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

Team ID : LTVIP2025TMID30165

Team Size : 4

Team Leader : Vinayak Chinimilli

Team member : Pedasingu Ramcharan Pavan Teja

Team member : N Sandeep

Team member : Shaik Suhail

---

1. INTRODUCTION

1.1 Project Overview

In this project, we simulate a mini project management system using ServiceNow, focusing on structured user, group, and role management. It involves the creation of two user personas: Alice (Project Manager) and Bob (Team Member). We establish a secure and automated task tracking system using access control lists (ACLs) and Flow Designer. The system enables Bob to create and update tasks and Alice to review them and grant approvals.

1.2 Purpose

The primary purpose of this project is to improve task management accountability and efficiency by implementing role-based access controls and automated workflows. It aims to eliminate confusion in task assignments, enforce permissions using roles and ACLs, and automate parts of the task lifecycle using Flow Designer.

---

2. IDEATION PHASE

2.1 Problem Statement

In a small project team, lack of defined roles and permissions can lead to miscommunication, duplicate work, and data inconsistency. This project addresses the need to: - Assign responsibilities based on roles - Restrict unauthorized field access - Enable approvals before marking tasks complete - Simplify and automate task status changes

2.2 Brainstorming Highlights

During brainstorming, we decided to: - Create two key users (Alice and Bob) - Build project and task tables for storing records - Use Flow Designer to automate task progress - Use impersonation to test ACLs and flows - Provide edit access only to authorized users via roles like `team_member` and `project_member`

---

3. REQUIREMENT ANALYSIS

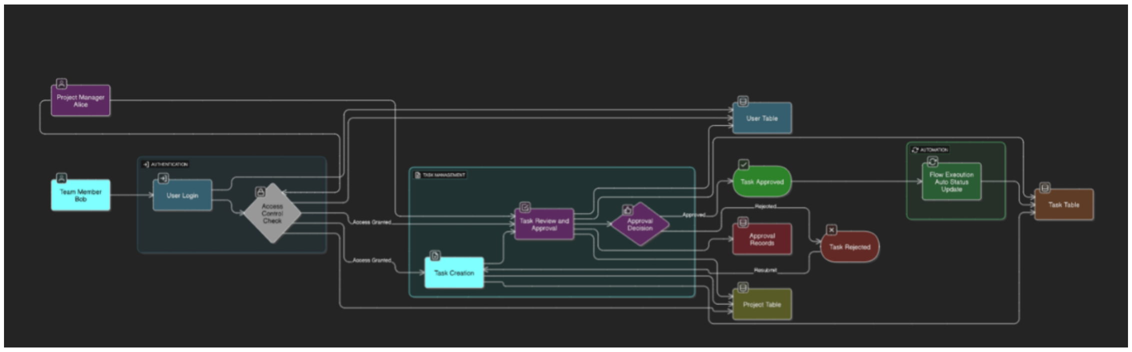
3.1 Functional Requirements

- Create two tables: `u_project_table` and `u_task_table_2`
- Create users: Alice and Bob
- Define groups and roles: `Project Team Group`, `team_member`, `project_member`, `u_task_table_2_user`
- Assign users to groups and roles appropriately
- Setup ACLs for edit rights to specific fields
- Create a Flow using Flow Designer to:
    - Trigger on record creation
    - Auto-update status to "completed"
    - Ask for approval from Alice

3.2 Non-Functional Requirements

- System should respond securely based on user roles
- Flows should execute reliably under defined conditions
- Design must be modular and scalable for future enhancements
- Navigation and testing should be user-friendly via impersonation

3.3 Data Flow Diagram



---

4. PROJECT DESIGN

4.1 Role Structure

- **Alice** is a Project Manager assigned the roles `project_member` and `u_task_table_2_user`. She can view, update, and approve tasks.
- **Bob** is a Team Member assigned the roles `team_member` and `u_task_table_2_user`. He can create and edit tasks but cannot approve them.
- Both are assigned to the **Project Team Group**, which facilitates group-based permission management.

