PROJECT TITLE:

Optimizing User, Group, and Role Management with Access Control and Workflows

TEAM ID: 005CF6AB0229EEBA2B747DB81FE82A19

TEAM MEMBERS:

- **SUBIKSHA. A**
- **NITHYASHREE. B**
- **GOPIKA. R**
- **MANGAIYARKARASI. P**

1. Ideation Phase

This phase focuses on identifying the problem and conceptualizing the solution.

- Modern project teams face challenges with unclear roles, lack of access control, and unstructured workflows.
- Problems identified include:
  - Ambiguity in task assignments
  - Inefficient progress tracking
  - Overlapping permissions
  - Lack of accountability
- The idea is to develop a role-based management framework to enforce proper permissions, ensure ownership, and improve collaboration.

Outcome of Ideation Phase:

A clear concept to create a system that optimizes user, group, and role management through access control and workflow automation.

2. Project Planning Phase

This phase involves setting the objectives and outlining the steps to achieve the project goals.

- Objectives:
  1. Define clear roles and responsibilities.
  2. Implement Role-Based Access Control (RBAC).
  3. Create structured workflows for task management.
  4. Improve visibility and accountability.
  5. Provide an intuitive dashboard for monitoring.
- System Overview:
  - Integration of user, group, and role management with access control and workflow automation.
  - Key components planned: User Management, Group Management, Role Management, Access Control, Workflow Management.

Outcome of Planning Phase:

A well-defined roadmap that outlines how the system will manage users, roles, and permissions to streamline project execution.

3. Project Design Phase

This phase deals with designing the system's structure and functionality.

- Design Components:
  1. User Management: Define user accounts and unique identifiers.

2. Group Management: Organize users based on roles or departments.
3. Role Management: Predefine roles (Manager, Member, Reviewer) with permissions.
4. Access Control: Apply RBAC to restrict user actions.
5. Workflow Management: Define states such as *To Do* → *In Progress* → *Review* → *Done*.

- Example Workflow:
  - Alice (Manager): Create and assign tasks, monitor progress.
  - Bob (Member): Update task status and add comments.

Outcome of Design Phase:
A blueprint of the system defining how components interact, ensuring secure, organized, and role-based task management.

4. Requirement Analysis Phase

This phase identifies and documents what is needed to implement the system.

- Functional Requirements:
  - Ability to create and manage users, roles, and groups.
  - Automated assignment of tasks and approvals via workflows.
  - Implementation of Access Control Lists (ACL) for defining permissions (read, write, delete).
  - Logging all actions and generating reports.
- Non-Functional Requirements:
  - Security: Restrict access based on user roles.
  - Usability: Provide an intuitive dashboard.
  - Efficiency: Automate processes to reduce manual intervention.

Outcome of Requirement Analysis Phase:
A comprehensive list of functional and non-functional requirements that serve as the foundation for system development.

