

Streamlining Ticket Assignment for Efficient Support Operations

Project Demonstration

1. Introduction

Project Title: *Streamlining Ticket Assignment for Efficient Support Operations*

Team ID: LTVIP2026TMIDS74504

Team Size: 5

- **Team Leader:** G Sravani – Project Coordination, Users & Groups Configuration
- **Team Member:** G Chandana – Roles & Custom Table Creation
- **Team Member:** J Meghana – Assigning Roles to Users & Groups
- **Team Member:** M Sanjana – Table-Level Role Assignment & ACL Configuration
- **Team Member:** S Swathi – Flow Designer Automation

2. Project Demonstration

2.1 Objective of Demonstration

The purpose of this demonstration is to show how the ticket assignment process is automated in ServiceNow to improve efficiency, reduce manual effort, and ensure proper role-based access control.

2.2 Implementation Overview

The project was implemented in the following structured manner:

Step 1: Custom Table Creation

A custom table **Operations Related (u_operations_related)** was created to store support tickets.

The table contains fields such as:

- Name
- Assigned to Group
- Assigned to User
- Issue (Choice Field)
- Priority
- Service Request Number
- Ticket Raised Date
- Comment

This table acts as the central repository for all support operations tickets.

The screenshot shows the ServiceNow interface for creating a new table named 'Operations related'. The 'Label' field is set to 'Operations related' and the 'Name' field is set to 'u_operations_related'. The table has 14 columns defined in the 'Dictionary Entries' section:

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

Figure 1: Table Creation - Operations Related

Step 2: Role Creation

Two roles were created to control access:

- Certification_role → For handling certification-related issues
- Platform_role → For handling platform-related issues

These roles ensure secure access control and prevent unauthorized modifications.

The screenshot shows the ServiceNow interface for creating a new role named 'Certification_role'. The 'Name' field is set to 'Certification_role' and the 'Application' field is set to 'Global'. The 'Description' field contains the text 'Can deal with certification issues'. The 'Contains' tab is selected, showing a single record: 'Role - Certification_role'.

Figure 2.1: Certification_role Creation

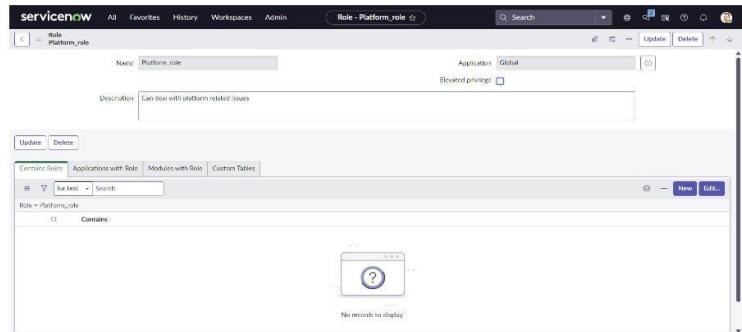


Figure 2.2: Platform_role - Creation

Step 3: Group Creation and Role Assignment

Two groups were created:

- certificates group → Assigned Certification_role
- Platform group → Assigned Platform_role

Users were added to respective groups to ensure proper ticket handling.

This enables role-based ticket visibility and access control.

