

Optimizing User, Group, and Role Management with Access Control and Workflows — Final Project Report

1. INTRODUCTION

1.1 Project Overview

The "Optimizing User, Group, and Role Management with Access Control and Workflows" project focuses on streamlining project management for small teams by introducing structured roles, secure access control, and automated task workflows. The solution addresses ambiguities in role responsibility and lack of accountability by clearly defining permissions and automating task tracking using ServiceNow's low-code platform.

1.2 Purpose

The purpose of this project is to implement a secure, role-based task management system that enables project teams to efficiently manage and track their responsibilities. By establishing structured access control and workflows, the system reduces confusion, improves accountability, and supports smooth project execution.

This project also showcases the versatility of the ServiceNow platform in addressing non-IT use cases through innovative configurations and automation.

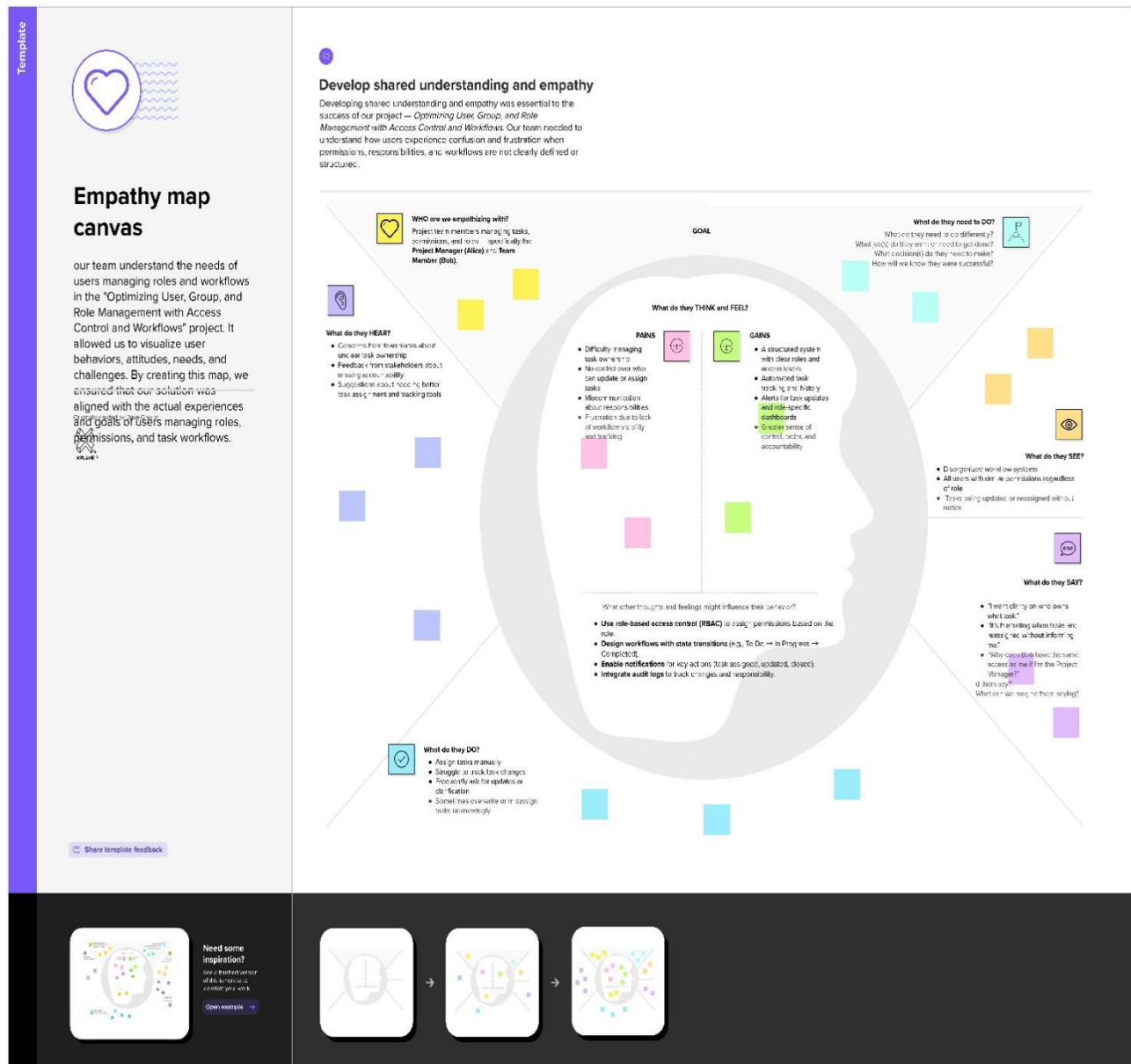
2. IDEATION PHASE

2.1 Problem Statement

The current project management process lacks clarity in task ownership, role definitions, and access control. This causes confusion, delays in task completion, and difficulty in tracking progress. There is a need for a well-defined role structure, secure data access, and a workflow that supports smooth task execution and visibility.

