

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

1. Introduction

In modern project environments, effective management of users, roles, and access controls is critical to ensure efficiency, transparency, and accountability. Small project teams often struggle with clearly defining responsibilities, maintaining security boundaries, and tracking task progress systematically.

This project aims to design and implement a structured system that optimizes user, group, and role management, with clearly defined workflows and access control mechanisms to streamline task management.

2. Problem Statement

In a small project management team comprising:

- Project Manager (Alice)
- Team Member (Bob)

the existing system lacks clear role definitions, proper access control, and structured workflows.

This results in:

- Ambiguity in task assignments
- Inefficient progress tracking
- Overlapping permissions and responsibilities
- Lack of accountability in project execution

Hence, there is a need to build a role-based management framework that enforces appropriate permissions, ensures task ownership, and improves collaboration.

3. Objectives

The primary objectives of this project are:

1. To define clear roles and responsibilities for users in the system.
2. To implement role-based access control (RBAC) for managing user privileges.
3. To create structured workflows for task creation, assignment, review, and completion.
4. To enhance task visibility and accountability among project team members.
5. To provide an intuitive dashboard for monitoring project status and user activities.

4. System Overview

The proposed system introduces user, group, and role management integrated with an access control mechanism and workflow automation.

Key Components:

1. User Management: Handles creation, modification, and removal of users.
2. Group Management: Groups users based on project teams or departments.
3. Role Management: Assigns predefined roles (e.g., Manager, Member, Reviewer) with specific permissions.
4. Access Control: Implements RBAC to restrict actions based on user roles.
5. Workflow Management: Defines task states (e.g., *To Do* → *In Progress* → *Review* → *Done*) to streamline operations.

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.

5. STEPS

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.

The screenshot shows the ServiceNow user management interface for a user named 'alice.p'. The interface includes a top navigation bar with tabs for 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. The user's name 'User - alice.p' is displayed in the top right corner. Below the navigation bar, there are buttons for 'Update', 'Set Password', and 'Delete'. The main form contains fields for 'First name' (alice), 'Last name' (p), 'Title' (empty), 'Department' (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). There are also checkboxes for 'Password needs reset', 'Locked out', 'Active' (checked), and 'Internal Integration User'. Below the form, there are links for 'Update', 'Set Password', and 'Delete'. The 'Related Links' section includes 'View linked accounts', 'View Subscriptions', and 'Reset a password'. The 'Entitled Custom Tables' section shows a table with columns 'Role', 'State', 'Inherited', and 'Inheritance Count'. The table contains three rows: 'u.task.table.user' (Active, false), 'u.project.table.user' (Active, false), and 'project.member' (Active, false).

servicenow All Favorites History Workspaces Admin User - Bob p

Search

User Bob p

First name: Bob
Last name: p
Title:
Department:
Password needs reset:
Locked out:
Active: ☒
Internal Integration User:
Identity type: Human
Language: -- None --
Calendar integration: Outlook
Time zone: System (America/Los_Angeles)
Date format: System (yyyy-MM-dd)
Business phone:
Mobile phone:
Photo: [Click to add...](#)

Update Set Password Delete

Related Links
[View linked accounts](#)
[View Subscriptions](#)
[Reset a password](#)

Entitled Custom Tables Roles (2) Groups Delegates Subscriptions User Client Certificates

Role Search

User = Bob p

Role	State	Inherited	Inheritance Count
u_task_table_user	Active	false	
team member	Active	false	

1 to 2 of 2

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

Job to add or remove role(s) from user(s) of group has been queued

Name: project team
Manager:
Description:
Group email:
Parent:
Update Delete

Roles Group Members (2) Groups

User Search

Group = project team

User
alice p
Bob p

1 to 2 of 2

servicenow All Favorites History Workspaces Admin Table - task table

Search

Table task 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: task table
Name: u_task_table
Application: Global

Columns Controls Application Access

Table Columns for text

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
Updated	Integer	(empty)	40		false
Updated	Date/Time	(empty)	40		false

