

Streamlining Ticket Assignment for Efficient Support Operations

Project Overview:

At **ABC Corporation**, the increasing volume of support requests has highlighted the need for a more efficient and automated ticket management process. Currently, manual assignment of tickets often leads to delays, misrouting, and uneven workload distribution across support teams.

This project introduces an **automated ticket assignment system** designed to streamline support operations. Leveraging predefined rules, intelligent routing mechanisms, and workflow automation, the system ensures that each ticket is directed to the most suitable team or individual.

By implementing this solution, ABC Corporation aims to:

- **Reduce resolution delays** by eliminating manual routing errors.
- **Enhance customer satisfaction** through faster and more accurate issue handling.
- **Optimize resource utilization** by balancing workloads across support teams.
- **Improve operational transparency** with clear assignment logic and reporting.

Ultimately, this initiative supports the organization's goal of delivering **high-quality, responsive, and efficient IT support services** while empowering teams to focus on problem resolution rather than administrative overhead.

Table of Contents

1. Introduction
2. Project Objectives
3. Key Features
4. ServiceNow Developer Setup
5. Project Implementation in ServiceNow
6. Screenshots of Output
7. Conclusion

1. Introduction

In today's fast-paced business environment, timely and accurate handling of IT support requests is essential for maintaining productivity and customer satisfaction. At **ABC Corporation**, the growing number of support tickets has made manual assignment inefficient, often resulting in delays, misrouted issues, and increased workload on support staff.

To overcome these challenges, this project focuses on implementing an **automated ticket assignment system**. By leveraging workflow automation and intelligent routing, the system will ensure that tickets are assigned to the right teams or individuals without manual intervention. This not only improves response and resolution times but also enhances overall efficiency, resource utilization, and service quality within the support department.

2. Project Objective

The primary objective of this project is to **implement an automated ticket assignment system** at **ABC Corporation** that enhances the efficiency of IT support operations. The solution is designed to:

- **Automate ticket routing** to ensure accurate and timely assignment to the appropriate support teams or individuals.
- **Reduce delays in issue resolution** by minimizing manual intervention and routing errors.
- **Improve customer satisfaction** through faster response times and consistent service delivery.
- **Optimize resource utilization** by balancing workloads across support teams.
- **Enhance operational transparency** with clear assignment logic, reporting, and monitoring capabilities.

By achieving these objectives, the project aims to transform the support process into a more **streamlined, reliable, and customer-centric system**.

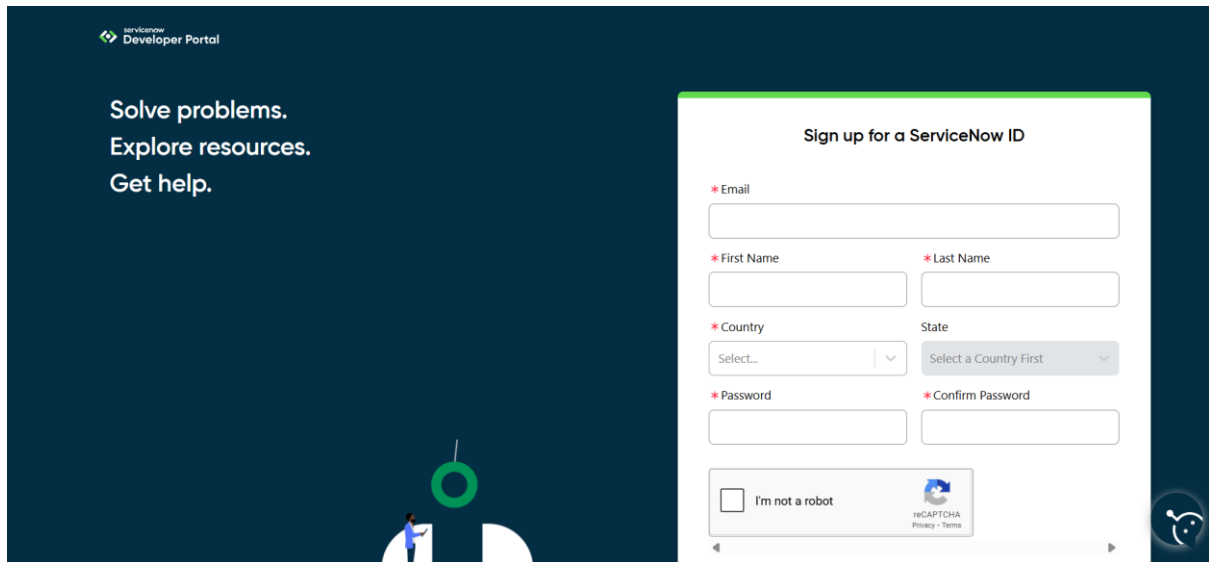
3. Key Features

- **Automated Routing** – Tickets assigned to the right team/person.
- **Dynamic Rules** – Configurable logic based on category, priority, etc.
- **Load Balancing** – Distributes workload evenly across teams.
- **Escalation Support** – Auto-escalates tickets nearing SLA breach.
- **Notifications** – Real-time alerts for quicker response.
- **Analytics** – Reports on ticket flow and team performance.

4. ServiceNow Developer Setup:

Create a Developer Account

1. Go to ServiceNow Developer Portal(<https://developer.servicenow.com/dev.do>).
- Sign up for a free developer account and fill the following details.