Use Case Example

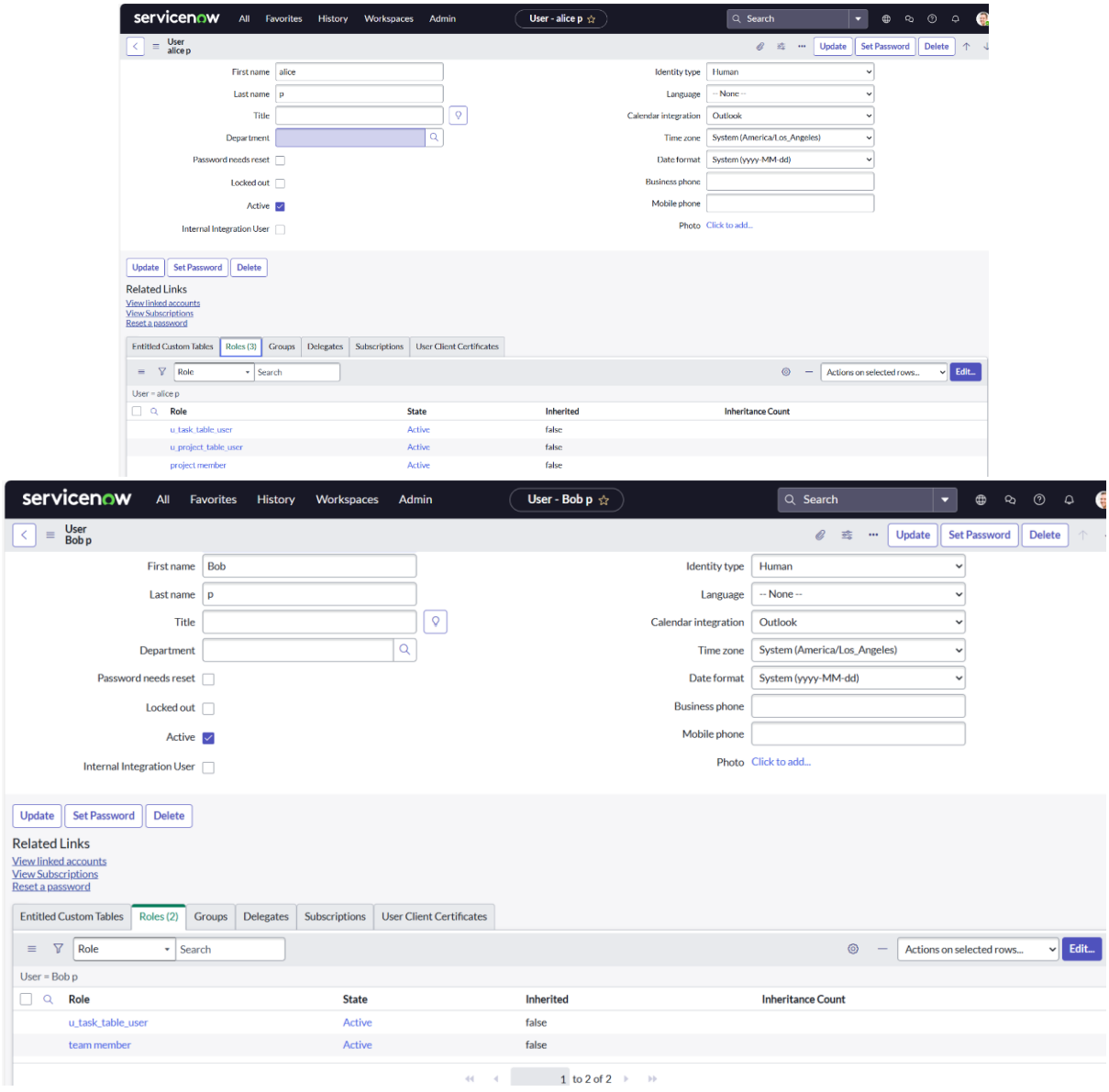
Actor	Action	Access Level
Alice (Manager)	Create task, assign to Bob, monitor progress	Full access
Bob (Team Member)	Update task status, add comments	Limited access
System	Log all actions, generate reports	Automated

This structured approach ensures that only authorized users can perform specific actions, reducing conflicts and improving clarity.

