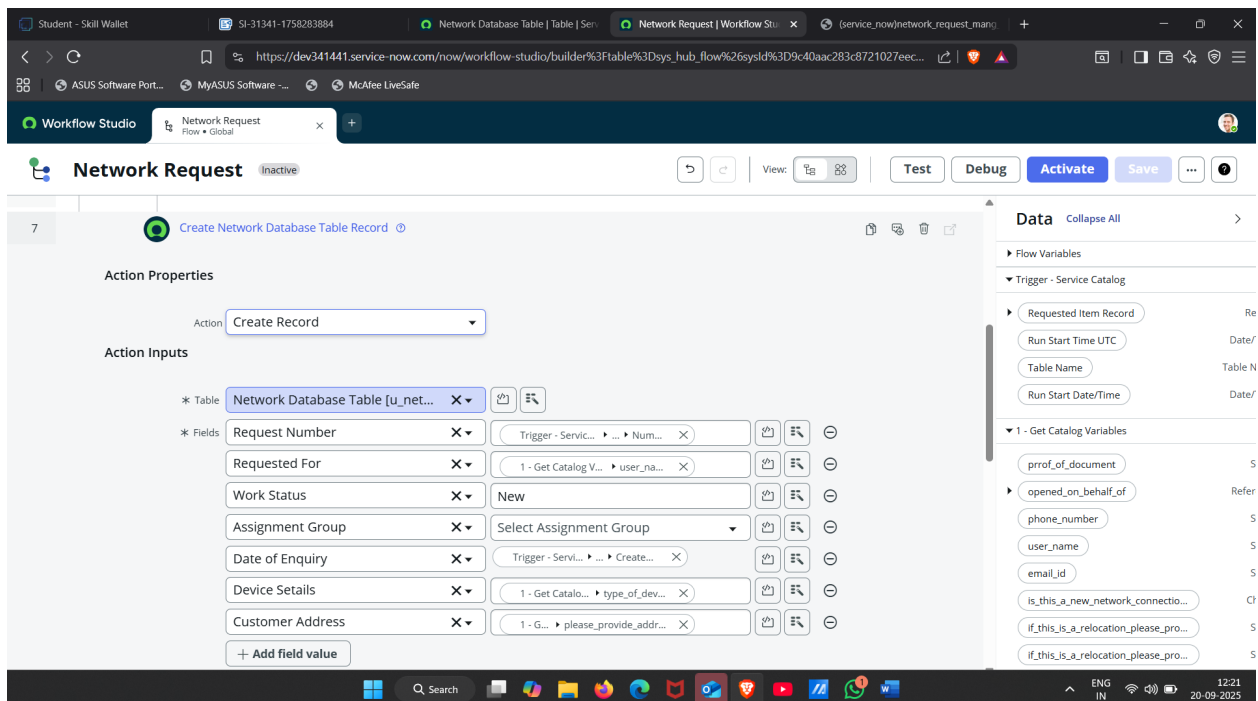
- i. Add a new action → **Update Record**.
- ii. **Target Record:** Create Record → NetworkDatabase Table (auto- selected).
- iii. Configure required fields (like Assigned to -> Abraham Lincoln Work Status -> Work in Progress).
- iv. Click **Done**.



