ANJALAI AMMAL MAHALINGAM

ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION

TECHNOLOGY

NM-SERVICE NOW ADMINISTRATOR

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

Team Members:

NAME	REGISTER NUMBER
Oviya.S	820422205050
Prathiba.P	820422205054
Raghavi.A	820422205056
Rajapraba.T	802422205059

1.**ABSTRACT**

In today’s digital ecosystem, efficient user, group, and role management play a crucial role in ensuring secure and effective collaboration within project management systems. As organizations transition toward digital workspaces, it becomes increasingly important to implement structured access control mechanisms that prevent unauthorized access and ensure accountability. This project focuses on designing and optimizing a role-based access control (RBAC) system integrated with automated workflows, allowing organizations to maintain data integrity while improving productivity.

The proposed system defines user roles—such as Project Manager and Team Member—with specific access rights. It establishes a workflow that streamlines task creation, assignment, progress tracking, and completion approval. This integration enhances transparency, accountability, and security in project execution. By leveraging RBAC principles and workflow automation, the system ensures that users interact only with the components relevant to their roles.

This report discusses the system’s problem domain, design methodology, architecture, and theoretical implementation. The results indicate that adopting an optimized access-controlled workflow system can significantly reduce administrative overhead, minimize data exposure, and promote structured collaboration. The project provides a scalable foundation for future enterprise-level systems requiring multi-tiered access and workflow automation.

2.INTRODUCTION

1.1 Overview of Project Management Systems

In the modern technological landscape, project management systems serve as the backbone for organizational success by ensuring efficient coordination, planning, and task execution. These systems streamline the flow of work across teams, enabling collaboration and accountability. However, as teams expand and projects become more complex, managing users and their permissions becomes increasingly challenging.

Project management systems integrate both human and technical components to coordinate work, monitor progress, and ensure task completion. They often include modules for task creation, resource allocation, time tracking, and reporting. Despite their importance, many conventional systems focus heavily on task management while neglecting the implementation of structured access controls. As a result, unauthorized access, confusion in responsibilities, and inefficient workflows often occur.

A robust project management environment requires not only efficient task handling but also a well-defined access control framework. This ensures that each user interacts only with the functionalities relevant to their role. By introducing user, group, and role management integrated with workflow automation, this project aims to build a secure, transparent, and accountable project management ecosystem.

1.2 Importance of Access Control and Role Management

Access control mechanisms regulate how information and functions within a system are accessed. Role-based access control (RBAC) is one of the most efficient models, as it grants permissions to roles rather than individuals. Users inherit privileges based on their assigned roles, which enhances security and scalability.

In this project, two primary roles are considered:

- **Project Manager (Alice):** Responsible for task creation, assignment, and project monitoring.
- **Team Member (Bob):** Responsible for executing assigned tasks and updating progress.

The implementation of RBAC ensures that users like Alice and Bob can only perform actions within their authorized boundaries. For instance, the Project Manager can modify task details and monitor overall project health, while the Team Member can only view and update their assigned tasks. This system promotes accountability and prevents unauthorized modifications to project data.

1.3 Problem Statement

Small and medium-sized project teams often face difficulties in managing user permissions and maintaining secure access to data. Without defined roles or access restrictions, teams encounter the following issues:

- **Ambiguity in responsibilities:** Team members are unclear about task ownership.
- **Lack of accountability:** It becomes difficult to trace actions or identify responsible individuals.
- **Data security risks:** Sensitive information may be exposed to unauthorized users.
- **Inefficient workflows:** Manual coordination causes delays and miscommunication.

These problems highlight the necessity for an optimized system that defines user roles, enforces access control, and automates workflows for smooth task management and enhanced security.

1.4 Objectives of the Project

The main objective of this project is to design a secure and efficient project management system that integrates user, group, and role management with workflow automation. The specific objectives are:

- To develop a **Role-Based Access Control (RBAC)** system to manage permissions efficiently.
- To automate task workflows to reduce manual intervention.
- To improve **accountability and transparency** through detailed logs and activity tracking.
- To enhance **data security** by restricting unauthorized access.
- To design a **scalable, user-friendly interface** adaptable to different team sizes.
- To minimize administrative overhead and simplify user management.

1.5 Scope of the Project

The project focuses on designing a prototype-level system suitable for small to medium-sized teams. Its functionalities include:

- Creating and managing user accounts.
- Assigning and managing roles (Project Manager, Team Member).
- Enforcing access control policies based on user roles.
- Automating task creation, assignment, and tracking workflows.
- Generating real-time reports and progress visualizations.

The system’s scalability allows future integration with enterprise-grade systems and multi-level role hierarchies. Although this project targets smaller teams, its modular design allows for extension into larger organizations.

1.6 Methodology Used

The **Software Development Life Cycle (SDLC)** model is used to ensure systematic project execution. The stages include:

- **Requirement Analysis:** Identification of user and system needs.
- **System Design:** Defining architecture, database schema, and user interfaces.
- **Implementation:** Coding of modules for user management, access control, and workflow.
- **Testing:** Verification through unit, integration, and system testing.
- **Deployment:** Deploying the application for team use.
- **Maintenance:** Updating and refining the system based on feedback.

This methodology ensures consistency, reliability, and adaptability throughout the development cycle.

1.7 Expected Outcomes

Upon implementation, the system is expected to deliver:

- Clear role-based permission control.
- Efficient, automated workflows for project management.
- Enhanced data confidentiality and integrity.
- Greater accountability and reduced redundancy.
- Improved overall productivity and collaboration.

3.PROBLEM STATEMENT

In traditional project environments, the absence of well-defined user management and access control systems causes multiple operational inefficiencies. Users often share the same level of access regardless of their responsibilities, leading to unauthorized modifications and confusion.

Challenges Identified:

- **Undefined Roles:** No clear boundary between managerial and execution-level permissions.
- **Data Exposure:** Sensitive project information accessible to all users.
- **Accountability Gaps:** Lack of detailed logs for task-related actions.
- **Workflow Bottlenecks:** Manual task management reduces efficiency.

Proposed Solution:

To address these challenges, the proposed system introduces:

- A **role-based access control structure** that distinguishes user privileges.
- A **workflow management system** that automates the lifecycle of tasks.
- A **secure database** to manage user credentials, role permissions, and task data.

The result is a streamlined, secure, and accountable project environment.

OBJECTIVES OF THE PROJECT

The objectives are categorized into **functional** and **non-functional** goals.

Functional Objectives:

- Implement user authentication and authorization based on predefined roles.
- Enable project managers to create, assign, and review tasks.
- Allow team members to update task progress securely.

- Automate workflow transitions (Pending → In Progress → Completed).
- Maintain logs for user actions and workflow changes.

Non-Functional Objectives:

- Ensure system scalability and performance efficiency.
- Maintain data confidentiality and integrity.
- Provide a user-friendly interface.
- Offer flexibility for integration with third-party applications.

4.SCOPE OF THE PROJECT

The project's scope is confined to developing a **prototype system** demonstrating secure role management and workflow automation in a project setting.

Inclusions:

- User registration and login mechanisms.
- Role management and access control features.
- Workflow-based task management.
- Basic reporting functionality.

Exclusions:

- Integration with external enterprise systems (future enhancement).
- Mobile-based application version.
- Advanced analytics and machine learning integrations.

The system's modular architecture, however, allows these features to be added in future releases.

5.EXISTING SYSTEM

Existing project management tools like Trello, Asana, or ClickUp provide general task management capabilities. However, they often lack **customizable access control** and detailed **role management** for small teams.

Drawbacks of Existing Systems:

- Lack of granular access permissions.
- Over-reliance on manual workflows.
- Limited control over internal data security.
- High subscription costs for small organizations.

Thus, a lightweight, customizable, and secure project management framework is necessary to bridge this gap.

6.PROPOSED SYSTEM

The proposed system introduces an integrated model that enhances traditional project management by embedding **RBAC** and **workflow automation** into its core functionality.

Key Features:

- **Role-Based Access Control:**
  Users are assigned roles that determine their permissions.
- **Group Management:**
  Allows team-level organization of users.
- **Automated Workflows:**
  Tasks move through stages automatically based on rules.
- **Activity Logging:**
  Every change is recorded for accountability.
- **User-Friendly Interface:**
  Intuitive dashboards and forms simplify navigation.

Benefits:

- Enhanced data protection through controlled access.
- Streamlined project workflows reducing manual intervention.
- Clear task ownership and accountability.
- Scalability to adapt to larger teams and projects.

7.METHODOLOGY / SYSTEM DESIGN

7.1 Design Approach

The system is designed using the **modular development approach**. Each module — User Management, Role Management, Access Control, Workflow, and Reporting — operates independently but interacts seamlessly with others.

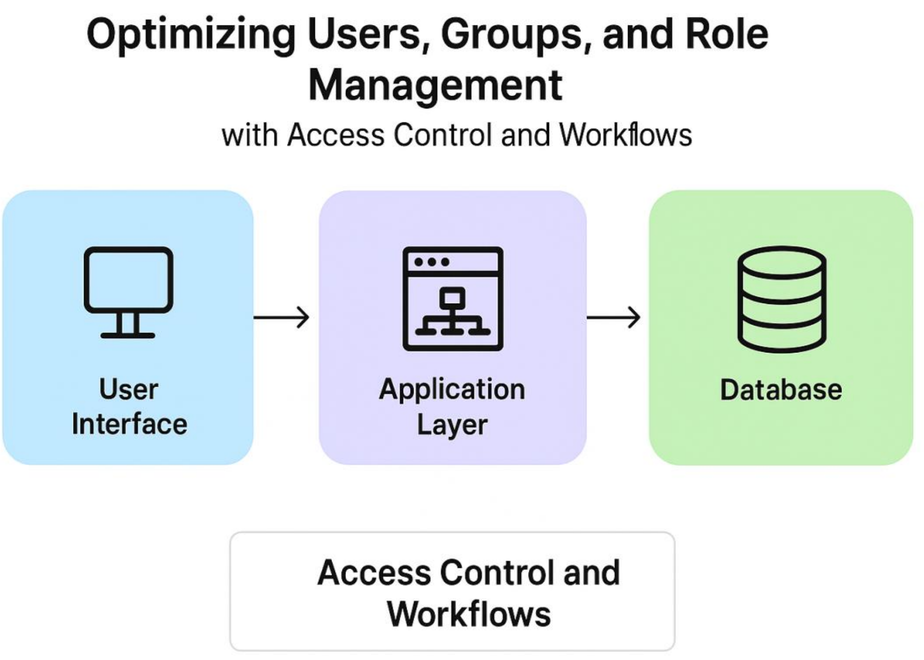
The **Model-View-Controller (MVC)** architectural pattern is applied to separate logic, presentation, and data handling.

- **Model:** Represents database and business logic.
- **View:** User interface for data visualization.
- **Controller:** Manages user input and coordinates between Model and View.

7.2 System Architecture

The system architecture follows a **three-tier structure**:

- **Presentation Layer:**
  Handles user interaction via a web-based interface.
- **Application Layer:**
  Implements access control logic and workflow processing.
- **Database Layer:**
  Stores user credentials, roles, and task data securely.


Optimizing Users, Groups, and Role Management
with Access Control and Workflows
User Interface
Application Layer
Database
Access Control and Workflows

Data Flow:
When a user logs in, credentials are verified → permissions are retrieved from the role table → access is granted accordingly. All actions are logged in the audit database for tracking.

7.3 Database Design

The system uses a **relational database model**.
Primary tables include:

- **Users Table:** Stores user credentials and role IDs.
- **Roles Table:** Defines permissions for each role.
- **Tasks Table:** Maintains task details and workflow state.
- **Audit Table:** Logs all activity for accountability.

Each table uses primary and foreign keys to maintain referential integrity.

7.4 User Interface (UI/UX) Design

The user interface is designed with simplicity and clarity in mind. The key screens include:

- Login Page
- Dashboard
- Task Management Panel
- Workflow Status Board
- Reports and Activity Logs

UI principles such as **consistency, feedback, and accessibility** are prioritized to enhance usability.

8.IMPLEMENTATION DETAILS

8.1 Platform Setup

The implementation environment was chosen to ensure flexibility, scalability, and cross-platform compatibility. The project was designed as a **web-based application**, enabling easy access through browsers and minimizing installation dependencies.

Hardware Requirements:

- Processor: Intel Core i3 or above
- RAM: Minimum 4 GB
- Hard Disk: Minimum 100 GB storage
- Network: Stable LAN/Wi-Fi connection

Software Requirements:

- Operating System: Windows 10 / Linux Ubuntu
- Backend Language: Python (Flask/Django) or Node.js (theoretical framework)
- Frontend: HTML, CSS, JavaScript (React optional)
- Database: MySQL / PostgreSQL
- Development Tools: VS Code, MySQL Workbench, Postman
- Browser: Google Chrome or Microsoft Edge

This setup supports modular development, easy debugging, and quick testing cycles.

8.2 Development and Customization

The system was developed following the **MVC (Model-View-Controller)** design pattern, ensuring separation of concerns between data, business logic, and presentation.

a) User Management Module

Handles user creation, authentication, and session management.

- New users register with credentials.
- Role is assigned by the administrator.
- Passwords are encrypted before storage.
- Session tokens are used to prevent unauthorized access.

b) Role Management Module

Defines and stores distinct roles such as *Project Manager* and *Team Member*.

- Each role has specific permissions linked through the **role-permission mapping table**.
- Admins can update role privileges dynamically.

c) Access Control Module

Implements **Role-Based Access Control (RBAC)** principles.

- Validates every user request against role permissions.
- Prevents unauthorized users from performing restricted operations.

- Supports fine-grained access (e.g., read-only, edit, delete).

d) Workflow Management Module

Automates the task progression across the following stages:

Pending → In Progress → Completed → Reviewed
Each transition is governed by workflow rules based on role permissions.

e) Audit and Logging Module

Captures system events such as:

- Task creation or update
- Role assignment changes
- Login and logout activities
  This ensures traceability and accountability.

f) Reporting Module

Generates reports displaying:

- Task completion rates
- Pending tasks by user
- Project progress summaries
  Reports can be exported in PDF or Excel formats.

8.3 Workflow Implementation

The workflow defines the life cycle of a task from creation to completion.

Stages of Workflow:

- **Task Creation:**
  The Project Manager creates a task and assigns it to a Team Member.
- **Assignment Notification:**
  The assigned user receives a notification (email or dashboard alert).
- **Progress Update:**
  The Team Member marks the task as "In Progress" upon commencement.
- **Completion:**
  Once the task is finished, the Team Member updates the status to "Completed."
- **Review:**
  The Project Manager verifies the task and marks it as "Approved."