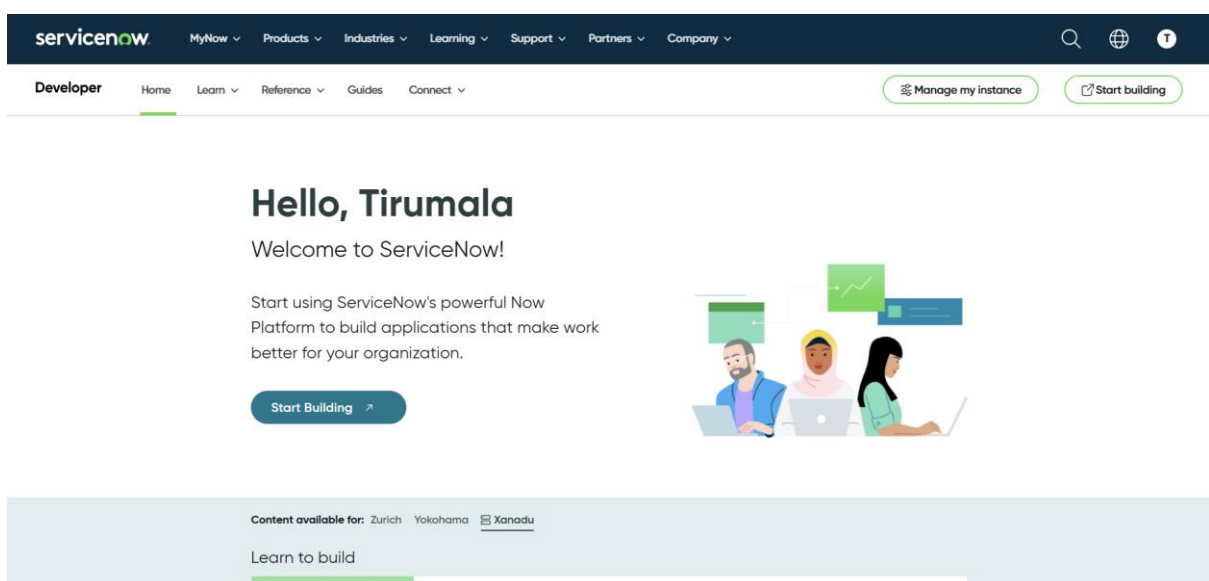


The screenshot shows the ServiceNow Developer Portal sign-up page. On the left, there's a dark blue banner with the text "Solve problems. Explore resources. Get help." and a small illustration of a person. On the right, there's a white sign-up form titled "Sign up for a ServiceNow ID". The form includes fields for Email, First Name, Last Name, Country (a dropdown menu), State (a dropdown menu with "Select a Country First" as the first option), Password, and Confirm Password. There is also a checkbox for "I'm not a robot" and a reCAPTCHA logo. The ServiceNow logo is in the top left corner of the page.

After signing up you will get an verification mail to you provided email id. After the verification your ServiceNow Developer Portal Home Page will appear

Now click on start building it will take you to the section where you can **request a Personal Developer Instance (PDI)** or start using **App Engine Studio** and other tools.

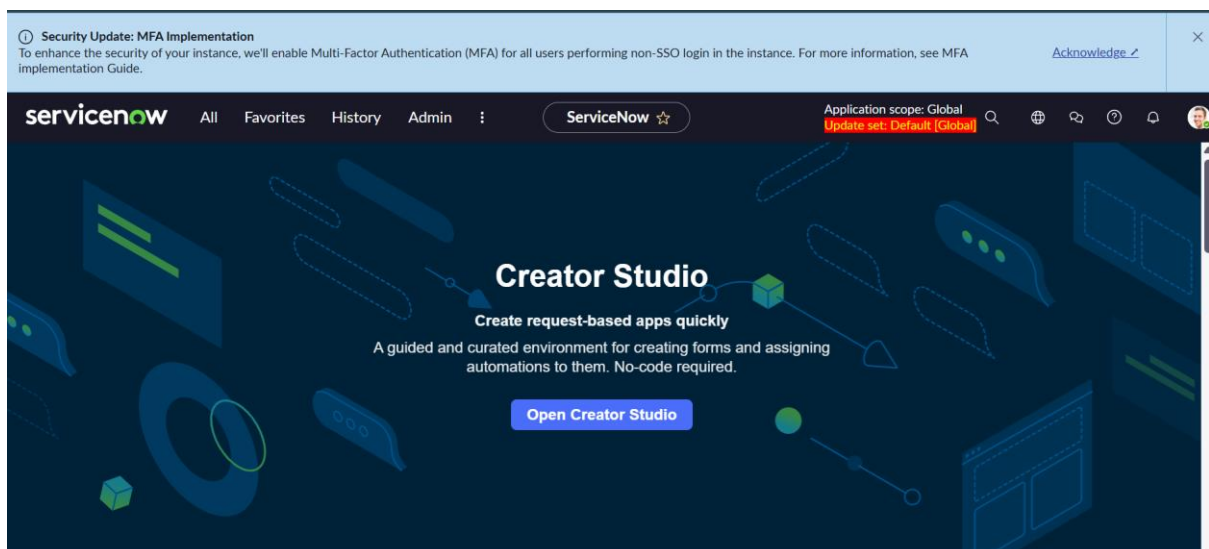
Profile Icon (Top Right Corner) → Manage your account, request instances, and check your developer profile.



5. Project Implementation in ServiceNow:

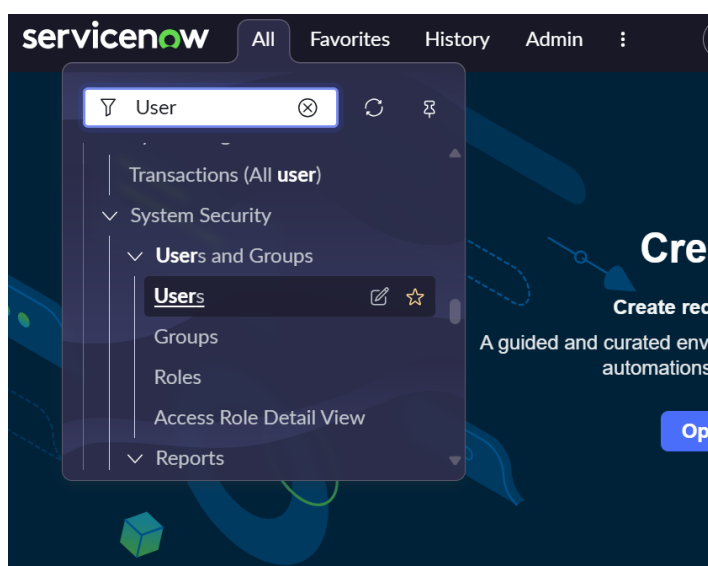
After the instance building is completed the page will be directed to your creator studio.

Creator Studio in ServiceNow provides a **guided, no-code environment** to build applications quickly. It is especially useful for creating **request-based applications** by defining forms, setting up tables, and automating workflows.



a. Creating Users:

1. In the left-hand navigation panel, click on **All** and search for **Users**.
2. Under **System Security**, select **Users**.



3. Click on **New** to create a new user record.
4. Fill in the required details (such as *First Name, Last Name, User ID, Email, Password, Roles*).
5. Click **Submit** to save the user.

