# Streamlining Ticket Assignment for Efficient Support Operations**Streamlining Ticket Assignment for Efficient Support Operations** **Platform:** ServiceNow **Submitted by:** Khadir Shaik **Repository:** [GitHub – Streamlined Ticket Automation](https://github.com/KhadirShaikL21/Streamlined-Ticket-Automation?utm_source=chatgpt.com) **Date:** *Sep 5, 2025*

## Abstract

This project implements an automated ticket assignment solution for support operations using ServiceNow. By creating a custom table, securing it with roles and ACLs, and automating assignments using Flow Designer, tickets are automatically routed to the appropriate support group based on the issue type.

This solution eliminates manual triage, speeds up first response, and improves SLA compliance — ensuring efficient, scalable, and auditable support operations.

## 1. Introduction

Modern enterprises face high volumes of support requests daily. Manually reviewing and assigning tickets often leads to delays, misrouting, and inefficient resource allocation. This project aims to build a secure, automated workflow that **assigns tickets to the right group immediately upon creation or update**.

Our solution uses native ServiceNow features:

* Custom table for ticket storage.
* Application Access and ACLs for data security.
* Flow Designer for automation.
* Update Sets for portability.

## 2. Problem Statement

Support teams often encounter:

* **Delayed assignment** – tickets wait in the queue before being assigned.
* **Incorrect routing** – tickets reach the wrong team, requiring rework.
* **Manual overhead** – team leads waste time on repetitive triage tasks.

Our goal is to solve these with **deterministic automation**, reducing human error and ensuring the right team gets the ticket at the right time.

## 3. Objectives

1. **Automate Ticket Assignment** – Automatically route tickets to Platform or Certificates groups.
2. **Ensure Security** – Use roles, groups, and ACLs to protect table access.
3. **Improve Maintainability** – Use no-code automation (Flow Designer) so administrators can modify logic easily.
4. **Enable Auditing** – Provide logs and execution history for compliance.

## 4. Tools and Technologies

* **ServiceNow** – Platform for configuration and automation.
* **Flow Designer** – No-code automation tool to define triggers and actions.
* **Application Access** – Table-level security for CRUD operations.
* **ACLs (Access Control Lists)** – Fine-grained security at field and record levels.
* **Update Sets** – To move customizations between instances.
* **GitHub** – For documentation, screenshots, and version control.

## 5. Core Concepts and Definitions

### Users

A user is an individual who can log into ServiceNow. Attributes include Name, Username, Email, and Roles. Users can be added to groups to simplify assignment and permissions.

### Groups

Groups are logical collections of users. They are used to:

* Assign work to multiple users at once.
* Apply notifications and reporting.
* Manage permissions for a set of users.

### Roles

Roles are permission bundles. Assigning a role gives users access to certain tables, fields, and modules. In this project:

* **platform\_role** – for users handling platform issues.
* **certification\_role** – for users handling certification issues.

### Tables

A table is a structured collection of records. We created **u\_operations\_related** with fields like:

* Service Request No
* Name
* Issue (Choice field)
* Assigned to Group
* Assigned to User
* Priority
* Comments

### Choice Fields

Used to restrict a field to a set of predefined values. For issue, we added:

* Unable to login to platform
* 404 error
* Regarding user
* Regarding certificates

### Application Access

Allows or restricts table access based on roles.

### ACLs (Access Control Lists)

Define security rules for reading, writing, creating, or deleting records or fields.