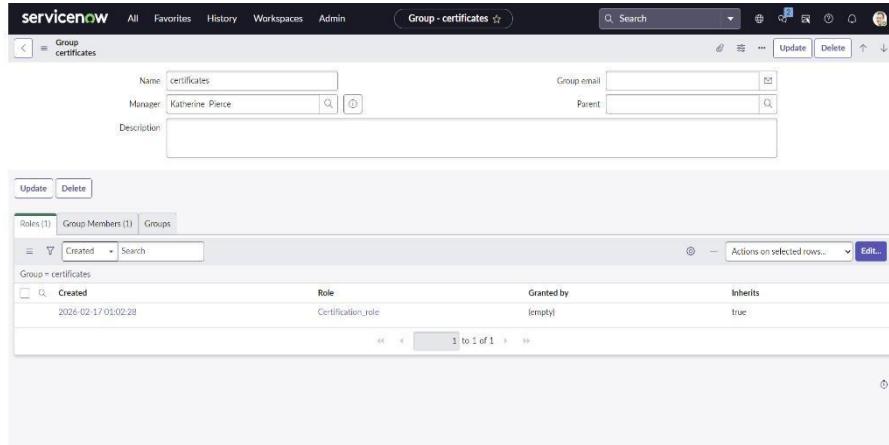


Figure 3.1: certificates group

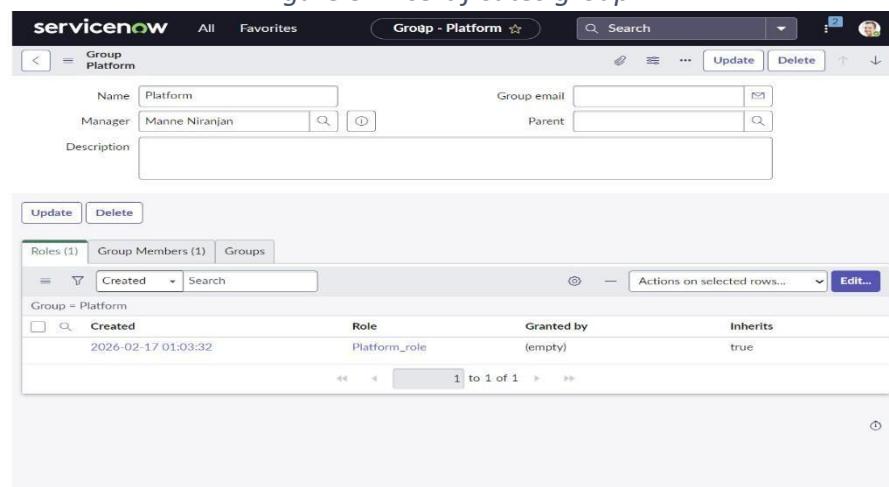


Figure 3.2: Platform group

The screenshot shows a ServiceNow interface titled 'Access Controls'. The table has columns: Name, Decision Type, Operation, Type, Active, Updated by, and Updated. The data includes various rules for operations like 'issue', 'name', 'ticket_raised_date', 'priority', 'service_request_no', and 'related'.

Name	Decision Type	Operation	Type	Active	Updated by	Updated
u_operations_related.u.issue	Allow If	write	record	true	admin	2026-02-17 01:15:39
u_operations_related.u.name	Allow If	write	record	true	admin	2026-02-17 01:14:59
u_operations_related.u.ticket_raised_date	Allow If	write	record	true	admin	2026-02-17 01:14:20
u_operations_related.u.priority	Allow If	write	record	true	admin	2026-02-17 01:13:38
u_operations_related.u.service_request_no	Allow If	write	record	true	admin	2026-02-17 01:11:45
u_operations_related	Allow If	write	record	true	admin	2026-02-17 00:51:08
u_operations_related	Allow If	read	record	true	admin	2026-02-17 00:51:08
u_operations_related	Allow If	delete	record	true	admin	2026-02-17 00:51:08
u_operations_related	Allow If	create	record	true	admin	2026-02-17 00:51:08

Figure 4: Access Control List

4: Access Control (ACL) Configuration

Access Control Rules were created to:

- Allow only authorized roles to Create, Read, Write, Delete records.
- Restrict unauthorized users from accessing ticket data. This ensures secure data handling and compliance.

Step 5: Flow Designer Implementation (Auto Assignment Logic)

Two separate flows were created in Flow Designer:

Flow 1 – Regarding Certificate

Trigger:

- When a record is Created or Updated
- Condition: Issue = "regarding certificates"

Action:

- Automatically updates the record
- Assigns ticket to certificates group

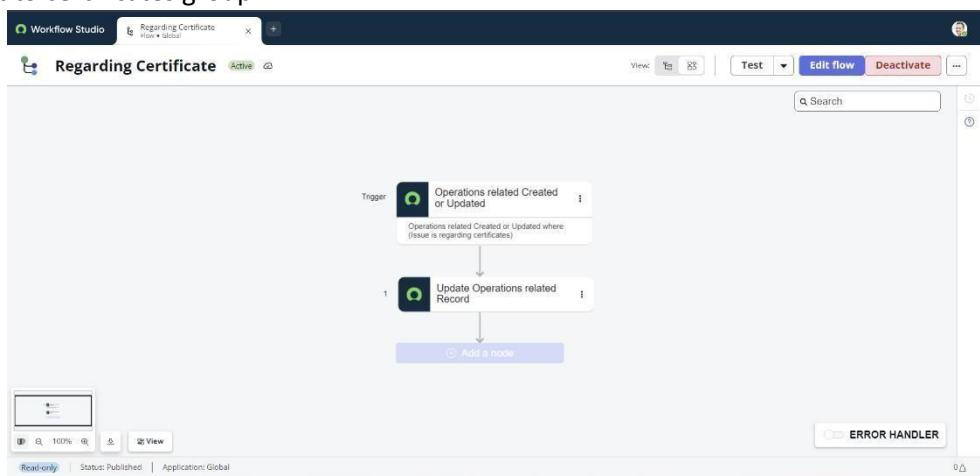


Figure 5.1: Flow Automation for “Regarding Certificate”

Flow 2 – Regarding Platform

Trigger:

- When record is Created or Updated
- Condition: Issue = "unable to login to platform" OR "404 error"

Action:

- Automatically assigns ticket to Platform group

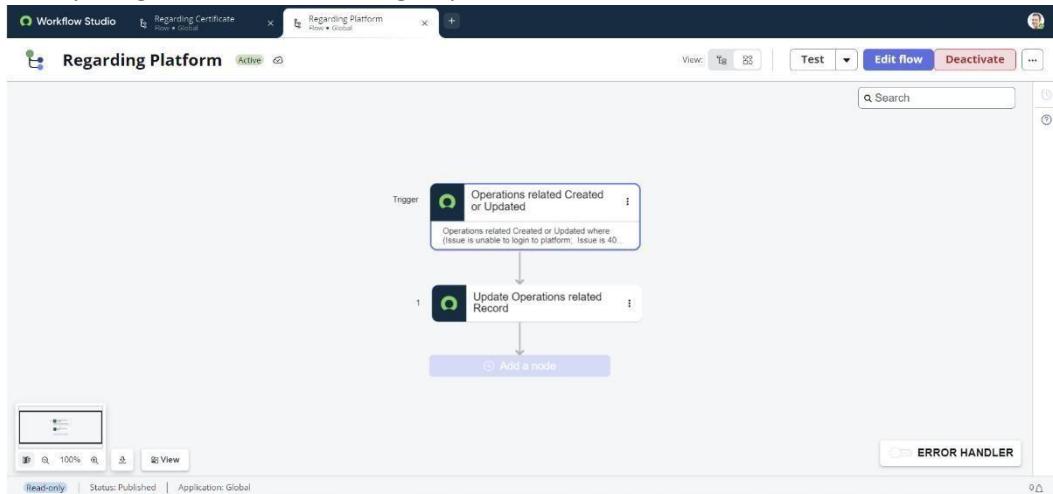


Figure 5.2: Flow Automation for "Regarding Platform"

This eliminates manual assignment and ensures faster ticket routing.

2.3 End-to-End Working Demonstration

- User creates a new ticket in Operations Related table.
- User selects Issue category.
- Record is submitted.
- Flow Designer triggers automatically.
- Based on Issue:
 - Ticket is auto-assigned to correct group.
- Only authorized users (based on roles) can modify the ticket.

This demonstrates automated ticket routing and secure access control.

Figure 6.1: Table Form filled successfully

Name	Assigned to group	Assigned to user	Comment	Issue	Priority	Service request No	Ticket raised Date
servicenow user	certificates	(empty)	Not Working properly	regarding certificates	(empty)	(empty)	(empty)
Chand	Platform	(empty)	Something user expired is getting	regarding user expired	(empty)	(empty)	(empty)
admin user	Platform	(empty)	unable to login to platform	(empty)	(empty)	(empty)	(empty)
sample user	Platform	(empty)	Getting this error repeatedly	404 error	(empty)	(empty)	(empty)

Figure 6.2: Automatically "Assigned Group" is created

2.4 Business Impact

The implementation provides:

- Reduced manual effort in ticket routing
- Faster response time
- Improved SLA compliance
- Secure role-based access
- Organized ticket management

Github Link : <https://github.com/srava123371/Streamlining-Ticket-Assignment-for-Efficient-Support-Operations.git>

Demo Video Link: <https://drive.google.com/file/d/17oMzpHHAosefjQNUXeFl3L9oXB2NTOdI/view?usp=sharing>