Create the user by filling the following details:

The screenshot shows the ServiceNow user management interface for a user named Katherine Pierce. The interface includes a header with the ServiceNow logo, navigation tabs (All, Favorites, History), and a breadcrumb trail showing the user's details. The main form is divided into two columns. The left column contains fields for User ID (Katherine Pierce), First name (Katherine), Last name (Pierce), Title (empty), and Department (empty). Below these are checkboxes for 'Password needs reset', 'Locked out', 'Active' (checked), and 'Internal Integration User'. The right column contains fields for Email (empty), Identity type (Human), Language (-- None --), Calendar integration (Outlook), Time zone (System (America/Los Angeles)), Date format (System (yyyy-MM-dd)), Business phone (empty), and Mobile phone (empty). At the bottom of the form are buttons for 'Update', 'Set Password', and 'Delete'. Below the form is a 'Related Links' section with a link to 'View linked accounts'.

Create another user:

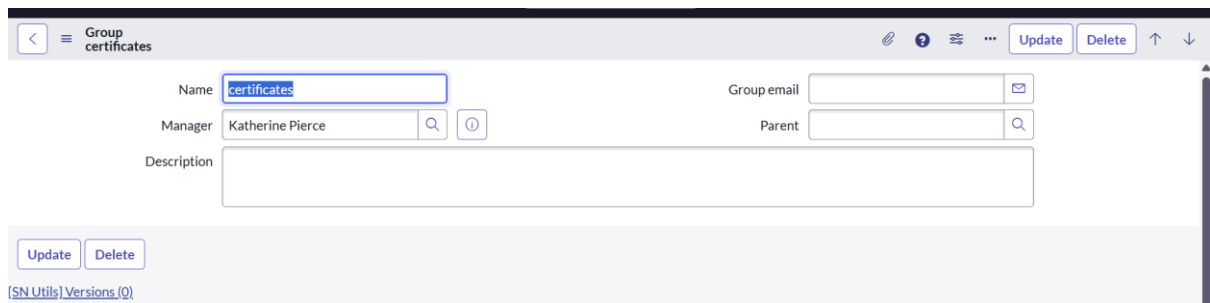
- Repeat the same steps to add a second user with different details.
- Click **Submit** again to save the second user.

The screenshot shows the ServiceNow user management interface for a user named Manne Niranjana. The interface is similar to the previous one, with the same header and navigation tabs. The main form is divided into two columns. The left column contains fields for User ID (manne.niranjana), First name (Manne), Last name (Niranjana), Title (empty), and Department (empty). Below these are checkboxes for 'Password needs reset', 'Locked out', 'Active' (checked), and 'Internal Integration User'. The right column contains fields for Email (niranjareddymanne2507@gmail.com), Identity type (Human), Language (-- None --), Calendar integration (Outlook), Time zone (System (America/Los Angeles)), Date format (System (yyyy-MM-dd)), Business phone (empty), and Mobile phone (empty). At the bottom of the form are buttons for 'Update', 'Set Password', and 'Delete'. Below the form is a 'Related Links' section with a link to 'View linked accounts'.

b. Create Groups:

1. In the left-hand navigation panel, click on **All** and search for **Groups**.
2. Under **System Security**, select **Groups**.
3. Click on **New** to create a new group record.
4. Fill in the required details such as:
 - **Name** – Group name.
 - **Description** – Short description of the group
 - **Manager** – Assign a manager if required.
5. Click **Submit** to save the group

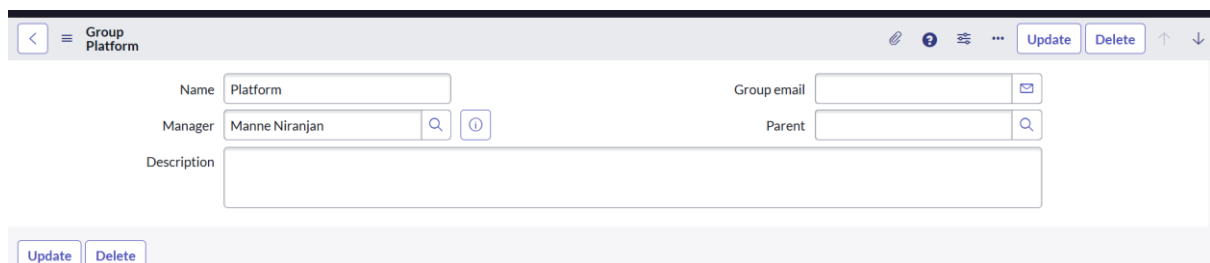
Create the group by filling the following details:



The screenshot shows a web application interface for creating a new group. The header bar includes a back arrow, a hamburger menu, the text 'Group certificates', and action buttons 'Update', 'Delete', and sort arrows. The form fields are: 'Name' with the value 'certificates', 'Group email' (empty), 'Manager' with the value 'Katherine Pierce' and a search icon, 'Parent' (empty), and 'Description' (empty). At the bottom left, there are 'Update' and 'Delete' buttons. A small link '[SN Utils] Versions (0)' is visible at the bottom left.

Create another group:

- Repeat the same steps to create a second group with different details.
- Click **Submit** again to save the second group.



The screenshot shows a web application interface for creating a new group. The header bar includes a back arrow, a hamburger menu, the text 'Group Platform', and action buttons 'Update', 'Delete', and sort arrows. The form fields are: 'Name' with the value 'Platform', 'Group email' (empty), 'Manager' with the value 'Manne Niranjan' and a search icon, 'Parent' (empty), and 'Description' (empty). At the bottom left, there are 'Update' and 'Delete' buttons.

c. Create Roles:

1. In the left navigation panel, select **All** → search for **Roles**.
2. Under **System Security**, click **Roles**.
3. Choose **New** to add a role.
4. Enter the required information, for example:
 - **Name** – A unique role identifier (e.g., *Certificate_role*).
 - **Description** – Brief details of what this role is meant for (e.g., Can deal with certification issues).
5. Click **Submit** to save.

The screenshot shows a web interface for creating a role. At the top, there's a header bar with a back arrow, a menu icon, the title 'Role Certification_role', and action buttons 'Update', 'Delete', and navigation arrows. Below the header, the form fields are: 'Name' with the value 'Certification_role', 'Application' with the value 'Global', and 'Elevated privilege' with an unchecked checkbox. The 'Description' field contains the text 'Can deal with certification issues'. Below the form fields are 'Update' and 'Delete' buttons. Underneath is a section titled 'Contains Roles' with a search bar and a table. The table has one row with the text 'Contains'. At the bottom of the table is a small icon of a computer monitor.