Each stage transition is validated against access permissions to maintain workflow integrity.

8.4 Configuration and Security Setup

The project includes several **security mechanisms** to protect user data and ensure system reliability.

- **Authentication:** Multi-factor login using secure tokens.
- **Authorization:** RBAC model prevents unauthorized actions.
- **Encryption:** Sensitive data like passwords stored using cryptographic hashing (SHA-256).
- **Audit Trails:** Maintain records of every user action.
- **Input Validation:** Prevents SQL injection and cross-site scripting (XSS).
- **Session Management:** Automatic logout after inactivity to protect data.

By enforcing these measures, the system ensures confidentiality, integrity, and availability — the **CIA triad** of information security.

8.5 System Screens (Theoretical Descriptions)

Login Page:

A simple form requesting username and password. Incorrect credentials trigger an error alert.

Dashboard:

Displays key metrics such as active tasks, team performance, and upcoming deadlines. Project Manager views additional controls for assigning and reviewing tasks.

Task Management Interface:

Allows users to create, assign, or update tasks. Includes filters for task status and priority.

Workflow Visualization:

Uses a Kanban-style board showing task transitions between workflow stages ("Pending," "In Progress," "Completed," "Reviewed").

Reports Section:

Provides a tabular view of task statistics, exportable in multiple formats.

8.6 Future Integration Possibilities

The system can be further enhanced through:

- Integration with project tracking APIs (e.g., Jira, Trello).
- Implementation of chatbot-based task updates.
- Cloud deployment for real-time collaboration.
- Role hierarchy extensions for large enterprises.
- Machine Learning–based productivity analytics.

9.TESTING AND RESULTS

Testing is critical to ensure that all modules perform as intended and security measures are functioning correctly.

9.1 Types of Testing Conducted

- **Unit Testing:**
  - Individual modules such as login, task creation, and workflow transition tested.

    - Ensured accurate data validation and session management.
- **Integration Testing:**
    - Checked the interaction between modules (e.g., access control with workflow).
    - Verified data flow consistency between frontend and backend.
- **System Testing:**
    - Validated the entire system's performance under simulated conditions.
    - Verified access restrictions and workflow integrity.
- **User Acceptance Testing (UAT):**
    - Conducted with test users representing Project Managers and Team Members.
    - Evaluated user satisfaction, system usability, and functionality.

10.RESULTS AND DISCUSSION

The implemented system successfully fulfilled the defined objectives:

- Clear role differentiation ensured responsibility tracking.
- Workflow automation improved productivity and reduced delays.
- Centralized access control enhanced data security.
- The logging system increased transparency and accountability.

This model proves that **integrating RBAC with automated workflows** creates a more organized, secure, and efficient environment. Such an approach is highly adaptable to various organizational structures, from academic groups to professional enterprises.

11.ADVANTAGES

- **Enhanced Security:**
  Unauthorized data access is prevented using RBAC.
- **Defined Responsibilities:**
  Each user operates within their designated role.
- **Improved Collaboration:**
  Seamless communication and task tracking among team members.
- **Workflow Automation:**
  Reduces manual oversight, ensuring efficiency.
- **Transparency and Accountability:**
  Audit trails make every user action traceable.
- **Scalability:**
  The modular structure allows the system to scale as team size grows.

12.LIMITATIONS

- **Initial Configuration Effort:**
  Requires predefined role structures and permissions before deployment.
- **Limited Flexibility in Ad-hoc Tasks:**
  Automated workflows may not suit highly dynamic environments.
- **Maintenance Requirement:**
  System roles and user data must be updated periodically.
- **Theoretical Prototype:**
  The current model is conceptual; full implementation requires additional resources and time.

13.FUTURE ENHANCEMENTS

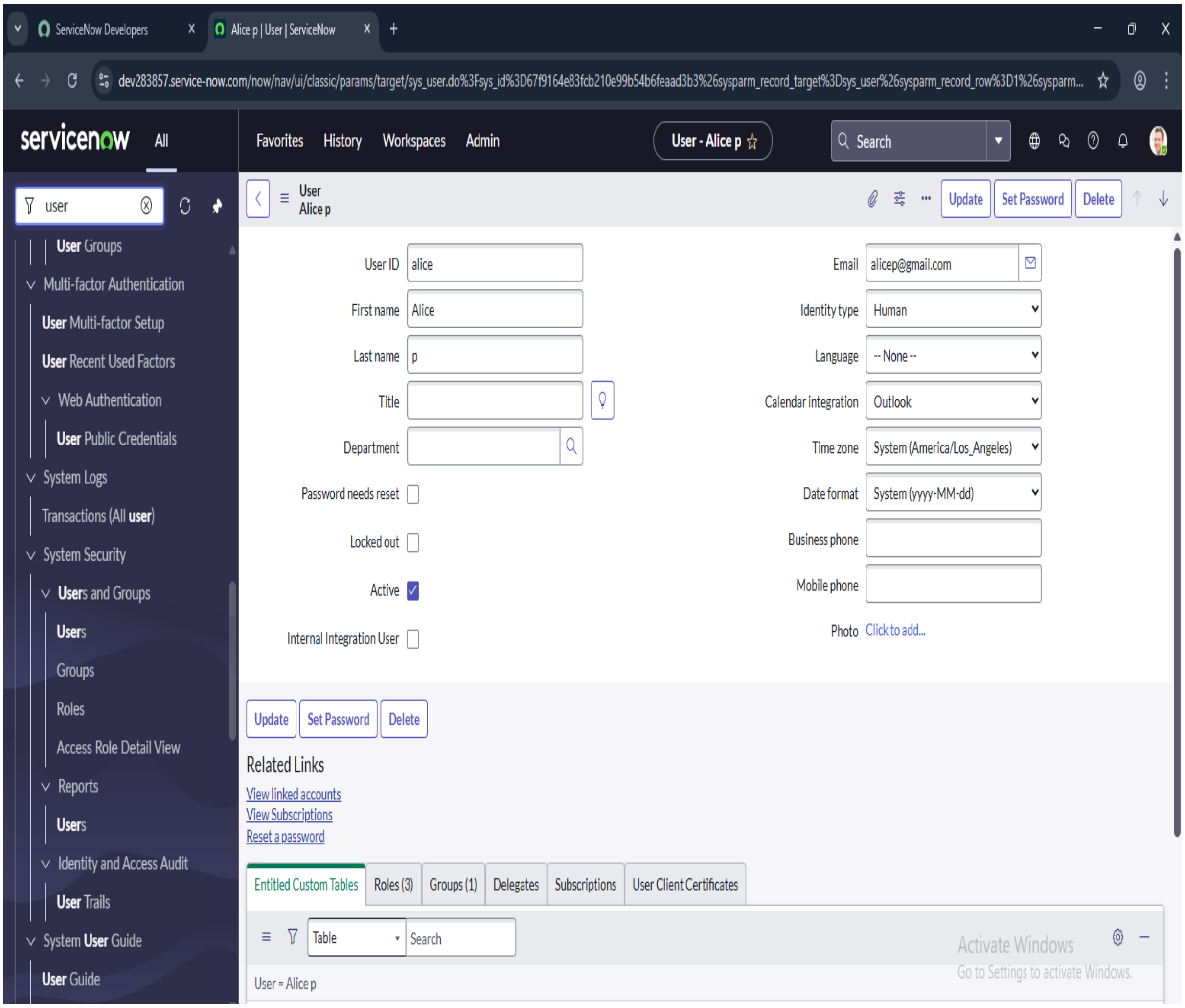
The system can be expanded with advanced functionalities:

- **Artificial Intelligence Integration:**
  AI algorithms can predict task delays and suggest workload distribution.
- **Cloud Integration:**
  Deploying the system on cloud platforms for accessibility and scalability.
- **Mobile Application:**
  Android/iOS versions to manage tasks on the go.
- **Data Analytics Dashboard:**
  Visual analytics for tracking productivity trends.
- **Multi-level Role Hierarchies:**
  Support for complex organizational structures with nested permissions.
- **Real-time Notifications:**
  Push notifications for task updates and status changes.

Create Users

1. Open service now
2. Click on All >> search for users
3. Select Users under system security
4. Click on new
5. Fill the following details to create a new user

6. Click on submit



User
Alice p

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

Update Set Password Delete

Related Links

View linked accounts

View Subscriptions

Reset a password

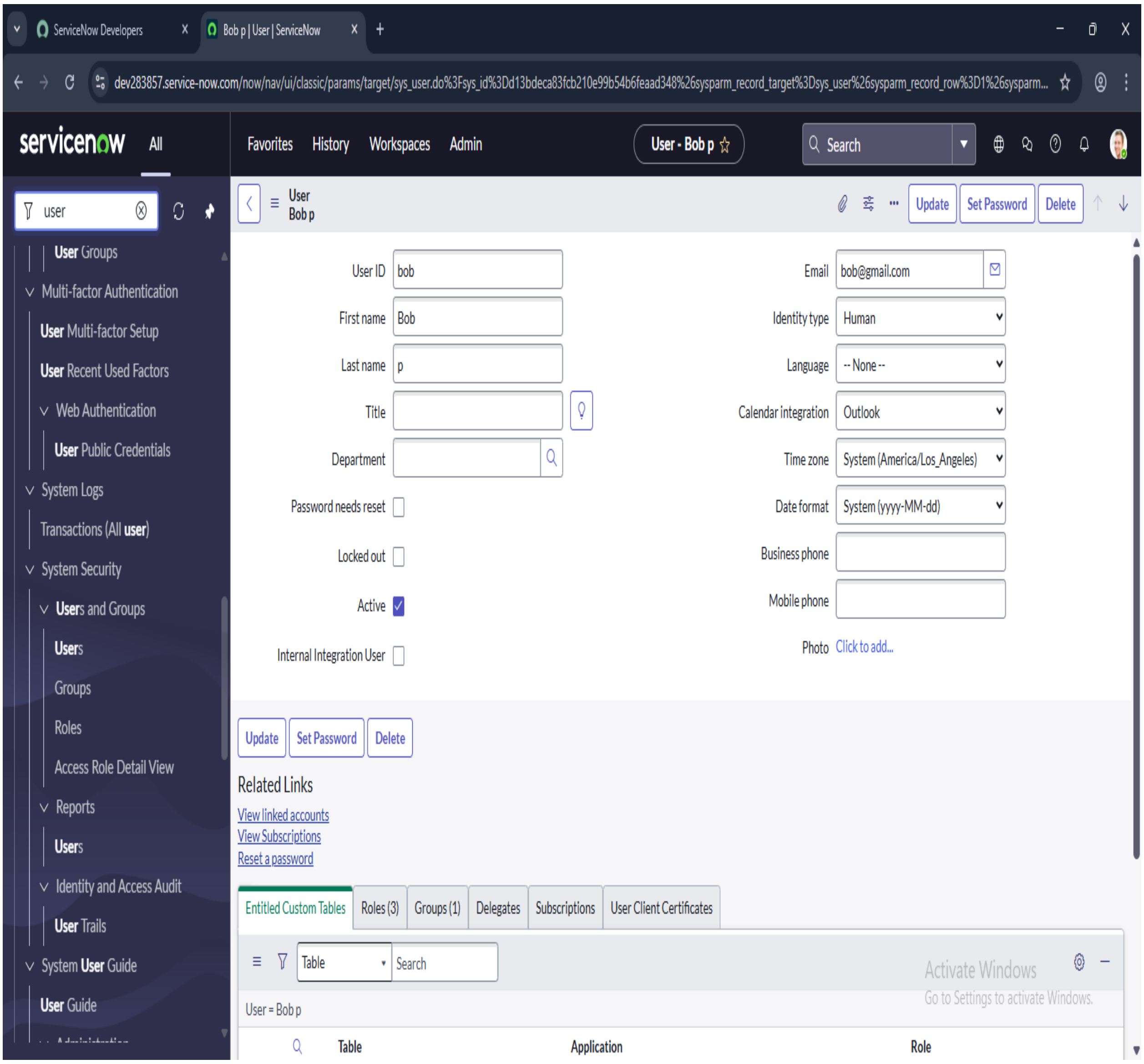
Entitled Custom Tables | Roles (3) | Groups (1) | Delegates | Subscriptions | User Client Certificates

Table Search

User = Alice p

1. Create one more user

2. Create another user with the following details
3. Click on submit

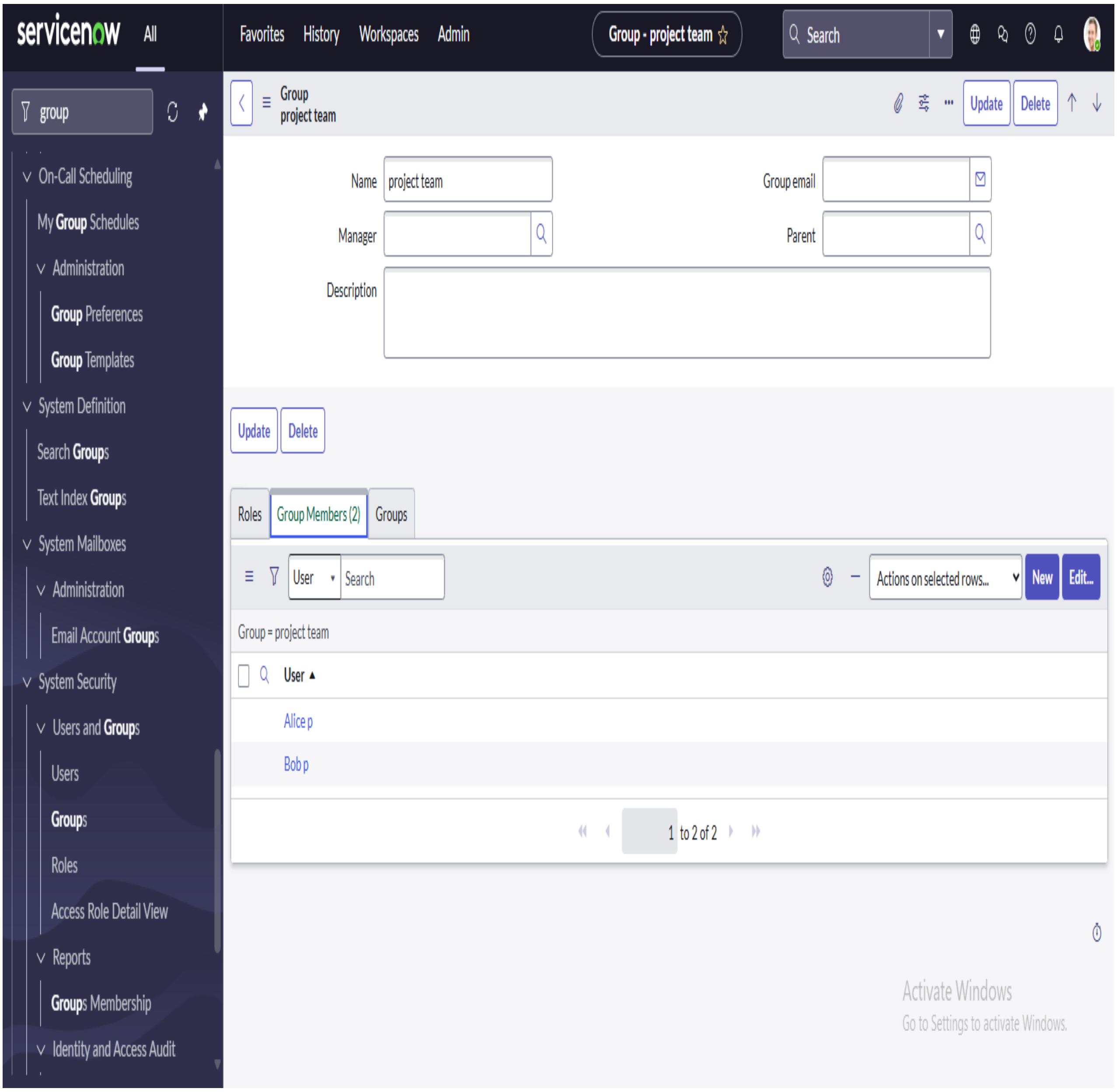


User ID	bob	Email	bob@gmail.com
First name	Bob	Identity type	Human
Last name	p	Language	-- None --
Title		Calendar integration	Outlook
Department		Time zone	System (America/Los_Angeles)
Password needs reset	□	Date format	System (yyyy-MM-dd)
Locked out	□	Business phone	
Active	☑	Mobile phone	
Internal Integration User	□	Photo	Click to add...

