OPTIMIZING USER, GROUP, AND ROLE MANAGEMENT WITH ACCESS CONTROL AND WORKFLOWS

ABSTRACT

This project aims to optimize user, group, and role management using ServiceNow by implementing Role-Based Access Control (RBAC) and automating workflows. The purpose of the project is to create a structured environment where user permissions are clearly defined, tasks are automatically assigned, and data security is maintained. Through ServiceNow's platform capabilities, the project demonstrates how automation, access control, and dashboards improve transparency and efficiency. The findings reveal that a well-configured RBAC system can significantly reduce manual work, prevent unauthorized actions, and improve collaboration among users.

INTRODUCTION

In organizations that handle multiple users, roles, and departments, maintaining a clear structure of responsibilities and permissions is essential for productivity and security. Without a centralized management system, users may face confusion about task ownership, leading to redundant work and data inconsistencies. ServiceNow provides a low-code platform that simplifies the creation of workflows, role-based access control, and process automation. This project focuses on using these capabilities to create a streamlined environment where each role has specific access and where workflows automatically update task statuses.

The implementation of this system benefits both managers and employees by ensuring that only authorized users perform certain operations. For instance, project managers can create and assign tasks, while team members can only view or update tasks assigned to them. This separation of duties not only enhances security but also improves accountability. The integration of dashboards and reports provides real-time insights into task completion and performance trends, ensuring that teams stay aligned with project objectives.

PROBLEM STATEMENT

Organizations often face challenges when managing users and roles manually. Without automated access control and workflows, there is a high risk of unauthorized actions, duplicated efforts, and poor visibility into project progress. Traditional management systems require manual updates, which can lead to errors and wasted time. Furthermore, without role-based restrictions, sensitive data can be exposed to unintended users.

The absence of workflow automation also affects team coordination. For instance, when a project manager assigns a task to a team member, the lack of an automated system means that the manager has to manually track progress. This can delay project execution. To overcome these limitations, this project proposes a solution built on ServiceNow that combines role-based access control with automated workflows to provide a transparent, efficient, and secure environment.

METHODOLOGY / SYSTEM DESIGN

The project follows a structured design and implementation approach using ServiceNow Studio, Flow Designer, and Platform Analytics. The methodology includes user creation, role definition, group setup, workflow automation, and dashboard reporting. Each step is integrated to form a comprehensive role-based system that automates task management.

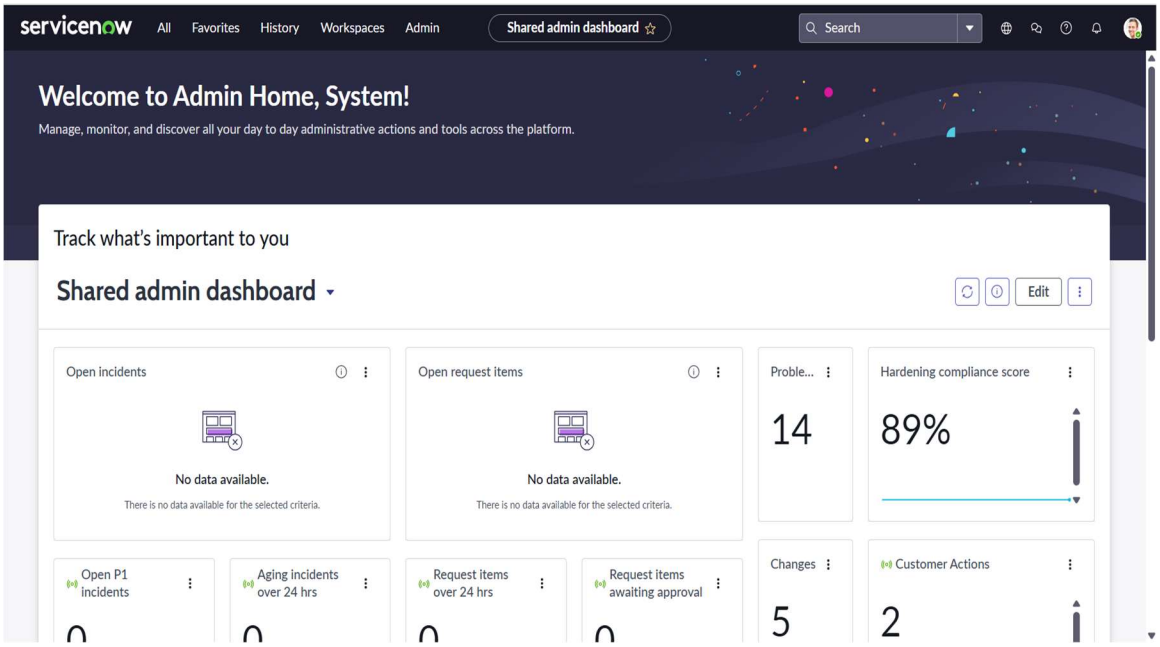
• Design Approach: The system uses ServiceNow's modular structure, where each function—user creation, access control, workflow automation, and analytics—is handled separately. ServiceNow Studio is used to build a custom scoped application called Project Task Tracker, which serves as the main workspace for managing project tasks.

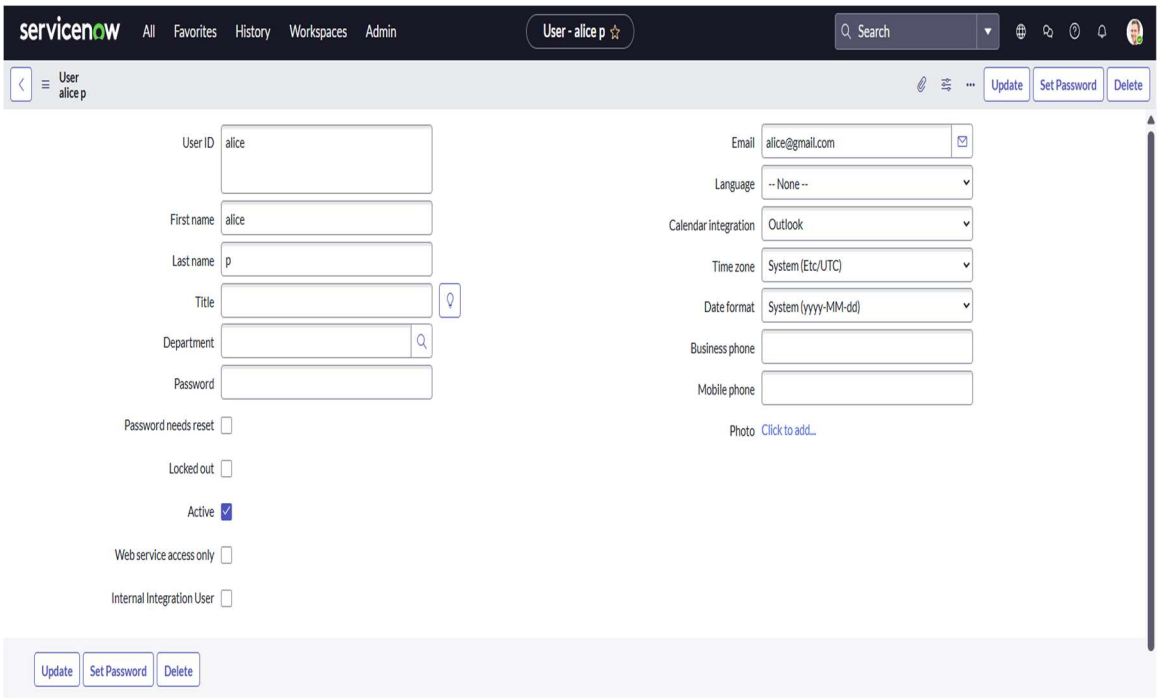
• System Architecture: The architecture is divided into three layers: User Layer, Application Layer, and Workflow Layer. The User Layer handles roles and permissions, the Application Layer manages task data using custom tables, and the Workflow Layer automates updates and notifications. Together, these layers ensure secure and smooth operations.

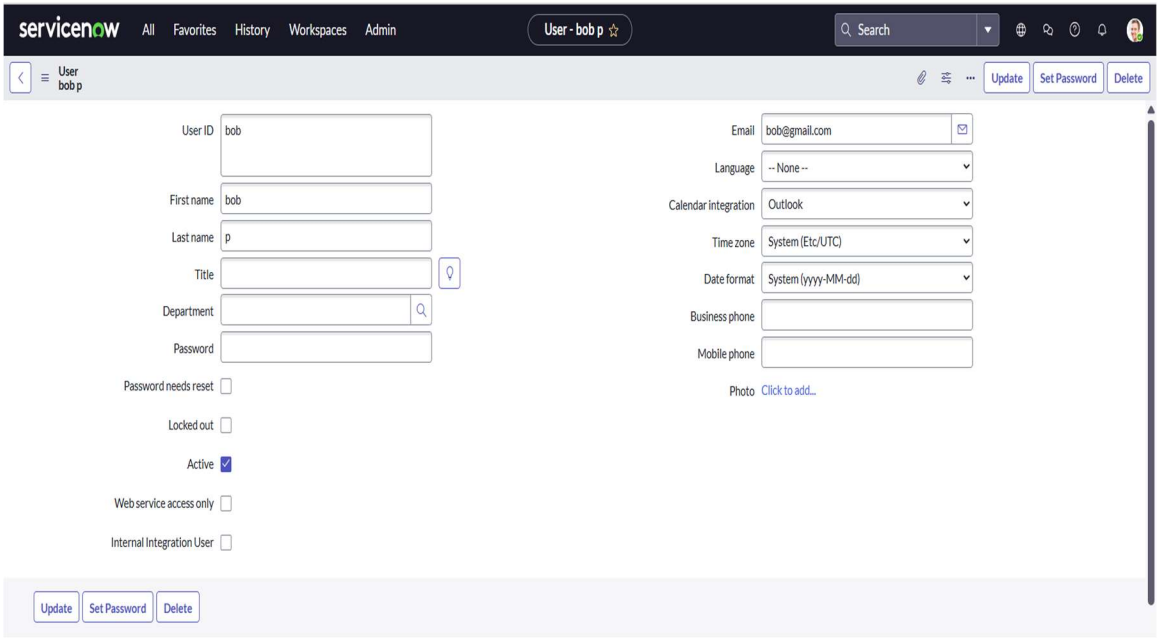
• User Interface (UI) and User Experience (UX): The application's forms and list layouts are designed for simplicity. The Project Manager view includes all tasks, while team members see only their assigned tasks. The interface includes dashboards that summarize ongoing work, pending tasks, and completed items using visual charts.