4.2 Tables Created

- **`u_project_table`**: Holds project metadata like Project Name, Start Date, Description, and Owner.
- **`u_task_table_2`**: Stores task-specific details including Task Name, Task ID, Status, Assigned To, Due Date, and Comments.

4.3 ACL Setup

Access Control Lists were defined as follows: - Write ACLs for fields like `status`, `comments`, `task_name`, and `assigned_to` - `team_member` role allows Bob to edit only select fields - Alice has additional access via `project_member` and table roles - ACLs were tested using impersonation to simulate role-based restrictions

---

5. PROJECT PLANNING & SCHEDULING

5.1 Planning Timeline

- **Week1**: Requirements gathering and user-role planning.
- **Week 2**: Creation of tables, users, roles, and groups.
- **Week 3**: ACL configuration and testing using impersonation.
- **Week 4**: Designing and implementing the flow in Flow Designer.
- **Week 5**: Testing and final validation with role-based users.

---

6. TESTING

6.1 User Testing

- **Bob**: Logged in via impersonation and successfully created a task
- **Alice**: Reviewed task, verified fields, and edited comments
- **Other users**: Lacked roles, received ACL errors on restricted fields

6.2 Flow Testing

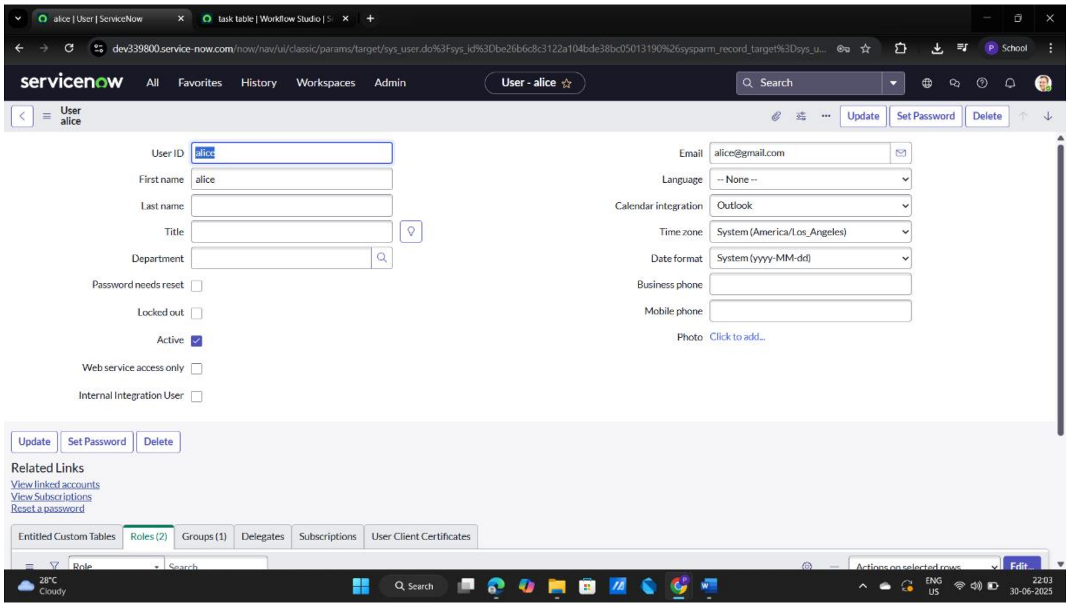
- Task created by Bob triggered the Flow

- Status was updated to “completed” automatically
- Flow logs confirmed execution of the update and approval steps