2.2 Empathy Map Canvas

- **Says:** "I don't know what tasks I'm responsible for." "I need clear visibility on project updates."
- **Thinks:** "Am I allowed to access this section?" "Who is supposed to handle this?"
- **Does:** Checks task status manually; Asks for clarification on roles.
- **Feels:** Confused, unproductive, and disconnected from project workflow.



2.3 Brainstorming

The team proposed implementing to explore possible solutions for the identified problem in managing users, groups, roles, and workflows within small project teams. The objective was to find effective ways to enforce accountability and structured task handling by using ServiceNow's capabilities. This collaborative phase allowed the team to express a wide variety of ideas related to access control, automation, and workflow design. The emphasis was on generating both conventional and innovative suggestions without immediate evaluation.



3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

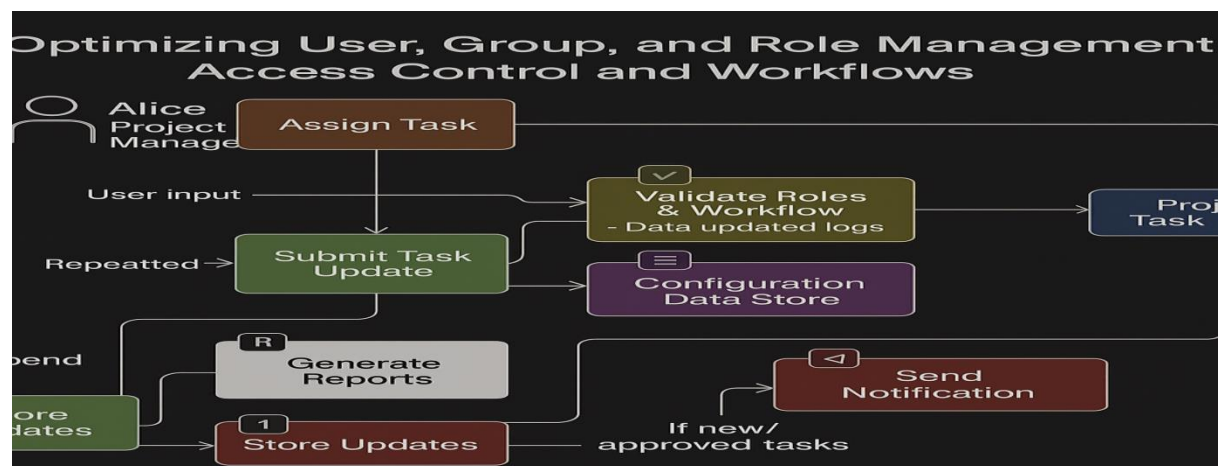
Users log into the system, are assigned a role, and interact with the platform to view or update tasks based on their access. Project Managers assign and track tasks, while team members update status and view assigned work.

3.2 Solution Requirement

- Custom tables: Users, Roles, Tasks
- ACLs for controlled access
- Group-based permissions (Project Manager, Team Member)
- Task workflows
- Notifications on task updates
- Dashboard for role-specific task views