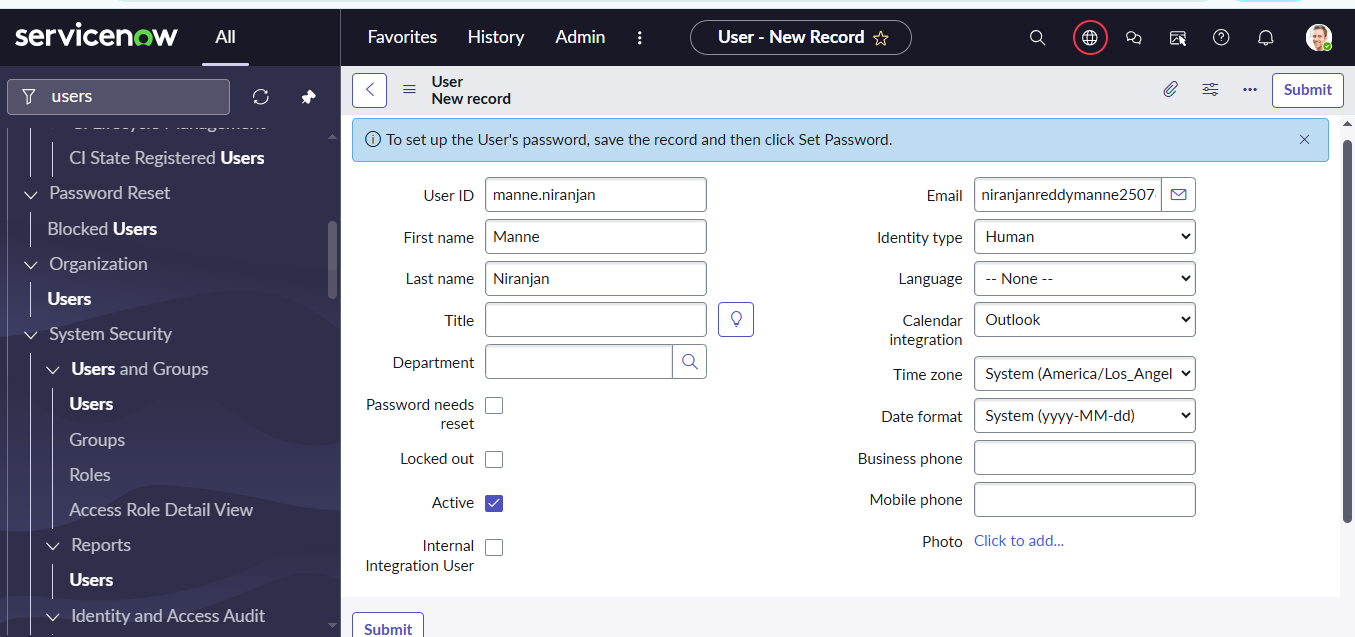
### Flow Designer

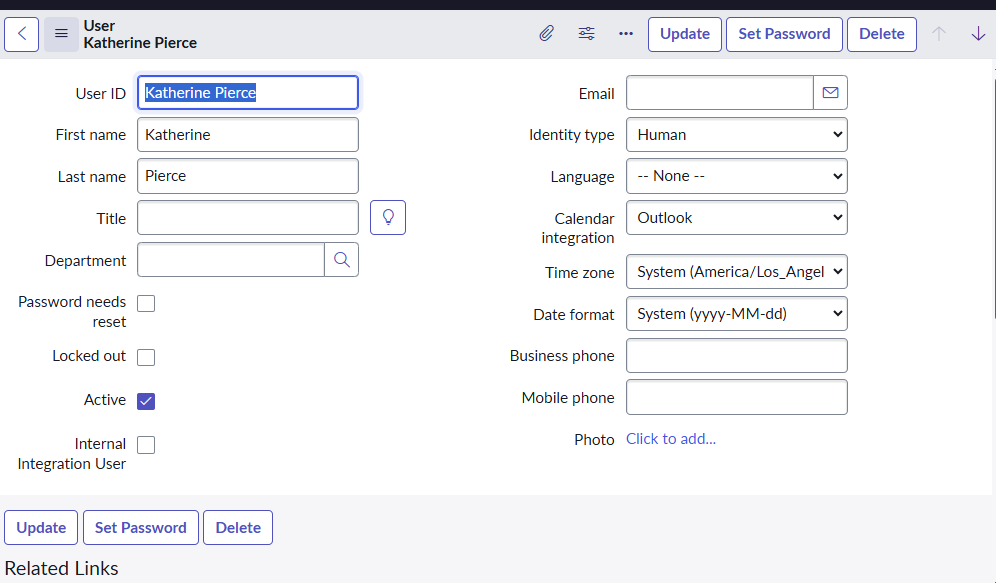
A no-code tool that automates record-based workflows with triggers, conditions, and actions.

## 6. Implementation

### Step 1: User Creation

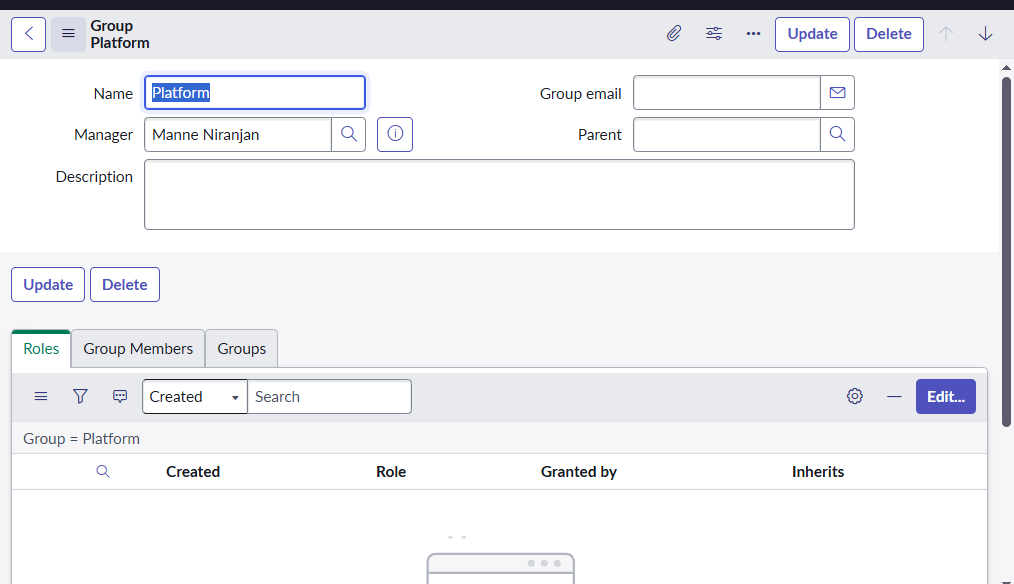
* Created two users:
  + Niranjan Manne (Platform team)
  + Katherine Pierce (Certificates team)