IMPLEMENTATION DETAILS

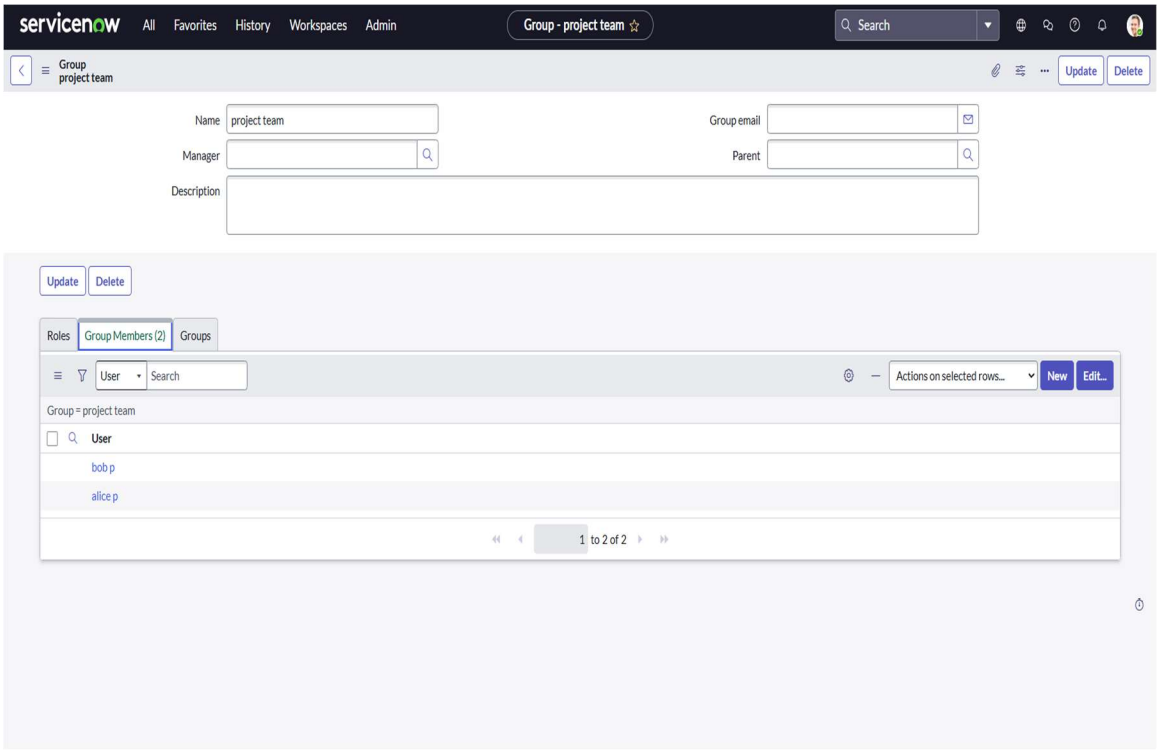
• Platform Setup: A ServiceNow instance is initialized, and users are created.

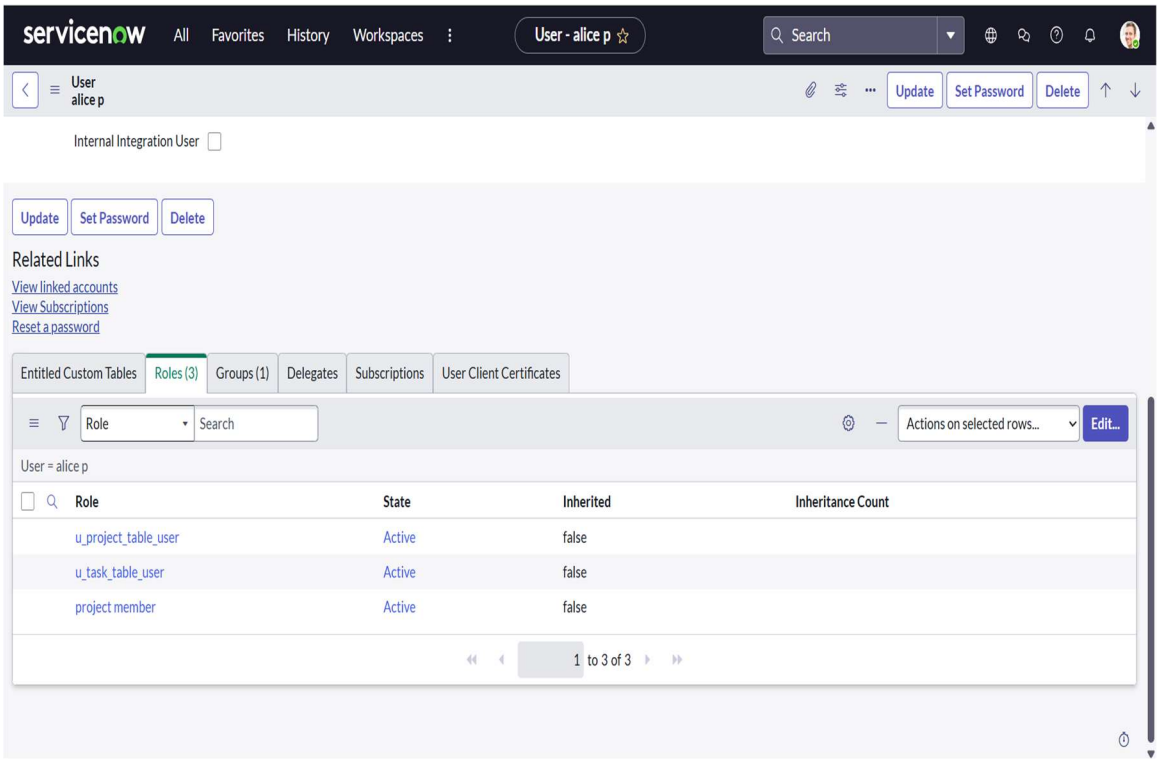


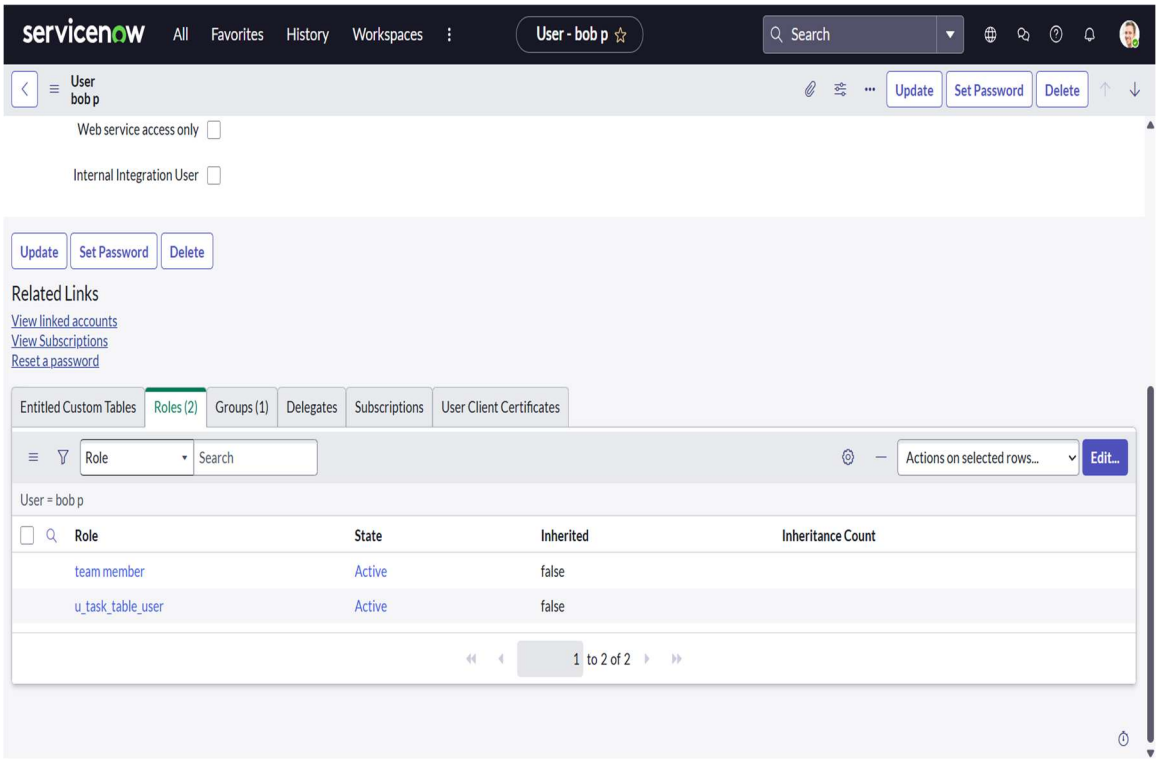




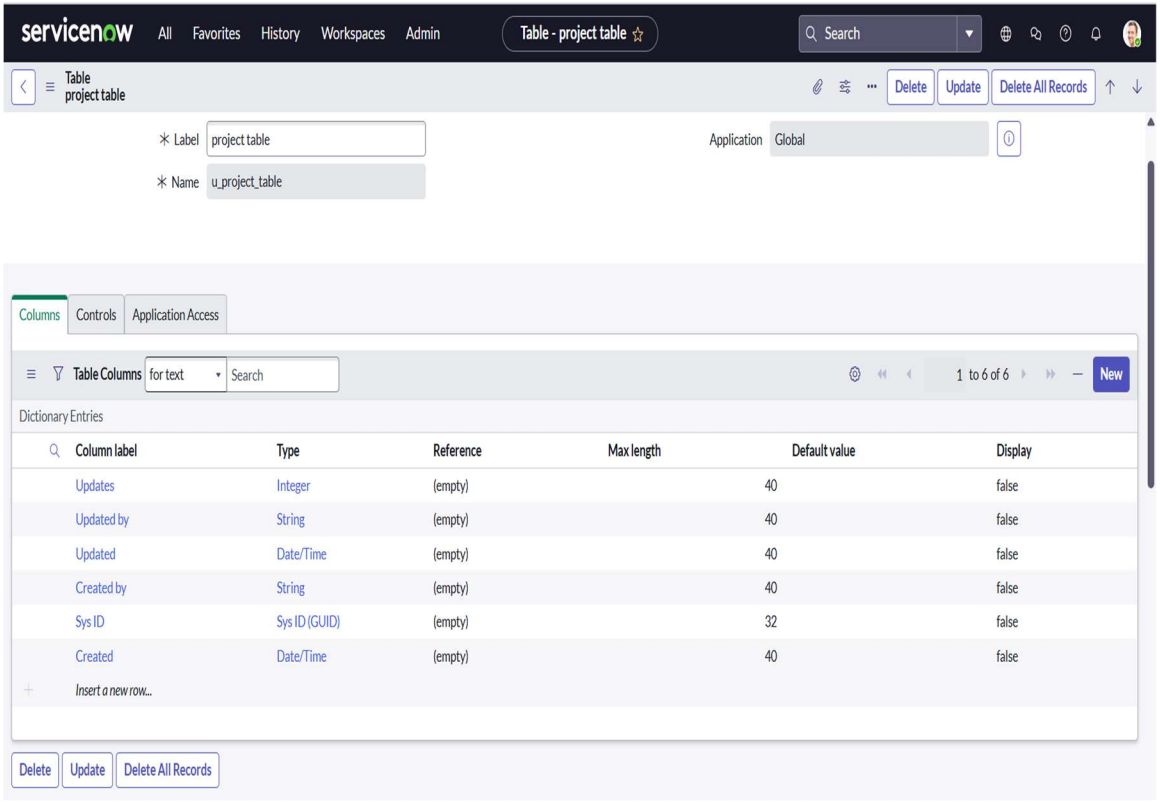
Alice acts as the Project Member and Bob as the Team Member. Custom groups Project Members and Team Members are formed, and roles are assigned. This establishes the foundation for RBAC.