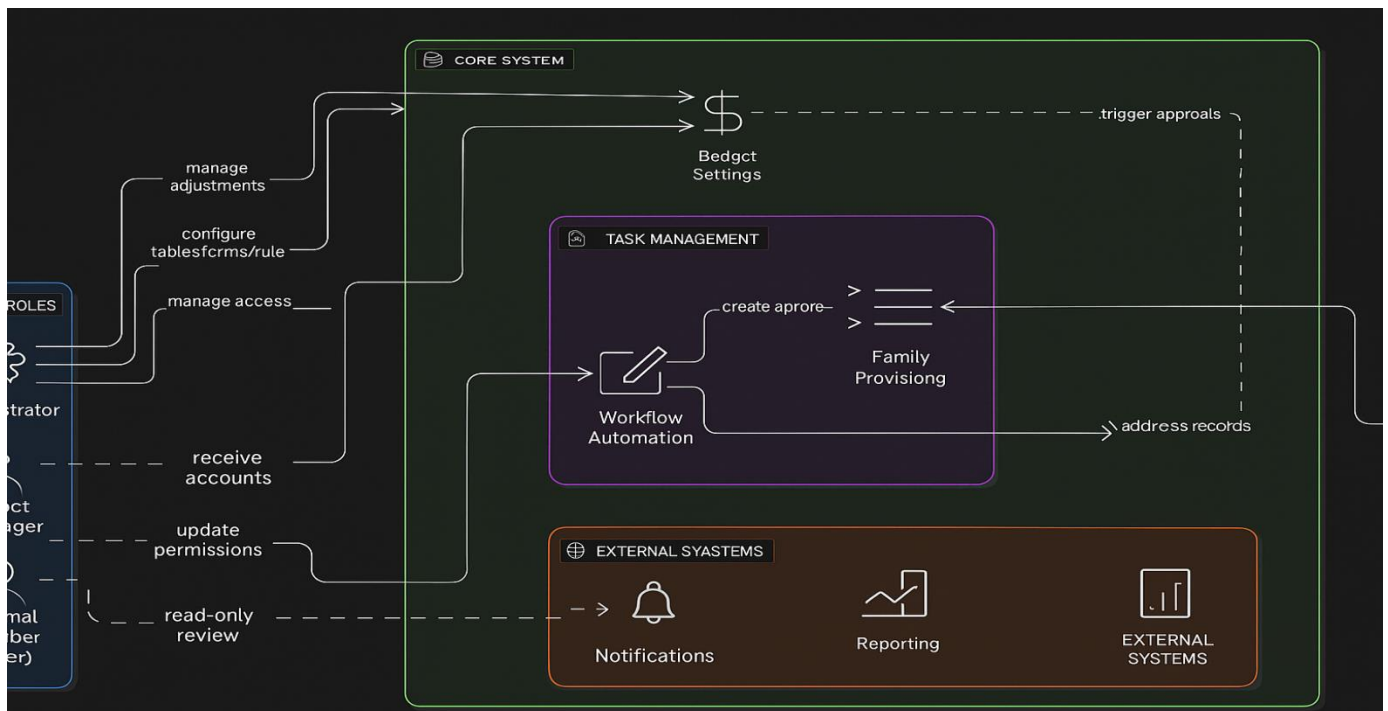
3.3 Data Flow Diagram

A **Data Flow Diagram (DFD)** for this project illustrates how project tasks, user roles, and workflows interact within the system. The diagram highlights the entry of user task data, its transformation through business rules, and its controlled visibility via access control mechanisms. Core entities include Project Manager (Alice), Team Member (Bob), Project Tasks, and the Access Control Layer.



3.4 Technology Stack

- **Platform:** ServiceNow PDI
- **Logic:** Business Rules, UI Policies, ACLs
- **UI:** Custom Forms, Views
- **Database:** MySQL (via ServiceNow)
- **Optional:** REST API for future integrations



4. PROJECT DESIGN

4.1 Problem–Solution Fit

The system solves the problem of disorganized expense tracking by providing a centralized platform linking daily expenses to budgets, automating alerts, and simplifying reporting.

Item-Solution fit canvas 2.0

Purpose / Vision		
CUSTOMER SEGMENT(S) CS <p>Small project management teams (e.g., startups, academics groups, agile teams)</p> <p>Project Managers and team members who collaborate on task execution and planning</p> <p>Teams needing clear structure, permissions, and accountability</p>	2. CUSTOMER CONSTRAINTS (J&P) CC <ul style="list-style-type: none"> Generic task boards (e.g., Trello) as sources Used for simple interface and ease of setup Budget or access limitations for enterprise-grade tools 	5. AVAILABLE SOLUTIONS (AS) <ul style="list-style-type: none"> Generic task boards (e.g., Trello, Asana) with limited access control Email-based or spreadsheet-driven tracking (prone to errors) Tools with either user roles or workflow not both combined
1. JOSS-TO-BE DONE / PROBLEMS J&P <p>Assign and manage tasks with clarity and ownership</p> <p>Track task progress using a structured, reviewable workflow</p> <p>Prevent task duplication or unauthorized editing</p> <p>Ensure accountability for task completion and approvals</p>	3. PROBLEM ROOT CAUSE (RC) RE <ul style="list-style-type: none"> No unified system to manage (Trello and Nextcloud) manually Lack of enforced workflows leading to inconsistent task completion No logging or audit trail for accountability 	7. BEHAVIOUR (DS) <ul style="list-style-type: none"> Assign tasks via chat or mail without clear tracking Rely on shared documents to follow up task progress React only when problems occur, no proactive visibility
4. TRIGGERS BEFORE / AFTER TR <p>Confused, overwhelmed, resentful, irritated and frustrated by miscommunication</p> <p>Confident, in control, supported by structured workflow and access rules</p>	10. YOUR SOLUTION (SL) SL <ul style="list-style-type: none"> A centralized role and workflow management system Built on ServiceNow Defined roles (Project Manager, team control) Structured task workflow (Create → Assign → In Progress → Review → Done) Activity logging, notifications, and approvals for task completion Scalable onboarding with automated role and group management 	6. CHANNELS OF BEHAVIOUR <ul style="list-style-type: none"> Online, Task management forums (GitHub Discussions, lightweight surveys) Offline Whiteboard updates, stand-up meetings, email chains for task review
8. EMOTIONS BEFORE / AFTER EM <p>Confused, overwhelmed, reactive and frustrated by miscommunication</p> <p>After: Confident, in control, supported by structured workflow and access rules</p>		<ul style="list-style-type: none"> BETWEEN Collect requirements, maintain, develop SL