Implementation

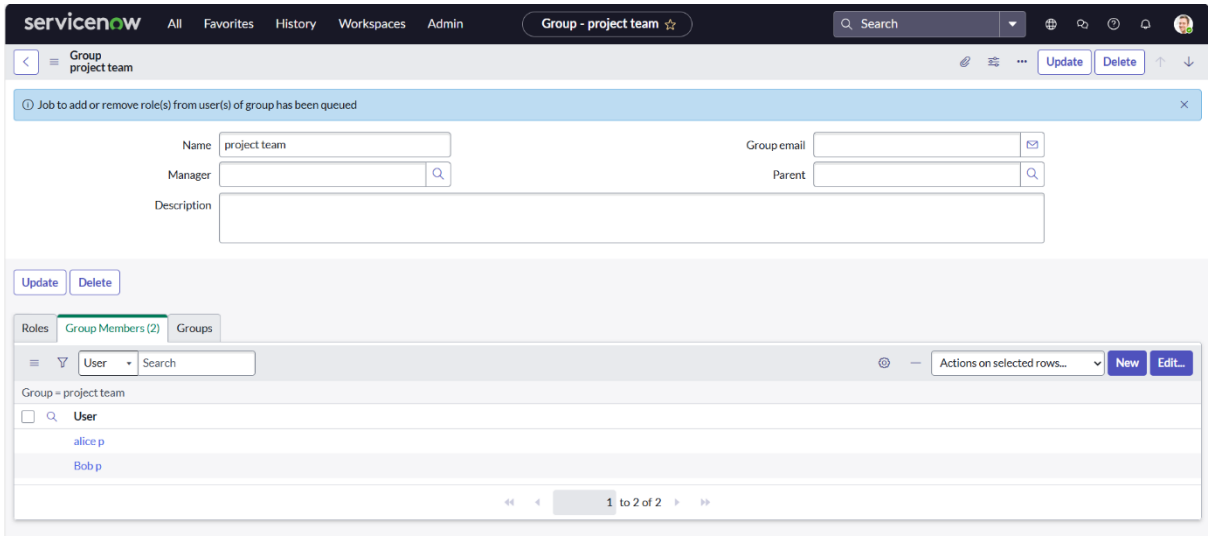
Step 1: User Management

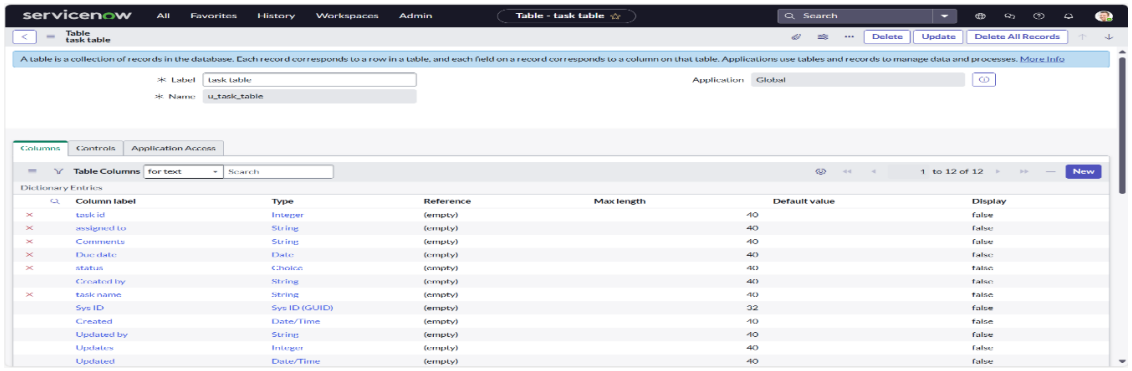
- Create and manage user accounts in the system.
- Define user details such as username, email, and role.
- Ensure each user has a unique identifier to maintain accountability.