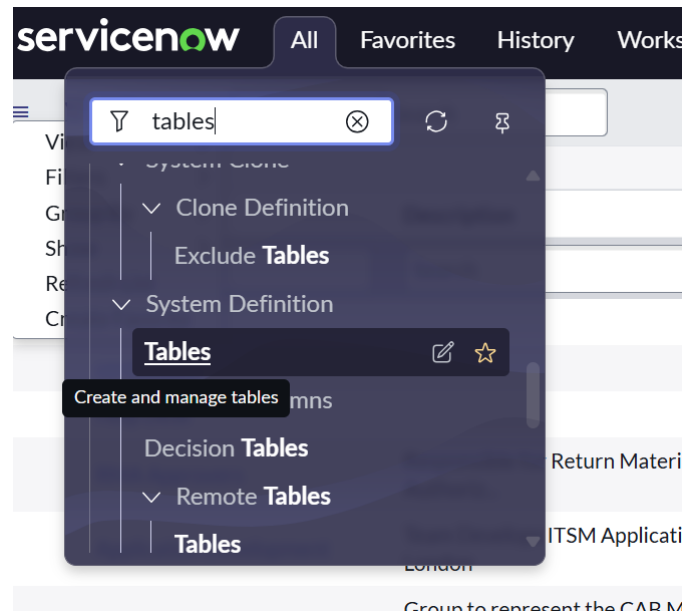
Add another role:

- Follow the same steps to define a second role.
- Role name as Platform_role and with suitable description.

The screenshot shows a web interface for creating a role. At the top, there's a header bar with a back arrow, a menu icon, the title 'Role Platform_role', and action buttons 'Update', 'Delete', and navigation arrows. Below the header, the form fields are: 'Name' with the value 'Platform_role', 'Application' with the value 'Global', and 'Elevated privilege' with an unchecked checkbox. The 'Description' field contains the text 'Can deal with platform related issues'. Below the form fields are 'Update' and 'Delete' buttons. Underneath is a section titled 'Contains Roles' with a search bar and a table. The table has one row with the text 'Contains'. At the bottom of the table is a small icon of a computer monitor.

d. Create Table:

1. In the left-hand navigation panel, click on **All** → search for **Tables**.
2. Under **System Definition**, select **Tables**.



3. Click on **New** to create a new table.
4. Fill in the required details:
 - **Label:** *Operations related*
 - Check the boxes **Create module** and **Create mobile module**.
 - **New menu name:** *Operations related*

A screenshot of the ServiceNow 'Table' form for 'Operations related'. The form is titled 'Table Operations related View: Tables'. It has several fields and checkboxes. On the left, there are fields for 'Label' (Operations related), 'Name' (u_operations_related), and 'Extends table'. On the right, there are fields for 'Application' (Global), 'Remote Table' (unchecked), 'Create module' (checked), 'Create mobile module' (checked), 'Add module to menu' (Create new), and 'New menu name'. At the top right, there are buttons for 'Delete', 'Update', and 'Delete All Records'.

5. Define the **table columns** as per the project requirements (e.g., *Issue, Description, Assigned To, Status*).
6. Click **Submit** to save the table.

Columns					
Table Columns					
Column label	Type	Reference	Max length	Default value	Display
Assigned to user	Reference	User		32	false
Comment	String	(empty)		40	false
Updated	Date/Time	(empty)		40	false
Service request No	String	(empty)		40	false
Assigned to group	Reference	Group		32	false
Name	String	(empty)		40	false
Created by	String	(empty)		40	false
Priority	String	(empty)		40	false
Created	Date/Time	(empty)		40	false
Sys ID	Sys ID (GUID)	(empty)		32	false
Issue	String	(empty)		40	false
Ticket raised Date	Date/Time	(empty)		40	false
Updated by	String	(empty)		40	false
Updates	Integer	(empty)		40	false

This custom table will act as the data storage for support tickets in our project. It allows us to capture details about issues raised by users and makes it possible to route them automatically to the right team. Without this table, we wouldn't have a centralized place to manage project-specific records.

Adding Choices for the *Issue* Field

1. Navigate to the created table and open **Form Design**.
2. Select the **Issue** field.
3. Add the following choices:
- *Unable to login to platform*

○ *404 error*

○ *Regarding certificates*

○ *Regarding user expired*

Form Design

Operations related [u_operations_related] 2 Column

Name

Assigned to user

Comment

Assigned to group

Priority

Issue

Properties

Dependent

Choices

Choice type Dropdown with none

404 Error

404 error

Regarding certificates

regarding certificates

Regrading User expired

regrading user expired

Unable to login to platform

unable to login to platform

4. Save the form design.

The *Issue* field with predefined choices ensures **standardization** when users log problems. This avoids ambiguity (e.g., someone typing “login issue” vs. “can’t login”) and makes it easier to set up **automation rules** for ticket assignment later.

e. Assign roles & users to groups:

i. Assign roles & users to certificate group:

1. In the left-hand navigation panel, click on **All** → search for **Groups**.
2. Under **System Security**, select **Groups**.
3. Open the **Certificates Group** that was created earlier.
4. In the **Group Members** tab:
 - Click **Edit**.
 - Select *Katherine Pierce* as a member.
 - Save the changes.

Add FilterRun filter?

-- choose field -- -- oper -- -- value --

Collection

Q

Abel Tuter
Abraham Lincoln
Adela Cervantsz
Aileen Mottern
Alejandra Prenatt
Alejandro Mascall
Alene Rabeck
Alfonso Griglen
Alissa Mountjoy
Allan Schwandt
Allie Pumphrey
Allyson Gillispie
Alva Pennigton
Alyssa Biasotti
Amelia Caputo
Amos Linnan
Andrew Jackson

Group Members List

certificates

Katherine Pierce

>
<

CancelSave

Roles (1)Group Members (1)Groups

User

Search

Actions on selected rows...NewEdit...

Group = certificates

Q

User

Katherine Pierce

5. In the **Roles** tab:

- Click **Edit**.
- Select *Certification_role*.
- Save the changes.

Add FilterRun filter?