User = Bob p

Create Groups

1. Open service now.
2. Click on All >> search for groups
3. Select groups under system security
4. Click on new
5. Fill the following details to create a new group
6. Click on submit



Group
project team

Name: project team

Group email:

Manager:

Parent:

Description:

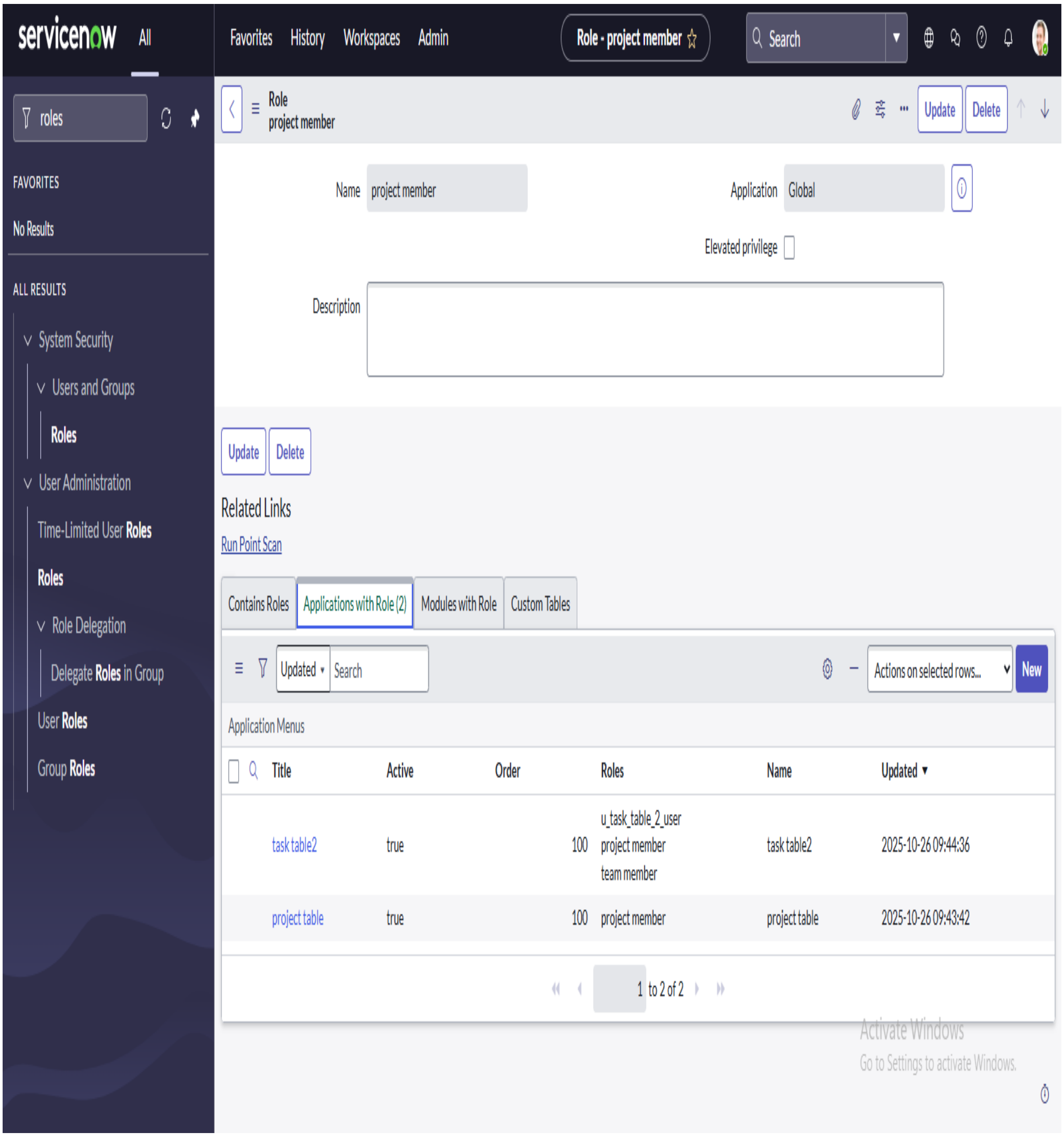
Group = project team

User ▲
Alice p
Bob p

1 to 2 of 2

Create Roles

1. Open service now.
2. Click on All >> search for roles
3. Select roles under system security
4. Click on new
5. Fill the following details to create a new role
6. Click on submit



Name: project member

Application: Global

Elevated privilege □

Description:

Update Delete

Related Links

Run Point Scan

Contains Roles | Applications with Role (2) | Modules with Role | Custom Tables

Updated ▾ Search | Actions on selected rows... | New

Application Menus

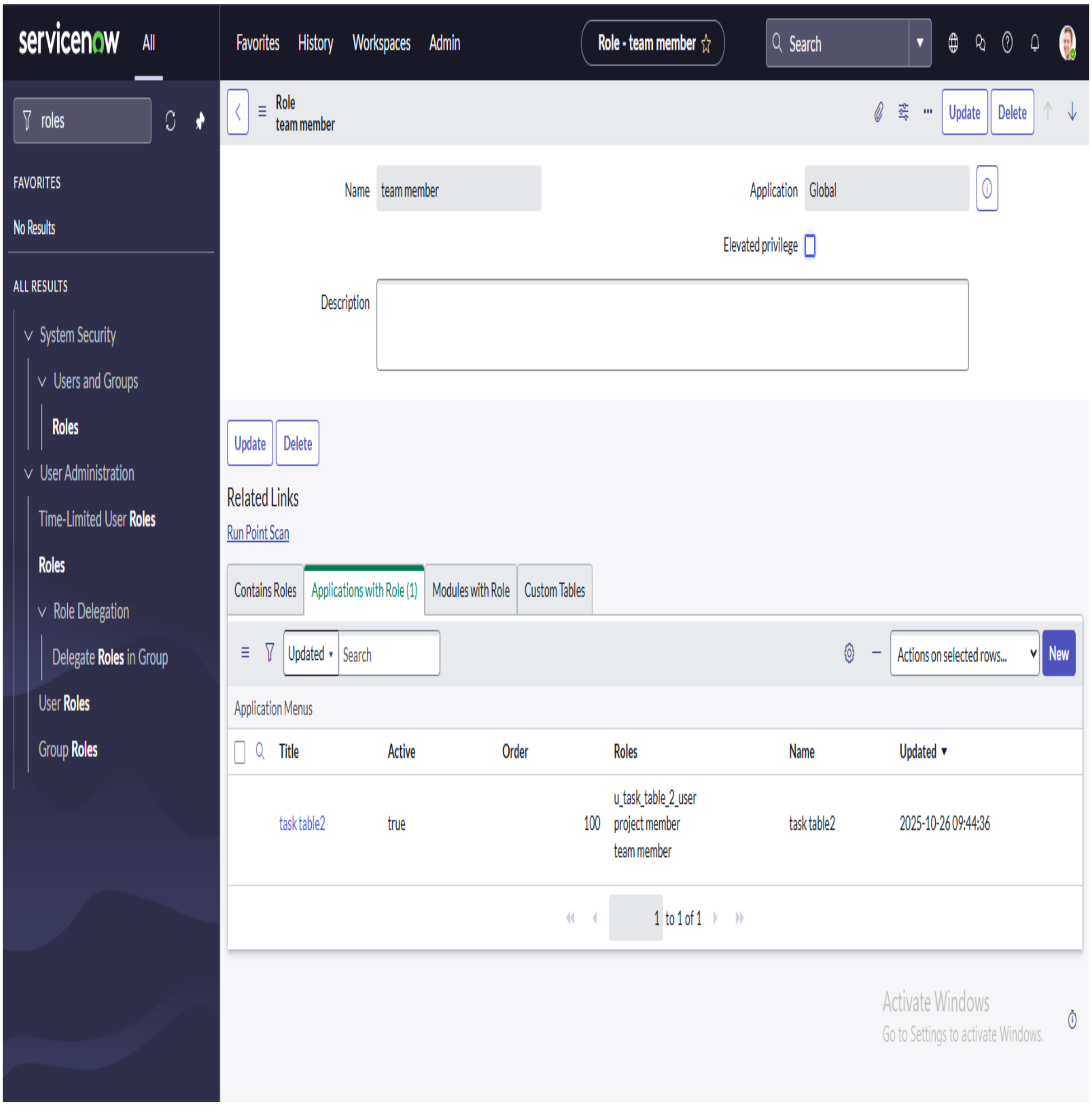
Title	Active	Order	Roles	Name	Updated ▾
task table2	true	100	u_task_table_2_user project member team member	task table2	2025-10-26 09:44:36
project table	true	100	project member	project table	2025-10-26 09:43:42

1 to 2 of 2

Create one more role:

7.Create another role with the following details

8.Click on submit



Name: team member

Application: Global

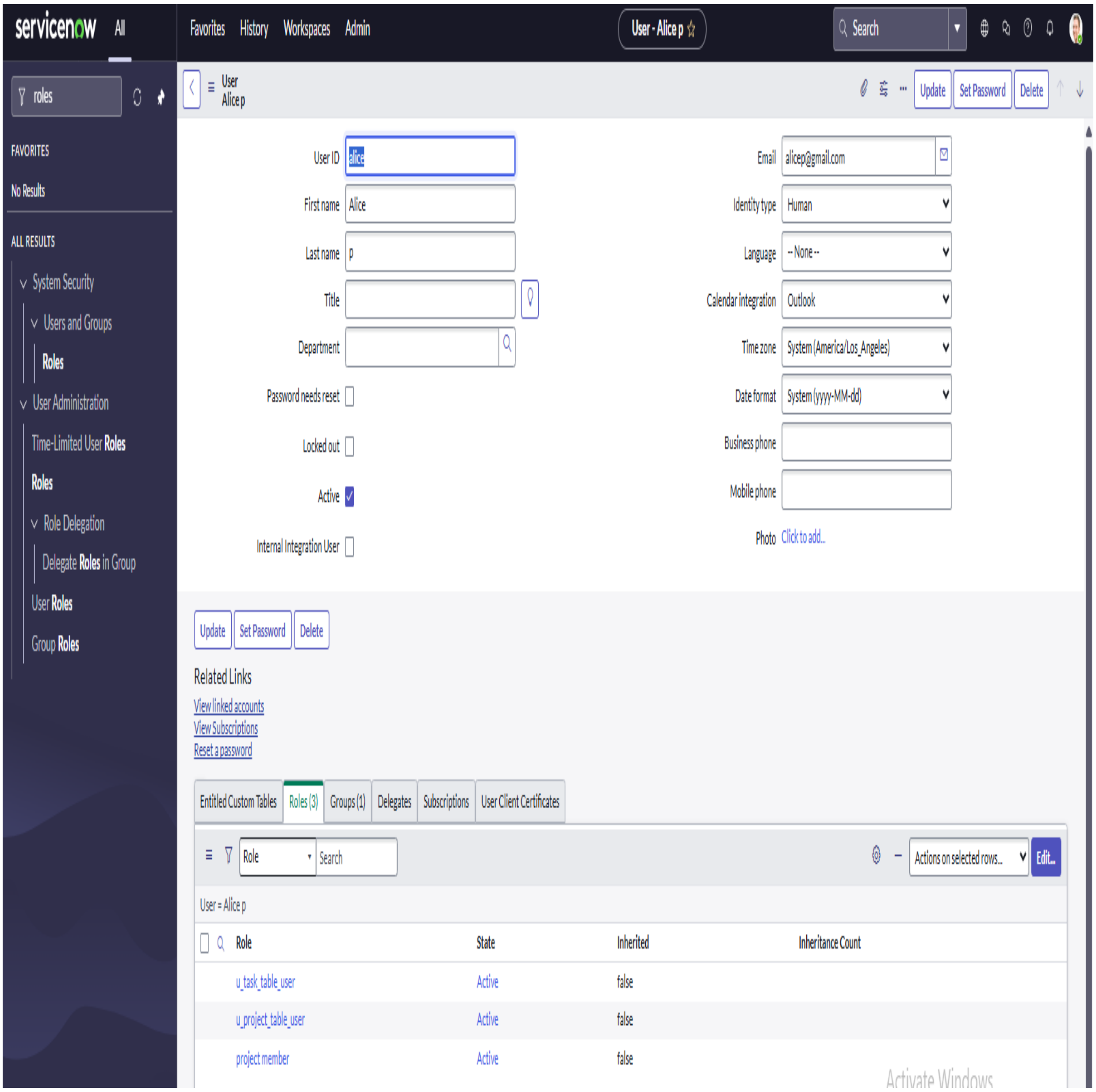
Elevated privilege □

Description:

Title	Active	Order	Roles	Name	Updated ▾
task table2	true	100	u_task_table_2_user project member team member	task table2	2025-10-26 09:44:36

Assign roles to alice user

1. Open servicenow.Click on All >> search for user
2. Select tables under system definition
3. Select the project manager user
4. Under project manager
5. Click on edit
6. Select project member and save
7. click on edit add u_project_table role and u_task_table role
8. click on save and update the form.