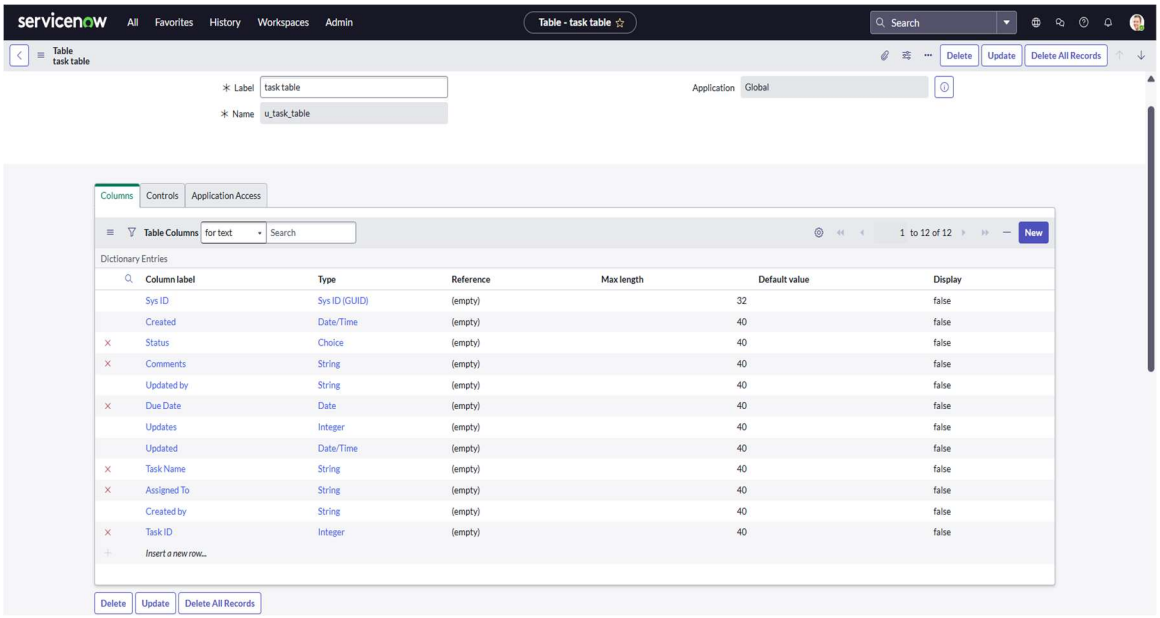

servicenow
All
Favorites
History
Workspaces
User - alice p
Search
User
alice p
Update
Set Password
Delete
Internal Integration User
Update
Set Password
Delete
Related Links
View linked accounts
View Subscriptions
Reset a password
Entitled Custom Tables
Roles (3)
Groups (1)
Delegates
Subscriptions
User Client Certificates
Role
Search
Actions on selected rows...
Edit...
User = alice p
Role
State
Inherited
Inheritance Count
u_project_table_user
Active
false
u_task_table_user
Active
false
project member
Active
false
1 to 3 of 3


servicenow
All
Favorites
History
Workspaces
User - bob p
Search
User
bob p
Update
Set Password
Delete
Web service access only
Internal Integration User
Update
Set Password
Delete
Related Links
View linked accounts
View Subscriptions
Reset a password
Entitled Custom Tables
Roles (2)
Groups (1)
Delegates
Subscriptions
User Client Certificates
Role
Search
Actions on selected rows...
Edit...
User = bob p
Role
State
Inherited
Inheritance Count
team member
Active
false
u_task_table_user
Active
false
1 to 2 of 2

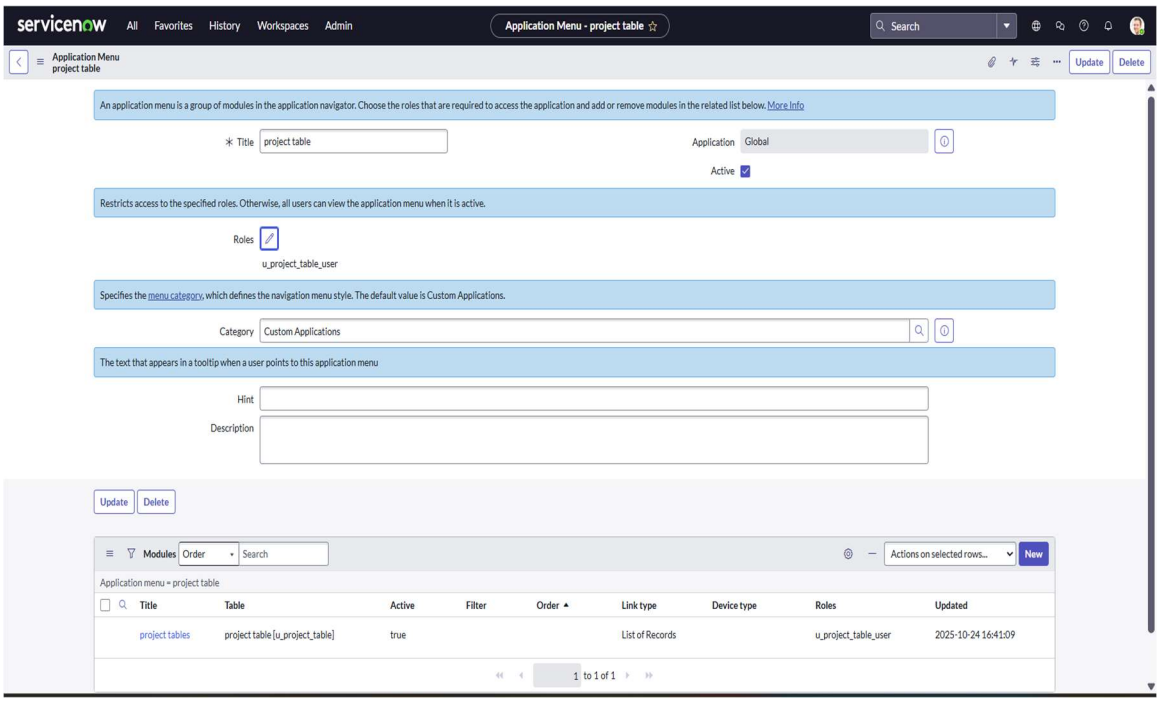
• Development and Customization: In ServiceNow Studio, the Project Task Tracker application is created to manage and monitor project tasks efficiently. A custom table named Project Table is designed with fields such as Task Name, Description, Status, Assigned To, Due Date, and Created By to store main project details.

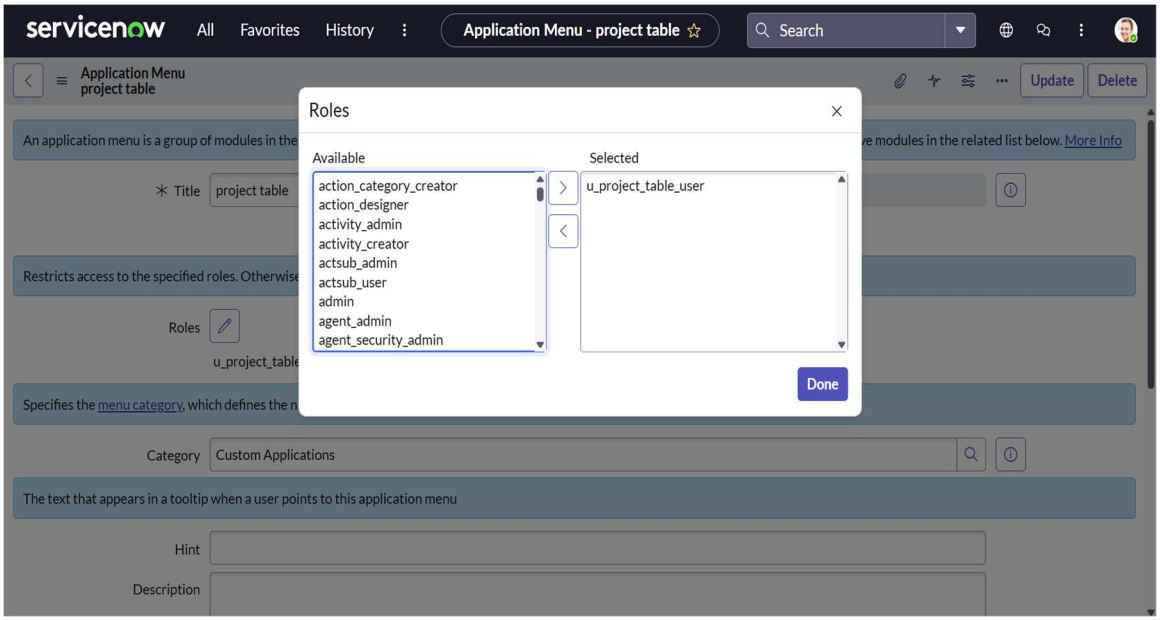


Another custom table named **Task Table** is also created to handle individual tasks related to each project. It includes fields like Task ID, Task Name, Project Name (reference to Project Table), Assigned To, Priority, Start Date, End Date, and Task Status. These tables help in organizing project data and tracking task progress effectively within the application.

