Project Documentation

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

(ServiceNow Administration)

Team Details

- **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

Managing users, groups, and roles effectively is crucial for any system with controlled access. This project, "Optimizing User, Group, and Role Management with Access Control and Workflows," focuses on the implementation of ServiceNow's user and role management features. It aims to create a secure and structured environment where permissions are assigned and controlled systematically using groups, roles, application access, and ACLs (Access Control Lists).

1.2 Purpose

The purpose of this project is to demonstrate how to configure user-role-group relationships and restrict access to ServiceNow applications using access control lists and workflow automation. The project ensures that each user has appropriate access depending on their responsibilities, thereby enforcing data security and operational clarity.

2. IDEATION PHASE

2.1 Problem Statement

Without proper user and access control, systems can become vulnerable to unauthorized data access or changes. Many organizations lack clarity on role-based permissions and often assign excessive privileges to users. This project addresses this by building a systematic user management model with proper access layers and automation using ServiceNow.

2.2 Empathy Map Canvas

Who? Admins, developers, and users
Think/Feel? Admins need better control over access
See? Confusion with permissions and unstructured data
Say/Do? Users request changes or report access issues
Hear? Complaints of unauthorized access or lack of access
Pain? Mismanagement of roles and privileges
Gain? Controlled and transparent access management

2.3 Brainstorming

We considered manual access configuration, script-based access control, and finally chose ServiceNow's role-based model with ACL and flow automation for its scalability and built-in support.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The user logs into the ServiceNow platform and performs the following steps:

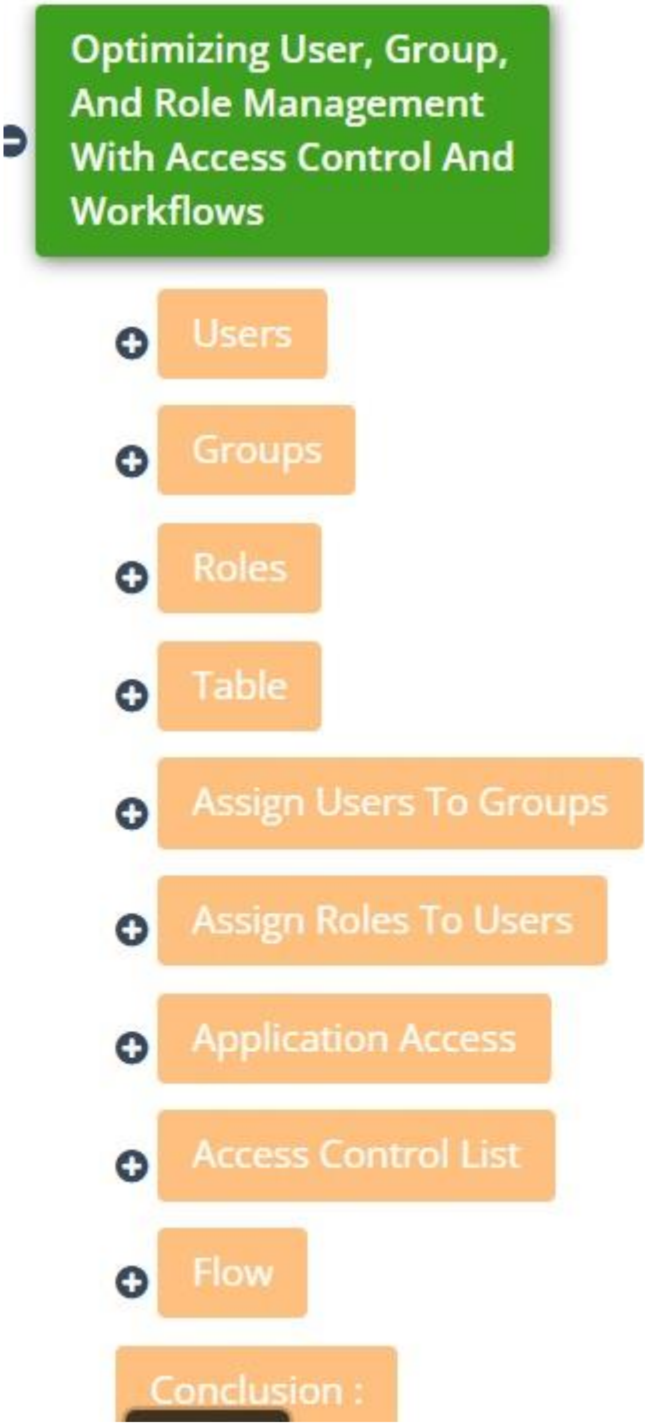
1. Create users in the system
2. Create groups for specific access levels
3. Define and assign roles
4. Create a custom table
5. Assign users to groups
6. Assign roles to users
7. Configure application access
8. Define Access Control Lists (ACLs)
9. Design and implement a workflow