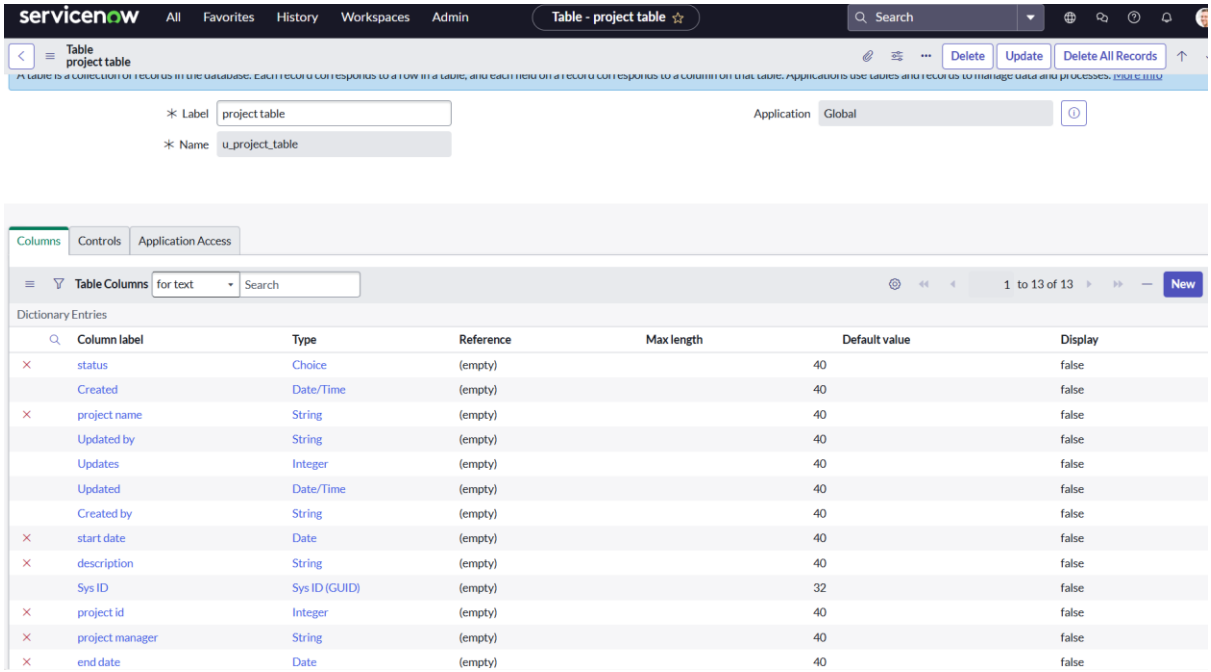


Step 2: Group Management

- Create groups to organize users based on their department, role, or project.
- Groups help in managing permissions collectively.
- Example: *Development Group*, *Operations Group*, *Testing Group.*


servicenow
All
Favorites
History
Workspaces
Admin
Group - project team
Search
Group
project team
Update
Delete
Job to add or remove role(s) from user(s) of group has been queued
Name
project team
Group email
Manager
Parent
Description
Update
Delete
Roles
Group Members (2)
Groups
User
Search
Actions on selected rows...
New
Edit...
Group = project team
User
alice p
Bob p
1 to 2 of 2


servicenow
Table - task table
Table
task table
Delete
Update
Delete All Records
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
task table
Name
u_task_table
Application
Global
Columns
Controls
Application Access
Table Columns
for text
Search
1 to 12 of 12
New
Dictionary Entries
Column label
Type
Reference
Max length
Default value
Display
task id
Integer
(empty)
40
false
assigned to
String
(empty)
40
false
Comments
String
(empty)
40
false
Due date
Date
(empty)
40
false
status
Choice
(empty)
40
false
Created by
String
(empty)
40
false
task name
String
(empty)
40
false
Sys ID
Sys ID (GUID)
(empty)
32
false
Created
Date/Time
(empty)
40
false
Updated by
String
(empty)
40
false
Updates
Integer
(empty)
40
false
Updated
Date/Time
(empty)
40
false


servicenow
All
Favorites
History
Workspaces
Admin
Table - project table
Search
Table
project table
Delete
Update
Delete All Records
Label
project table
Name
u_project_table
Application
Global
Columns
Controls
Application Access
Table Columns
for text
Search
1 to 13 of 13
New
Dictionary Entries
Column label
Type
Reference
Max length
Default value
Display
status
Choice
(empty)
40
false
Created
Date/Time
(empty)
40
false
project name
String
(empty)
40
false
Updated by
String
(empty)
40
false
Updates
Integer
(empty)
40
false
Updated
Date/Time
(empty)
40
false
Created by
String
(empty)
40
false
start date
Date
(empty)
40
false
description
String
(empty)
40
false
Sys ID
Sys ID (GUID)
(empty)
32
false
project id
Integer
(empty)
40
false
project manager
String
(empty)
40
false
end date
Date
(empty)
40
false

Step 3: Role Management

- Define different roles such as *Project Manager*, *Team Member*, *Reviewer*, etc.
- Each role determines what actions a user can perform in the system.
- Example:
    - Manager: Create and assign tasks.
    - Member: Update and complete assigned tasks.