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

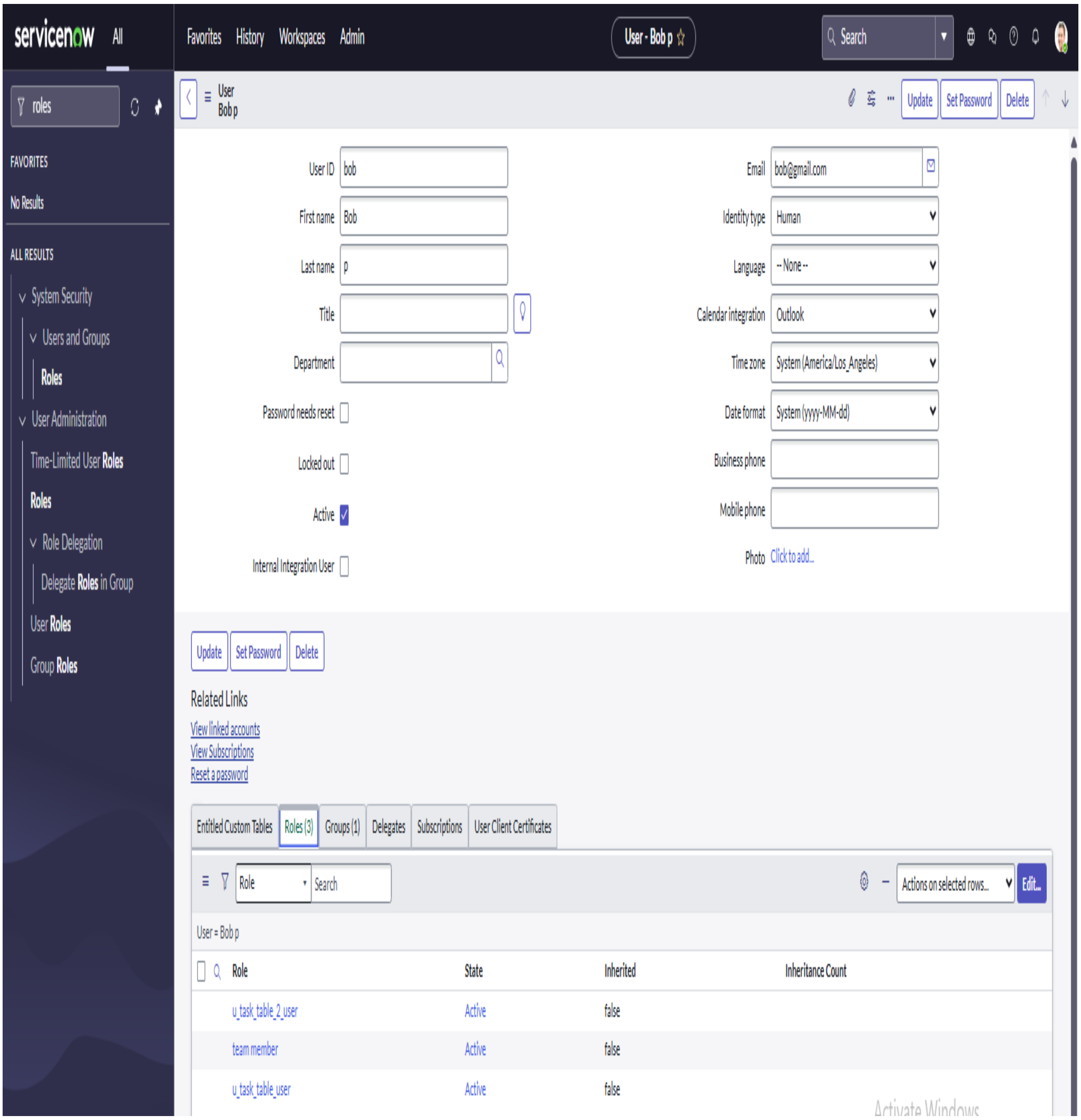
User = Alice p

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

Assign roles to bob user

1. Open servicenow.Click on All >> search for user
2. Select tables under system definition
3. Select the bob p user
4. Under team member
5. Click on edit

6. Select team member and give table role and save
7. Click on profile icon Impersonate user to bob
8. We can see the task table2.



User ID	bob	Email	bob@gmail.com
First name	Bob	Identity type	Human
Last name	p	Language	-- None --
Title		Calendar integration	Outlook
Department		Time zone	System (America/Los_Angeles)
Password needs reset	□	Date format	System (yyyy-MM-dd)
Locked out	□	Business phone	
Active	☑	Mobile phone	
Internal Integration User	□	Photo	Click to add...

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

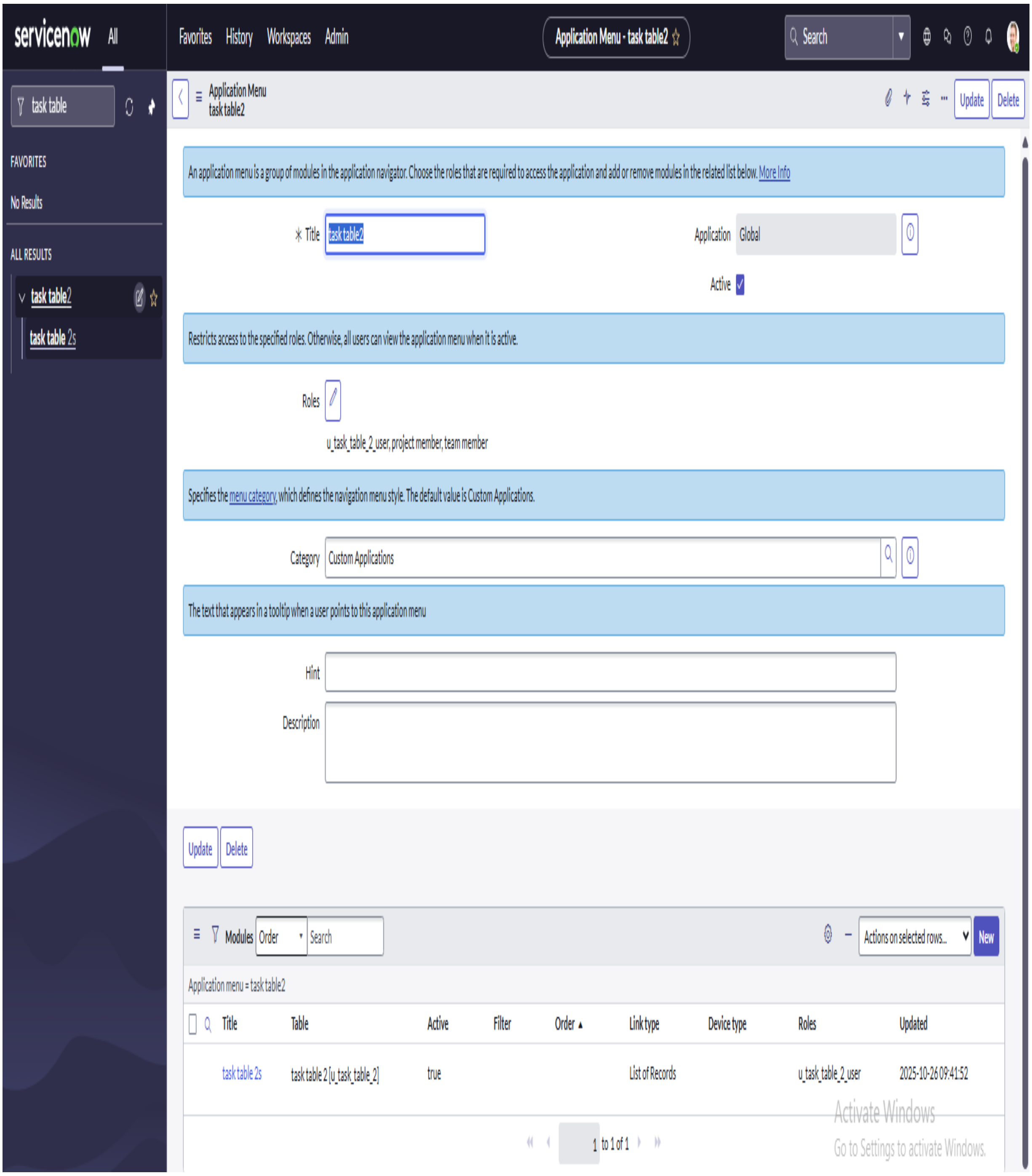
Role | Search | Actions on selected rows... | Edit...

User = Bob p

Role	State	Inherited	Inheritance Count
u_task_table_2_user	Active	false	
team member	Active	false	
u_task_table_user	Active	false	

Assign table access to application

1. while creating a table it automatically create a application and module for that table
2. Go to application navigator search for search project table application
3. Click on edit module
4. Give project member roles to that application
5. Search for task table2 and click on edit application.
6. Give the project member and team member role for task table 2 application



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	task table2	Application	Global
		Active	☑

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

Roles: u_task_table_2_user,project member,team 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

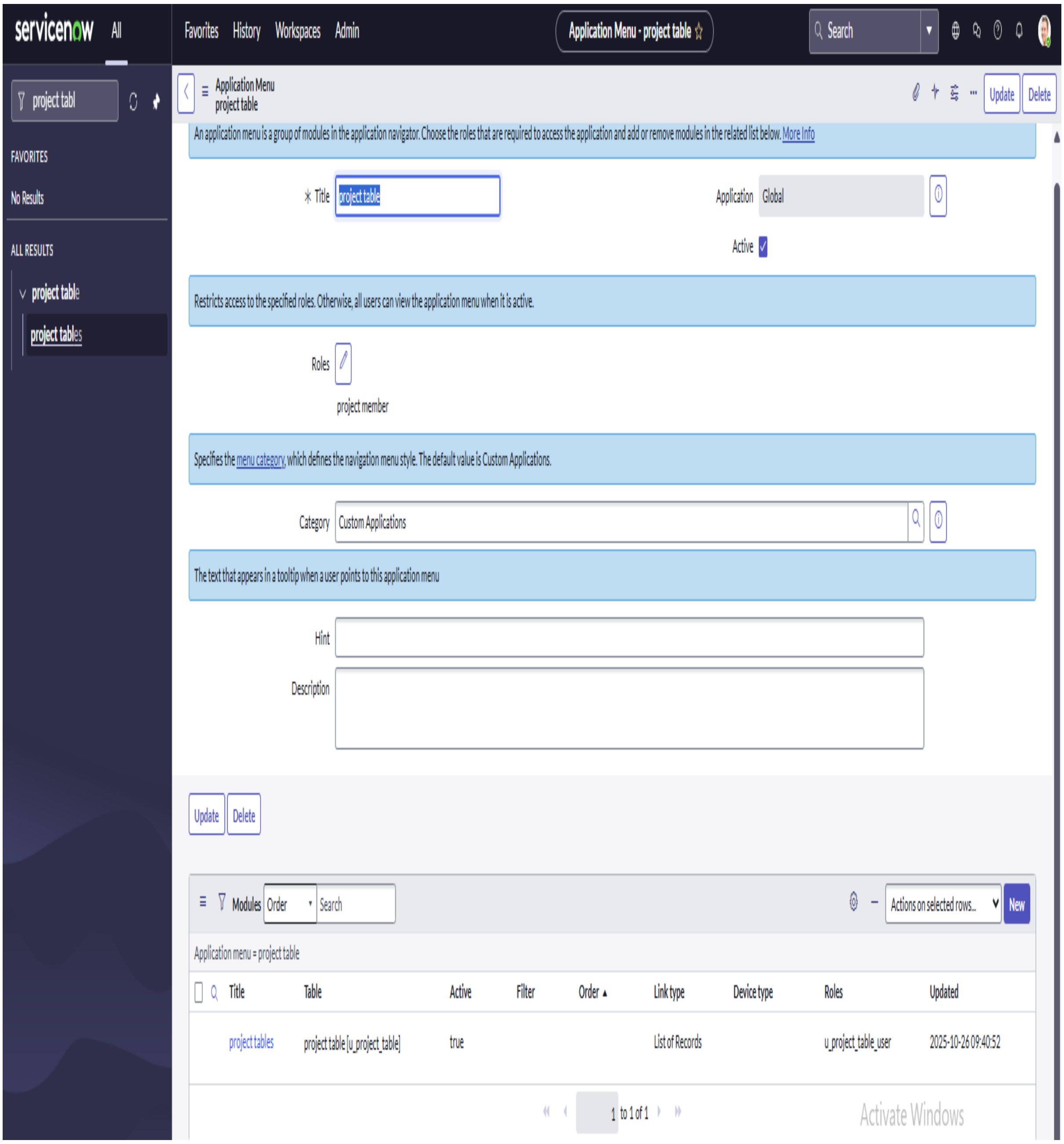
Hint:

Description:

Application menu = task table2

Title	Table	Active	Filter	Order ▲	Link type	Device type	Roles	Updated
task table 2s	task table 2 [u_task_table_2]	true			List of Records		u_task_table_2_user	2025-10-26 09:41:52

1 to 1 of 1



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:

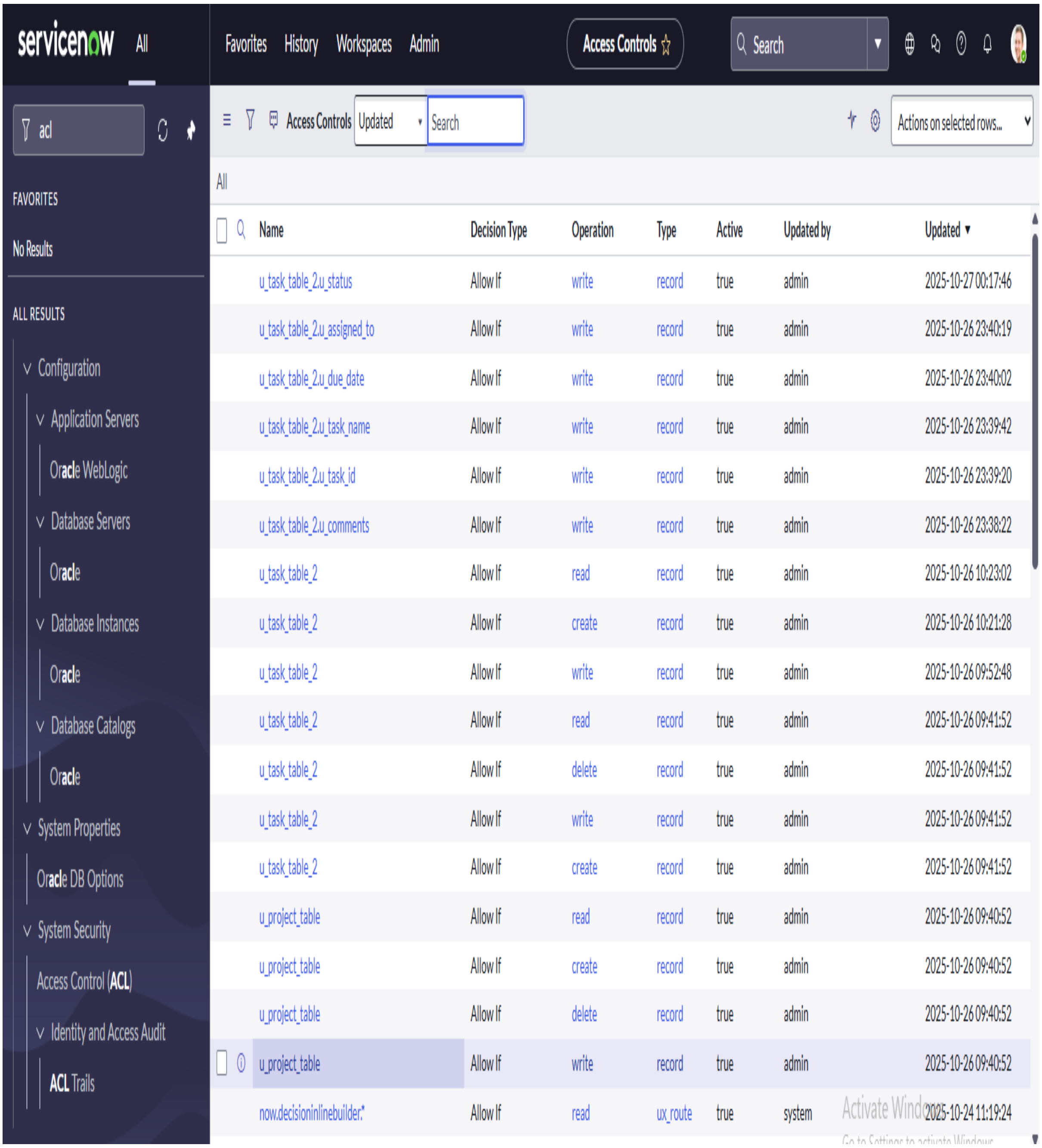
Application menu = project table

Title	Table	Active	Filter	Order ▲	Link type	Device type	Roles	Updated
project tables	project table [u_project_table]	true			List of Records		u_project_table_user	2025-10-26 09:40:52

1 to 1 of 1

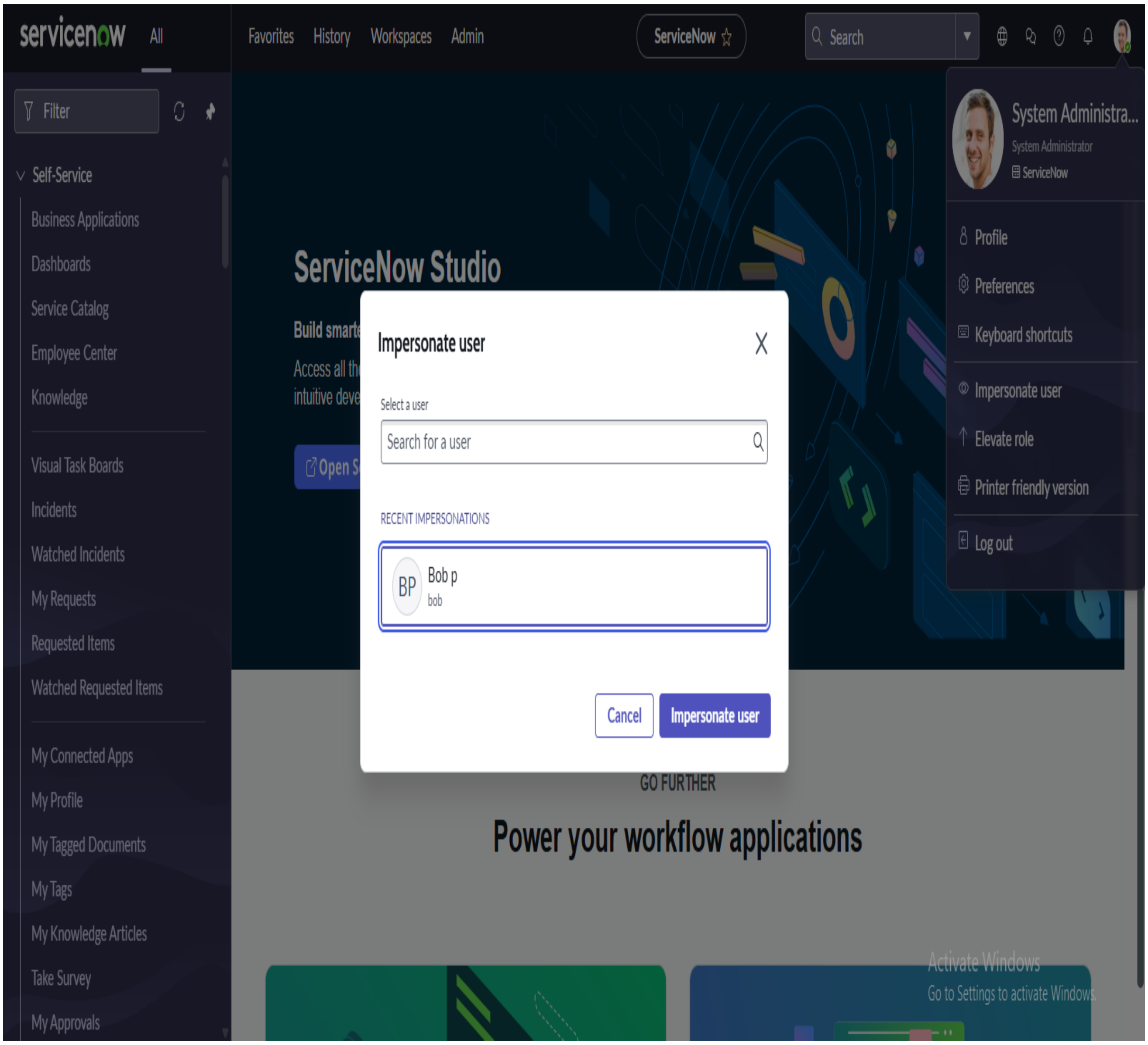
Create ACL

1. Open service now.
2. Click on All >> search for ACL
3. Select Access Control(ACL) under system security
4. Click on elevate role
5. Click on new
6. Fill the following details to create a new ACL
7. Scroll down under requires role
8. Double click on insert a new row
9. Give task table and team member role
10. Click on submit
11. Similarly create 4 acl for the following fields



Name	Decision Type	Operation	Type	Active	Updated by	Updated ▾
u_task_table_2.u_status	Allow If	write	record	true	admin	2025-10-27 00:17:46
u_task_table_2.u_assigned_to	Allow If	write	record	true	admin	2025-10-26 23:40:19
u_task_table_2.u_due_date	Allow If	write	record	true	admin	2025-10-26 23:40:02
u_task_table_2.u_task_name	Allow If	write	record	true	admin	2025-10-26 23:39:42
u_task_table_2.u_task_id	Allow If	write	record	true	admin	2025-10-26 23:39:20
u_task_table_2.u_comments	Allow If	write	record	true	admin	2025-10-26 23:38:22
u_task_table_2	Allow If	read	record	true	admin	2025-10-26 10:23:02
u_task_table_2	Allow If	create	record	true	admin	2025-10-26 10:21:28
u_task_table_2	Allow If	write	record	true	admin	2025-10-26 09:52:48
u_task_table_2	Allow If	read	record	true	admin	2025-10-26 09:41:52
u_task_table_2	Allow If	delete	record	true	admin	2025-10-26 09:41:52
u_task_table_2	Allow If	write	record	true	admin	2025-10-26 09:41:52
u_task_table_2	Allow If	create	record	true	admin	2025-10-26 09:41:52
u_project_table	Allow If	read	record	true	admin	2025-10-26 09:40:52
u_project_table	Allow If	create	record	true	admin	2025-10-26 09:40:52
u_project_table	Allow If	delete	record	true	admin	2025-10-26 09:40:52
u_project_table	Allow If	write	record	true	admin	2025-10-26 09:40:52
now.decisioninlinebuilder*	Allow If	read	ux_route	true	system	2025-10-24 11:19:24

1. Click on profile on top right side
2. Click on impersonate user
3. Select bob user
4. Go to all and select task table2 in the application menu bar
5. Comment and status fields are have the edit access