---

7. RESULTS

Successfully created users, roles, tables, and flows

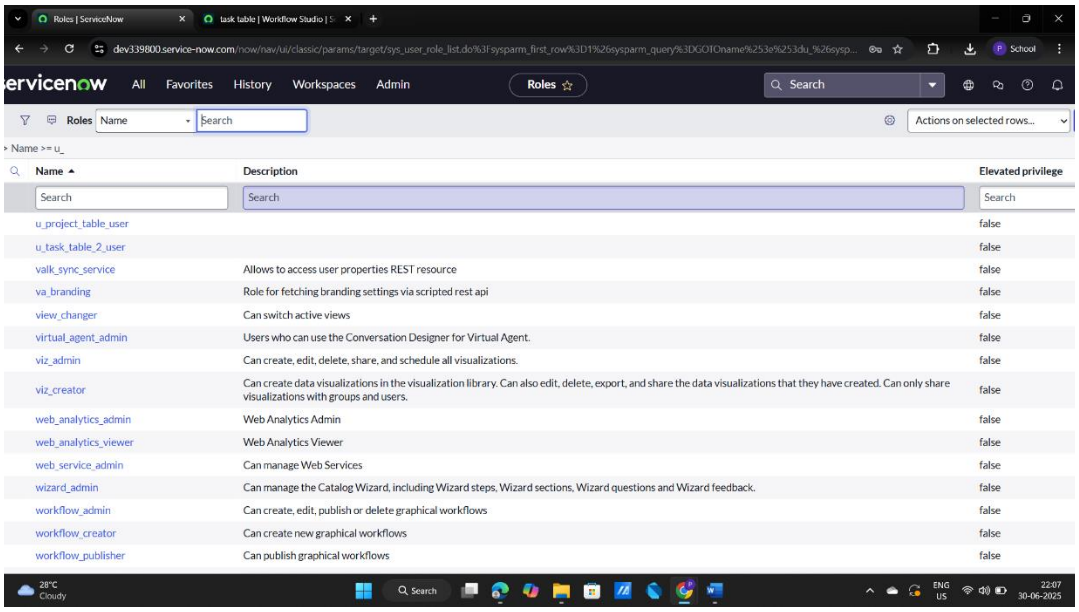

alice | User | ServiceNow
task table | Workflow Studio | S
dev339800.service-now.com/now/nav/ui/classic/params/target/sys_user.do%3Fsys_id%3Dbe26b6c8c3122a104bde38bc05013190%26sysparm_record_target%3Dsys_u...
servicenow
All
Favorites
History
Workspaces
Admin
User - alice
Search
Update
Set Password
Delete

User
alice

User ID	alice	Email	alice@gmail.com
First name	alice	Language	-- None --
Last name		Calendar integration	Outlook
Title		Time zone	System (America/Los_Angeles)
Department		Date format	System (yyyy-MM-dd)
Password needs reset	☐	Business phone	
Locked out	☐	Mobile phone	
Active	☑	Photo	Click to add...
Web service access only	☐		
Internal Integration User	☐		