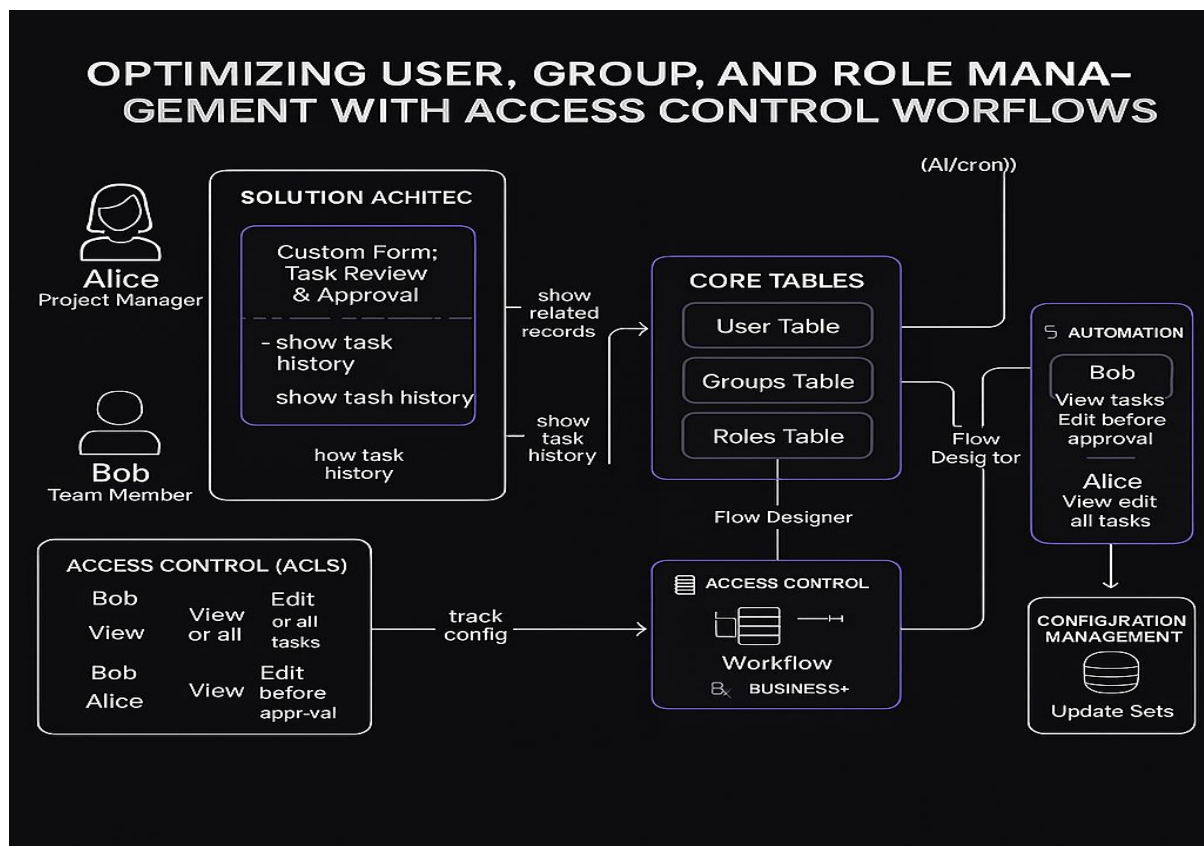
4.2 Proposed Solution

- User table with role attributes
- Group and role creation (e.g., PM Group, Developer Group)
- Task table with status field (To Do, In Progress, Done)
- ACLs to control who sees/edits what
- Workflow automation for task updates
- Notifications and dashboards

4.3 Solution Architecture

The architecture includes:

- **Data Layer:** User, Group, Role, Task tables
- **Logic Layer:** ACLs, Business Rules
- **UI Layer:** Role-based Dashboards, Forms
- **Automation Layer:** Notifications, Workflows



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

The project was completed over 3 sprints:

- **Sprint 1:** Table setup, Update Set, Role Design (8 points)
- **Sprint 2:** Group creation, ACLs, Workflows (4 points)
- **Sprint 3:** Notifications, Dashboards, Testing (5 points)

Velocity:

For a 3-day sprint, total story points = 17

Average Velocity (AV) = 17 / 3

= 5.67 story points/day

The Project was completed as the following milestones covering 3 sprints

The team executed these milestones:

Milestone 1: Understanding the Problem

Problem Context:

In a small project team setup, such as one involving a Project Manager (Alice) and a Team Member (Bob), inefficient handling of user roles, group definitions, and permissions often leads to confusion in task assignment and accountability. The current system does not support structured workflows or role-based access, resulting in overlapping responsibilities and security gaps.