3.2 Solution Requirement

- Custom Users and Roles
- ServiceNow Groups with user membership
- Role-based access to tables and applications
- ACL rules for table-level, field-level, and record-level access
- Workflow for access or permission request management

- Screens for managing relationships (users, groups, roles)
- Business logic automation via flows

3.3 Data Flow Diagram



3.4 Technology Stack

- Platform: ServiceNow
- Language: JavaScript (for business rules)
- Modules Used: User Administration, Roles, Groups, ACL, Flow Designer
- Testing Method: Manual testing for each user-role-group access scenario

4. PROJECT DESIGN

4.1 Problem Solution Fit

Unstructured access can lead to security gaps. This project introduces a hierarchical role-based structure that manages access efficiently through group membership, ACLs, and automation workflows.

4.2 Proposed Solution

- Create users, groups, and roles in ServiceNow
- Define application access
- Establish custom ACL rules
- Assign users to groups and map roles to users
- Design a simple approval workflow using Flow Designer
- Set up related lists to view role and group mapping

4.3 Solution Architecture

Frontend:
- User interface to manage access settings and group/role assignments

Logic Layer:
- Flow Designer for automation
- ACL engine for conditional access control

Data Layer:
- System tables: User, Role, Group, ACL, and custom table

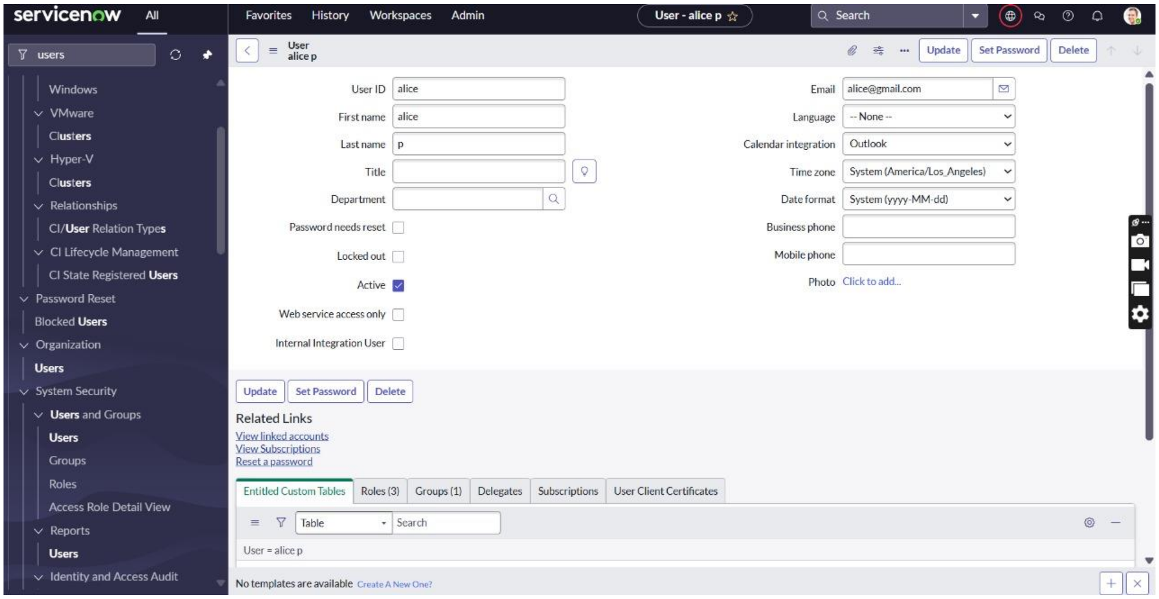
---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project followed a structured implementation and testing plan, as outlined below:

- Create a new Update Set in ServiceNow to track configuration changes.
- Add Users to the instance using the User Administration module.
- Create Groups based on functional or access needs.
- Define custom Roles (like viewer, editor, approver) for modular access.
- Design a Custom Table (e.g., custom_access_table) and assign proper access controls.
- Assign Users to appropriate Groups.
- Assign Roles either directly to Users or via Groups.
- Configure Application Access settings for visibility control.
- Apply ACLs (Access Control Lists) at the table, field, and record level.
- Design a simple approval Flow using Flow Designer.
- Test each module after configuration.
- Document each step with screenshots.



User
alice p

User ID	alice	Email	alice@gmail.com
First name	alice	Language	-- None --
Last name	p	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	☐		

User = alice p

No templates are available

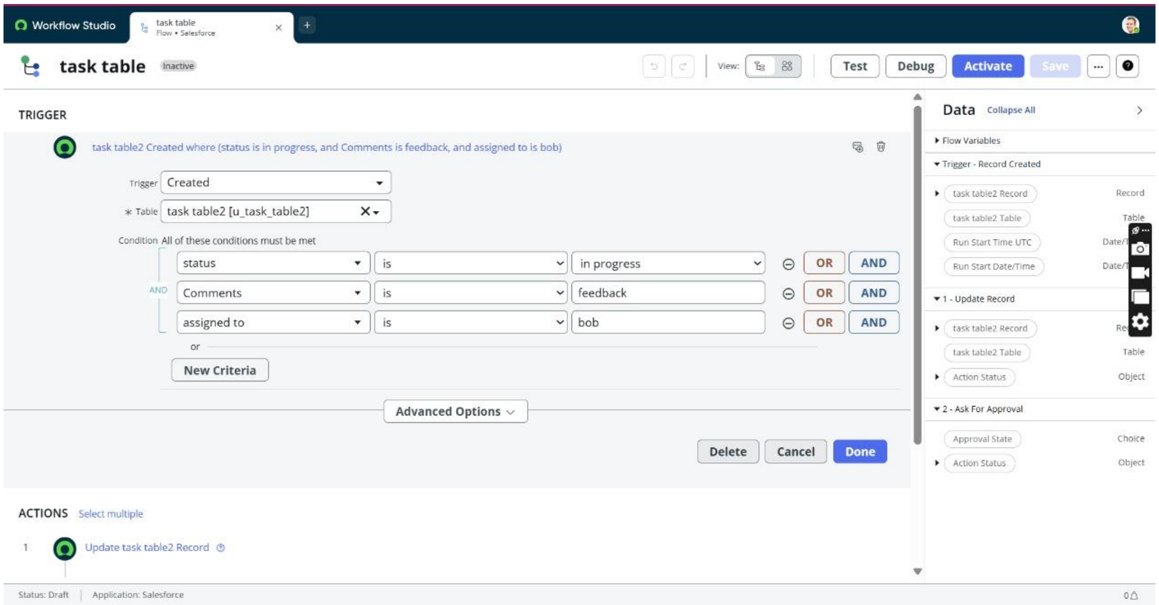
---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Functional Testing

- ✅ Users created successfully and reflected in the User table.
- ✅ Groups properly linked with associated users.

- ✅ Roles assigned through both direct and group inheritance.
- ✅ Custom table visibility and access worked as per ACL rules.
- ✅ Application access was restricted based on assigned roles.
- ✅ Flow executed correctly for access approval scenarios.