Impersonate user

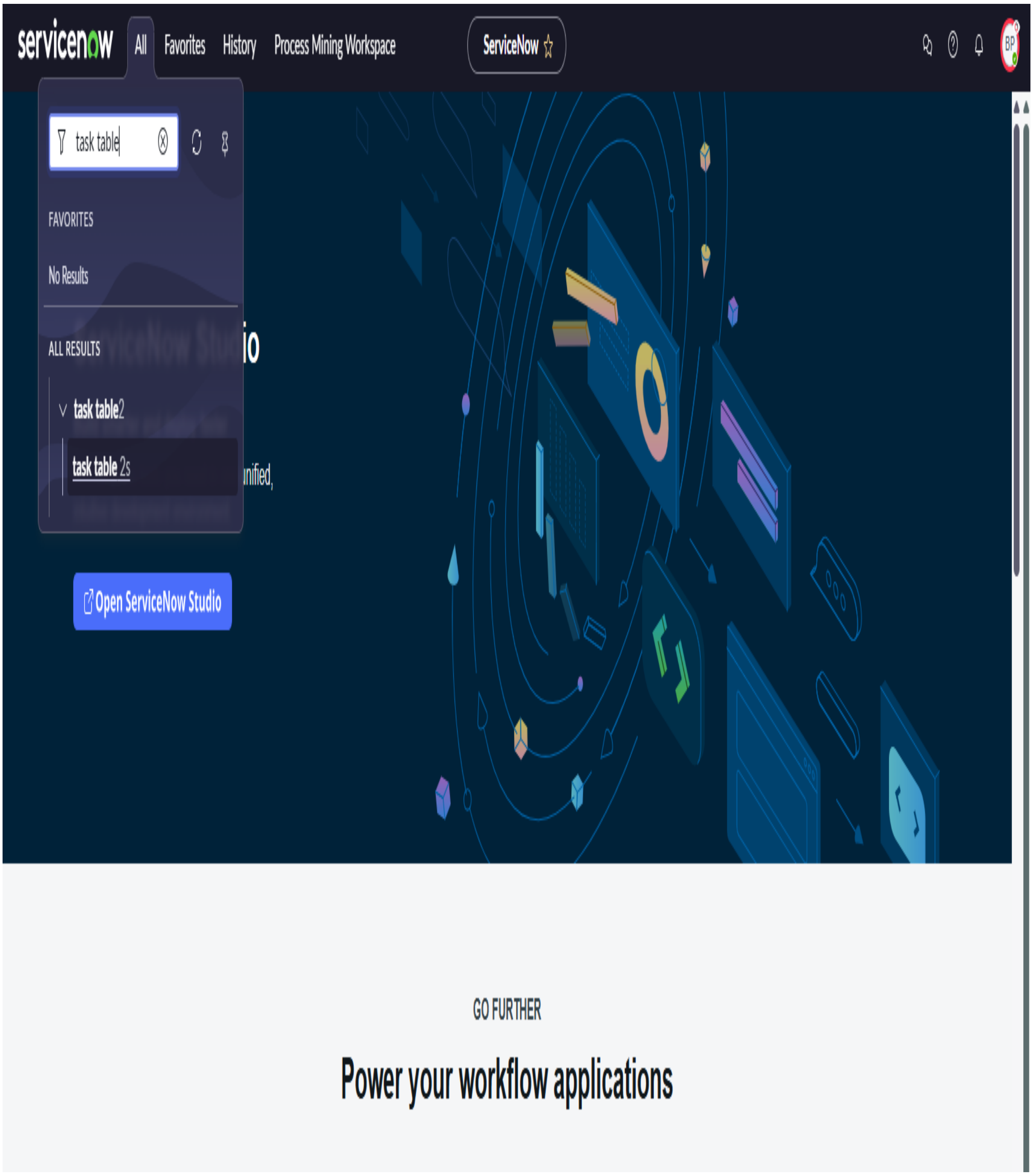
Select a user

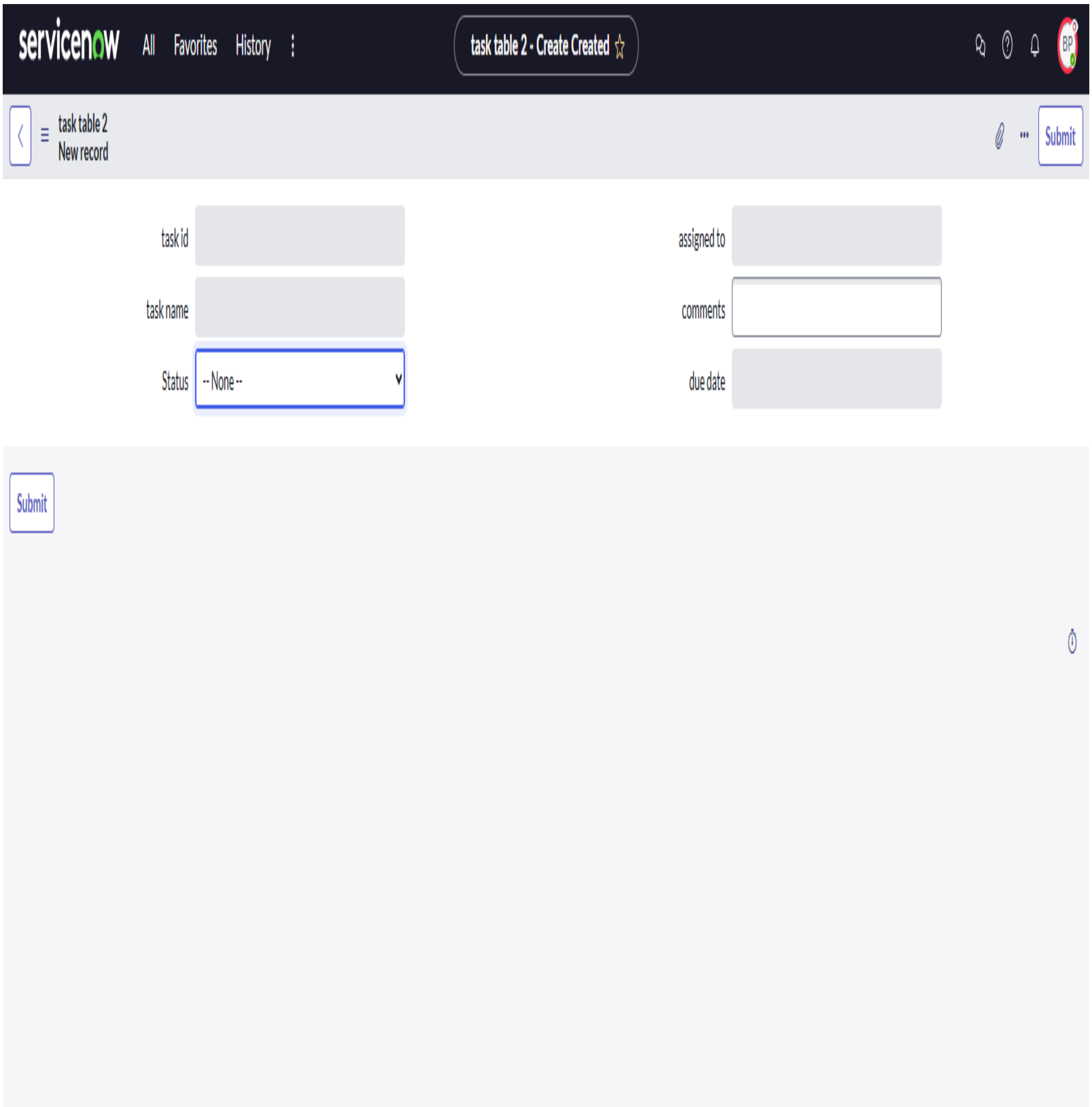
Search for a user

RECENT IMPERSONATIONS

BP Bob p
bob

Cancel
Impersonate user

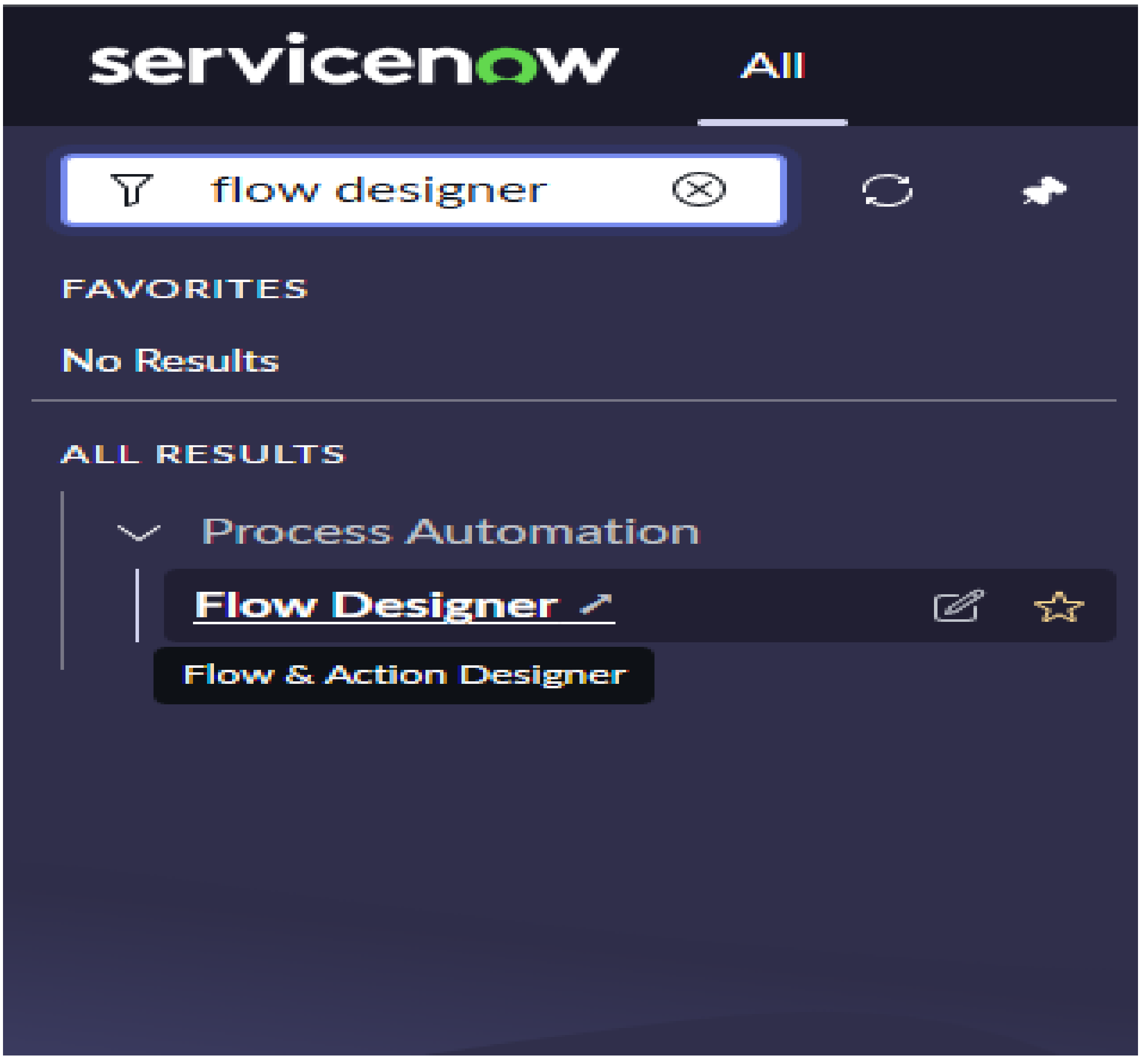

servicenow
All
Favorites
History
Process Mining Workspace
ServiceNow ☆
task table
FAVORITES
No Results
ALL RESULTS
task table2
task table 2s
io
nified,
Open ServiceNow Studio
GO FURTHER
Power your workflow applications



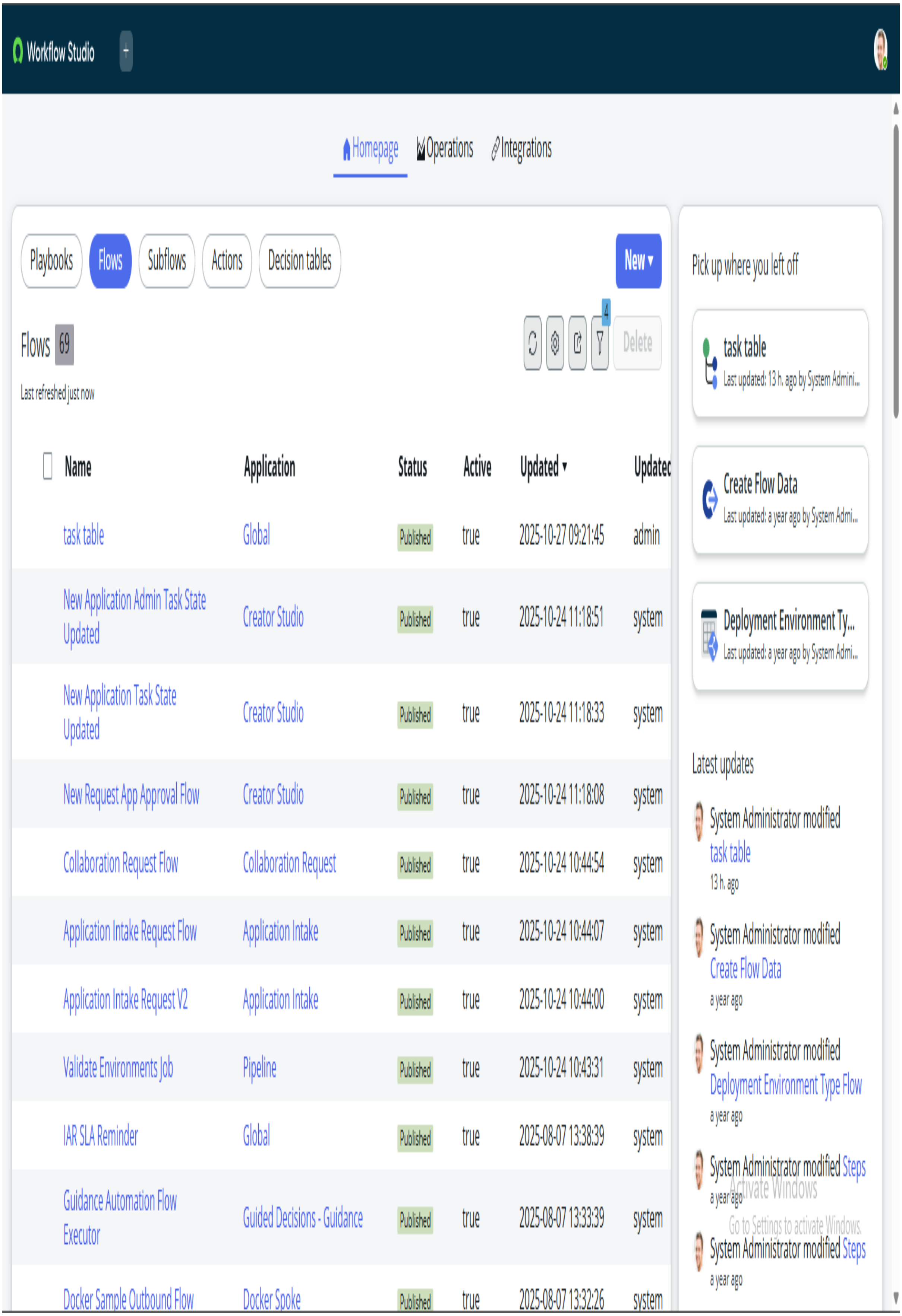
task id		assigned to	
task name		comments	
Status	-- None --	due date	

Create a Flow to Assign operations ticket to group

1. Open service now.
2. Click on All >> search for Flow Designer
3. Click on Flow Designer under Process Automation.
4. After opening Flow Designer Click on new and select Flow.
5. Under Flow properties Give Flow Name as “ task table”.
6. Application should be Global.
7. Click build flow.



servicenow
All
flow designer
FAVORITES
No Results
ALL RESULTS
Process Automation
Flow Designer
Flow & Action Designer



Flows 69

Last refreshed just now

□	Name	Application	Status	Active	Updated ▾	Updatec
	task table	Global	Published	true	2025-10-27 09:21:45	admin
	New Application Admin Task State Updated	Creator Studio	Published	true	2025-10-24 11:18:51	system
	New Application Task State Updated	Creator Studio	Published	true	2025-10-24 11:18:33	system
	New Request App Approval Flow	Creator Studio	Published	true	2025-10-24 11:18:08	system
	Collaboration Request Flow	Collaboration Request	Published	true	2025-10-24 10:44:54	system
	Application Intake Request Flow	Application Intake	Published	true	2025-10-24 10:44:07	system
	Application Intake Request V2	Application Intake	Published	true	2025-10-24 10:44:00	system
	Validate Environments Job	Pipeline	Published	true	2025-10-24 10:43:31	system
	IAR SLA Reminder	Global	Published	true	2025-08-07 13:38:39	system
	Guidance Automation Flow Executor	Guided Decisions - Guidance	Published	true	2025-08-07 13:33:39	system
	Docker Sample Outbound Flow	Docker Spoke	Published	true	2025-08-07 13:32:26	system

Pick up where you left off

task table
Last updated: 13 h. ago by System Admini...

Create Flow Data
Last updated: a year ago by System Admi...

Deployment Environment Ty...
Last updated: a year ago by System Admi...

Latest updates

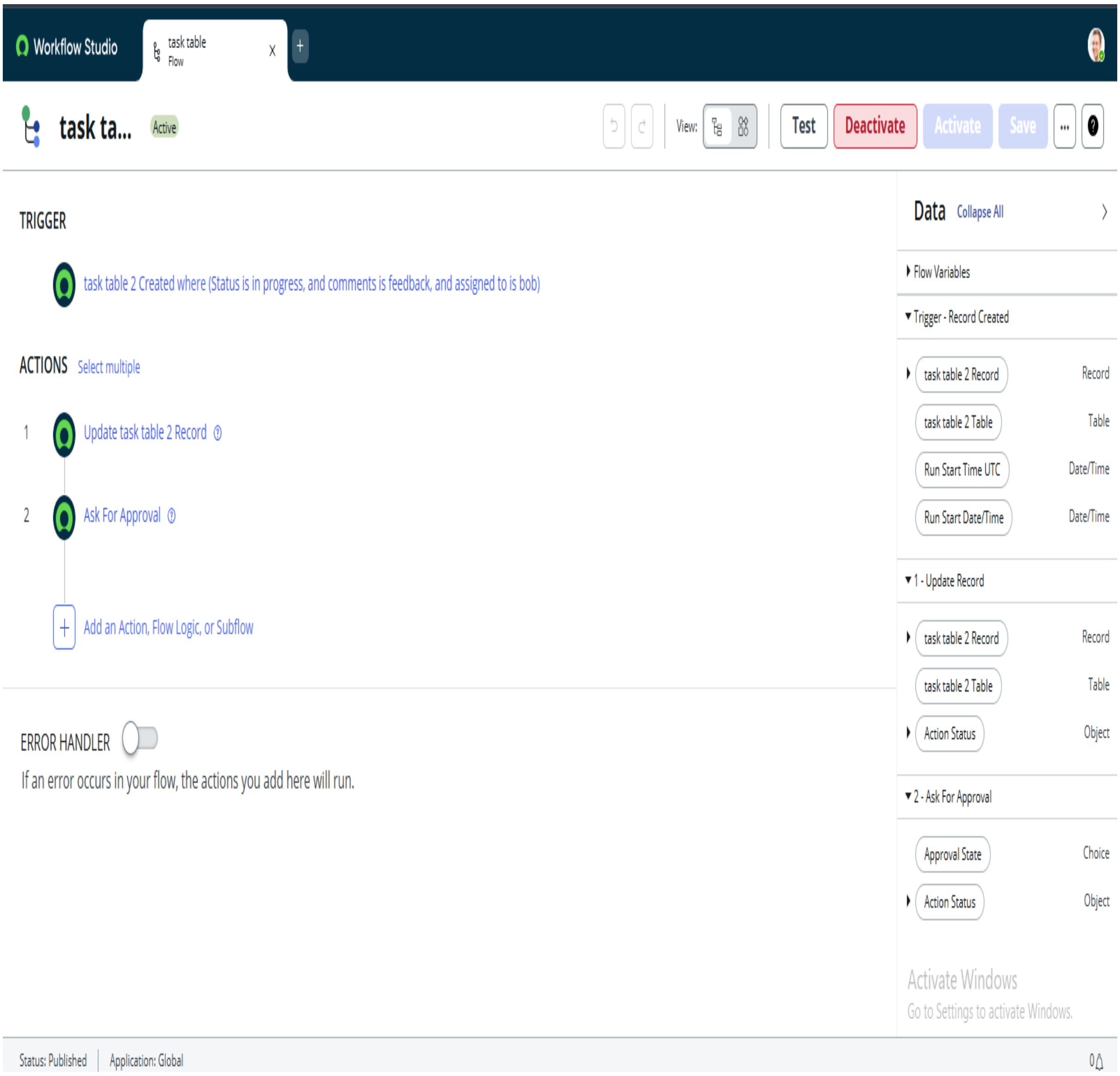
System Administrator modified task table
13 h. ago

