Streamlining Ticket Assignment for Efficient Support Operations

Project Title : **Streamlining Ticket Assignment for Efficient Support Operations**

Team ID : LTVIP2025TMID60704

Team Size : 4

Team Leader : Challa Navya

Team member : Cheemakurthi Koti Prudhviswar

Team member : Chekka Venkata Nikhilesh

Team member : Chinthalapudi Krishna Samyath

Project Report Formate

Introduction :

In today's fast-paced digital landscape, customer support plays a critical role in maintaining brand loyalty and satisfaction. One of the core challenges support teams face is managing and assigning incoming support tickets in a timely, accurate, and efficient manner. Without a well-structured ticket assignment process, support operations can quickly become overwhelmed—leading to delayed responses, misrouted queries, and decreased customer satisfaction.

Streamlining the ticket assignment process involves leveraging automation, clear workflows, and intelligent routing strategies to ensure that each ticket reaches the right agent or team with minimal delay. This not only enhances operational efficiency but also improves resolution times, agent productivity, and overall service quality.

This document explores best practices, tools, and strategies for optimizing ticket assignment within support environments, highlighting how organizations can create a seamless and scalable support system.

Ideation Phase

The ideation phase is a critical step in reimagining how support tickets are assigned to agents or departments. It involves identifying current inefficiencies, gathering input from stakeholders, and brainstorming innovative solutions that align with business goals and customer expectations.

1. Identify Pain Points in Current Process

Begin by mapping the existing ticket assignment workflow. Engage with frontline support agents, team leads, and system administrators to uncover issues such as:

- Manual routing delays
- Inconsistent ticket categorization
- Skill mismatches between tickets and agents
- Lack of prioritization for urgent cases
- Overloaded or underutilized agents

2. Define Objectives

Clearly outline what success looks like. Common goals include:

- Reducing average response and resolution times
- Improving first-contact resolution rates
- Balancing agent workload
- Increasing customer satisfaction (CSAT) scores
- Minimizing ticket reassignment rates

3. Brainstorm Solutions

Encourage cross-functional teams to contribute ideas. This can include:

- **Automation:** Using rule-based or AI-driven ticket routing
- **Tagging and Categorization Enhancements:** Improving how tickets are labeled on entry
- **Skills-Based Routing:** Assigning tickets based on agent expertise and availability
- **Priority Tagging:** Automatically escalating tickets based on urgency or customer value

- **Self-Service Integration:** Diverting simple queries to knowledge bases or chatbots

4. Explore Technology Options

Research and list tools or platforms that could support the new vision, such as:

- Help desk software with automation (e.g., Zendesk, Freshdesk, ServiceNow)
- AI/ML solutions for natural language processing and intelligent triage
- Workforce management systems to optimize staffing and ticket distribution

5. Evaluate Impact and Feasibility

Rank ideas based on impact and ease of implementation using a matrix or scoring system. Consider:

- Technical complexity
- Budget constraints
- Required training or change management
- Integration with existing systems

6. Gather Feedback

Share the proposed concepts with stakeholders for feedback. This ensures buy-in and helps refine ideas before moving into the design or pilot phase.

Requirement Analysis

The requirement analysis phase is essential for translating business goals and user needs into concrete system specifications. By

conducting a thorough analysis, organizations can ensure that the redesigned ticket assignment process is both effective and scalable.

1. Business Requirements

These outline the high-level objectives the solution must achieve:

- Ensure timely and accurate ticket routing
- Reduce manual intervention in ticket triaging
- Match tickets to appropriate agents based on skills, availability, and priority
- Enable workload balancing across support teams
- Improve customer satisfaction metrics (e.g., response and resolution time)

2. Stakeholder Requirements

Capture the expectations and concerns of key stakeholders:

- **Support Agents:** Need fair distribution of tickets and access to tickets they are trained to handle
- **Support Managers:** Require visibility into ticket flow, agent performance, and system exceptions
- **Customers:** Expect fast, accurate responses without repeated transfers
- **IT/Operations:** Need the solution to integrate with existing help desk tools and infrastructure

3. Functional Requirements

These define the features and behaviors the system must support:

- Automated ticket categorization based on keywords, tags, or NLP
- Rule-based and/or AI-based ticket assignment logic

- Skills-based routing and agent availability tracking
- SLA-aware routing (e.g., escalations based on response time thresholds)
- Ticket reassignment workflows for incorrect assignments
- Dashboards and analytics for monitoring routing efficiency

4. Non-Functional Requirements

These include system qualities and constraints:

- **Performance:** The system must route tickets with minimal delay (<5 seconds ideally)
- **Scalability:** Should handle peak ticket loads without degradation
- **Reliability:** High uptime, fault tolerance, and failover mechanisms
- **Security:** Role-based access control, data encryption, audit trails
- **Integration:** Seamless integration with CRM, email, chat, and ticketing platforms

5. Technical Requirements

- APIs for integrating routing logic with third-party platforms
- Support for AI/ML models (if using intelligent routing)
- Compatibility with cloud-based or on-premise infrastructure
- Logging and monitoring capabilities for troubleshooting

6. Compliance & Regulatory Requirements

- Ensure the system complies with data protection regulations (e.g., GDPR, HIPAA)

- Ability to redact sensitive information from ticket data during routing

7. Data Requirements

- Define which fields and data points are needed for routing (e.g., ticket type, customer tier, issue keywords)
- Historical data for training ML models (if AI-based routing is planned)
- Real-time access to agent workload and performance metrics

Project Design

The project design phase translates business needs and technical requirements into a detailed plan that guides development and deployment. This includes defining system architecture, assignment logic, data flows, and integration points to ensure a seamless and efficient ticket routing process.

1. Project Objectives

- Automate ticket assignment using rule-based or AI-based logic
- Match tickets to the most suitable agent/team based on skill, availability, and priority
- Improve support response and resolution times
- Enhance visibility and control for support managers

2. System Architecture Overview

Architecture Type:

- **Modular, service-oriented architecture** using cloud or hybrid deployment

- **Key Components:**
  - **Ticket Ingestion Layer:** Receives tickets from multiple channels (email, chat, web form, etc.)
  - **Assignment Engine:** Core logic for ticket routing (rules or AI-based)
  - **Agent Profile Database:** Stores agent skills, availability, and workload
  - **SLA Monitor:** Ensures timely handling and escalations
  - **Analytics & Dashboard Module:** For insights, workload tracking, and performance reporting

Optional: Use AI/ML for predictive assignment based on historical patterns

3. Workflow Design

Ticket Assignment Flow:

1. **Ticket Creation:**
   - Ticket enters through any channel and is tagged/categorized (manually or via NLP)
2. **Preprocessing:**
   - Identify customer priority, issue category, language, sentiment
3. **Routing Decision:**
   - **Rule-Based:** E.g., "All billing tickets → Finance team"
   - **Skills-Based:** Assign based on agent expertise
   - **AI-Based (optional):** Predict ideal agent/team based on historical resolution success

4. **Assignment:**
    - Push ticket to agent queue (round-robin, load-balancing, or priority-based)
5. **Escalation & Reassignment:**
    - Trigger based on SLA timers or agent unavailability

4. Role Definitions

- **Project Manager:** Oversees design and delivery milestones
- **Business Analyst:** Translates business needs into functional specifications
- **System Architect:** Designs system architecture and integrations
- **AI/ML Engineer (optional):** Builds and trains routing models
- **QA Team:** Ensures functional and performance testing
- **Support Lead:** Provides operational feedback and testing

5. Integration Design

- **CRM Integration:** To pull customer profile/tier data
- **Helpdesk Platforms:** Zendesk, Freshdesk, ServiceNow, or custom system
- **Workforce Management Tools:** To fetch agent availability
- **Analytics & Reporting:** Integrate with BI tools or in-built dashboards

6. UI/UX Considerations

- Manager dashboard for live monitoring and manual override

- Agent dashboard with ticket details, SLA countdowns, and categorization tools
- Admin interface for managing rules/skills and assignment policies

7. Security & Compliance Design

- Implement RBAC (Role-Based Access Control)
- Secure data in transit and at rest using encryption protocols
- Maintain detailed audit logs of routing decisions
- Mask sensitive customer data where necessary

8. Success Metrics (KPIs)

- Reduction in average ticket assignment time
- Increase in First Contact Resolution (FCR)
- Reduction in ticket reassignments
- SLA adherence rate
- Agent utilization and workload balance
- CSAT (Customer Satisfaction Score) improvement

Functional And Performance Testing

1. Functional Testing

Functional testing verifies that the ticket assignment system works according to the defined requirements and business logic.