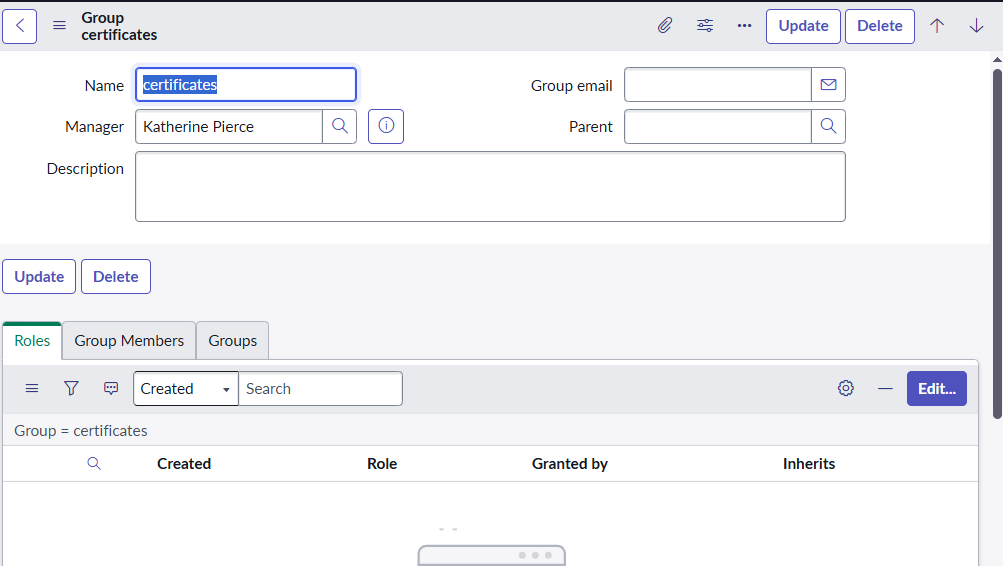




### Step 2: Group Creation

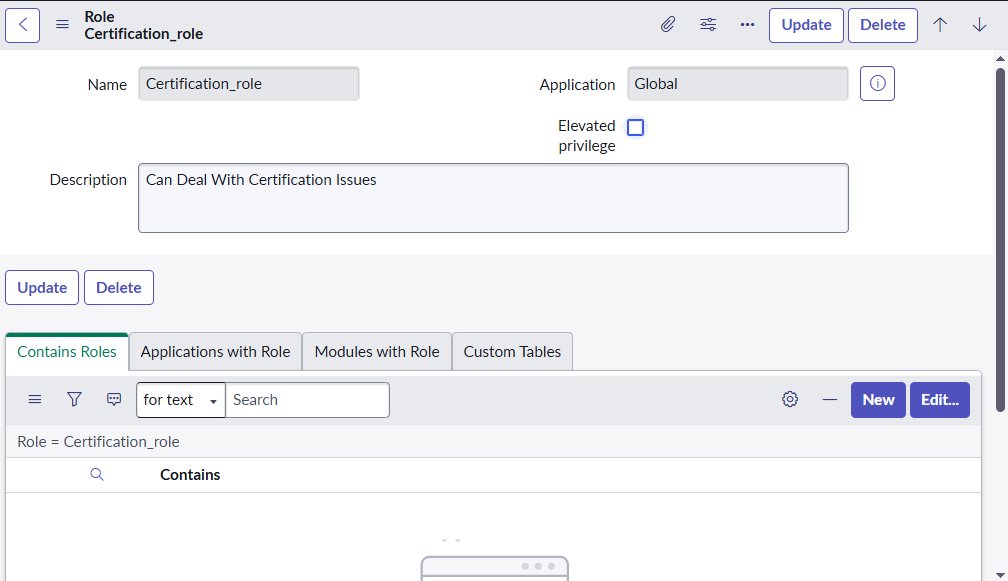
* Created:
  + Platform Group – Handles platform-related issues.
  + Certificates Group – Handles certification-related issues.