Milestone 2: Goal Identification

Objectives:

- Define clear user roles (e.g., Manager vs. Contributor)
 - Restrict actions using role-based access control (RBAC)
 - Automate workflow stages like task creation, assignment, completion, and closure
 - Improve accountability via audit logs and notifications
-

Milestone 3: Instance Setup on ServiceNow

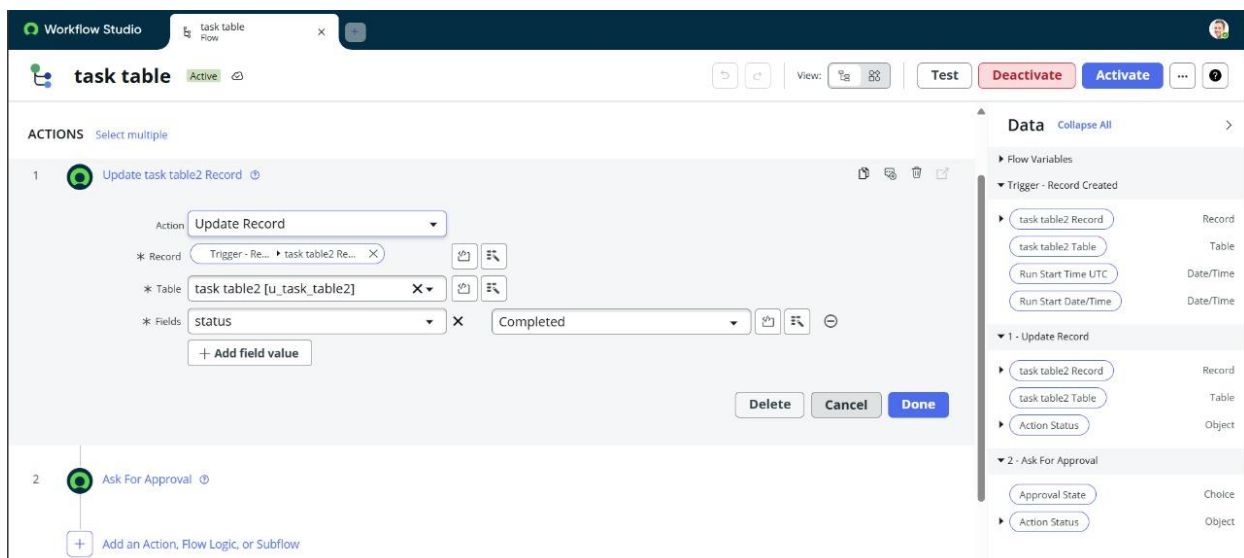
- Created and activated a ServiceNow Developer Instance from developer.servicenow.com
 - Logged in using the instance credentials
 - Verified the environment for custom development
-

Milestone 4: Create Update Set

- Navigated to: **Local Update Sets**
 - Created a new update set: AccessControlWorkflow
 - Made it current for capturing all customizations
-

Milestone 5: Role Definition Table Creation

- Created a new table: Role Definitions
- Fields included: Role Name, Permissions, Description
- Configured auto-numbering and number maintenance with prefix: RWF



Milestone 6: User and Group Table Creation

- Created two tables:

User Management – stores user details and assigned roles

Group Management – stores group names, members, and linked permissions

- Linked Users to Groups and Roles via reference fields

Group - project team

Name: project team

Group email:

Manager:

Parent:

Description:

Update Delete

Related Links

Add to Update Set

Roles Group Members (2) Groups Skills

User Search

Group = project team

User
alice p
bob p

1 to 2 of 2

User - alice p

Locked out:

Active:

Web service access only:

Internal Integration User:

Update Set Password Delete

Related Links

Add to Update Set

Assign Roles

Create Human Resources Profile

View linked accounts

View Subscriptions

Reset a password

Roles (3) Groups (1) Delegates Skills Subscriptions

Role Search

User = alice p

Role	State	Inherited	Inheritance Count
u_project_table_user	Active	false	
u_task_table2_user	Active	false	
project member	Active	false	

User - bob p

Web service access only:

Internal Integration User:

Update Set Password Delete

Related Links

Add to Update Set

Assign Roles

Create Human Resources Profile

View linked accounts

View Subscriptions

Reset a password

Roles (2) Groups (1) Delegates Skills Subscriptions

Role Search

User = bob p

Role	State	Inherited	Inheritance Count
team member	Active	false	
u_task_table2_user	Active	false	