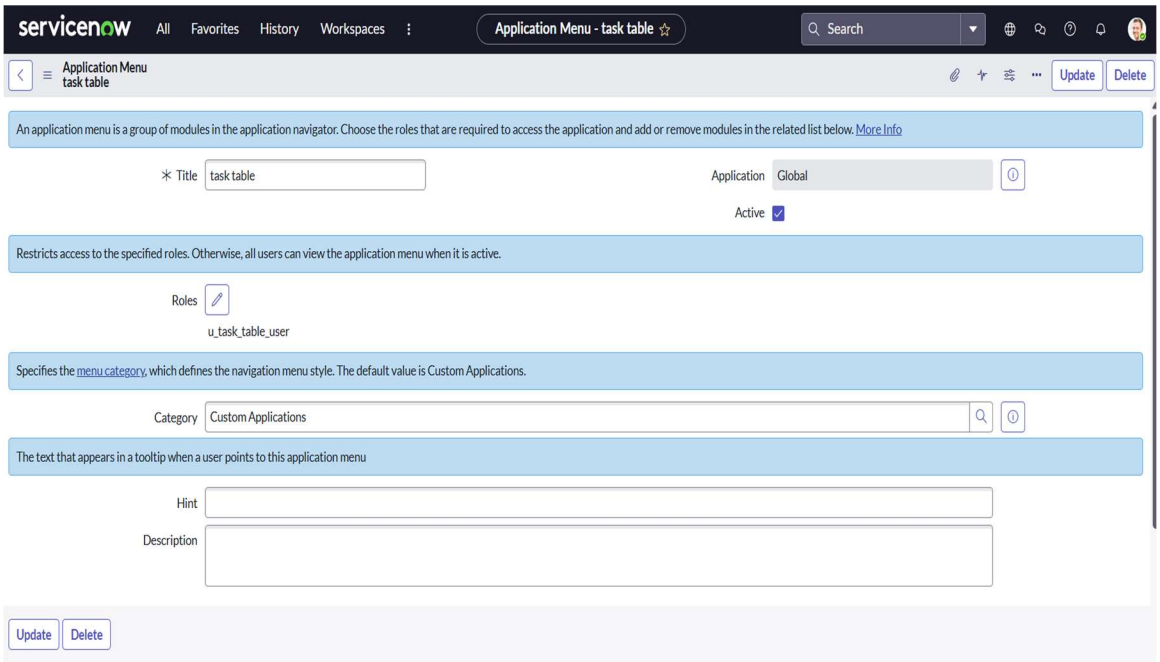


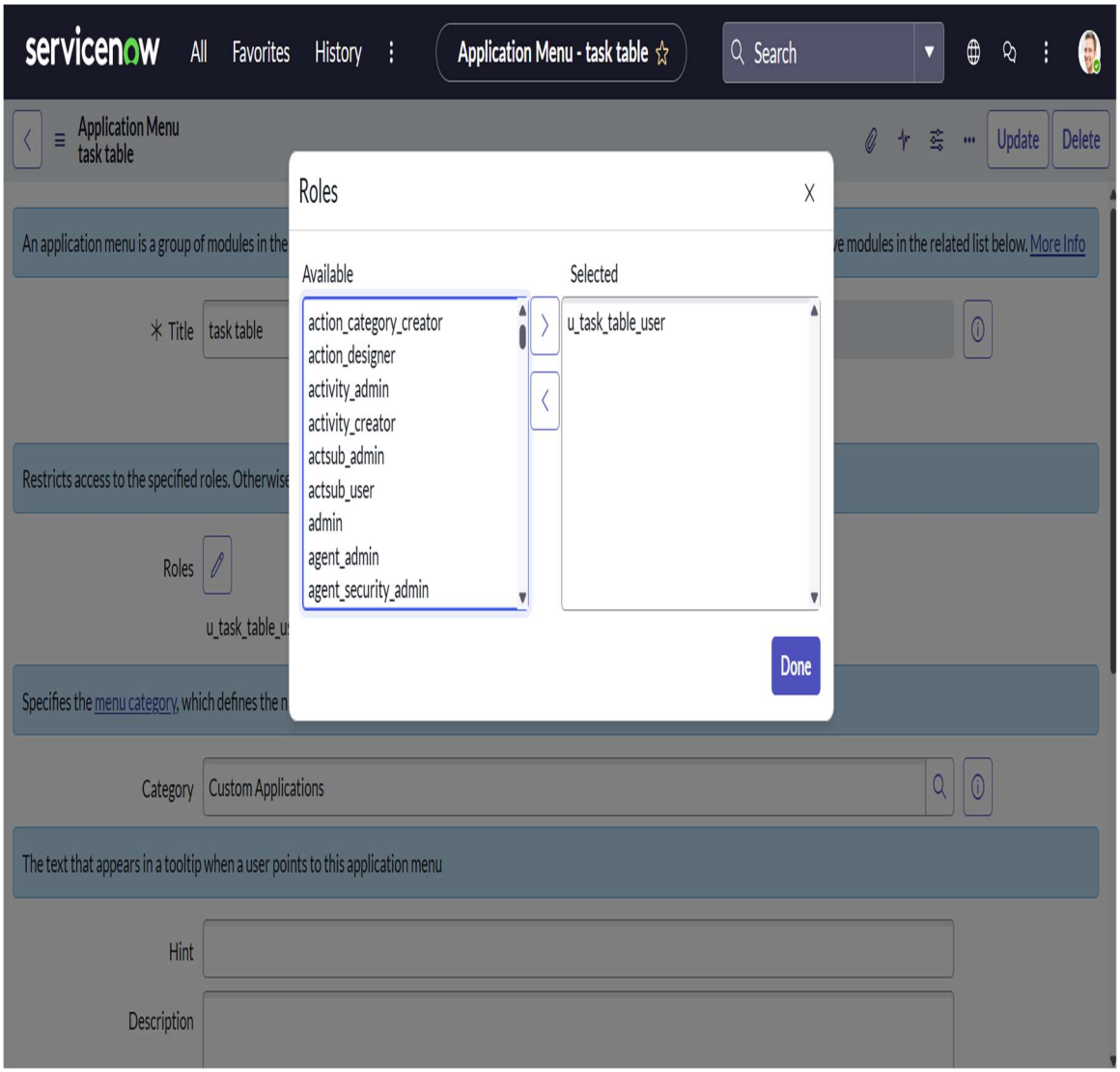
In the Project Task Tracker application, table access was assigned to control user permissions. When a new table is created in ServiceNow, an application and module are automatically generated for that table. The **Project Table** application was configured by editing its module and assigning the **Project Member** role to ensure only authorized members can access and manage project-related information.



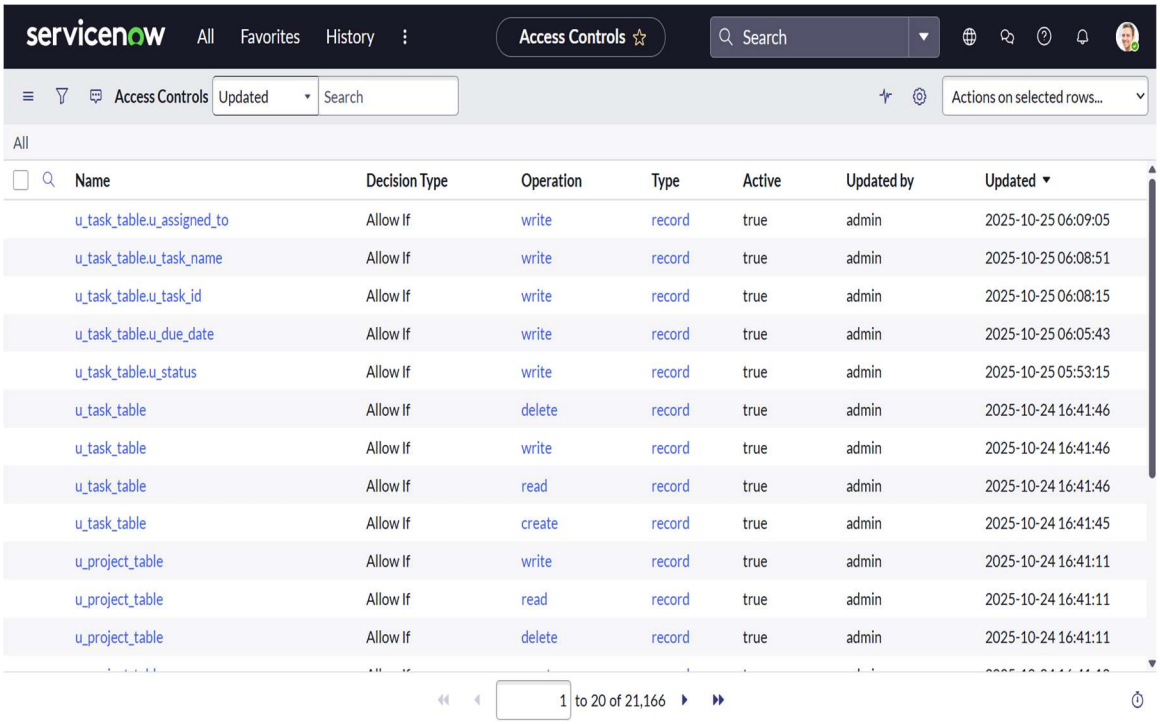


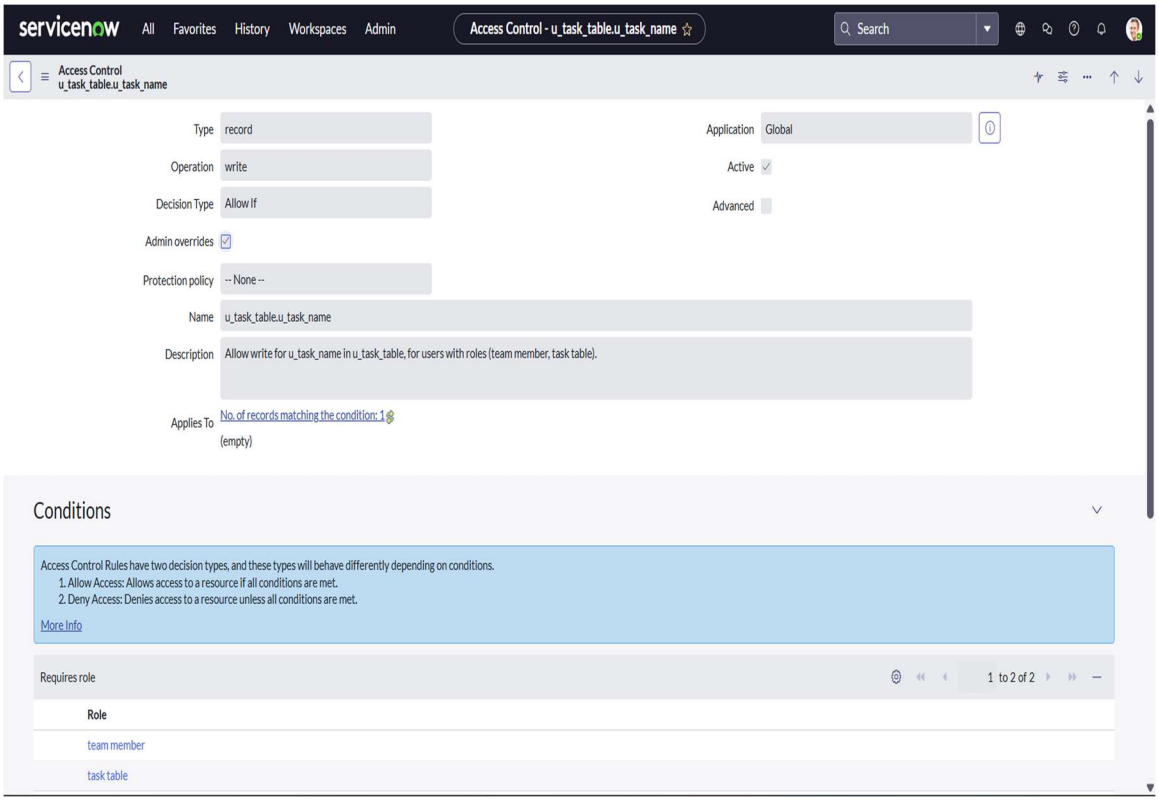
Similarly, the **Task Table** application was customized by editing its application settings and assigning both **Project Member** and **Team Member** roles. This allows project members and team members to access, update, and track task details efficiently, ensuring proper role-based access within the application.