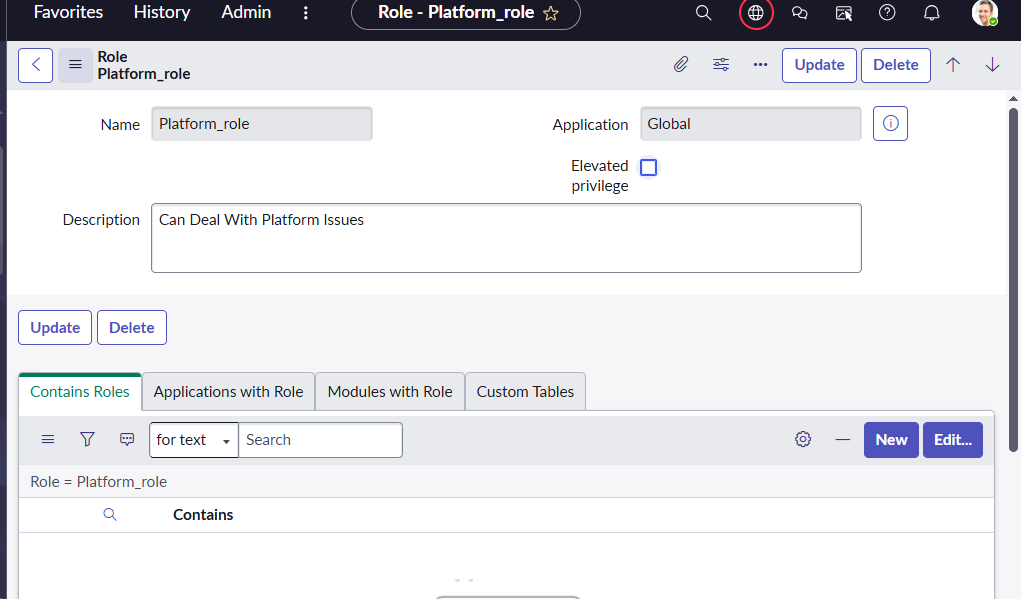




### Step 3: Role Creation

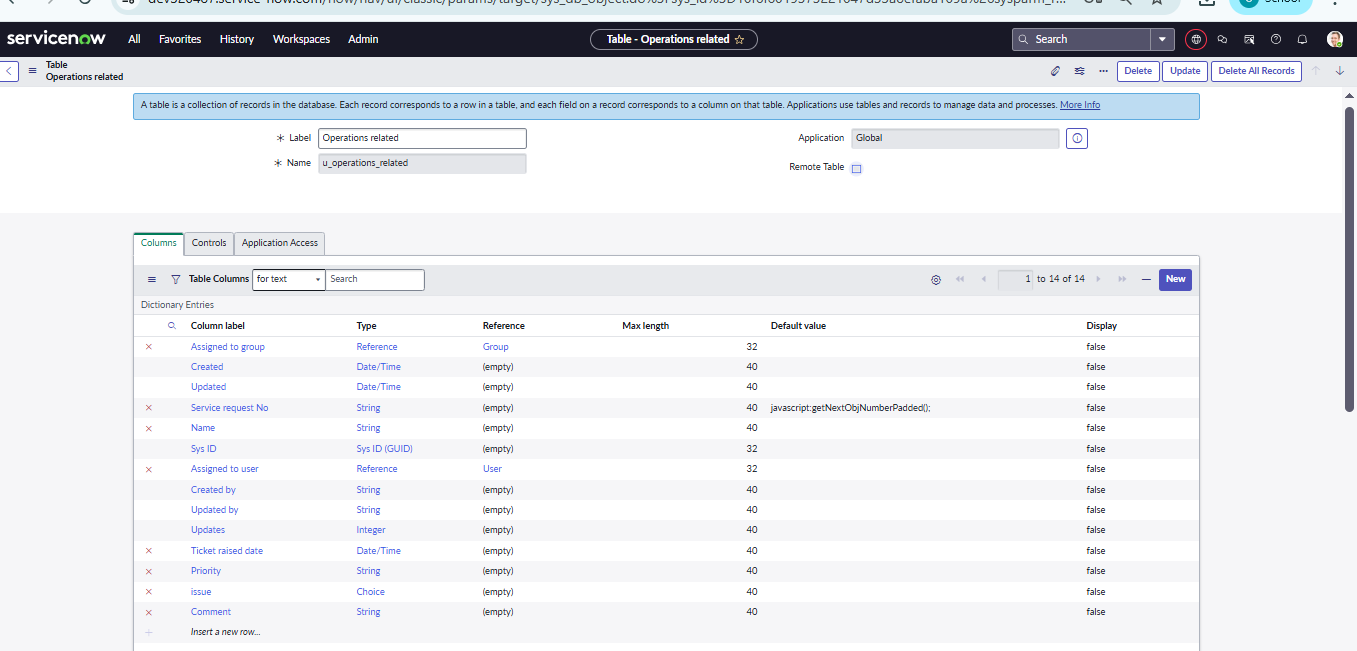
* Created:
  + platform\_role – Assigned to Platform Group.
  + certification\_role – Assigned to Certificates Group.