Step 4: Assign Users to Groups

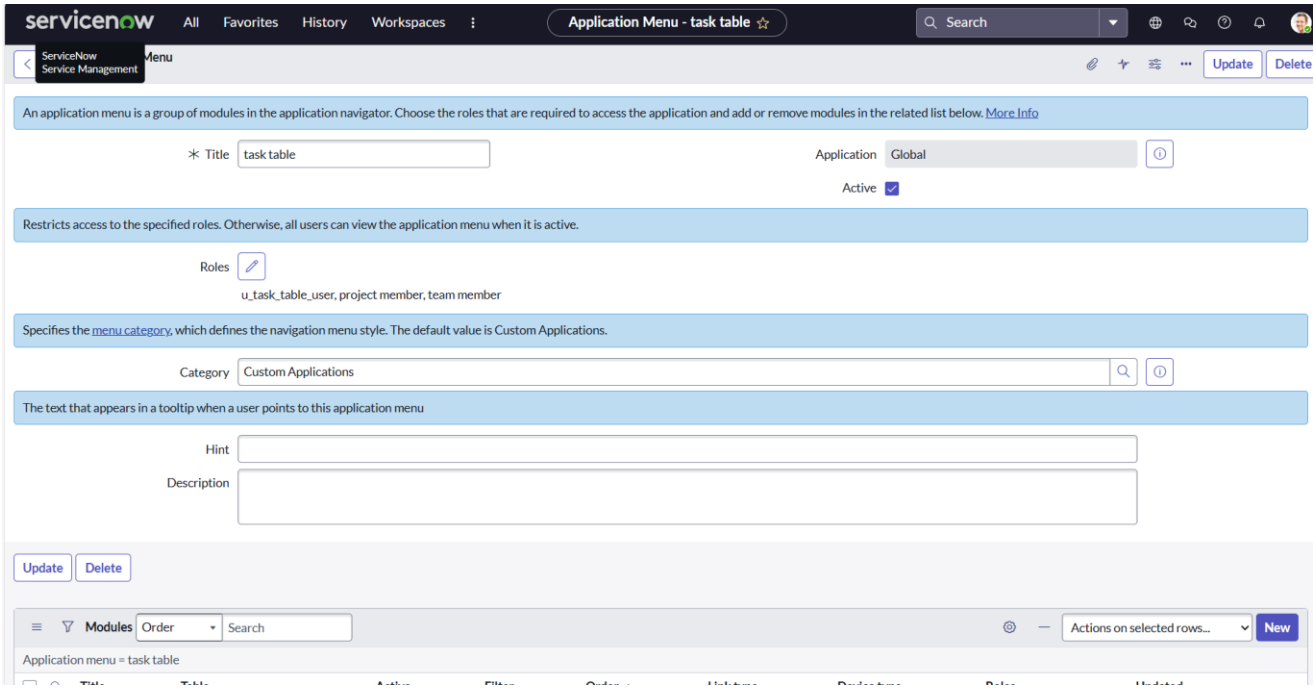
- Assign each user to one or more groups depending on their responsibilities.
- This ensures that group-level permissions are automatically applied to the users.
- Example: Alice → Project Managers group, Bob → Team Members group.