ServiceNow Table - project table

* Label: project table

* Name: u_project_table

Application: Global

Columns Controls Application Access

Table Columns for text Search 1 to 13 of 13 New

Column label	Type	Reference	Max length	Default value	Display
status	Choice	(empty)	40	false	false
Created	Date/Time	(empty)	40	false	false
project name	String	(empty)	40	false	false
Updated by	String	(empty)	40	false	false
Updates	Integer	(empty)	40	false	false
Updated	Date/Time	(empty)	40	false	false
Created by	String	(empty)	40	false	false
start date	Date	(empty)	40	false	false
description	String	(empty)	40	false	false
Sys ID	Sys ID (GUID)	(empty)	32	false	false
project id	Integer	(empty)	40	false	false
project manager	String	(empty)	40	false	false
end date	Date	(empty)	40	false	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.

servicenow All Favorites History Workspaces Application Menu - task table Search

ServiceNow Service Management Menu Update Delete

An application menu is a group of modules in the application navigator. Choose the roles that are required to access the application and add or remove modules in the related list below. [More Info](#)

* Title Application

Active ☒

Restricts access to the specified roles. Otherwise, all users can view the application menu when it is active.

Roles
u_task_table_user, project member, team member

Specifies the [menu.category](#), which defines the navigation menu style. The default value is Custom Applications.

Category

The text that appears in a tooltip when a user points to this application menu

Hint

Description

Modules Order Search Actions on selected rows...

Application menu = task table

servicenow All Favorites History Workspaces Application Menu - project table Search

Application Menu project table Update Delete

An application menu is a group of modules in the application navigator. Choose the roles that are required to access the application and add or remove modules in the related list below. [More Info](#)

* Title Application

Active ☒

Restricts access to the specified roles. Otherwise, all users can view the application menu when it is active.

Roles
project member

Specifies the [menu.category](#), which defines the navigation menu style. The default value is Custom Applications.

Category

The text that appears in a tooltip when a user points to this application menu

Hint

Description

Modules Order Search Actions on selected rows...

Application menu = project table

Title	Table	Active	Filter	Order	Link type	Device type	Roles	Updated

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.

servicenow							
All Favorites History Workspaces Admin Access Controls							
Access Controls Updated Search							
All							
	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
	gsn_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.

Workflow Studio
task table
Flow
task table
Active
View:
Test
Deactivate
Activate
Save
Delete
Cancel
Done

task table Created where (status is In progress, and Comments is feedback, and assigned to is bob)

Trigger
Created
* Table
task table [u_task_table]
Condition All of these conditions must be met

status
is
In progress

Comments
is
feedback

assigned to
is
bob

New Criteria

Advanced Options

Data
Collapse All

Flow Variables

Trigger - Record Created

task table Record
Record
task table Table
Table
Run Start Time UTC
Date/Time
Run Start Date/Time
Date/Time

1 - Update Record

task table Record
Record
task table Table
Table
Action Status
Object

2 - Ask For Approval

Approval State
Choice
Action Status
Object

ACTIONS Select multiple

1

Update task table Record

Action

Update Record

* Record

Trigger - Rec... → task table Re...

* Table

task table [u_task_table]

* Fields

status

Completed

+ Add field value

Delete

Cancel

Done

2

Ask For Approval

Action

Ask For Approval

* Record

1 - Update ... → task table Rec...

Table

task table [u_task_table]

Approval Field

status

Journal Field

Select a field

* Rules

Approve

When:

All users approve

alice p X

OR

AND

Due Date

None

Add another OR rule set

Delete

Cancel

Done

Data Collapse All

Flow Variables

Trigger - Record Created

task table Record

Record

task table Table

Table

Run Start Time UTC

Date/Time

Run Start Date/Time

Date/Time

1 - Update Record

task table Record

Record

task table Table

Table

Action Status

Object

2 - Ask For Approval

Approval State

Choice

Action Status

Object

6. 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.

servicenow All Favorites History Workspaces Admin Approvals Search					
Approvals Created Search Actions on selected rows...					
All	State	Approver	Comments	Approval for	Created
<input type="checkbox"/>	Approved	alice p		(empty)	2025-10-28 08:14:09
<input type="checkbox"/>	No Longer Required	alice p		(empty)	2025-10-28 02:33:24
<input type="checkbox"/>	No Longer Required	alice p		(empty)	2025-10-28 02:07:36
<input type="checkbox"/>	Approved	alice p		(empty)	2025-10-28 00:03:18

7. 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.