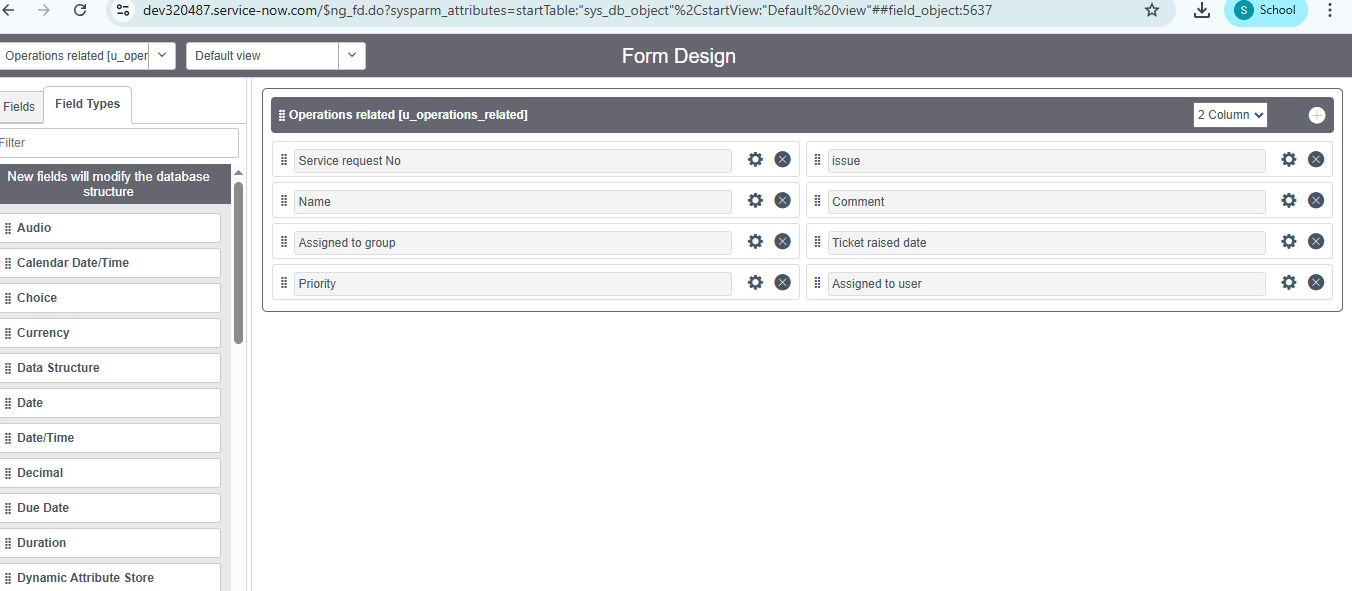
### Step 4: Table Creation

* Created custom table: **Operations Related (u\_operations\_related)**.
* Added fields for assignment, priority, ticket date, issue, and comments.
* Enabled Create Module & Create Mobile Module.