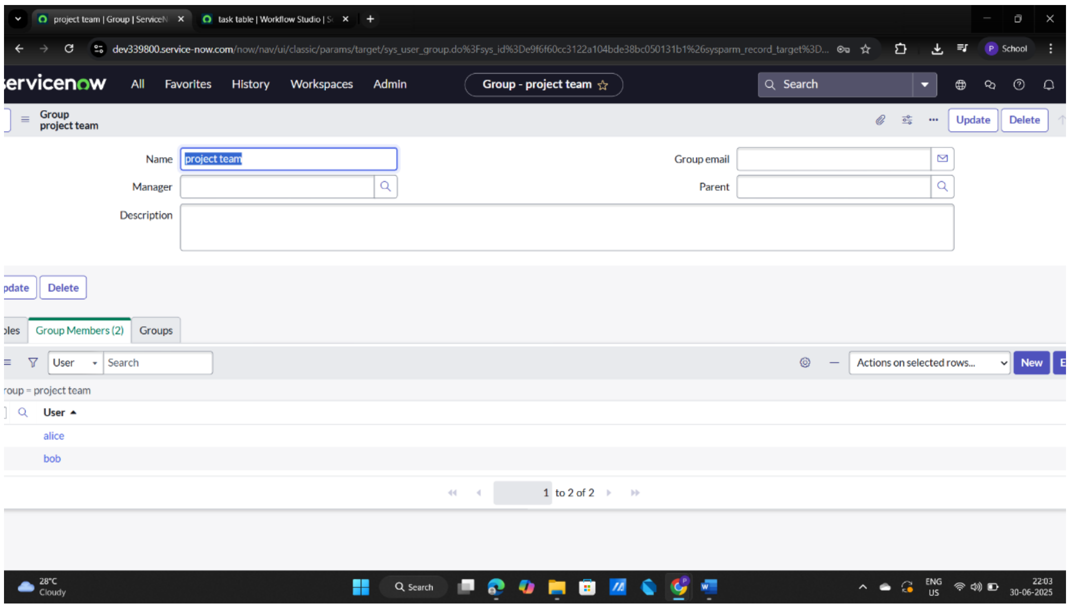
Update
Set Password
Delete

Related Links
View linked accounts
View Subscriptions
Reset a password


Roles | ServiceNow
task table | Workflow Studio | S
dev339800.service-now.com/now/nav/ui/classic/params/target/sys_user_role_list.do%3Fsysparm_first_row%3D1%26sysparm_query%3DGOTOname%253e%253du_%26sysp...
servicenow
All
Favorites
History
Workspaces
Admin
Roles
Search
Roles
Name
Search
Actions on selected rows...

> Name >= u_

Name ▲	Description	Elevated privilege
u_project_table_user		false
u_task_table_2_user		false
valk_sync_service	Allows to access user properties REST resource	false
va_branding	Role for fetching branding settings via scripted rest api	false
view_changer	Can switch active views	false
virtual_agent_admin	Users who can use the Conversation Designer for Virtual Agent.	false
viz_admin	Can create, edit, delete, share, and schedule all visualizations.	false
viz_creator	Can create data visualizations in the visualization library. Can also edit, delete, export, and share the data visualizations that they have created. Can only share visualizations with groups and users.	false
web_analytics_admin	Web Analytics Admin	false
web_analytics_viewer	Web Analytics Viewer	false
web_service_admin	Can manage Web Services	false
wizard_admin	Can manage the Catalog Wizard, including Wizard steps, Wizard sections, Wizard questions and Wizard feedback.	false
workflow_admin	Can create, edit, publish or delete graphical workflows	false
workflow_creator	Can create new graphical workflows	false
workflow_publisher	Can publish graphical workflows	false



Group
project team

Name	project team
Manager	
Description	
Group email	
Parent	

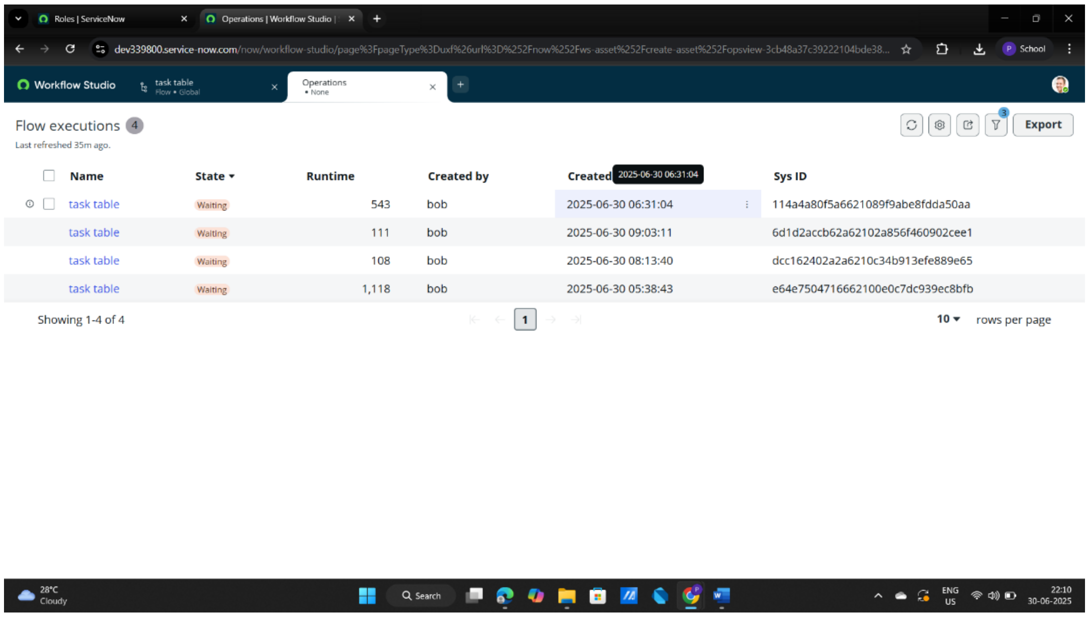
Group Members (2)