-- choose field --

-- oper --

-- value --

Collection

access_analyzer_admin
action_category_creator
action_designer
activity_admin
activity_creator
actsub_admin
actsub_user
admin
agent_admin
agent_security_admin
agent_workspace_user
ais_admin
ais_high_security_admin
aisa_admin
analytics_admin
analytics_categories_admin
analytics_filter_admin

Roles List

certificates

Certification_role

>

<

Cancel

Save

Roles (1) Group Members (1) Groups			
Created Search Actions on selected rows... Edit...			
Group = certificates			
<input type="checkbox"/>	Created	Role	Inherits
<input type="checkbox"/>	2025-08-31 08:23:08	Certification_role	true
1 to 1 of 1			

Assigning users and roles to the **Certificates Group** ensures that only authorized members can handle **certificate-related support tickets**. This allows proper **access control**, ensures **responsibility assignment**, and enables the ticket assignment automation to work correctly.

ii. Assign roles & users to platform group:

1. In the left-hand navigation panel, click on **All** → search for **Groups**.
2. Under **System Security**, select **Groups**.
3. From the list of groups, open the **platform group**.
4. In the **Group Members** tab:
 - Click **Edit**.
 - Select *Manne Niranjana* from the available users.
 - Save the changes.

Add Filter **Run filter** ?

-- choose field -- -- oper -- -- value --

Collection

Q

Abel Tuter
Abraham Lincoln
Adela Cervantsz
Aileen Mottern
Alejandra Prenatt
Alejandro Mascall
Alene Rabeck
Alfonso Griglen
Alissa Mountjoy
Allan Schwantd
Allie Pumphrey
Allyson Gillispie
Alva Pennigton
Alyssa Biasotti
Amelia Caputo
Amos Linnan
Andrew Jackson

>
<

Group Members List

Platform

Manne Niranjana

Cancel Save

Roles (1) **Group Members (1)** Groups

≡ User Search

Group = Platform

☐ User

Manne Niranjana

1 to 1 of 1

5. In the **Roles** tab:

- Click **Edit**.
- Select *platform group* from the available roles.
- Save the changes.

Add Filter
Run filter ⓘ

-- choose field --
-- oper --
-- value --

Collection

Q

access_analyzer_admin
action_category_creator
action_designer
activity_admin
activity_creator
actsub_admin
actsub_user
admin
agent_admin
agent_security_admin
agent_workspace_user
ais_admin
ais_high_security_admin
aisa_admin
analytics_admin
analytics_categories_admin
analytics_filter_admin

>
<

Roles List
Platform

Platform_role

Cancel
Save

Roles (1)
Group Members (1)
Groups

Created
Search

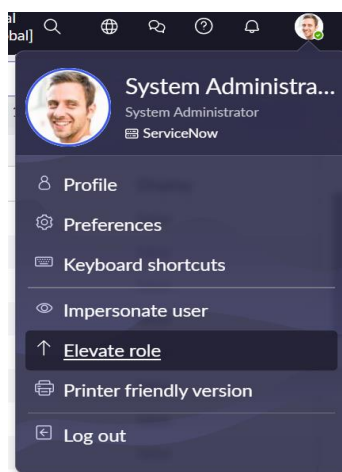
Group = Platform

	Created	Role	Granted by	Inherits
<input type="checkbox"/>	2025-08-31 08:26:37	Platform_role	(empty)	true

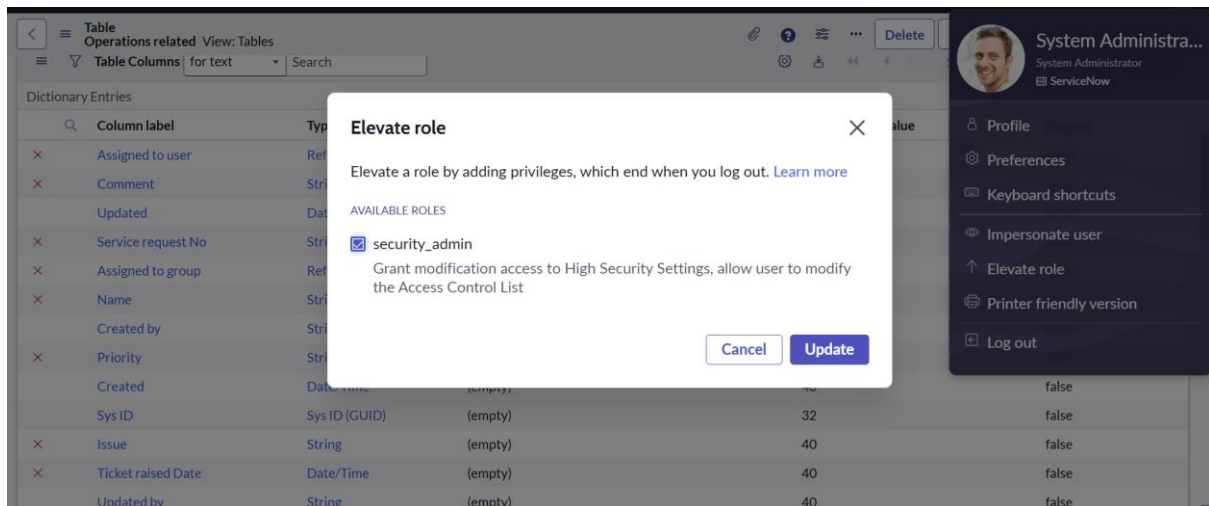
1 to 1 of 1

f. Assign role to table:

1. In the left-hand navigation panel, click on **All** → search for **Tables**.
2. From the list, select the **Operations related** table.
3. Go to the **Application Access** tab.
4. Click on your profile (top-right corner).



5. Select **Elevate Role** → choose **security_admin** → click **Update**.



6. Under **u_operations_related** [Read] operation:

- In **Requires Role**, double-click to insert a new row.
- Add **platform_role** and **certificate_role**.
- Click **Update**.