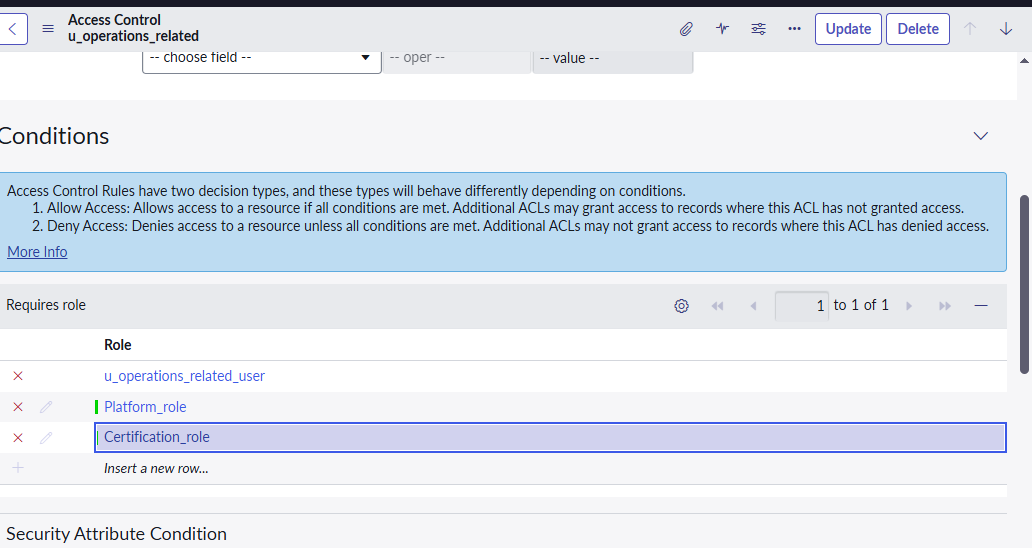
### Step 5: Choice Values

* Configured choices for issue field (login issues, 404 errors, certificates, etc.).



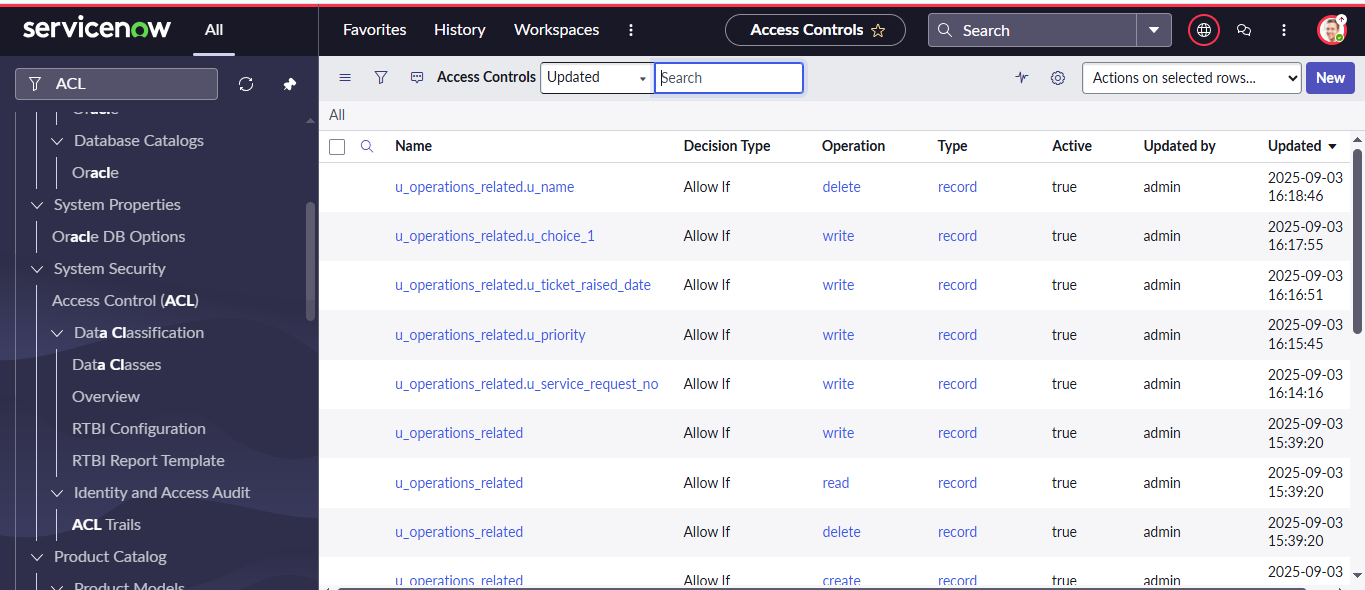
### Step 6: Assign Users and Roles

* Added users to their respective groups.
* Assigned roles to groups for access control.