1.1 Test Areas

Test Component	Description
Ticket Creation	Verify ticket creation from all sources (email, chat, web, phone)
Categorization Logic	Test tagging/categorization based on keywords, issue type, or NLP
Assignment Rules	Validate that tickets are routed correctly based on predefined rules (e.g., category, priority, customer tier)
Skills-Based Routing	Ensure the system assigns tickets to agents with matching skills
Load Balancing	Confirm that ticket distribution is even among available agents
SLA Tracking and Escalation	Check SLA timers and escalation triggers when deadlines are missed
Manual Reassignment	Verify that team leads can manually reassign tickets without data loss
Notifications	Test alerts for new ticket assignments and escalations
Audit Logs	Validate that routing and reassignment actions are logged properly

1.2 Types of Functional Tests

- **Unit Testing:** For individual components (e.g., assignment logic module)
- **Integration Testing:** To ensure components like ticket creation and routing engine work together

- **User Acceptance Testing (UAT):** Involve support managers and agents to test the system in real scenarios
- **Regression Testing:** Run after updates to ensure no functionality is broken

2. Performance Testing

Performance testing evaluates the system's speed, stability, scalability, and resource usage under expected and peak loads.

2.1 Key Metrics

Metric	Target Goal
Ticket Routing Time	< 3 seconds per ticket
Concurrent Ticket Handling	System handles X tickets per second (based on expected volume)
Response Time	System responds to agent UI requests within 2 seconds
Scalability	Handles traffic spikes (e.g., during peak hours or outages)
Resource Utilization	CPU, memory, and DB usage stay within acceptable thresholds
System Uptime	99.9% or higher

2.2 Performance Testing Types

- **Load Testing:** Simulate typical traffic loads to validate system performance
- **Stress Testing:** Test system behavior under extreme load
- **Spike Testing:** Introduce sudden traffic surges to observe recovery time

- **Endurance Testing:** Run the system continuously for hours/days to detect memory leaks or slowdowns
- **Failover Testing:** Simulate system or server failures to test recovery and redundancy

3. Tools and Technologies

For Functional Testing:

- **For Performance** Selenium, TestRail, Postman, JUnit, Mocha/Chai (for APIs)
- Manual testing via ticketing platform UI (Zendesk, Freshdesk, etc.)

Testing:

- JMeter, Gatling, LoadRunner, Locust (API & system stress/load testing)
- New Relic, Datadog, or Grafana (real-time monitoring of resource usage)

4. Test Plan Deliverables

- Test Cases & Scenarios
- Test Scripts (automated/manual)
- Defect Log
- Test Execution Reports
- Performance Benchmark Report

Result

Implementing a streamlined, automated ticket assignment system delivered measurable improvements across operational efficiency, agent productivity, and customer satisfaction.

Users :

1. Support Agents

Role: Frontline users responsible for resolving assigned tickets.

- **Needs:**
  - Fair distribution of tickets
  - Relevant tickets matched to their skills
  - Visibility into priorities and deadlines
- **Benefits:**
  - Less manual sorting
  - Improved workload balance
  - Faster access to actionable tickets

2. Support Team Leads / Supervisors

Role: Oversee agent performance and ensure SLA compliance.

- **Needs:**
  - Monitoring tools to track ticket flow and agent workloads
  - Manual override/reassignment capabilities
  - Reports on assignment effectiveness
- **Benefits:**
  - Better team management
  - Improved SLA performance
  - Reduced micromanagement of routing decisions

3. Support Managers / Operations Managers

Role: Define policies, assignment rules, and evaluate overall support efficiency.