roup = project team

User ▲
alice
bob

1 to 2 of 2

Task status updated automatically based on flow conditions.



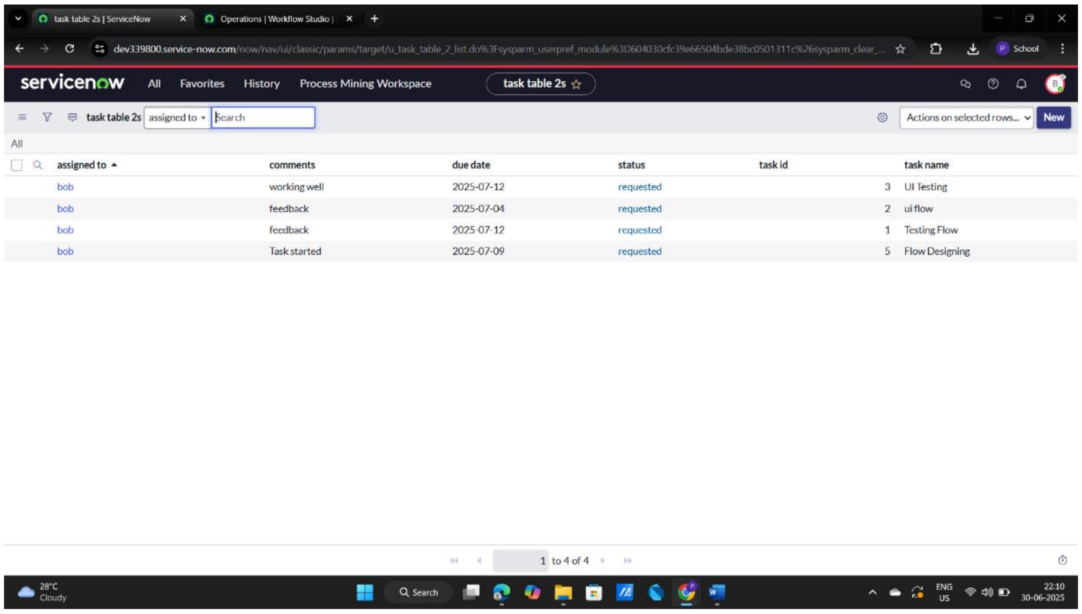
Flow executions 4

Last refreshed 35m ago.

Name	State	Runtime	Created by	Created	Sys ID
task table	Waiting	543	bob	2025-06-30 06:31:04	114a4a80f5a6621089f9abe8fdda50aa
task table	Waiting	111	bob	2025-06-30 09:03:11	6d1d2accb62a62102a856f460902cee1
task table	Waiting	108	bob	2025-06-30 08:13:40	dcc162402a2a6210c34b913efe889e65
task table	Waiting	1,118	bob	2025-06-30 05:38:43	e64e7504716662100e0c7dc939ec8bfb

Showing 1-4 of 4

Flow execution logs confirmed all steps were triggered in sequence.

