Phase 3: Project Design Phase

ServiceNow Project: Prevent User Deletion if Assigned to an Incident

Team ID	LTVIP2026TMIDS84378
Team Leader	Valluri Reethi
Team Member	Pothula Lalitha Sri Pani
Team Member	Ramena Mohana Satya Vamsi Kowsik
Team Member	Rudraraju Jogi Subrahmanyam Raju
Date	14 Feb 2026
Document Version	1.0

1. Design Overview

This phase defines the high-level and low-level design for the ServiceNow solution that prevents deletion of a user if the user is assigned to one or more incidents. The design focuses on using a server-side Business Rule on the **User [sys_user]** table and validating dependencies in the **Incident [incident]** table.

2. Architecture Design

The solution is implemented completely within ServiceNow using a Business Rule. No external systems or integrations are required.

Component	ServiceNow Area	Purpose
User Record	sys_user table	Stores users that may be deleted
Incident Records	incident table	Stores incidents assigned to users
Business Rule	System Definition → Business Rules	Prevents deletion if incidents exist
GlideRecord API	Server-side scripting	Queries incident table for assigned_to references
UI Error Message	gs.addErrorMessage()	Displays reason for deletion block

3. Data Design

The design uses existing ServiceNow tables and relationships. The key relationship is between **incident.assigned_to** and **sys_user.sys_id**.

Table	Key Fields Used	Description
sys_user	sys_id, user_name, email	User master table
incident	number, assigned_to, state, active	Incident records assigned to users

Relationship Mapping

incident.assigned_to → references → **sys_user.sys_id**

4. Functional Design

The Business Rule executes when an administrator attempts to delete a user record. The logic checks if at least one incident exists for that user.

1. Trigger: Before Delete on sys_user
2. Input: current.sys_id (user being deleted)
3. Query incident table where assigned_to = current.sys_id
4. If record exists: show error message and abort deletion
5. Else: allow deletion

Pseudo Flow Diagram (Text Representation)

• Start → Admin clicks Delete on User

• Business Rule executes (Before Delete)

• Check incident table for assigned_to = user

• If incident found → Block delete + Show error

• If no incident found → Allow delete

• End

5. Business Rule Design

Attribute	Value
Name	Prevent User Deletion
Table	User [sys_user]
Active	true
When	Before
Insert	false
Update	false
Delete	true
Advanced	true
Condition	None (handled in script)

6. Security Design

In ServiceNow, deleting user records is generally restricted to administrators. However, this project ensures that even administrators cannot delete users if those users are assigned to incidents.

• Deletion restricted by admin permissions (default ServiceNow).

• Business Rule enforces server-side validation (cannot be bypassed by UI).

• Prevents data loss and maintains audit trails.

7. Performance Design

To ensure good performance, the incident query is limited to a single record using **setLimit(1)**. This avoids scanning all incidents unnecessarily.

• Query uses indexed reference field: incident.assigned_to.

• setLimit(1) ensures minimal database load.

• Business Rule runs only on delete events (low frequency).

8. Design Summary

Design Aspect	Summary
Solution Type	Server-side Business Rule
Primary Table	sys_user
Validation Table	incident
Trigger	Before Delete
Validation Condition	Incident exists where assigned_to = current.sys_id
Outcome	Abort delete + show error message

This design ensures the project meets functional requirements, maintains ITSM data integrity, and is scalable for future enhancements.