## G: Create Network Task Table Record

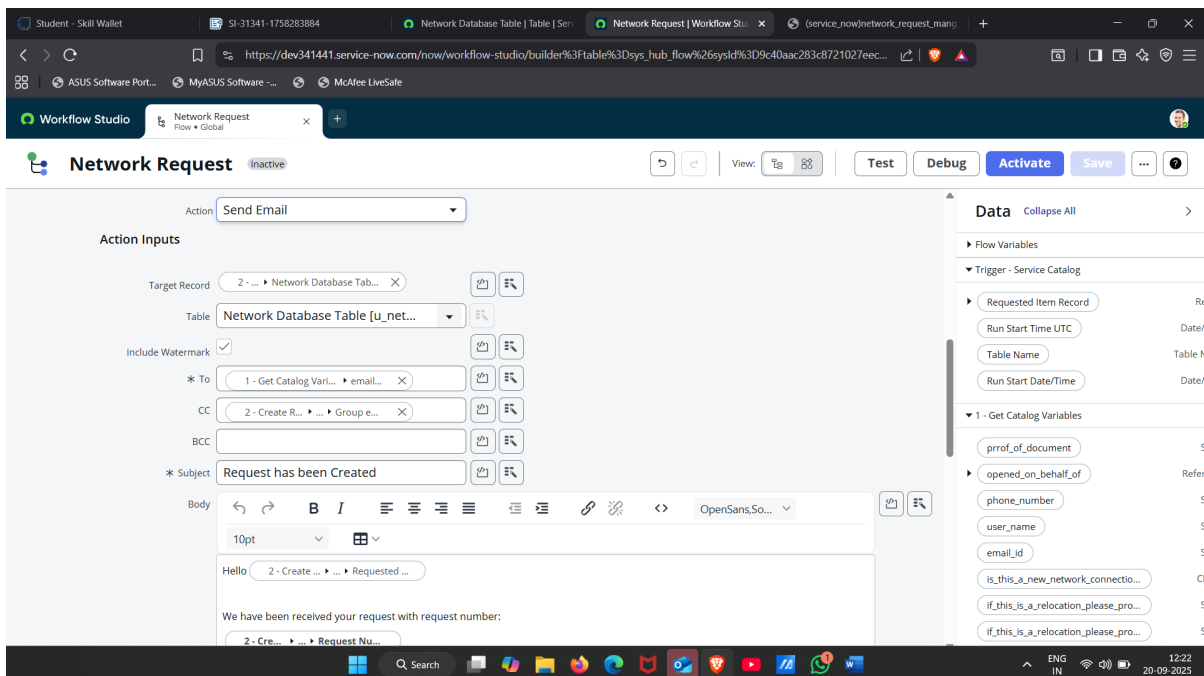
1. Add a new action → **Create Record**.
2. Select **Table** → *Network Task Table [u\_network\_task]*.

3. Under **Fields**, map Service Catalog variables to the table fields:
  - a. **Database Number** → Auto-populated (Number Maintenance / Business Rule).
  - b. **Request Number** → Map from CatalogVariable (e.g., *Request Number*).
  - c. **Requested For** → Map from CatalogVariable (Requested For).
  - d. **Description** → Map from CatalogVariable (Description of request).
  - e. **Priority** → Map from CatalogVariable (Priority).
  - f. **Assignment Group** → Network Assignment Group (static or from variable).
  - g. **Assigned To** → Leave blank initially (will be set later after approval).
4. Click **Done**.