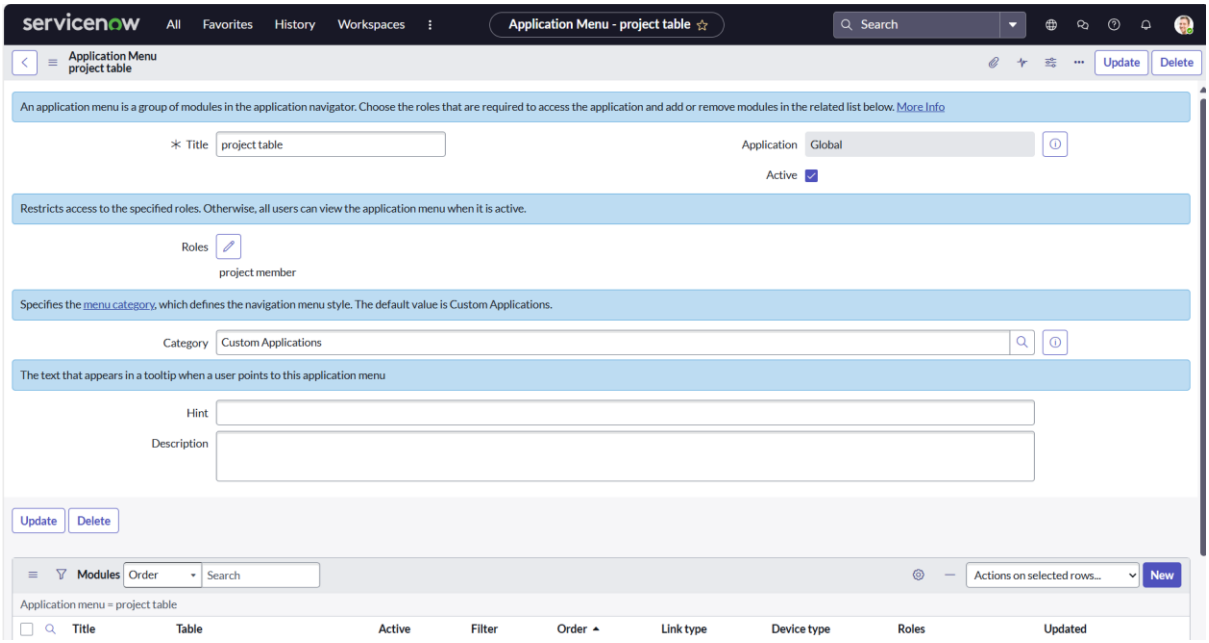
Step 5: Application Access

- Define access levels for different applications within the system.
- This step ensures that only authorized groups or roles can access specific modules or data.

Sub-step: Assign Table Access to Application

- Grant access to database tables or records required by the application.
- Restrict read, write, and delete permissions based on the user's role or group.



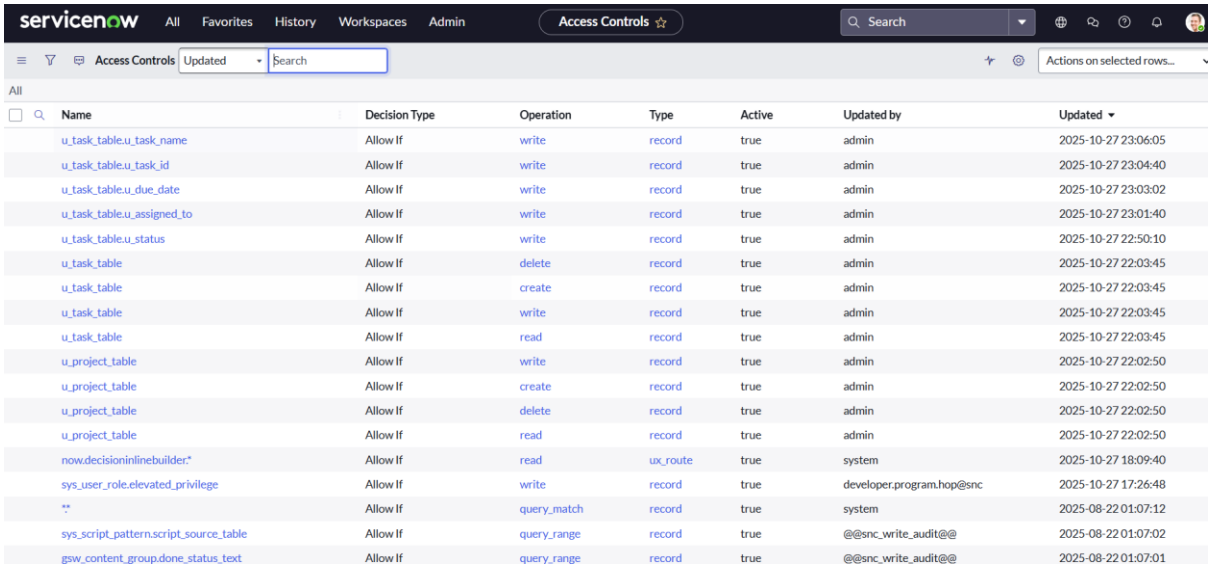


Step 6: Access Control List (ACL)

- Create an Access Control List (ACL) to manage what actions each role can perform on the system resources.

Sub-step: Create ACL

- Define explicit permissions in the ACL such as:
    - Read access → Allowed for all users.
    - Write access → Allowed for Managers only.
    - Delete access → Restricted to Admins.