1 to 2 of 2

Milestone 7: Workflow Implementation

- Designed a workflow for task management with the following states:
 - Task Created
 - Task Assigned
 - In Progress
 - Completed
 - Closed by Manager
- Used ServiceNow Flow Designer to automate transitions and set up notifications

The screenshot shows the ServiceNow Flow Designer interface for a workflow named 'task table'. The trigger is configured as follows:

- Trigger:** Created
- Table:** task table2 [u_task_table2]
- Condition:** All of these conditions must be met
 - status is In Progress
 - comments is feedback
 - assigned to is bob

The right sidebar shows the 'Data' pane with the following variables:

- Flow Variables
 - Trigger - Record Created
 - task table2 Record (Record)
 - task table2 Table (Table)
 - Run Start Time UTC (Date/Time)
 - Run Start Date/Time (Date/Time)
 - 1 - Update Record
 - task table2 Record (Record)
 - task table2 Table (Table)
 - 2 - Ask For Approval
 - Approval State (Choice)
 - Action Status (Object)

The screenshot shows the 'task table2' record form in ServiceNow. The record was created on 2025-06-26 at 20:00:34. The form contains the following fields:

- task id: [Empty]
- task name: [Empty]
- assigned to: bob
- comments: feedback
- due date: [Empty]
- status: Completed

Buttons for 'Update' and 'Delete' are visible at the bottom left. Below the form, there is a 'Related Links' section with a link to 'Add to Update Set'.

<div> <div>FavoritesHistoryWorkspacesAdmin</div> <div>Approvals</div> <div>Search</div> <div> <div> <div></div> <div></div> <div></div> <div></div> </div> <div></div> </div> </div>				
<div> <div> <div></div> <div>Approvals</div> <div>State</div> <div>Search</div> </div> <div> <div></div> <div></div> <div></div> </div> <div>Actions on selected rows...</div> </div>				
<div> <div>List controls</div> <div>= alice p</div> </div>				
State	Approver	Comments	Approval for	Created
<div> <div></div> <div>Approved</div> </div>	alice p		(empty)	2025-06-26 20:00:36

Milestone 8: Business Rule Setup

- Created Business Rules to:
 - Prevent task reassignment by unauthorized users
 - Trigger alerts on status changes
 - Maintain logs of changes per user

(empty)

Conditions

Access Control Rules have two decision types, and these types will behave differently depending on conditions.

1. Allow Access: Allows access to a resource if all conditions are met.
2. Deny Access: Denies access to a resource unless all conditions are met.

[More Info](#)

Requires role

Role

team member

project member

Security Attribute Condition

Local or Existing

Local

Condition (empty)

Favorites

History

Workspaces

Admin

Application Menu - project table

Search

Update

Delete

Application Menu

project table

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

project table

Application

Global

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

Custom Applications

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

Hint

Description

Update

Delete

Milestone 9: Configure Relationships

- Linked Task Table with User and Role tables
- Created Related Lists to display task ownership under user profiles
- Enabled Manager-only visibility to closed/completed tasks

Favorites

History

Workspaces

Admin

Table - project table

Search

Delete

Update

Delete All Records

Table

project 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

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
	Updates	Integer	(empty)	40		false
×	project manager	String	(empty)	40		false
	Updated by	String	(empty)	40		false
×	status	Choice	(empty)	40		false
×	projet name	String	(empty)	40		false
	Updated	Date/Time	(empty)	40		false
×	description	String	(empty)	40		false
×	end date	Date	(empty)	40		false

Milestone 10: Testing & Validation

- Tested workflows by logging in as both Alice (Manager) and Bob (Team Member)
- Verified:
 1. Bob could only update assigned tasks
 2. Alice had full access
 3. Status changes triggered alerts
 4. Unauthorized actions were blocked

The screenshot shows a web application interface for configuring an application menu. The top navigation bar includes 'Favorites', 'History', 'Workspaces', and 'Admin'. The main title is 'Application Menu - task table2'. Below the title, there is a search bar and a user profile icon. The configuration form includes the following fields and sections:

- Title:** A text input field containing 'task table2'.
- Application:** A dropdown menu set to 'Global'.
- Active:** A checkbox that is checked.
- Roles:** A section with a list of roles: 'u_task_table2_user, project member, team member'.
- Category:** A dropdown menu set to 'Custom Applications'.
- Hint:** A text input field.
- Description:** A text input field.