7. Under **u_operations_related** [Write] operation:

- In **Requires Role**, double-click to insert a new row.
- Add **platform_role** and **certificate_role**.
- Click **Update**

Name	Decision Type	Operation	Type	Active	Updated by	Updated
u_operations_related	Allow If	create	record	true	admin	2025-08-31 08:12:16
u_operations_related	Allow If	delete	record	true	admin	2025-08-31 08:12:17
u_operations_related	Allow If	write	record	true	admin	2025-08-31 08:12:17
u_operations_related	Allow If	read	record	true	admin	2025-08-31 08:12:17

Role
u_operations_related_user
Platform_role
Certification_role

Assigning roles to the **Operations related** table ensures that only authorized users (those with **Platform Role** or **Certificate Role**) can **read** and **write** records. This provides **data security, controlled access**, and ensures tickets are handled only by the correct teams.

g. Creating Access Control Rules (ACL):

1. **Open ServiceNow** and log in to your Personal Developer Instance (PDI).
2. In the left-hand navigation panel, click on **All** → search for **ACL**.
3. Under **System Security**, select **Access Control (ACL)**.
4. Click on **New** to create a new ACL.
5. Fill in the required details for the ACL rule (such as *Table, Operation, Field*).

The screenshot shows the 'Access Control - New Record' form in ServiceNow. At the top, there's a warning message: 'Warning: A role, security attribute, data condition, script or ACL control via reference fields is required to properly secure access with this ACL.' The form fields include: Type (record), Operation (write), Decision Type (Allow If), Application (Global), Active (checked), Admin overrides (checked), Protection policy (None), Name (Operations related [u_operations_related]), Service request No (selected from a dropdown), Description (empty), and Applies To (No. of records matching the condition: 4). There are also buttons for 'Add Filter Condition' and 'Add OR Clause'.

6. Scroll down to the **Requires Role** section.

- Double-click on **Insert a new row**.
- Add the **admin** role.

This screenshot shows the 'Requires role' section of the ACL form. A search bar contains the text 'admin'. Below it, a dropdown menu shows 'Showing 1 through 1 of 1' with 'admin' listed. The 'Security Attribute' section is also visible, showing 'admin'. At the bottom, there are radio buttons for 'Local or Existing', 'Existing', and 'Local' (which is selected).

This screenshot is identical to the previous one, showing the 'Requires role' section with the 'admin' role selected in the search bar and dropdown menu. The 'Security Attribute' section shows 'admin', and the 'Local or Existing', 'Existing', and 'Local' radio buttons are at the bottom.

7. Click **Submit** to save the ACL.

Similarly, create four ACLs for the following fields:

- Field 1: Issue
- Field 2: Priority
- Field 3: Ticket raised Date

Access Control New record

Warning: A role, security attribute, data condition, script or ACL control via reference fields is required to properly secure access with this ACL.

* Type: record ⓘ

* Operation: write ⓘ

Decision Type: Allow If

Application: Global ⓘ

Active: ☒

Advanced: ☐

Admin overrides: ☒

Protection policy: -- None --

* Name: Operations related [u_operations_related] Issue

Description: </>

Applies To: No. of records matching the condition: 4 ⓘ

Add Filter Condition Add OR Clause

-- choose field -- -- oper -- -- value --

Access Control New record

Warning: A role, security attribute, data condition, script or ACL control via reference fields is required to properly secure access with this ACL.

* Type: record ⓘ

* Operation: write ⓘ

Decision Type: Allow If

Application: Global ⓘ

Active: ☒

Advanced: ☐

Admin overrides: ☒

Protection policy: -- None --

* Name: Operations related [u_operations_related] Priority

Description: </>

Applies To: No. of records matching the condition: 4 ⓘ

Add Filter Condition Add OR Clause

-- choose field -- -- oper -- -- value --

Access Control New record

Warning: A role, security attribute, data condition, script or ACL control via reference fields is required to properly secure access with this ACL.

* Type: record ⓘ

* Operation: write ⓘ

Decision Type: Allow If

Application: Global ⓘ

Active: ☒

Advanced: ☐

Admin overrides: ☒

Protection policy: -- None --

* Name: Operations related [u_operations_related] Ticket raised Date

Description: </>

Applies To: No. of records matching the condition: 4 ⓘ

Add Filter Condition Add OR Clause

-- choose field -- -- oper -- -- value --

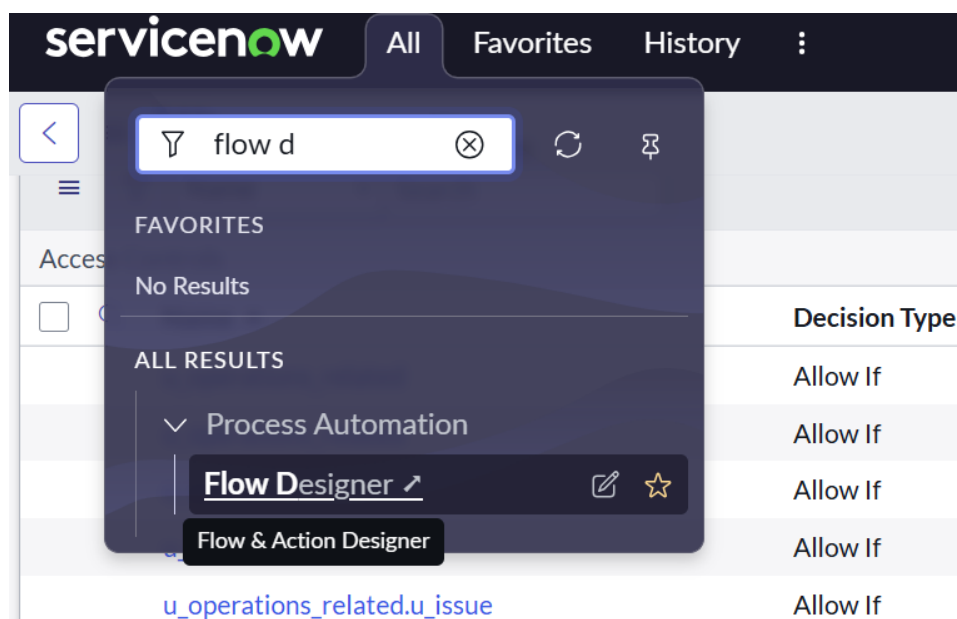
u_operations_related.u_name	Allow If	write	record	true	admin	2025-08-31 08:38:59
u_operations_related.u_priority	Allow If	write	record	true	admin	2025-08-31 08:37:12
u_operations_related.u_service_request_no	Allow If	write	record	true	admin	2025-08-31 08:35:05
u_operations_related.u_ticket_raised_date	Allow If	write	record	true	admin	2025-08-31 08:38:04

ACLs (Access Control Rules) provide **granular security** at the **table and field level**. By creating these ACLs, we ensure that only users with the **admin role** (or other required roles) can access or modify sensitive fields. This prevents unauthorized actions and protects critical data in the **Operations related** table.

h. Flow:

i. Create a Flow to Assign operations ticket to group:

1. **Open ServiceNow** and log in to your instance.
2. In the left navigation panel, click on **All** → search for **Flow Designer**.
3. Select **Flow Designer** under *Process Automation*.