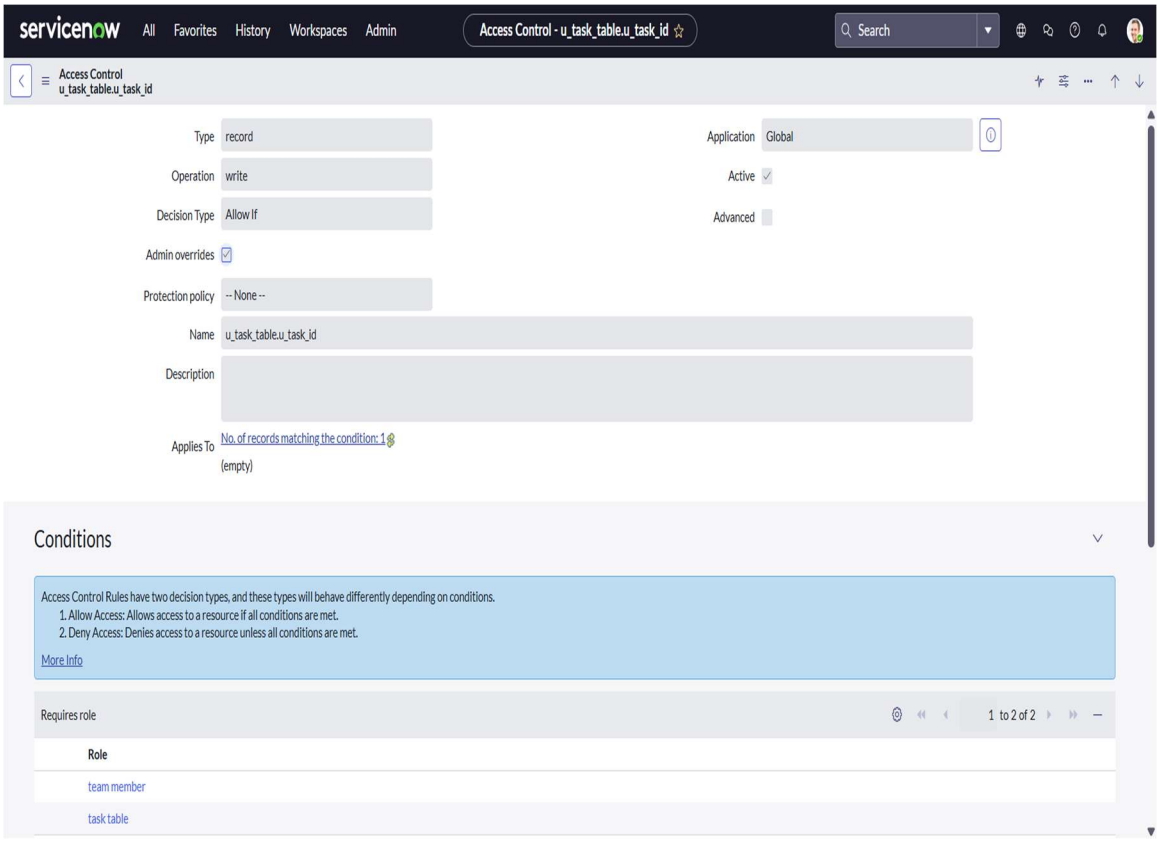


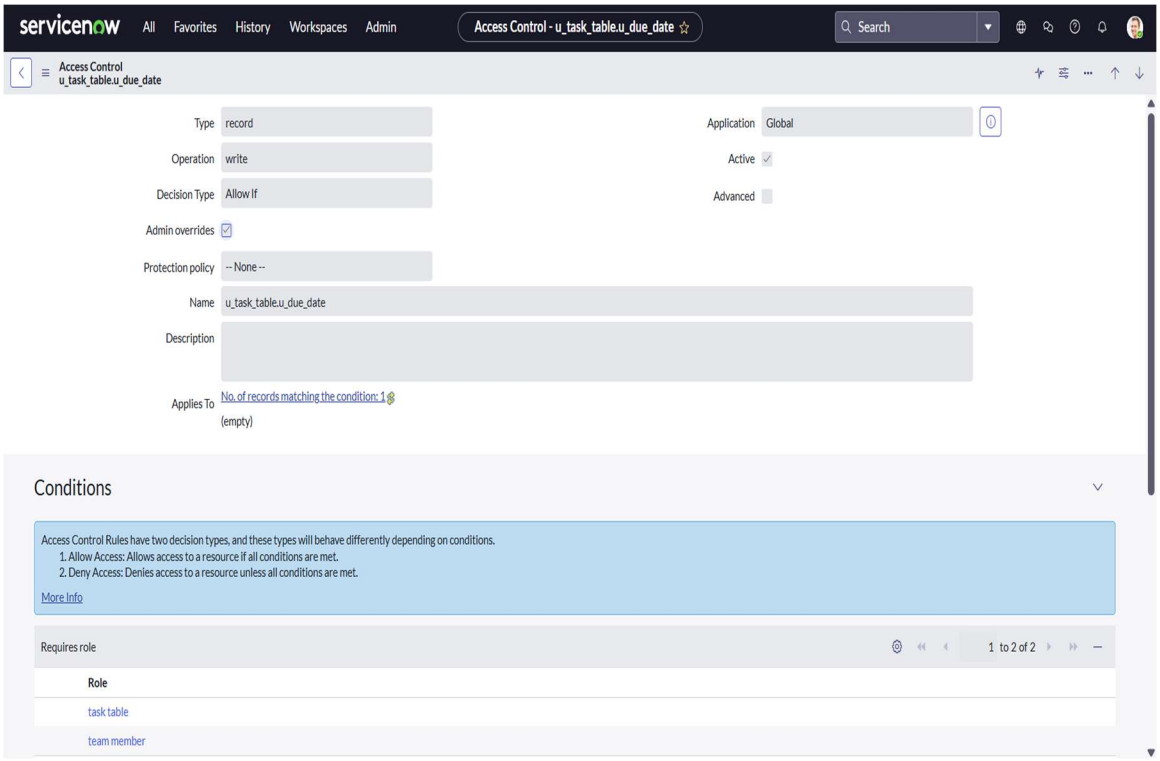


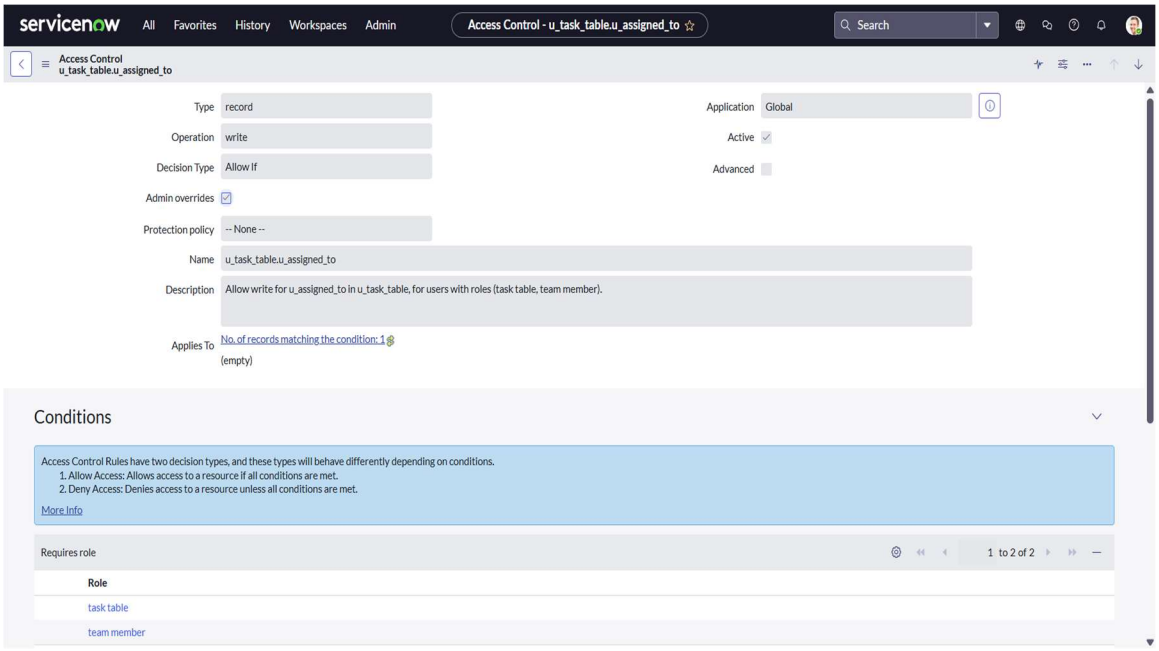
Custom forms were created for task entry, and Access Control Lists (ACLs) were configured to ensure secure and role-based access to data. The **Task Table** was assigned the **Team Member** role, allowing members to view and update only their assigned tasks, while the **Project Manager** role was given full access to create, modify, and delete tasks. Four ACLs were created for key fields such as **Task Name**, **Due Date**, **Assigned To**, **Status**, and **Comments** to control edit permissions. This configuration ensures that only authorized users can make changes, maintaining data integrity and proper access control within the Project Task Tracker application.

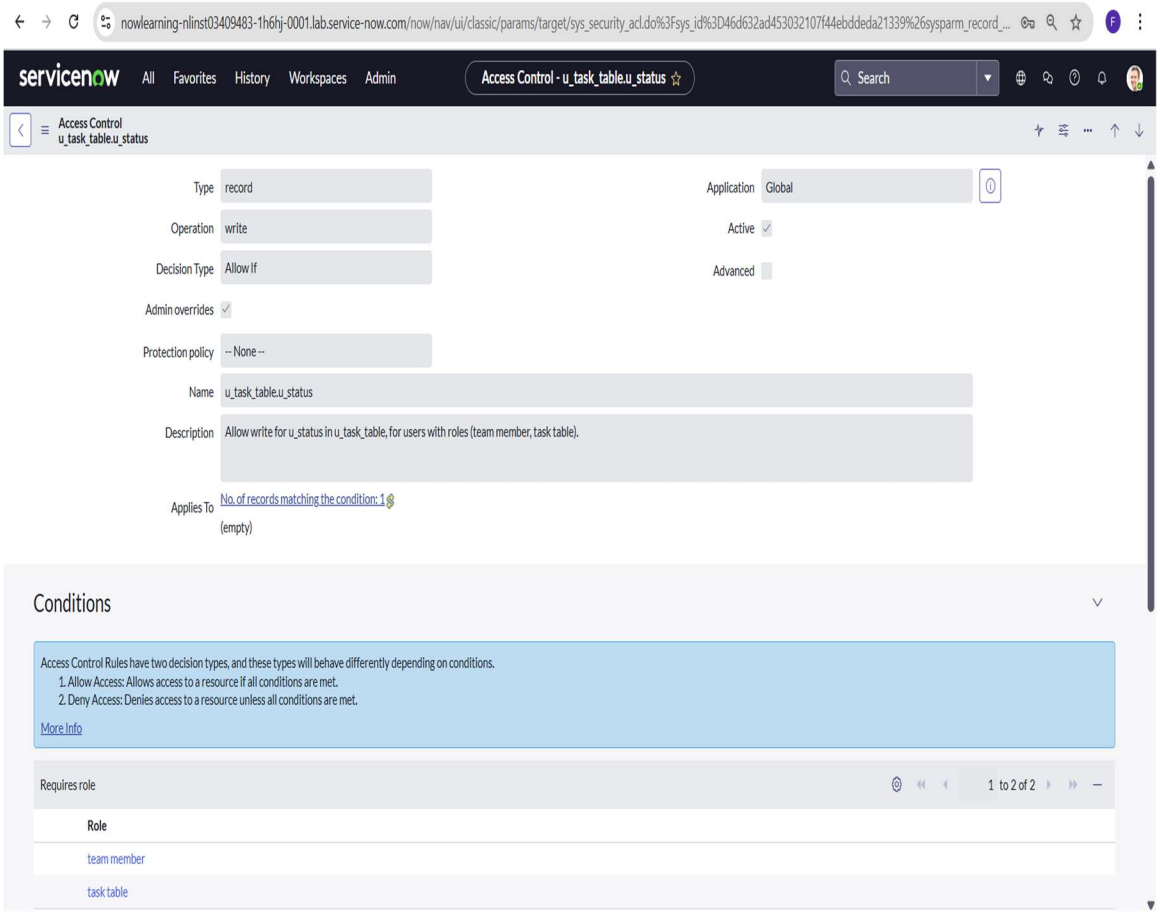

servicenow
All
Favorites
History
Access Controls
Search
Access Controls
Updated
Search
Actions on selected rows...
All
Name
Decision Type
Operation
Type
Active
Updated by
Updated
u_task_table.u_assigned_to Allow If write record true admin 2025-10-25 06:09:05
u_task_table.u_task_name Allow If write record true admin 2025-10-25 06:08:51
u_task_table.u_task_id Allow If write record true admin 2025-10-25 06:08:15
u_task_table.u_due_date Allow If write record true admin 2025-10-25 06:05:43
u_task_table.u_status Allow If write record true admin 2025-10-25 05:53:15
u_task_table Allow If delete record true admin 2025-10-24 16:41:46
u_task_table Allow If write record true admin 2025-10-24 16:41:46
u_task_table Allow If read record true admin 2025-10-24 16:41:46
u_task_table Allow If create record true admin 2025-10-24 16:41:45
u_project_table Allow If write record true admin 2025-10-24 16:41:11
u_project_table Allow If read record true admin 2025-10-24 16:41:11
u_project_table Allow If delete record true admin 2025-10-24 16:41:11
1 to 20 of 21,166