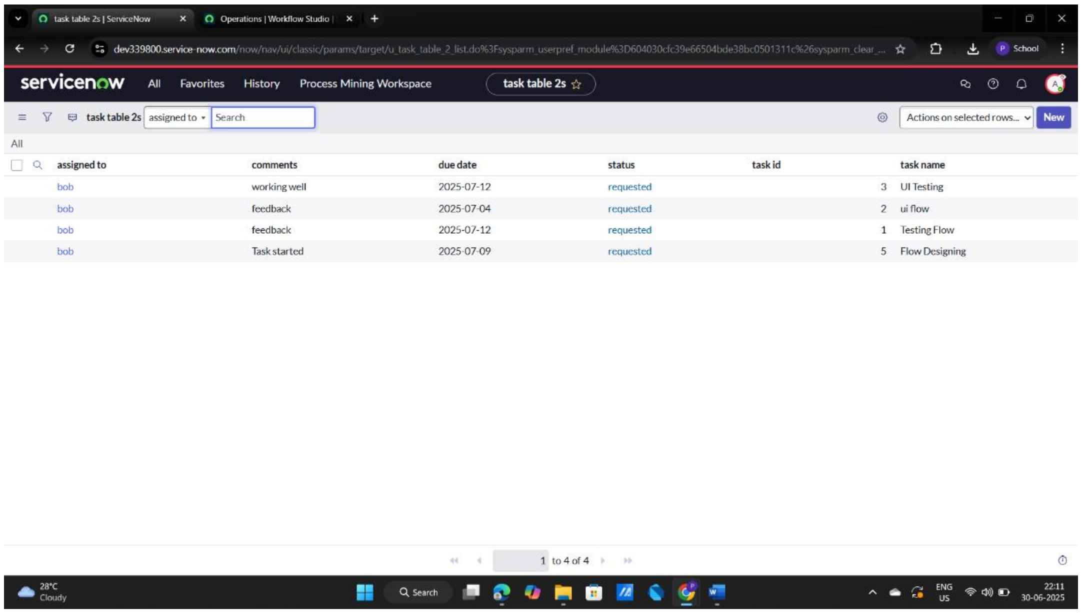


task table 2s | assigned to | Search

All

assigned to	comments	due date	status	task id	task name
bob	working well	2025-07-12	requested	3	UI Testing
bob	feedback	2025-07-04	requested	2	ui flow
bob	feedback	2025-07-12	requested	1	Testing Flow
bob	Task started	2025-07-09	requested	5	Flow Designing

1 to 4 of 4



task table 2s | assigned to | Search

All

assigned to	comments	due date	status	task id	task name
bob	working well	2025-07-12	requested	3	UI Testing
bob	feedback	2025-07-04	requested	2	ui flow
bob	feedback	2025-07-12	requested	1	Testing Flow
bob	Task started	2025-07-09	requested	5	Flow Designing

1 to 4 of 4

---

7. ADVANTAGES & CHALLENGES

Advantages

- Realistic simulation of enterprise task management
- Strong security model using ACLs and roles

- Efficient automation with Flow Designer
- Easy impersonation-based testing

Disadvantages

- Requires detailed knowledge of ServiceNow internals (ACL, modules)
- Approval modules can be difficult to configure correctly
- Visual module access may depend on roles and application menus

---

9. CONCLUSION

This project successfully demonstrates how ServiceNow can be used to create a secure, role-based project management system. With defined roles for Alice and Bob, and by leveraging ACLs and Flow Designer automation, we enabled structured task handling. Though minor UI visibility issues arose with approvals, core flow logic executed correctly, ensuring that task operations were secured, automated, and well-structured.

---

10. FUTURE SCOPE

- Introduce multi-level approvals (e.g., senior manager)
- Set SLA rules based on task priority
- Add reports and dashboards to visualize task and project status
- Integrate email or mobile push notifications for assigned tasks

---

11. APPENDIX

- **Users Created**: Alice (Project Manager), Bob (Team Member)
- **Groups**: Project Team Group
- **Tables**: u_project_table, u_task_table_2
- **Roles**: project_member, team_member, u_task_table_2_user
- **Flow Name**: task table flow

- **Tested ACL Fields**: status, comments, task_id, task_name, assigned_to

GitHub & Project Demo Link

• GitHub Repo : https://github.com/ramcharanpavanteja/Optimizing-User-Group-and-Role-Management-with-Access-Control-and-Workflows/tree/main

• Demo Video:
https://drive.google.com/file/d/1fCgnPSPrRTAGKZYLA5nX1V008MugmVLU/view?usp=driv
e_link