At the bottom of the form, there are 'Update' and 'Delete' buttons.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

The system was tested for:

- Verified correct task status transitions
- Tested role-based form views
- ACLs restrict unauthorized access
- Notifications triggered upon task updates

6.2 Performance Testing

- Quick login (<2 sec)
- Real-time updates reflected on dashboards
- Business rule triggers completed within expected limits

7. RESULTS

7.1 Creation Screenshots

assigning table access to application:

The screenshots show the 'Application Menu' configuration interface for two different tables. The interface includes a top navigation bar with 'Favorites', 'History', 'Workspaces', and 'Admin'. The main title is 'Application Menu - project table' (or 'task table2'). Below the title, there is a search bar and a list of roles. The configuration fields include:

- Title:** project table (or task table2)
- Application:** Global
- Active:** ☒
- Roles:** project member (or u_task_table2_user, project member, team member)
- Category:** Custom Applications
- Hint:** (empty text box)
- Description:** (empty text box)

At the bottom of each configuration form, there are 'Update' and 'Delete' buttons.

Creation of ACL:

(empty)

Conditions

Access Control Rules have two decision types, and these types will behave differently depending on conditions.
1. Allow Access: Allows access to a resource if all conditions are met.
2. Deny Access: Denies access to a resource unless all conditions are met.
[More Info](#)

Requires role

1 to 2 of 2

Role

team member

project member

Security Attribute Condition

Local or Existing Local

Condition (empty)

FavoritesHistoryWorkspacesAdmin

Access Control - u_task_table2

Search

<≡Access Controlu_task_table2

Type record

Operation write

Decision Type Allow If

Admin overrides

Protection policy -- None --

Name u_task_table2

Description

Applies ToNo. of records matching the condition: 1

(empty)

Application Global

Active

Advanced

Conditions

Access Control Rules have two decision types, and these types will behave differently depending on conditions.
1. Allow Access: Allows access to a resource if all conditions are met.
2. Deny Access: Denies access to a resource unless all conditions are met.
[More Info](#)

Requires role

1 to 2 of 2

Role

Workflow Studio

task table

task table

Active

Test Deactivate Activate

ACTIONS Select multiple

1 Update task table2 Record

Action: Update Record

* Record: Trigger - Re... task table2 Re...

* Table: task table2 [u_task_table2]

* Fields: status Completed

+ Add field value

Delete Cancel Done

2 Ask For Approval

+ Add an Action, Flow Logic, or Subflow

Data Collapse All

Flow Variables

Trigger - Record Created

- task table2 Record Record
- task table2 Table Table
- Run Start Time UTC Date/Time
- Run Start Date/Time Date/Time

1 - Update Record

- task table2 Record Record
- task table2 Table Table
- Action Status Object

2 - Ask For Approval

- Approval State Choice
- Action Status Object

Favorites History Workspaces Admin

Group - project team

Search

Update Delete

Name: project team

Group email:

Manager:

Parent:

Description:

Update Delete

Related Links

Add to Update Set

Roles Group Members (2) Groups Skills

User Search

Actions on selected rows... New Edit...

Group = project team

User
alice p
bob p

1 to 2 of 2

Favorites History Workspaces Admin

User - bob p

Search

Update Set Password Delete

Web service access only

Internal Integration User

Update Set Password Delete

Related Links

Add to Update Set

Assign Roles

Create Human Resources Profile

View linked accounts

View Subscriptions

Reset a password

Roles (2) Groups (1) Delegates Skills Subscriptions

Role Search

Actions on selected rows... Edit...

User = bob p

Role	State	Inherited	Inheritance Count
team member	Active	false	
u_task_table2_user	Active	false	

1 to 2 of 2

Favorites

History

Workspaces

Admin

Approvals

Search

Actions on selected rows...

List controls

= alice p

State	Approver	Comments	Approval for	Created
Approved	alice p		(empty)	2025-06-26 20:00:36