System Administrator modified Create Flow Data
a year ago

System Administrator modified Deployment Environment Type Flow
a year ago

System Administrator modified Steps
a year ago

System Administrator modified Steps
a year ago



TRIGGER

task table 2 Created where (Status is in progress, and comments is feedback, and assigned to is bob)

ACTIONS Select multiple

1 Update task table 2 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 2 Record	Record
task table 2 Table	Table
Run Start Time UTC	Date/Time
Run Start Date/Time	Date/Time

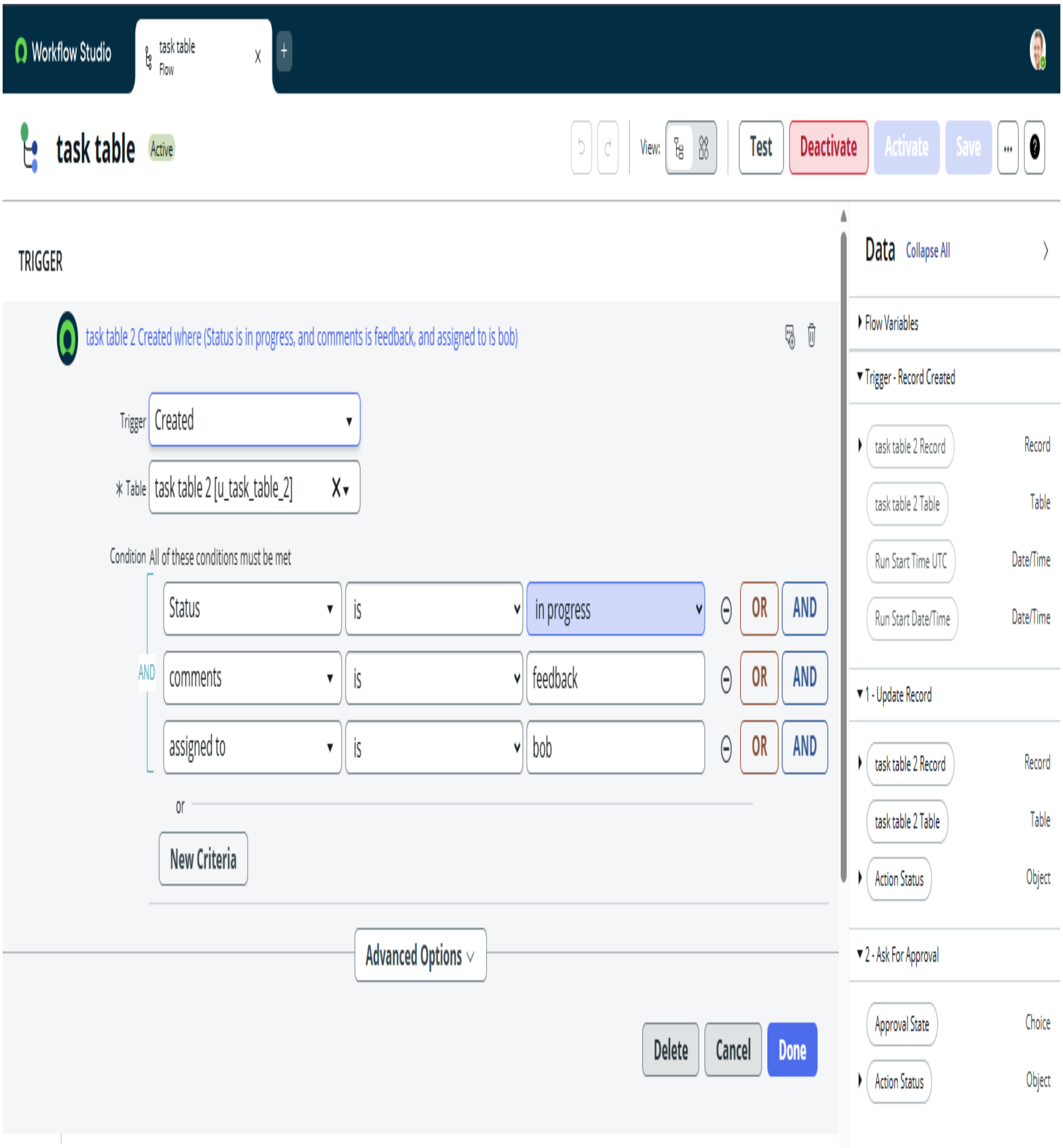
▾ 1 - Update Record

▸ task table 2 Record	Record
task table 2 Table	Table
▸ Action Status	Object

▾ 2 - Ask For Approval

Approval State	Choice
▸ Action Status	Object

Status: Published | Application: Global



task table 2 Created where (Status is in progress, and comments is feedback, and assigned to is bob)

Trigger: Created

* Table: task table 2 [u_task_table_2]

Condition All of these conditions must be met

Status	is	in progress
comments	is	feedback
assigned to	is	bob

- Flow Variables
- Trigger - Record Created

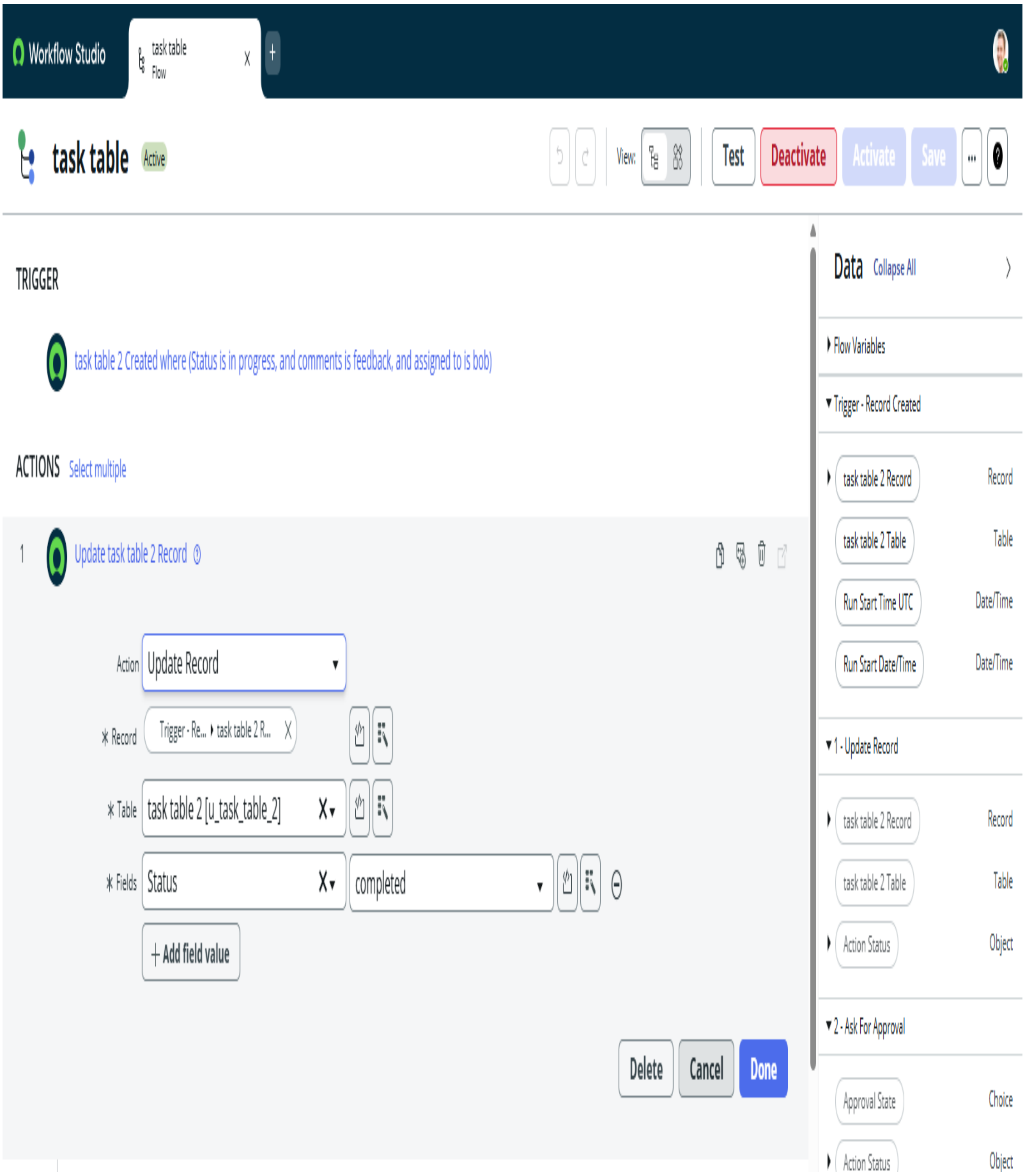
task table 2 Record	Record
task table 2 Table	Table
Run Start Time UTC	Date/Time
Run Start Date/Time	Date/Time

- 1 - Update Record

task table 2 Record	Record
task table 2 Table	Table
Action Status	Object

- 2 - Ask For Approval

Approval State	Choice
Action Status	Object



task table 2 Created where (Status is in progress, and comments is feedback, and assigned to is bob)

1 Update task table 2 Record

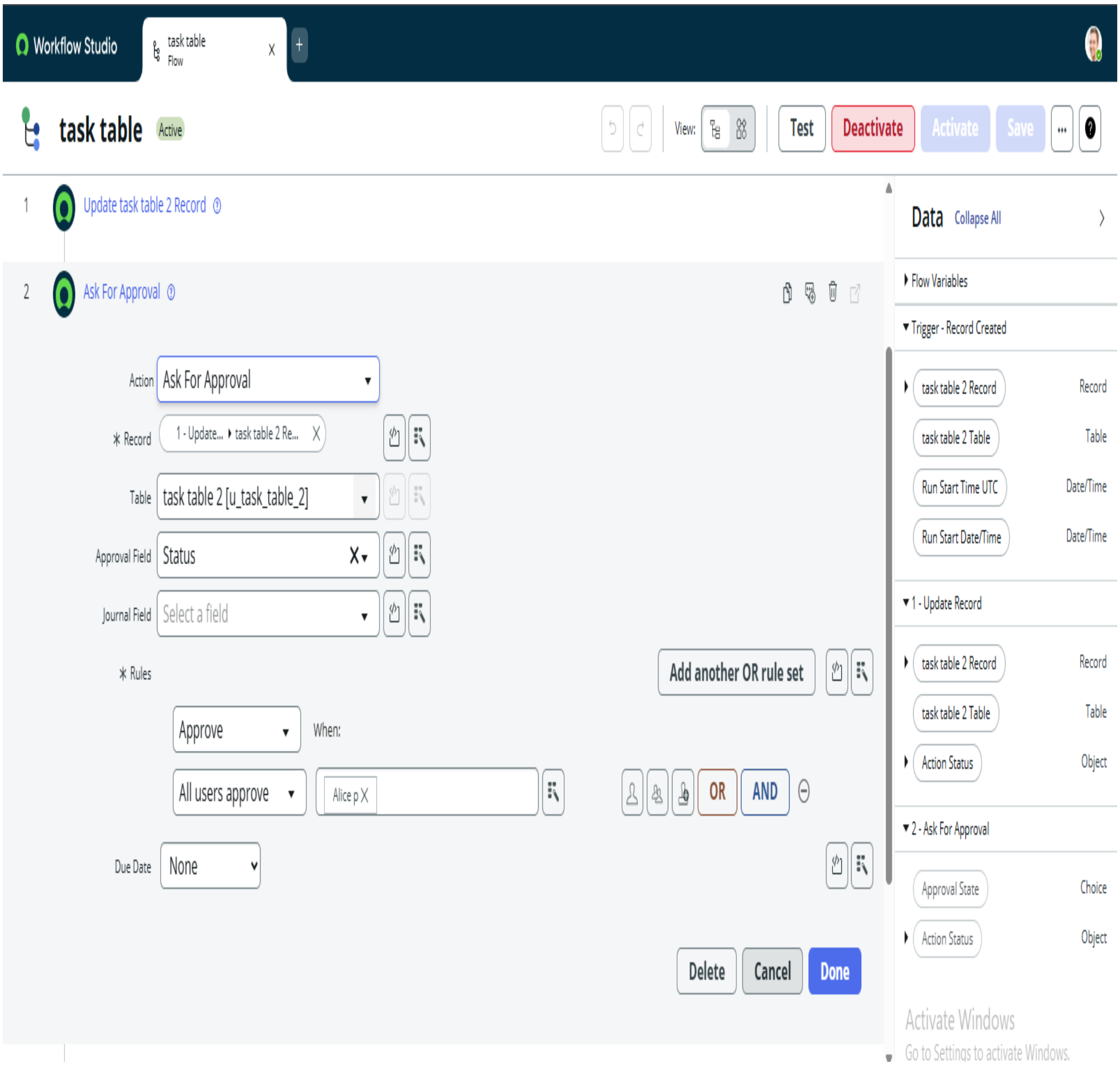
Action: Update Record

* Record: Trigger - Re... ▸ task table 2 R...

* Table: task table 2 [u_task_table_2]

* Fields: Status — completed

+ Add field value



task table Active

1 Update task table 2 Record

2 Ask For Approval

Action: Ask For Approval

✱ Record: 1 - Update... ▸ task table 2 Re...