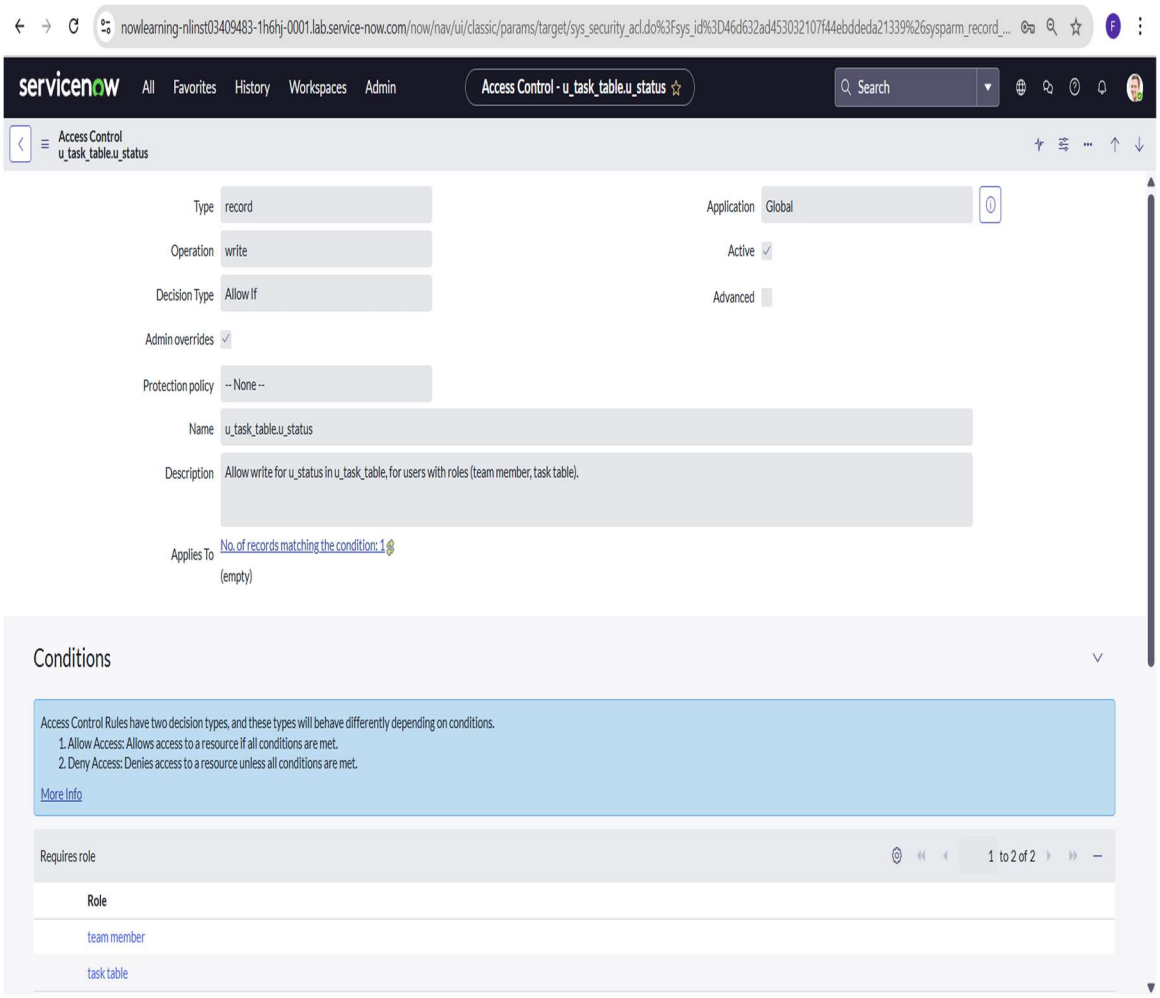

servicenow
All
Favorites
History
Workspaces
Admin
Access Control - u_task_table.u_task_name
Search
Access Control
u_task_table.u_task_name
Type record
Application Global
Operation write
Active
Decision Type Allow If
Advanced
Admin overrides
Protection policy -- None --
Name u_task_table.u_task_name
Description Allow write for u_task_name in u_task_table, for users with roles (team member, task table).
Applies To No. of records matching the condition: 1
(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
task table


servicenow
All
Favorites
History
Workspaces
Admin
Access Control - u_task_table.u_task_id
Search
Access Control
u_task_table.u_task_id
Type
record
Application
Global
Operation
write
Active
Decision Type
Allow If
Advanced
Admin overrides
Protection policy
-- None --
Name
u_task_table.u_task_id
Description
Applies To
No. of records matching the condition: 1
(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
task table


servicenow
All
Favorites
History
Workspaces
Admin
Access Control - u_task_table.u_due_date
Search
Access Control
u_task_table.u_due_date
Type
record
Application
Global
Operation
write
Active
Decision Type
Allow If
Advanced
Admin overrides
Protection policy
-- None --
Name
u_task_table.u_due_date
Description
Applies To
No. of records matching the condition: 1
(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
task table
team member

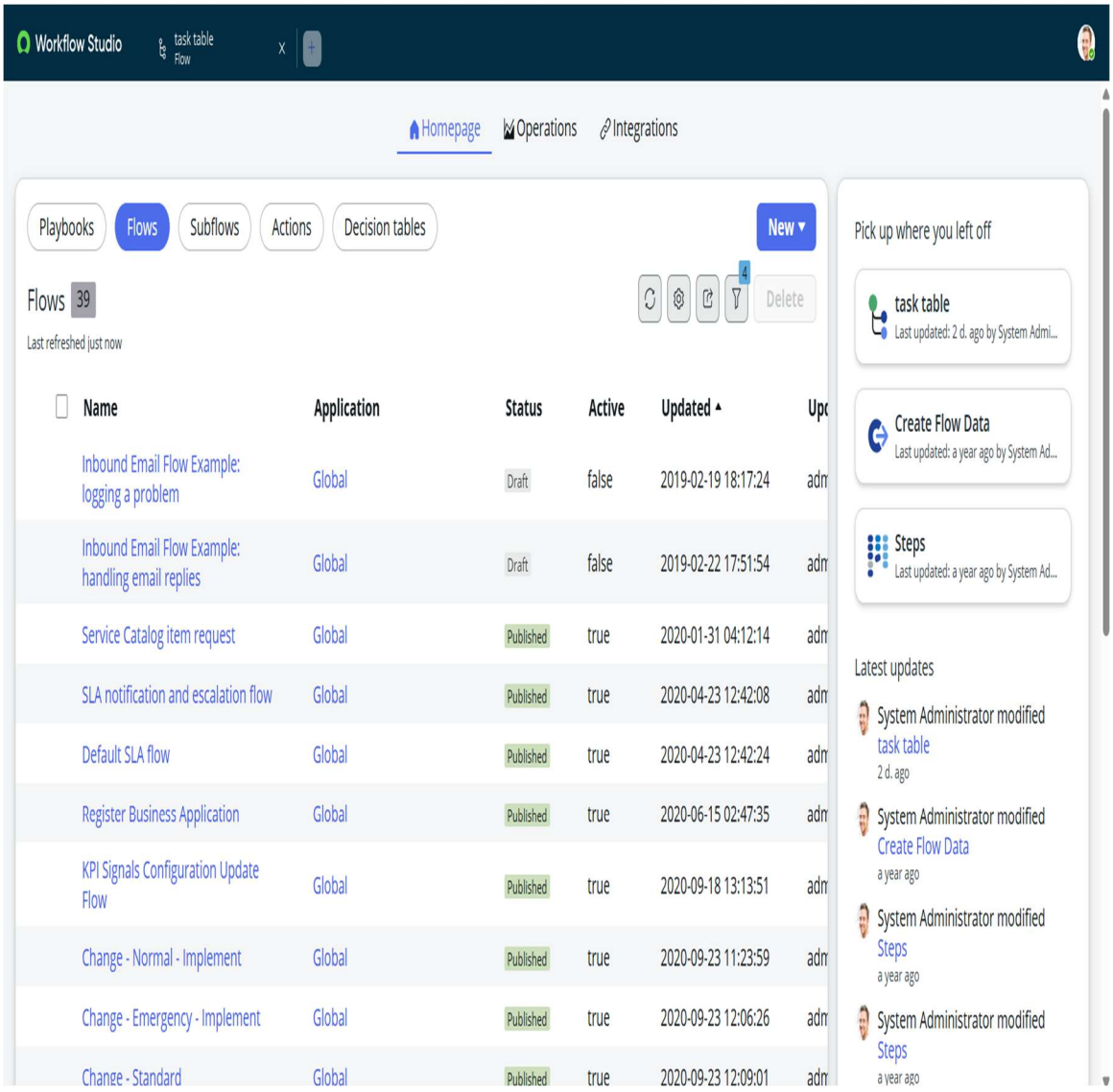

servicenow
All
Favorites
History
Workspaces
Admin
Access Control - u_task_table.u_assigned_to
Search
Access Control
u_task_table.u_assigned_to
Type
record
Application
Global
Operation
write
Active
Decision Type
Allow If
Advanced
Admin overrides
Protection policy
-- None --
Name
u_task_table.u_assigned_to
Description
Allow write for u_assigned_to in u_task_table, for users with roles (task table, team member).
Applies To
No. of records matching the condition: 1
(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
task table
team member

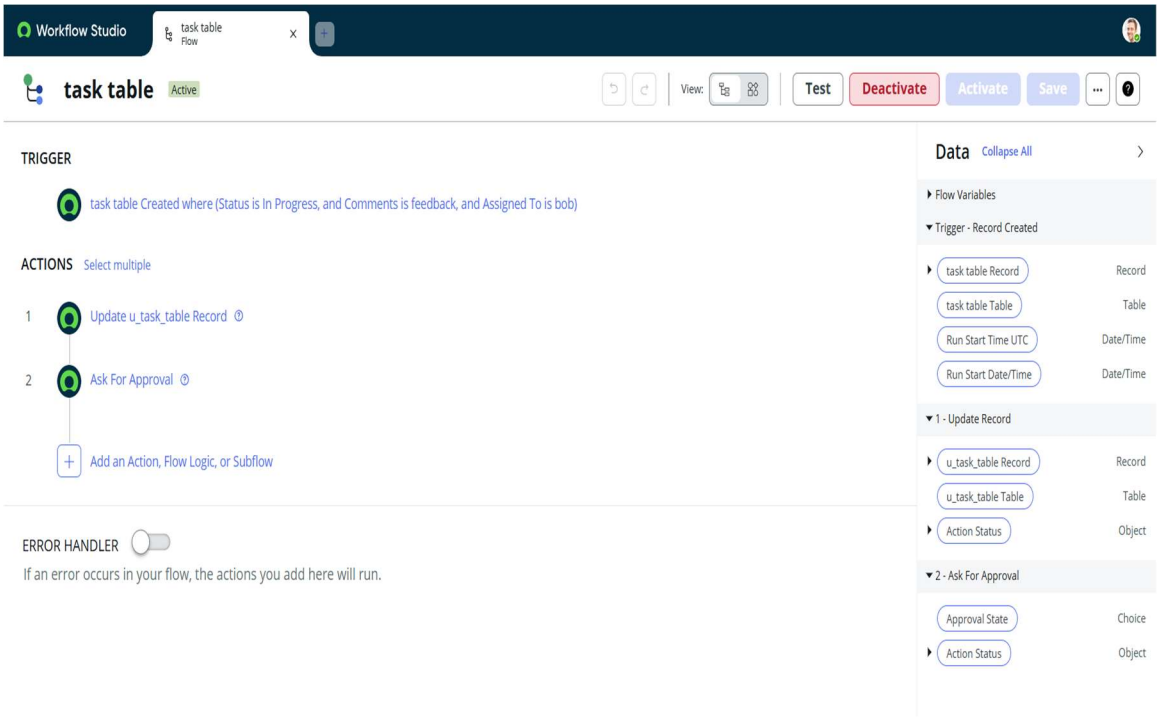

nowlearning-nlinst03409483-1h6hj-0001.lab.service-now.com/now/nav/ui/classic/params/target/sys_security_acl.do%3Fsys_id%3D46d632ad453032107f44ebddeda21339%26sysparm_record_...
servicenow
All
Favorites
History
Workspaces
Admin
Access Control - u_task_table.u_status
Search
Access Control
u_task_table.u_status
Type
record
Application
Global
Operation
write
Active
Decision Type
Allow If
Advanced
Admin overrides
Protection policy
-- None --
Name
u_task_table.u_status
Description
Allow write for u_status in u_task_table, for users with roles (team member, task table).
Applies To
No. of records matching the condition: 1
(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
task table