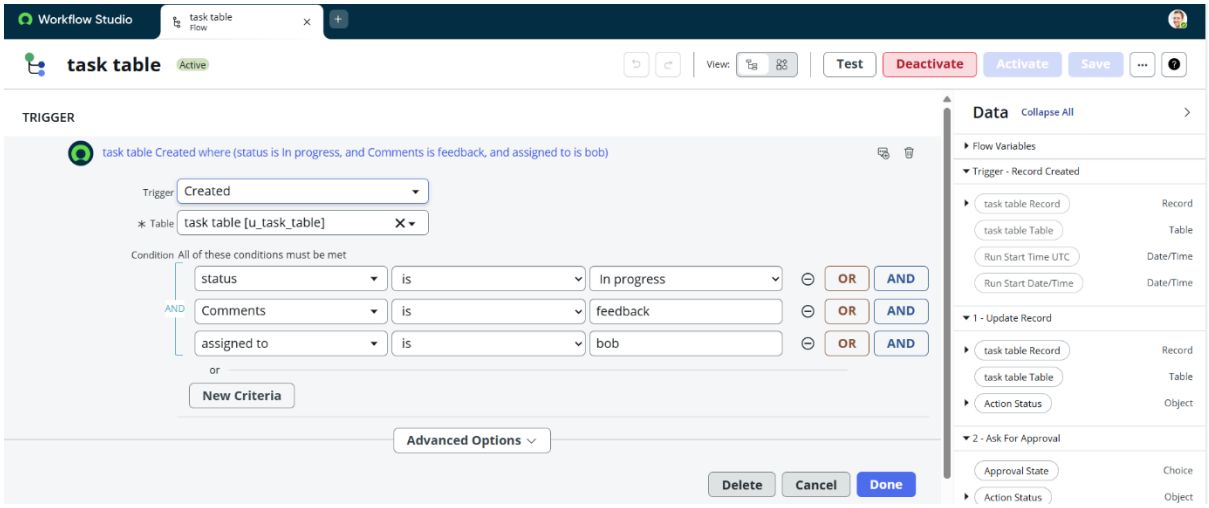
Name	Decision Type	Operation	Type	Active	Updated by	Updated
u_task_table.u_task_name	Allow If	write	record	true	admin	2025-10-27 23:06:05
u_task_table.u_task_id	Allow If	write	record	true	admin	2025-10-27 23:04:40
u_task_table.u_due_date	Allow If	write	record	true	admin	2025-10-27 23:03:02
u_task_table.u_assigned_to	Allow If	write	record	true	admin	2025-10-27 23:01:40
u_task_table.u_status	Allow If	write	record	true	admin	2025-10-27 22:50:10
u_task_table	Allow If	delete	record	true	admin	2025-10-27 22:03:45
u_task_table	Allow If	create	record	true	admin	2025-10-27 22:03:45
u_task_table	Allow If	write	record	true	admin	2025-10-27 22:03:45
u_task_table	Allow If	read	record	true	admin	2025-10-27 22:03:45
u_project_table	Allow If	write	record	true	admin	2025-10-27 22:02:50
u_project_table	Allow If	create	record	true	admin	2025-10-27 22:02:50
u_project_table	Allow If	delete	record	true	admin	2025-10-27 22:02:50
u_project_table	Allow If	read	record	true	admin	2025-10-27 22:02:50
now.decisioninlinebuilder.*	Allow If	read	ux_route	true	system	2025-10-27 18:09:40
sys_user_role.elevated_privilege	Allow If	write	record	true	developer.program.hop@snc	2025-10-27 17:26:48
**	Allow If	query_match	record	true	system	2025-08-22 01:07:12
sys_script_pattern.script_source_table	Allow If	query_range	record	true	@@snc_write_audit@@	2025-08-22 01:07:02
gsw_content_group.done_status_text	Allow If	query_range	record	true	@@snc_write_audit@@	2025-08-22 01:07:01