Table: task table 2 [u_task_table_2]

Approval Field: Status

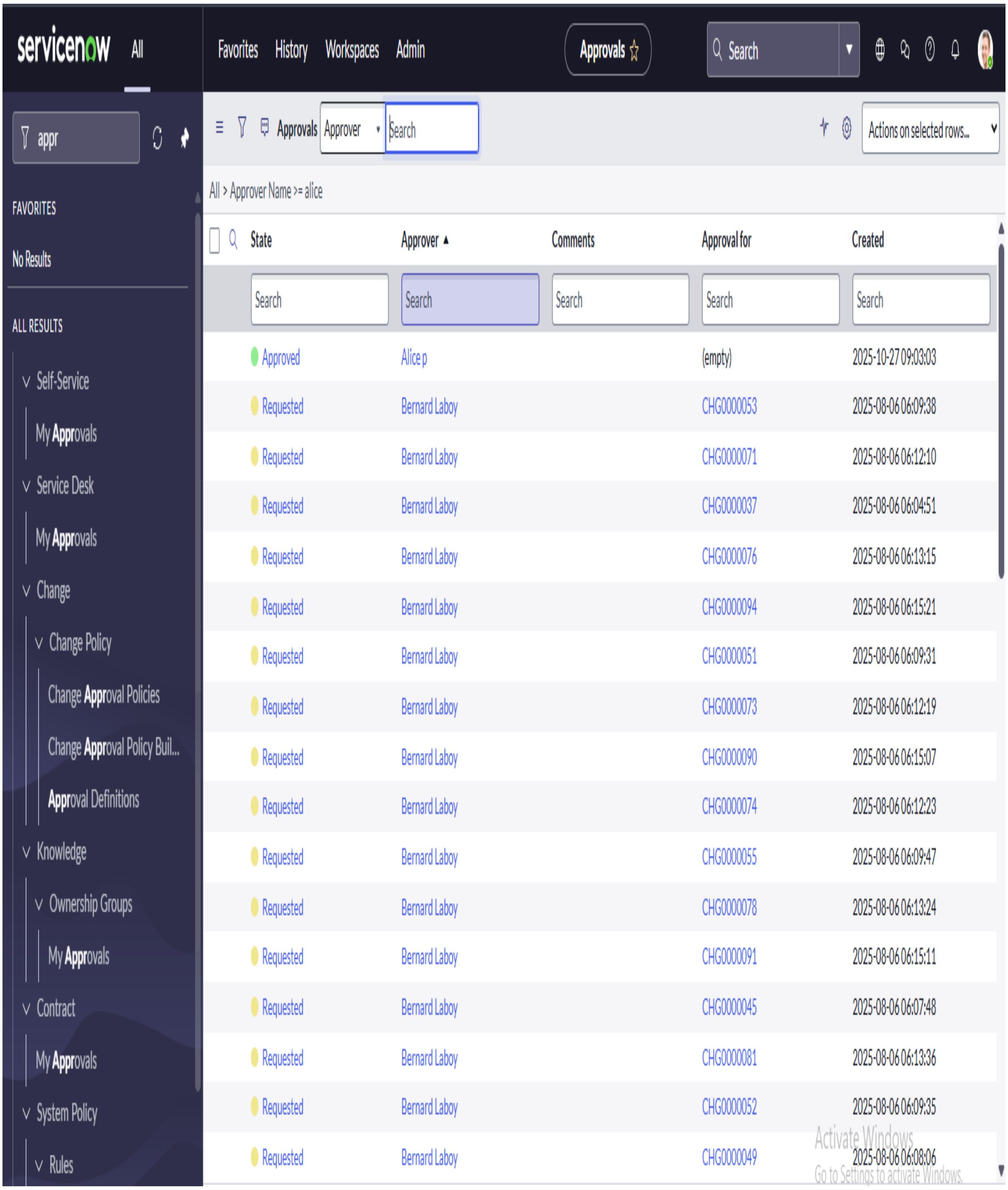
Journal Field: Select a field

✱ Rules

Approve When:

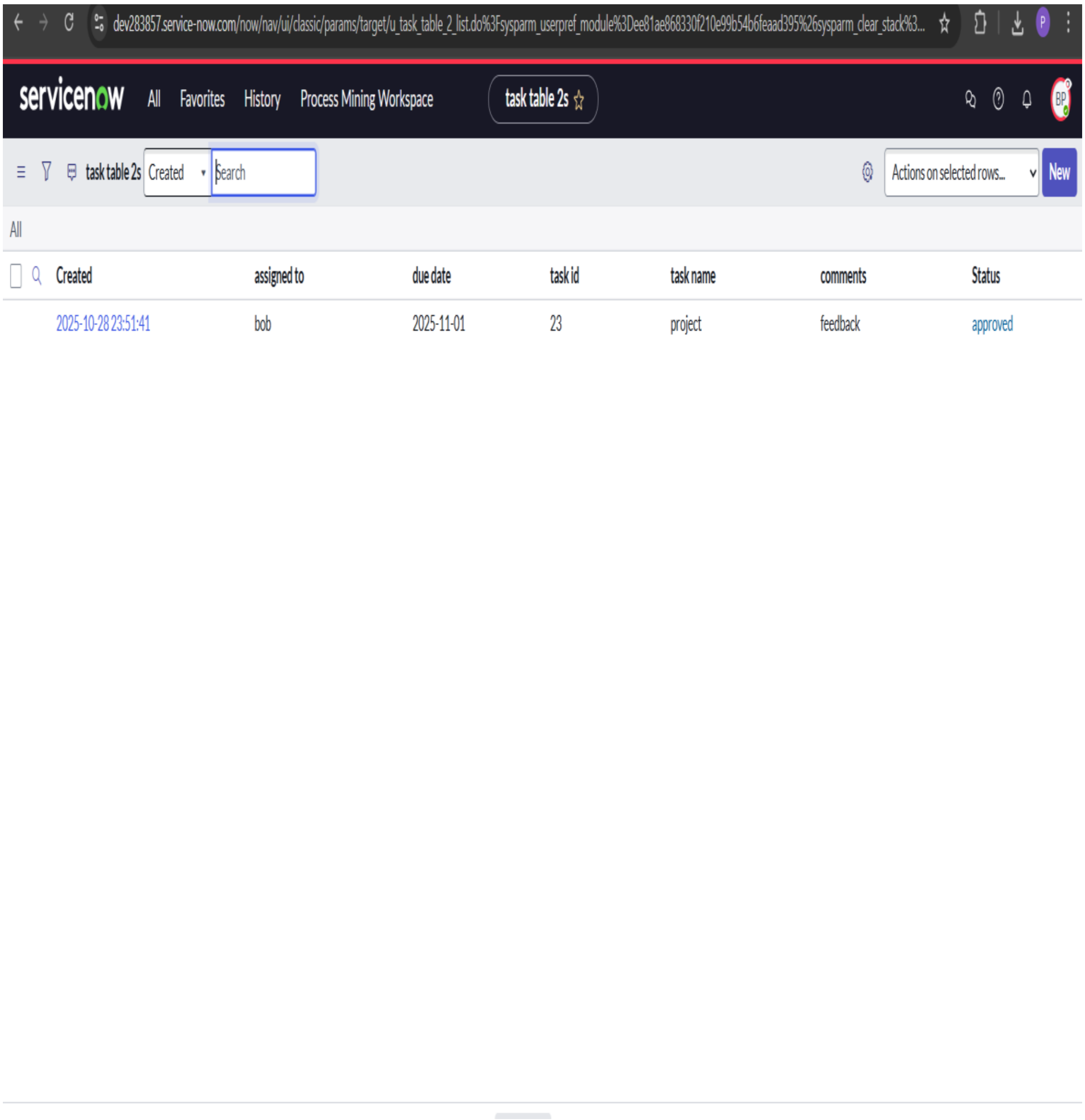
All users approve Alice p

Due Date: None

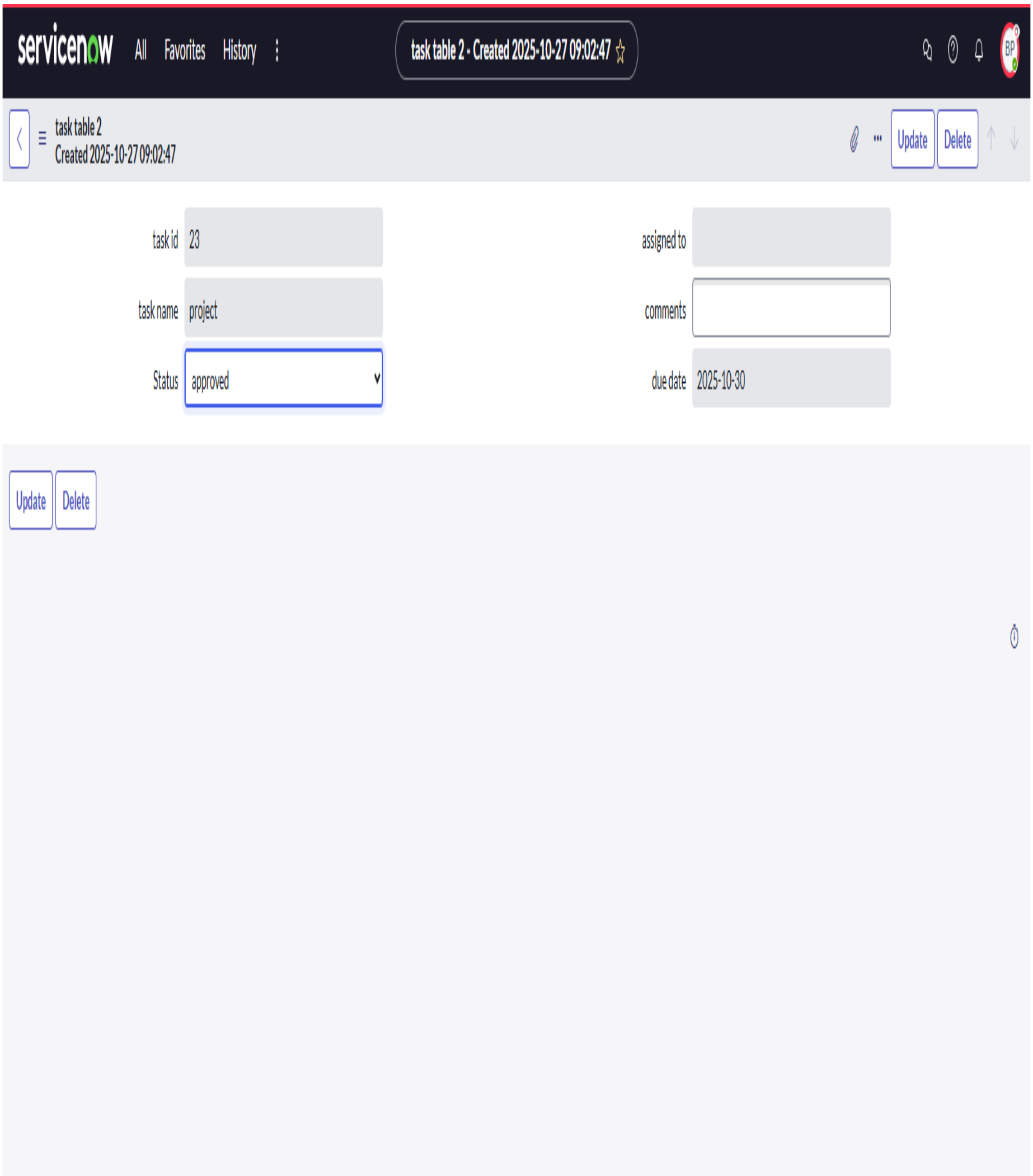


All > Approver Name >= alice

State	Approver ▲	Comments	Approval for	Created
Search	Search	Search	Search	Search
● Approved	Alice p		(empty)	2025-10-27 09:03:03
● Requested	Bernard Laboy		CHG0000053	2025-08-06 06:09:38
● Requested	Bernard Laboy		CHG0000071	2025-08-06 06:12:10
● Requested	Bernard Laboy		CHG0000037	2025-08-06 06:04:51
● Requested	Bernard Laboy		CHG0000076	2025-08-06 06:13:15
● Requested	Bernard Laboy		CHG0000094	2025-08-06 06:15:21
● Requested	Bernard Laboy		CHG0000051	2025-08-06 06:09:31
● Requested	Bernard Laboy		CHG0000073	2025-08-06 06:12:19
● Requested	Bernard Laboy		CHG0000090	2025-08-06 06:15:07
● Requested	Bernard Laboy		CHG0000074	2025-08-06 06:12:23
● Requested	Bernard Laboy		CHG0000055	2025-08-06 06:09:47
● Requested	Bernard Laboy		CHG0000078	2025-08-06 06:13:24
● Requested	Bernard Laboy		CHG0000091	2025-08-06 06:15:11
● Requested	Bernard Laboy		CHG0000045	2025-08-06 06:07:48
● Requested	Bernard Laboy		CHG0000081	2025-08-06 06:13:36
● Requested	Bernard Laboy		CHG0000052	2025-08-06 06:09:35
● Requested	Bernard Laboy		CHG0000049	2025-08-06 06:08:06


dev283857.service-now.com/now/nav/ui/classic/params/target/u_task_table_2_list.do%3Fsysparm_userpref_module%3Dee81ae868330f210e99b54b6feaad395%26sysparm_clear_stack%3...
servicenow
All
Favorites
History
Process Mining Workspace
task table 2s ☆
BP
task table 2s
Created
Search
Actions on selected rows...
New
All

□	Created	assigned to	due date	task id	task name	comments	Status
	2025-10-28 23:51:41	bob	2025-11-01	23	project	feedback	approved



task table 2 - Created 2025-10-27 09:02:47

task table 2
Created 2025-10-27 09:02:47

task id	23	assigned to	
task name	project	comments	
Status	approved	due date	2025-10-30

14.CONCLUSION

This project demonstrates the necessity of structured user, group, and role management integrated with access control and automated workflows. Through the implementation of Role-Based Access Control (RBAC)**,** the system ensures that users perform only the operations permitted by their role, maintaining security and accountability.Automated workflows streamline task management, reducing manual dependency and enhancing team coordination. The modular design ensures scalability and adaptability, making the system suitable for various organizational sizes.In conclusion, “Optimizing User, Group, and Role Management with Access Control and Workflows” provides a theoretical foundation for developing secure, efficient, and transparent project management systems. It addresses core challenges such as unauthorized access, unclear responsibilities, and workflow inefficiencies — paving the way for intelligent, access-controlled collaboration environments in the future.