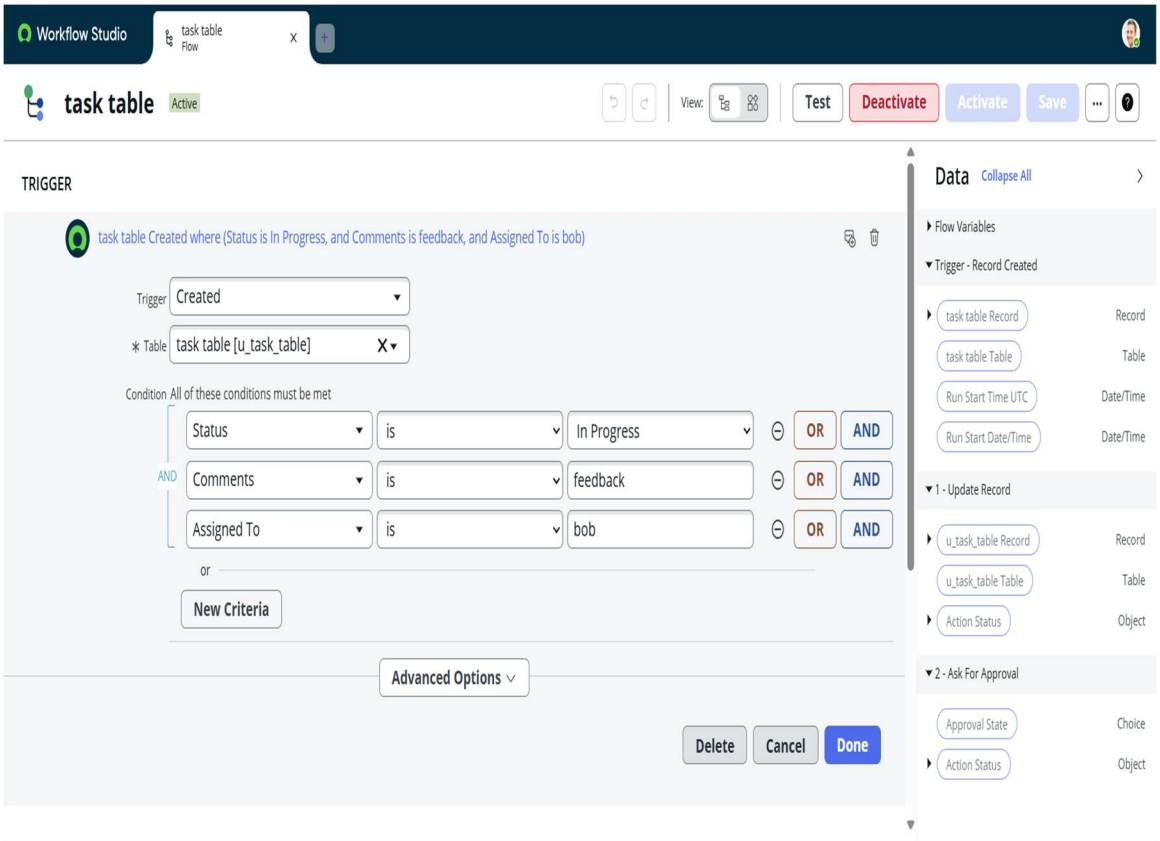

nowlearning-nlinst03409483-1h6hj-0001.lab.service-now.com/now/nav/ui/classic/params/target/sys_security_acl.do%3Fsys_id%3D46d632ad453032107f44ebddeda21339%26sysparm_record_...
servicenow
All
Favorites
History
Workspaces
Admin
Access Control - u_task_table.u_status
Search
Access Control
u_task_table.u_status
Type
record
Application
Global
Operation
write
Active
Decision Type
Allow If
Advanced
Admin overrides
Protection policy
-- None --
Name
u_task_table.u_status
Description
Allow write for u_status in u_task_table, for users with roles (team member, task table).
Applies To
No. of records matching the condition: 1
(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
task table

flow designer implementation:

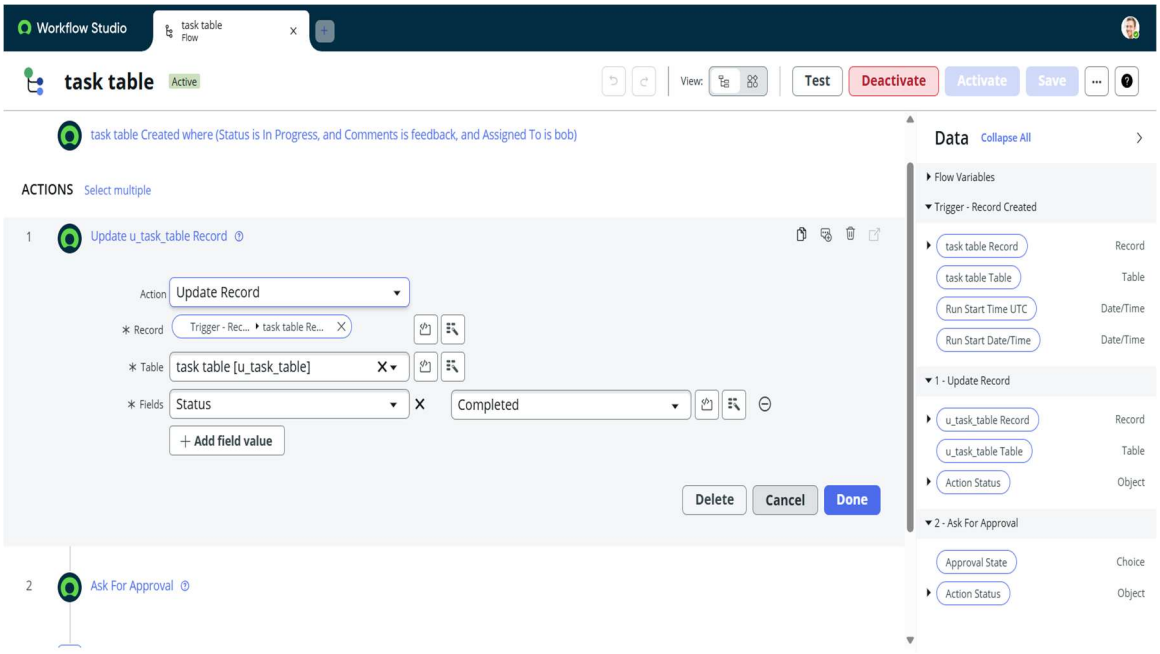
Flow Designer in ServiceNow is used to automate processes and enhance workflow efficiency without the need for coding. In the Project Task Tracker application, Flow Designer was utilized to automate task updates and approvals. A new flow was created with the **Task Table** as the trigger, which activates whenever a task is assigned or updated.

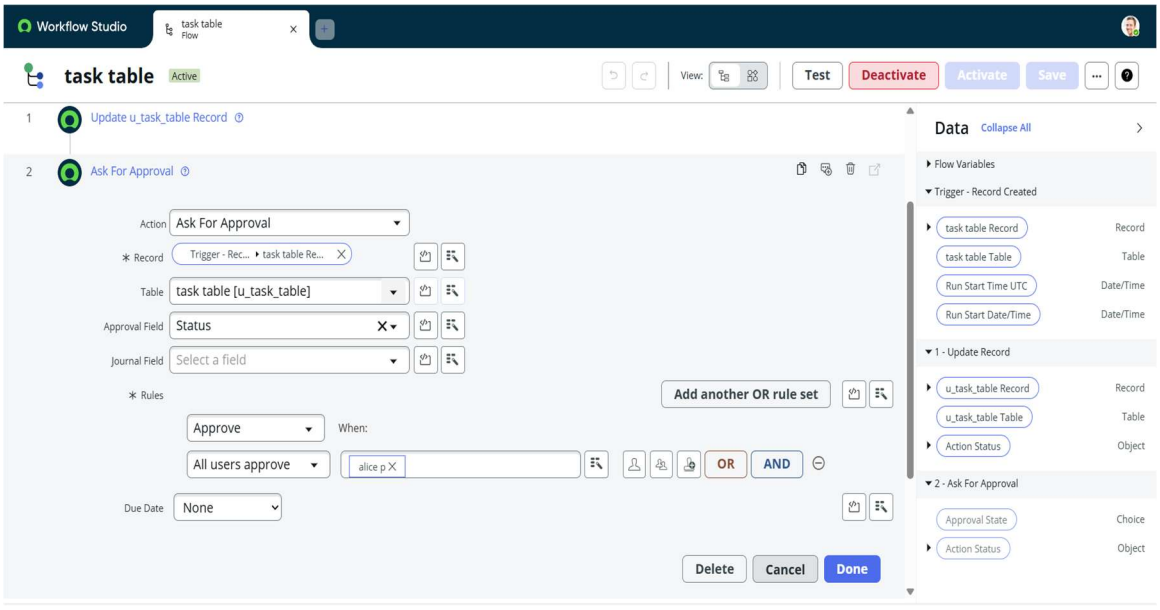



Workflow Studio
task table
Flow
task table
Active
View:
Test
Deactivate
Activate
Save
TRIGGER
task table Created where (Status is In Progress, and Comments is feedback, and Assigned To is bob)
ACTIONS
Select multiple
1
Update u_task_table Record
2
Ask For Approval
Add an Action, Flow Logic, or Subflow
ERROR HANDLER
If an error occurs in your flow, the actions you add here will run.
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
u_task_table Record
Record
u_task_table Table
Table
Action Status
Object
2 - Ask For Approval
Approval State
Choice
Action Status
Object