4. Once Flow Designer opens, click on **New** → select **Flow**.

Configure Flow Properties:

- **Flow Name:** *Regarding Certificate*
- **Application:** *Global*
- **Run As User:** *System User*
- Click **Submit**.

Let's get the details for your flow

Flow name * ⓘ

Application * ⓘ

Description ⓘ


▼ Hide additional properties

Protection ⓘ


Run as ⓘ

Add a Trigger:

1. Click **Add a trigger**.
2. Search and select **Create or update a record**.
3. In the configuration:
 - **Table:** *Operations related*
 - **Condition:**
 - *Field:* Issue
 - *Operator:* is
 - *Value:* Regarding Certificates
4. Click **Done**.

 **Regarding Certificate** Active View:

TRIGGER

 Operations related Created or Updated where (Issue is Regarding certificates)

Trigger

* Table

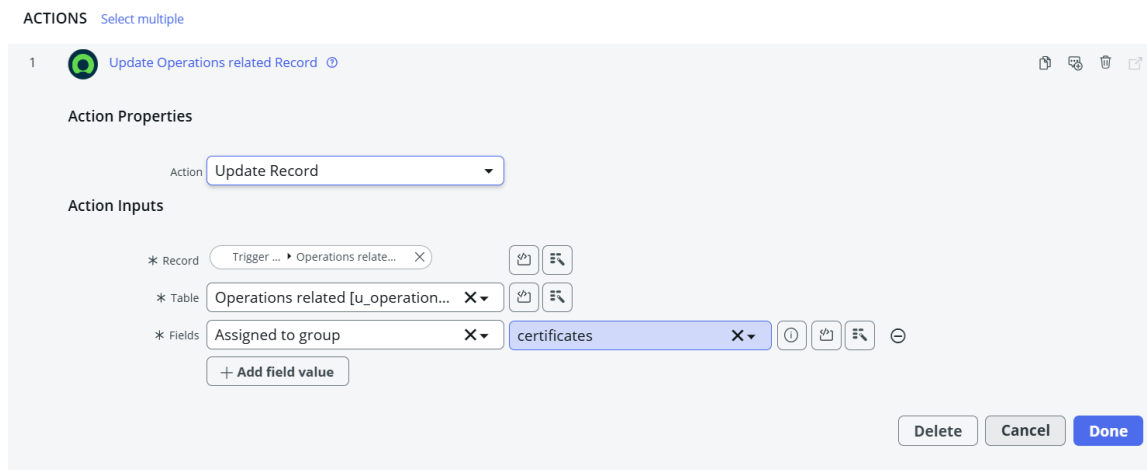
Condition All of these conditions must be met

or

Run Trigger

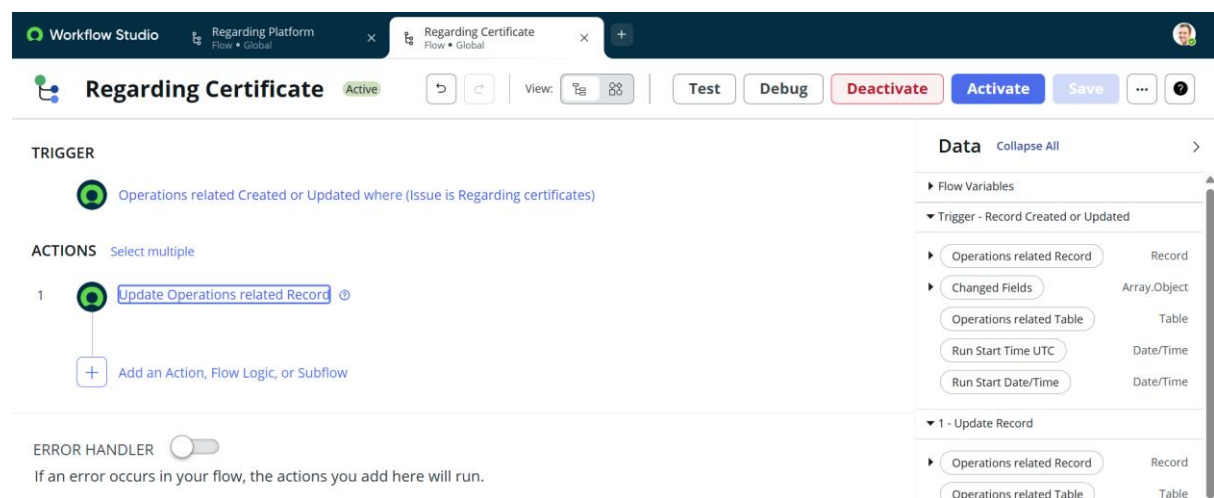
Add an Action:

1. Click **Add an action**.
2. Search for **Update Record** and select it.
3. In the **Record field**, drag the required fields from the **Data Panel** on the left.
4. Configure the update:
 - **Table:** Auto-assigned from trigger
 - **Field:** Assigned to Group
 - **Value:** Certificates
5. Click **Done**.



Final Steps:

- Click **Save** to save the Flow.
- Click **Activate** to enable it.



This Flow ensures that whenever a new **Operations related** ticket is created with the issue "*Regarding Certificates*", it will be **automatically routed** to the **Certificates group**. This eliminates manual assignment, speeds up issue resolution, and improves efficiency in ticket handling.

ii. Create a Flow to Assign operations ticket to Platform group:

1. **Open ServiceNow** and log in to your instance.
2. In the left navigation panel, click on **All** → search for **Flow Designer**.
3. Select **Flow Designer** under *Process Automation*.
4. Once Flow Designer opens, click on **New** → select **Flow**.

Configure Flow Properties:

- **Flow Name:** *Regarding Platform*
- **Application:** *Global*
- **Run As User:** *System User*
- Click **Submit**.