TRIGGER

task table2 Created where (status is in progress, and Comments is feedback, and assigned to is bob)

Trigger: Created

∗ Table: task table2 [u_task_table2]

Condition: All of these conditions must be met

status | is | in progress
AND Comments | is | feedback
assigned to | is | bob

or

ACTIONS

1 Update task table2 Record

Status: Draft | Application: Salesforce

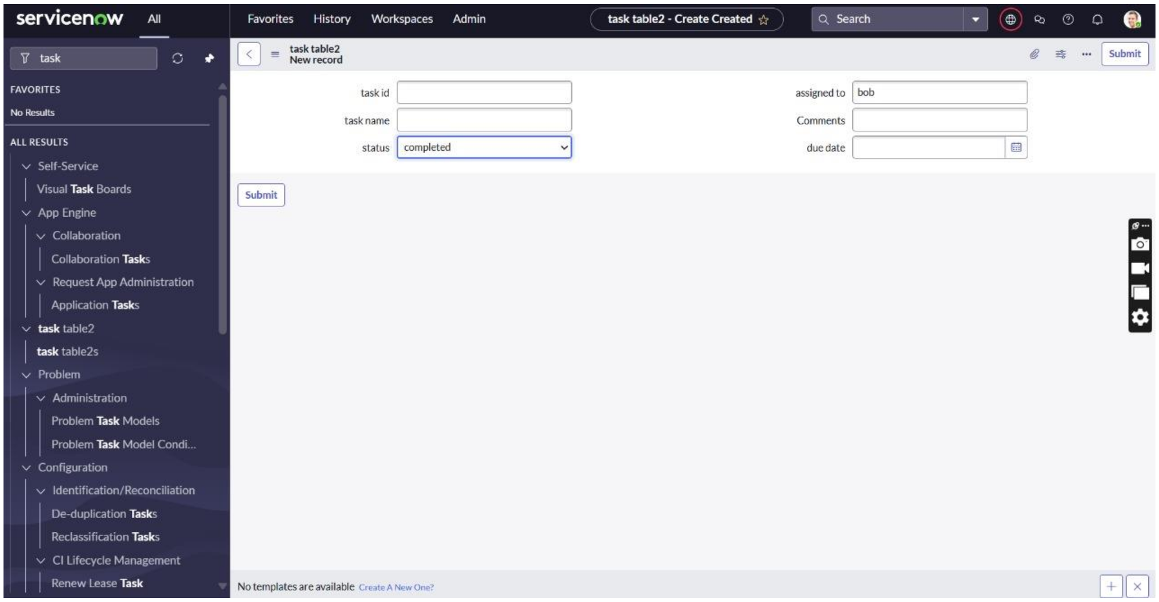
6.2 Performance Testing

- Tested login times with different role privileges.
- Validated table access latency with multiple concurrent users.
- Ensured ACL rules didn't delay access unnecessarily.
- Confirmed system stability with bulk role assignments.

---

7. RESULTS

7.1 Output Screenshots



task table2
New record

task id		assigned to	bob
task name		Comments	
status	completed	due date	

- Created Users in the system
- Group Membership View
- Assigned Roles to Users
- Application Access Permission Settings
- ACL rule configuration
- Custom Table form and list layout
- Flow Designer setup
- Flow execution logs and results

These results confirm that the system correctly restricts and permits access based on the roles assigned and follows the designed business logic.

---

8. ADVANTAGES & DISADVANTAGES

Advantages

- ✅ **Centralized user access management** using roles and groups.
- ✅ **Granular control** via ACLs allows tailored permissions.
- ✅ **Automated approval process** reduces manual errors.
- ✅ **Auditability** improves with well-defined flows and assignments.

- ✅ **Scalability**—easy to add users/roles as requirements grow.

Disadvantages

- ❌ **ACL configuration complexity** requires deep understanding.
- ❌ **Misconfiguration risks** can lead to excessive or blocked access.
- ❌ **Flow debugging** may be non-trivial in multi-role environments.
- ❌ **Initial setup time** may be high for beginners in ServiceNow.

---

9. CONCLUSION

This project demonstrates how organizations can improve security and access efficiency through proper user, group, and role management using ServiceNow. With structured role-based access, flow automation, and ACL enforcement, the system ensures only authorized users interact with data and applications. The use of Flow Designer further enhances the automation of access approval processes.

Implementing this architecture reduces the risk of unauthorized access, improves maintainability, and sets a solid foundation for future scalability in role-based environments.

---

10. APPENDIX

Github Link: https://github.com/suhail-cmd/Optimizing-User-Group-and-Role-Management-with-Access-Control-and-Workflows

Youtube link: https://youtu.be/DgcZ7gM-zfI