## H. SendEmail (Request Created)

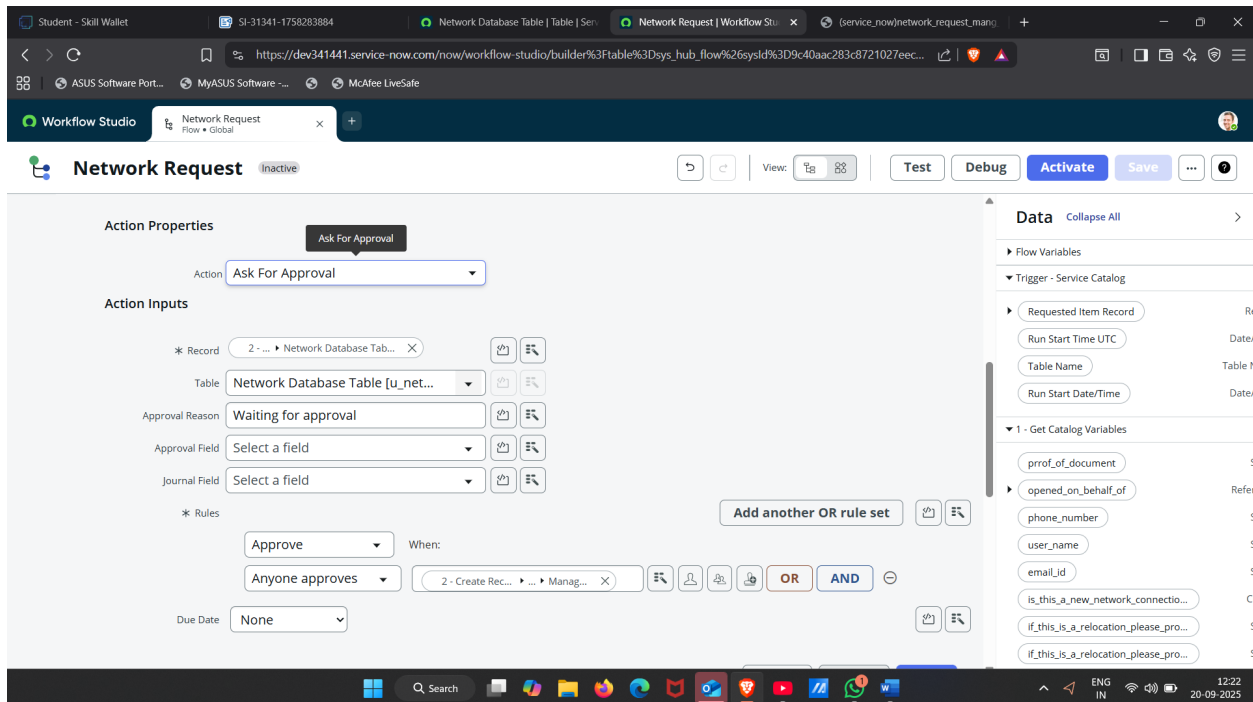
1. Add a new action → **Send Email**.
2. Target Record → *Create Network Task Table Record*.
3. Configure:
  - a. **To:** Requestor / RequestedFor.
  - b. **Subject:** "Your Network Task has been created."
  - c. **Body:** Include Task Number, Database Number, RequestNumber.
4. Click **Done**.



## I. Ask for Approval

1. Add a new action → **Ask For Approval**.
2. Target Record → *Network Task TableRecord*.
3. Configure:
  - a. **Approval Reason:** "Waiting for Network Task approval".

- b. **Approval Rules:** Approve / Reject.
  - c. **Approval Type:** Choose (e.g., *Anyone Approves*).
4. Click **Done**.



## J.If Condition – Approval StatusChanges

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

### Update Record

- a. Target Record → *Network Task TableRecord*.
- b. Update fields:
  - i. Assigned To → *Adam Ringle*.
  - ii. Work Status → *Work in Progress*.



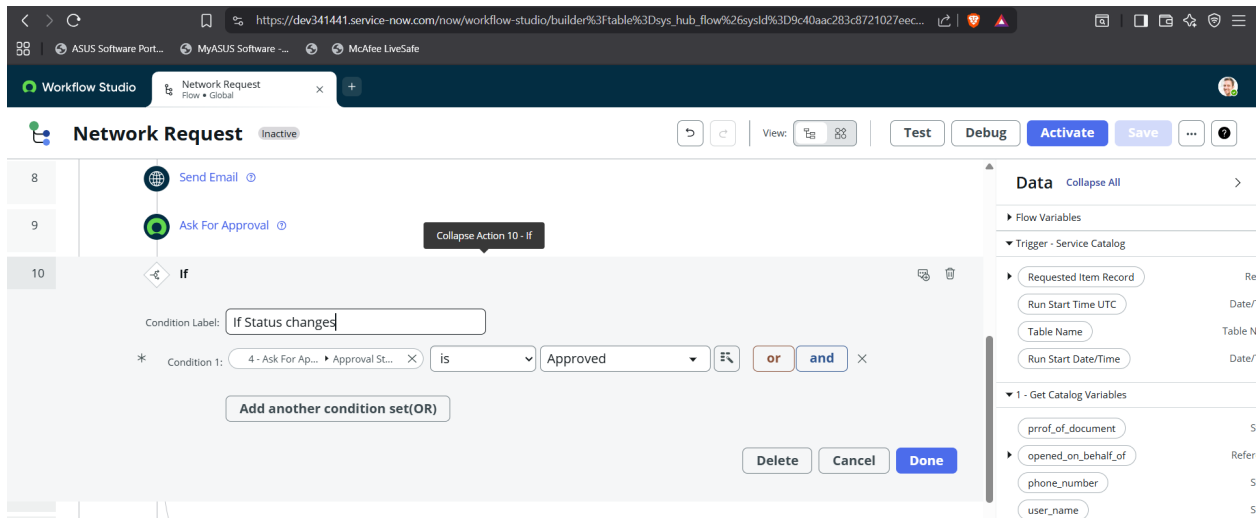
c. Click **Done**.