Let's get the details for your flow

Flow name * ⓘ

Regarding platform

Application * ⓘ

Global

Description ⓘ

Describe your flow.

▼ Hide additional properties

Protection ⓘ

-- None --

Run as ⓘ

System user

Cancel

Build flow

Add a Trigger:

1. Click **Add a trigger**.
2. Search and select **Create or update a record**.

3. Configure the trigger:

- **Table:** *Operations related*
- **Conditions:**
 - *Field: Issue → is → Unable to login to platform*
 - Click **New Criteria** → *Issue → is → 404 Error*
 - Click **New Criteria** → *Issue → is → Regarding User expired*

4. Click **Done**.

TRIGGER

Operations related Created or Updated where (Issue is Unable to login to platform; Issue is 404 Error; Issue is Regrading User expired)

Trigger: Created or Updated

* Table: Operations related [u_operation... X]

Condition All of these conditions must be met

Issue is Unable to login to platform

or

All of these conditions must be met

Issue is 404 Error

or

All of these conditions must be met

Issue is Regrading User expired

New Criteria


Run Trigger: Once

Advanced Options

Add an Action:

1. Click **Add an action**.
2. Search for **Update Record** and select it.
3. In the **Record field**, drag the fields from the **Data Panel** on the left.
4. Configure the update:
 - **Table:** *Auto-assigned from trigger*
 - **Field:** Assigned to Group
 - **Value:** Platform
5. Click **Done**.



ACTIONS [Select multiple](#)



1  Update Operations related Record ⓘ





Action Properties

Action Update Record

Action Inputs

* Record Trigger ... ▶ Operations relate... X  

* Table Operations related [u_operation... X  

* Fields Assigned to group X Platform X    

+ Add field value

Final Steps:


- Click **Save** to save the Flow.
- Click **Activate** to enable it.

Workflow Studio

Regarding Platform



Flow • Global

+



Regarding Platform

Active

View:  

Test


Debug

Deactivate


Activate

Save


...



TRIGGER

 Operations related Created or Updated where (Issue is Unable to login to platform; Issue is 404 Error; Issue is Re...

ACTIONS [Select multiple](#)

1  Update Operations related Record ⓘ

+ Add an Action, Flow Logic, or Subflow

ERROR HANDLER

☐ If an error occurs in your flow, the actions you add here will run.

Status: Published | Application: Global

0/1

Data [Collapse All](#)

Flow Variables

Trigger - Record Created or Updated

Operations related Record

Record

Changed Fields

Array.Object

Operations related Table

Table

Run Start Time UTC

Date/Time

Run Start Date/Time

Date/Time

1 - Update Record

Operations related Record

Record

Operations related Table

Table

Action Status

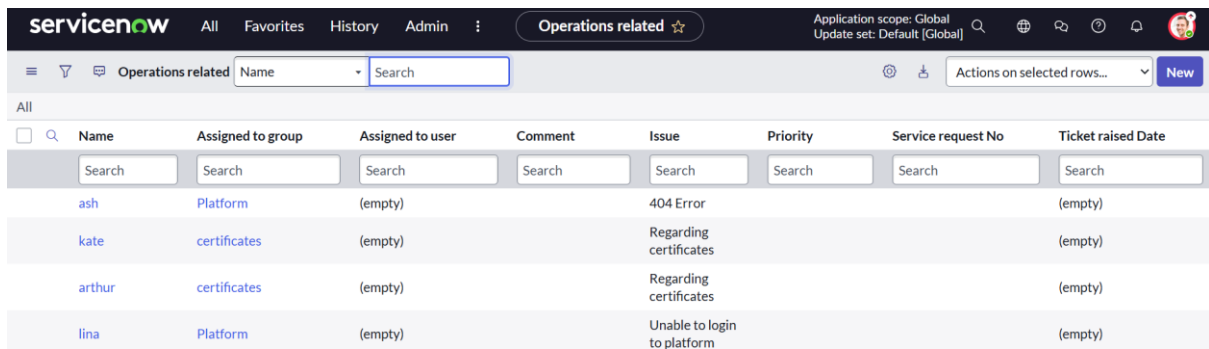
Object

This Flow ensures that all tickets in the **Operations related** table with issues like “Unable to login to platform”, “404 Error”, or “Regarding User expired” are **automatically assigned** to the **Platform group**. This reduces manual intervention, speeds up resolution times, and ensures tickets reach the correct support team without delays.

6. Screenshots of Output:

After implementing the Flows in ServiceNow, the ticket assignment works as follows:

- When a ticket is created in the **Operations related** table with the issue **“Regarding Certificates”**, it is **automatically assigned** to the **Certificates group**.
- When a ticket is created with issues like **“Unable to login to platform”**, **“404 Error”**, or **“Regarding User expired”**, it is **automatically assigned** to the **Platform group**



The screenshot shows the ServiceNow interface for the 'Operations related' table. The table has columns for Name, Assigned to group, Assigned to user, Comment, Issue, Priority, Service request No, and Ticket raised Date. Four rows are visible, showing tickets assigned to either the 'Platform' or 'certificates' group based on the issue type.

Name	Assigned to group	Assigned to user	Comment	Issue	Priority	Service request No	Ticket raised Date
ash	Platform	(empty)		404 Error			(empty)
kate	certificates	(empty)		Regarding certificates			(empty)
arthur	certificates	(empty)		Regarding certificates			(empty)
lina	Platform	(empty)		Unable to login to platform			(empty)

7. Conclusion:

The implementation of automated ticket assignment in ServiceNow has streamlined the support operations at **ABC Corporation**. By leveraging **Flow Designer**, tickets are now intelligently routed to the correct support groups based on the issue type. This eliminates manual intervention, reduces delays, and ensures efficient handling of incidents.

With the setup of **users, groups, roles, tables, ACLs, and automated flows**, the support process is now:

- **Faster** – Tickets reach the right team instantly.
- **Accurate** – Reduced chances of misrouting.
- **Efficient** – Optimized resource utilization across support groups.
- **Customer-focused** – Improved resolution times lead to higher customer satisfaction.

In summary, this project demonstrates how ServiceNow can be used to **enhance IT service management (ITSM)** by automating repetitive tasks and empowering support teams to focus on resolving issues rather than managing ticket assignments.

GitHub link: <https://github.com/tirumalaraju18/Streamlining-Ticket-Assignment-for-Efficient-Support-Operations>