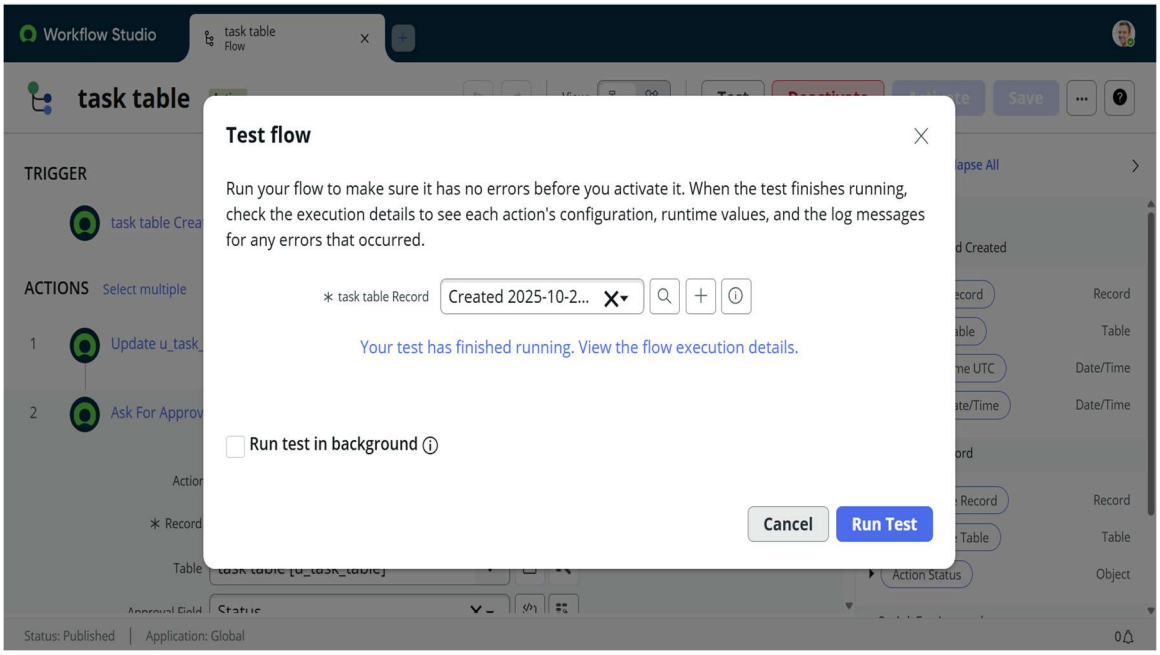

Workflow Studio
task table
Flow
task table
Active
View:
Test
Deactivate
Activate
Save
TRIGGER
task table Created where (Status is In Progress, and Comments is feedback, and Assigned To is bob)
Trigger
Created
* Table
task table [u_task_table]
Condition
All of these conditions must be met
Status
is
In Progress
OR
AND
AND
Comments
is
feedback
OR
AND
Assigned To
is
bob
OR
AND
or
New Criteria
Advanced Options
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
u_task_table Record
Record
u_task_table Table
Table
Action Status
Object
2 - Ask For Approval
Approval State
Choice
Action Status
Object

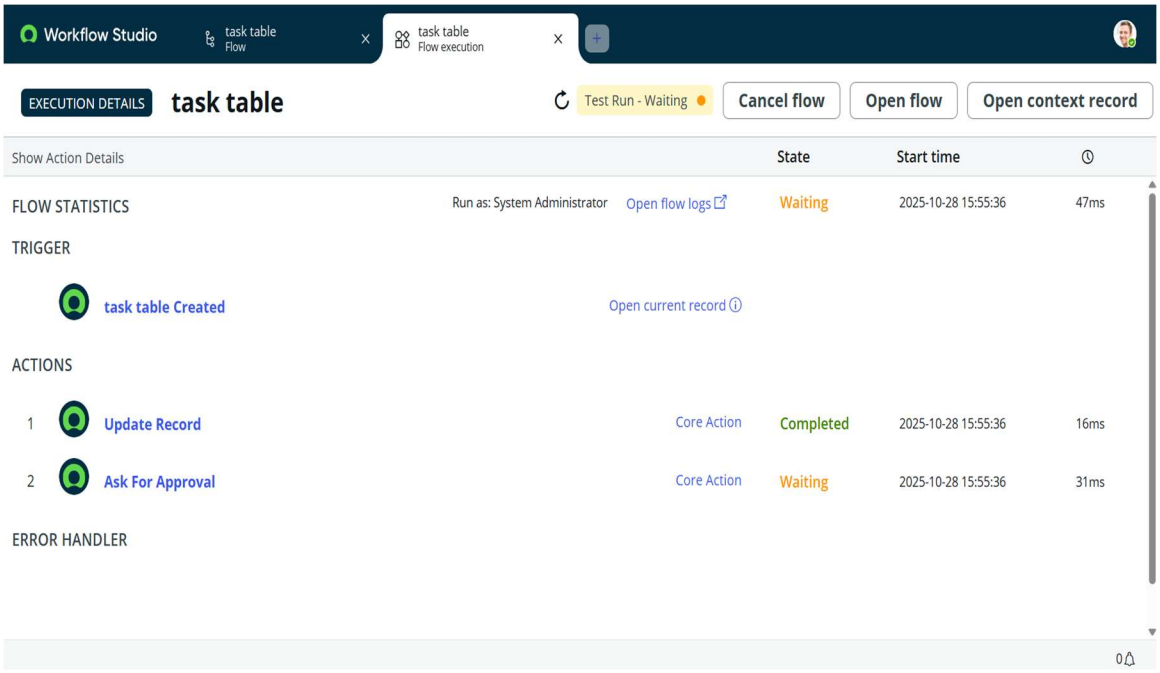
The flow includes several **actions** to define the automation steps. The first action checks if the **Assigned To** field is not empty. If true, the next **Update Record** action automatically changes the **Status** field to *In Progress*. Following this, an **Ask for Approval** action is added to request approval from the **Project Manager** once the task reaches completion. This ensures that every task is reviewed and approved before being marked as completed.



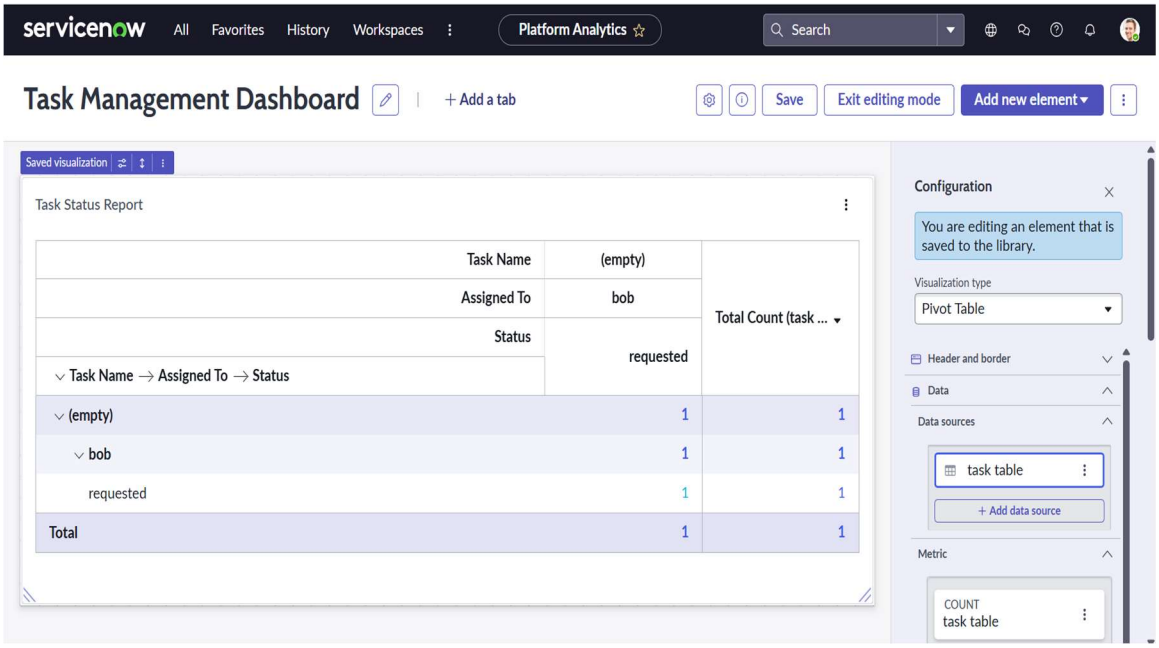


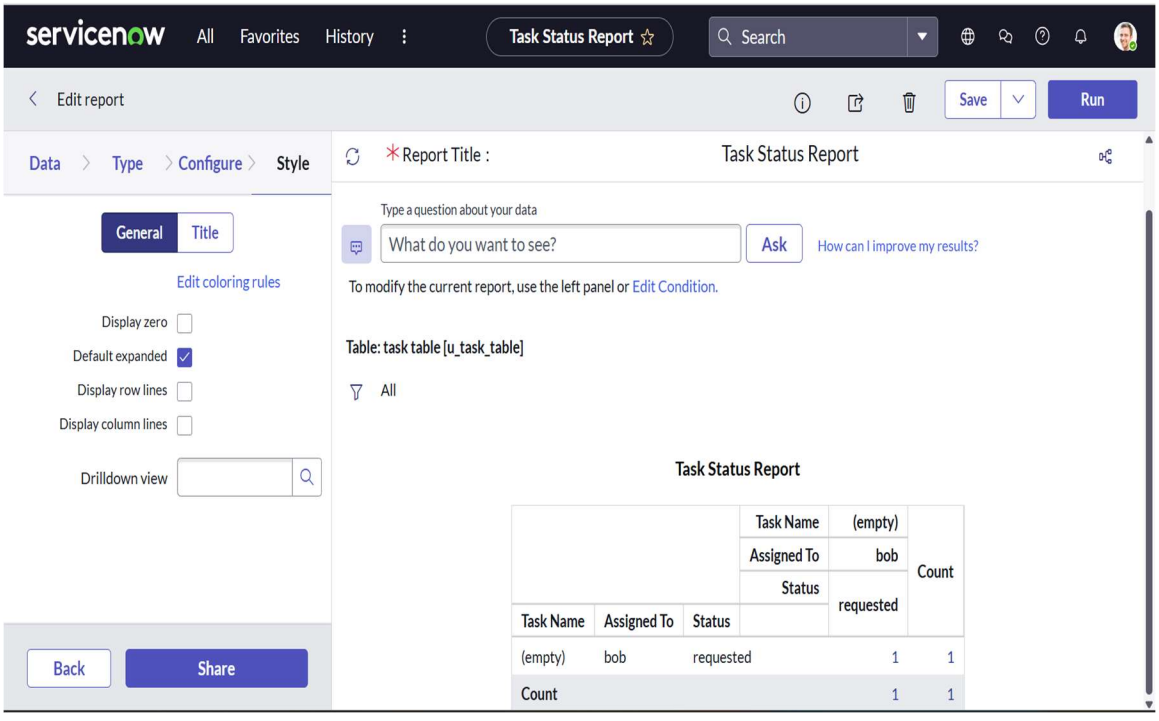
The **Test** feature in Flow Designer was used to validate the workflow. A sample record was selected from the **Task Table** to confirm that the flow triggered correctly, updated the record automatically, and sent the approval request as expected. This automation improves accuracy, ensures timely approvals, and reduces manual intervention in the task management process.





• Dashboard and Reporting: The Platform Analytics module in ServiceNow provides visual dashboards showing project performance. Reports such as pie charts by task status, bar graphs of user workloads, and pivot tables for project distribution are created. Managers use these insights to allocate resources efficiently and track progress.





CONCLUSION AND FUTURE SCOPE

The ServiceNow-based role management system successfully demonstrates how access control and workflow automation can transform traditional project management practices. The system ensures that tasks are securely assigned, monitored, and updated without manual intervention. Managers benefit from real-time analytics and dashboards that provide visibility into project performance, while team members gain clarity in their assigned responsibilities.

In the future, this system can be expanded to include integrations with collaboration tools such as Microsoft Teams, Slack, and Jira. Machine learning features can be added to predict task delays and recommend workload balancing. The automation logic can be further improved to include escalations and approvals. By continuously refining the workflows, organizations can achieve even greater productivity and security through the ServiceNow platform.