## Send Email (Approved)

d. Add action → Send Email.

e. 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)

a. Notify requestor that their request was rejected.

b. Optionally include rejection comments.

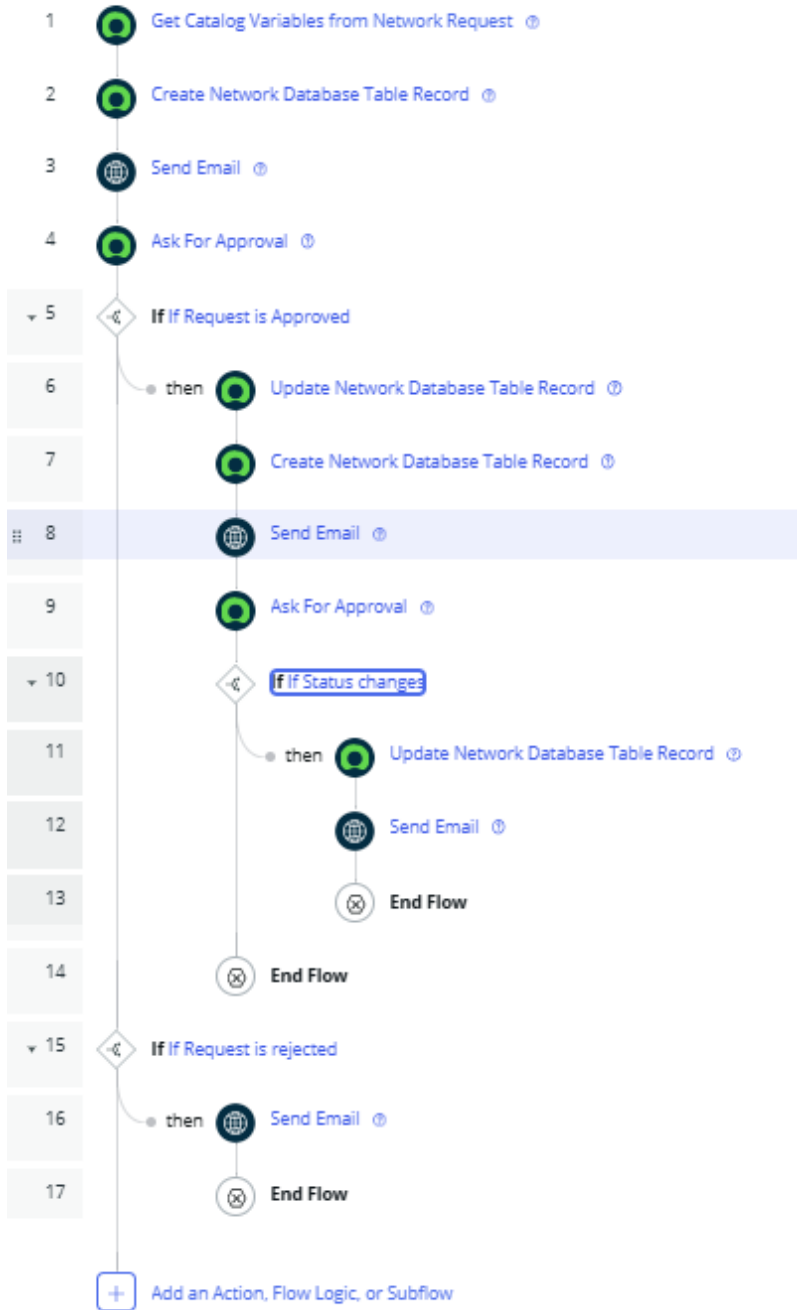
## OVERALL FLOW:

## Network Request Inactive

### TRIGGER

 Service Catalog

### ACTIONS Select multiple



## **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.