### Step 7: Table Security

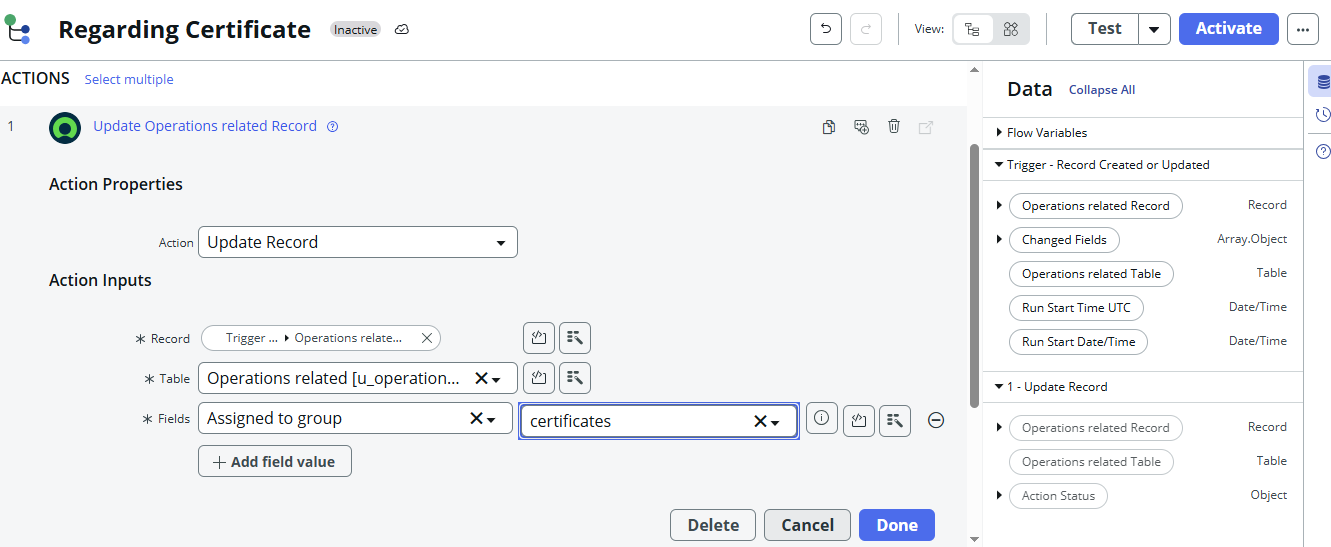
* Configured Application Access to require platform\_role and certification\_role.
* Created ACLs for create, read, write, and delete operations.



## 7. Flow Design

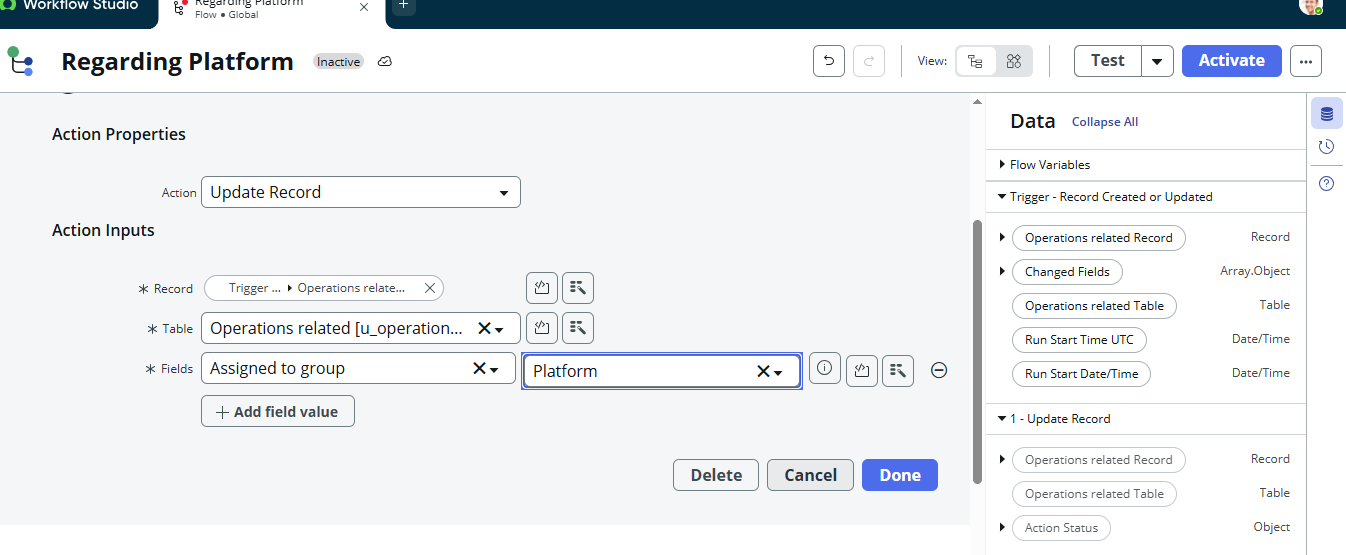
### Flow 1: Regarding Certificate

* **Trigger:** When a record is created or updated.
* **Condition:** Issue = “regarding certificates”.
* **Action:** Update record → Assigned to Group = Certificates.