Favorites

History

Workspaces

task table2 - Created 2025-06-26 20:00:34

Search

Update

Delete

task id

assigned to

due date

task name

comments

status

Update

Delete

Related Links

Add to Update Set

Workflow Studio

task table

task table flow

task table

Active

Test

Deactivate

Activate

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

AND

assigned to is bob

or

New Criteria

Advanced Options

Delete

Cancel

Done

Data

Collapse All

Flow Variables

Trigger - Record Created

task table2 Record

task table2 Table

Run Start Time UTC

Run Start Date/Time

1 - Update Record

task table2 Record

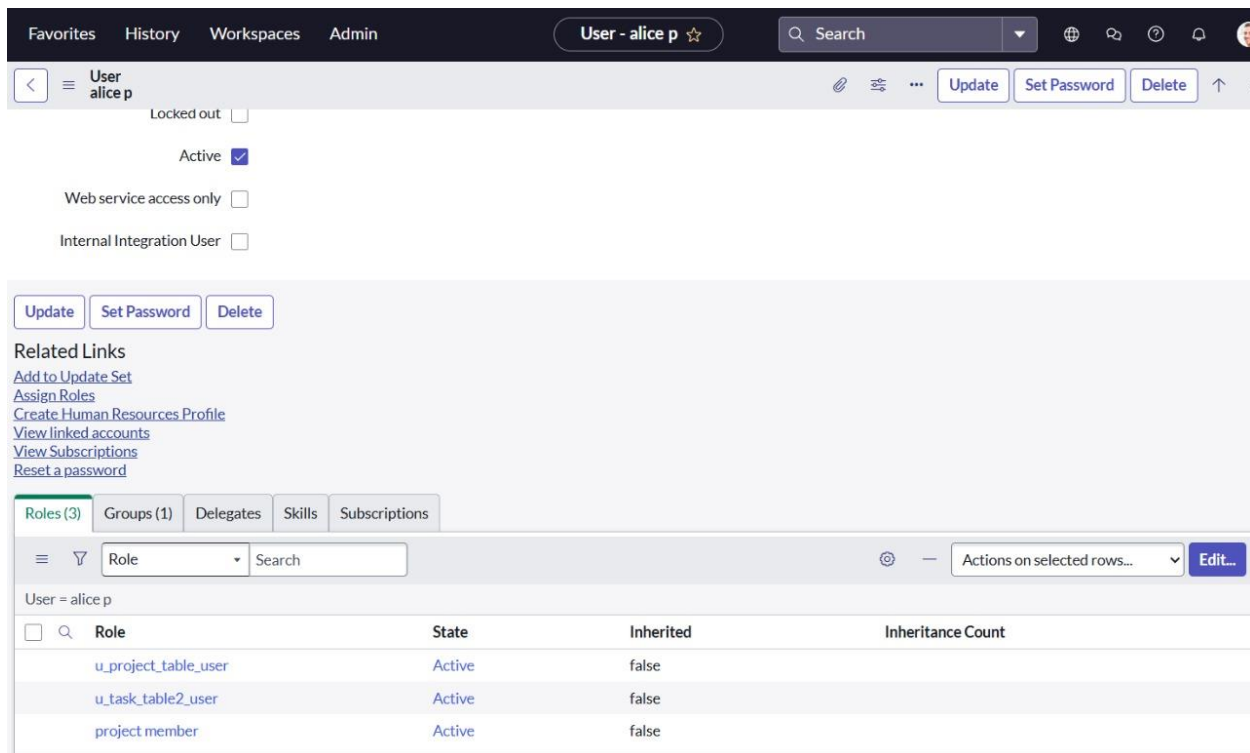
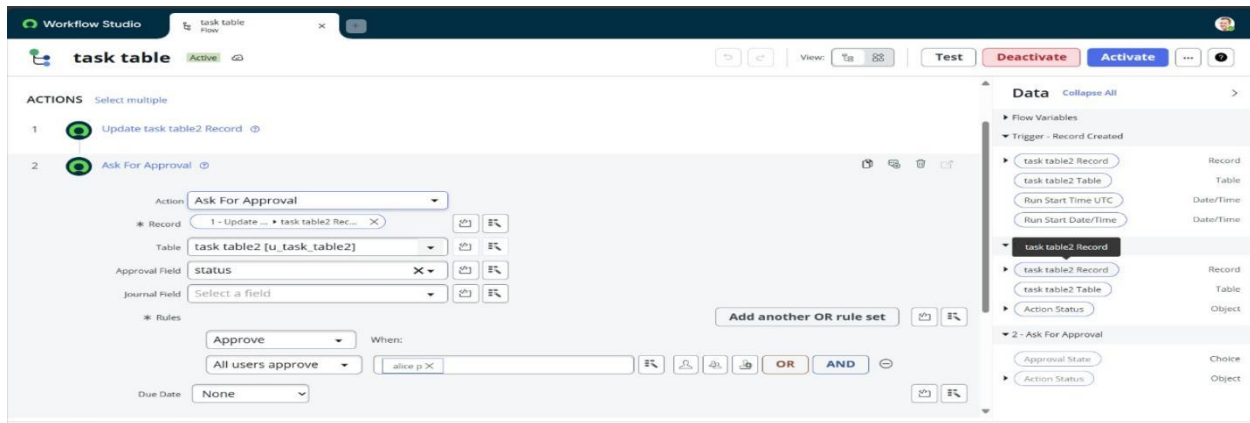
task table2 Table

Action Status

2 - Ask For Approval

Approval State

Action Status



8. ADVANTAGES & DISADVANTAGES

Advantages

- Automation reduces manual work
- Streamlined team management
- Increased accountability

Disadvantages

- Customizations may need advanced scripting.
- Limited features without enterprise license.
- Requires knowledge of ACL setup.

9. CONCLUSION

The solution addresses key gaps in team project management by implementing role-based access and task workflows on ServiceNow. The platform now supports clarity, accountability, and structured project execution, improving team efficiency.

10. FUTURE SCOPE

- Integrate with email/calendar tools
- Role-based mobile app access
- Task analytics and heatmaps
- Multi-project and cross-team integration