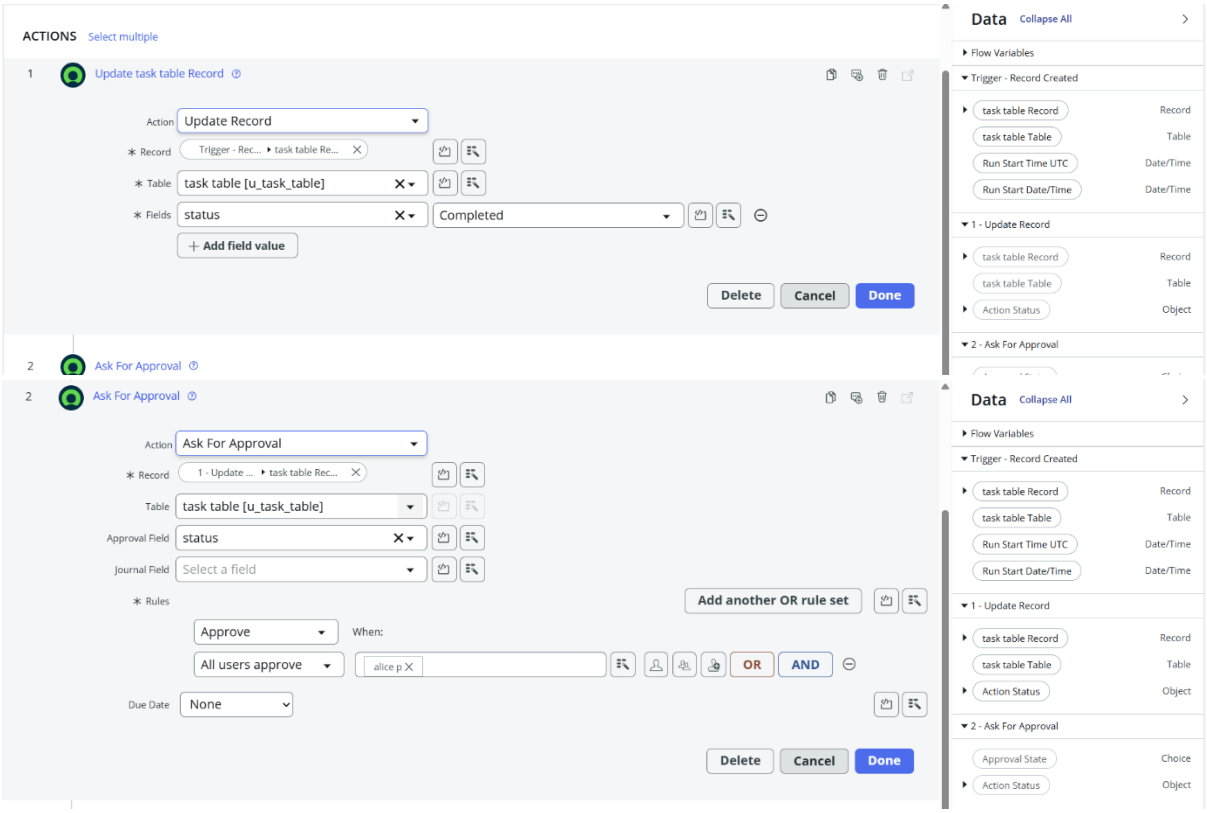
Step 7: Flow (Workflow Automation)

- Design workflows to automate task assignments and approvals.
- Workflows ensure that the right person receives the right task at the right time.

Sub-step: Create a Flow to Assign Operations Ticket to Group

- Create an automated flow that assigns operational tickets to a specific group (e.g., Operations Group).
- When a new operations request is created, the system automatically routes it to the designated group.
- Improves efficiency and eliminates manual intervention.

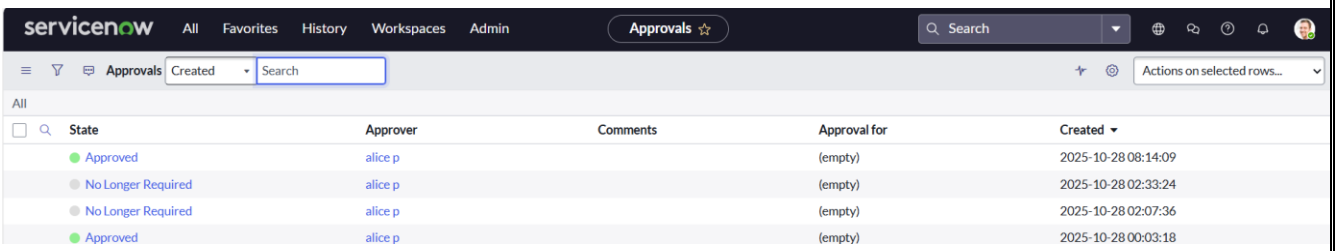




1. Outcome

After completing all the above steps:

- Roles and access are clearly defined.
- Users and groups are organized logically.
- Access is controlled securely through ACLs.
- Tasks and workflows are automated for better accountability and efficiency.



2. Conclusion

This project successfully demonstrates how optimizing user, group, and role management using access control and workflow automation enhances efficiency in project management.

By defining structured roles, secure permissions, and task workflows, the system ensures transparency, accountability, and scalability for growing project teams.