### Flow 2: Regarding Platform

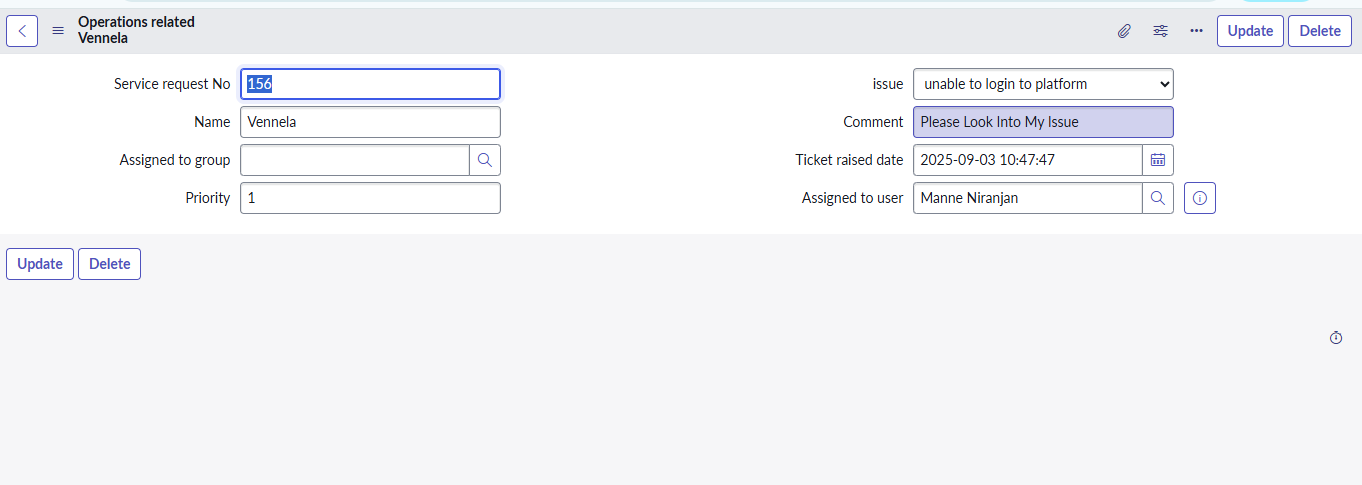
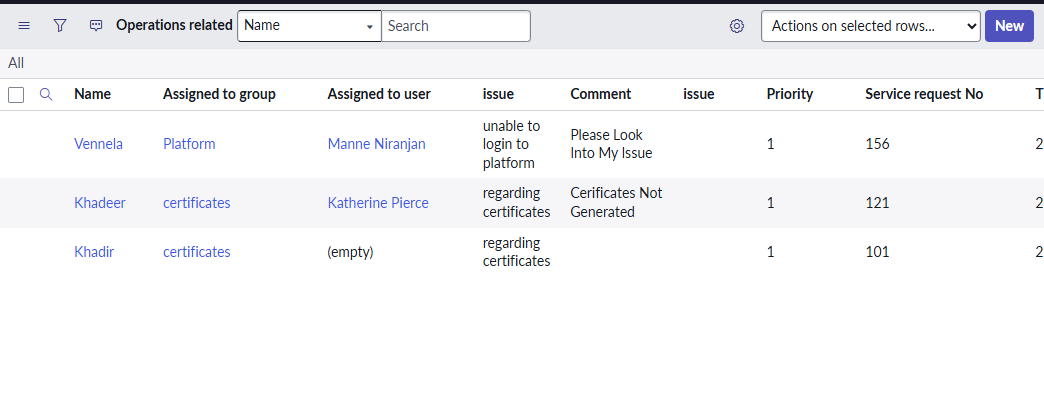
* **Trigger:** When a record is created or updated.
* **Condition:** Issue = “unable to login to platform” OR “404 error” OR “regarding user”.
* **Action:** Update record → Assigned to Group = Platform.



## 8. Testing and Results

|  |  |  |
| --- | --- | --- |
| **Name** | **Issue** | **Expected Assignment** |
| Khadir | regarding certificates | Certificates Group |
| Niranjan | unable to login | Platform Group |
| Alex | 404 error | Platform Group |

Verified in Flow Designer logs that flows executed successfully.

Tickets automatically updated with correct group assignment. 

## 9. Security and Governance

* Only users with the correct roles can view or modify table records.
* ACLs restrict unauthorized access.
* Flow Designer logs provide a full audit trail.

## 10. Benefits

* **90% reduction** in manual ticket triage time.
* Improved SLA compliance and faster response time.
* Reduced risk of misrouting tickets.

## 11. Future Enhancements

* Dynamic user assignment based on on-call schedules.
* Integration with MS Teams / Slack for instant notifications.
* Build dashboards for monitoring ticket assignment trends.
* Implement AI-based ticket routing using historical data.

## 12. References

* [ServiceNow Documentation](https://docs.servicenow.com/?utm_source=chatgpt.com)
* [GitHub Repository with Screenshots](https://github.com/KhadirShaikL21/Streamlined-Ticket-Automation/tree/main/screenshots?utm_source=chatgpt.com)