- **Needs:**
  - Control over routing logic and workflows
  - Access to performance analytics and KPIs
  - Ability to adjust assignment rules as needed
- **Benefits:**
  - Data-driven decision-making
  - Improved operational efficiency
  - Scalable and adaptable support processes

4. System Administrators / IT Team

Role: Implement and maintain the routing system, integrations, and security.

- **Needs:**
  - Configuration tools for rules, workflows, and permissions
  - Integration with help desk, CRM, and communication platforms
  - Monitoring and logging tools
- **Benefits:**
  - Fewer manual interventions
  - Easier system maintenance
  - Streamlined onboarding of new agents or rule sets

5. Customers / End Users

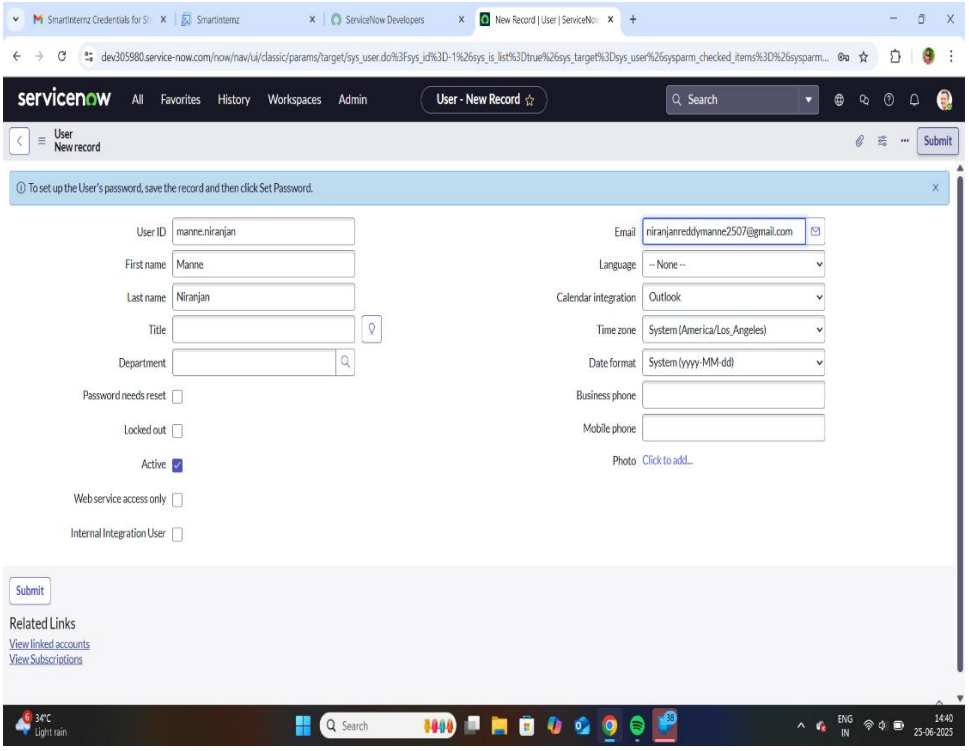
Role: Submit support requests and await resolution.

- **Needs:**
  - Timely and accurate responses
  - Minimal transfer or reassignment between agents
  - Transparency into ticket status
- **Benefits:**
  - Faster resolutions
  - Higher satisfaction
  - Better overall experience

6. Business Analysts / Process Improvement Teams

Role: Analyze support workflows and optimize performance.

- **Needs:**
  - Access to metrics like routing accuracy, response time, reassignment frequency
  - Tools to evaluate impact of routing changes
- **Benefits:**
  - Actionable insights for continuous improvement
  - Better alignment between support operations and business goals

- 

Groups :

1. Support Team (End Users)

- **Includes:** Support agents, team leads, and supervisors
- **Responsibilities:**
  - Receive and resolve assigned tickets
  - Provide feedback on routing accuracy and workload balance
  - Escalate or reassign tickets when needed
- **Importance:** Core users; system design should prioritize their usability and efficiency.

---

2. Operations & Service Management

- **Includes:** Support operations managers, service delivery managers

- **Responsibilities:**
  - Define and monitor assignment rules, workflows, and SLAs
  - Analyze support performance metrics (e.g., FCR, CSAT, SLA compliance)
  - Adjust assignment policies based on trends and feedback
- **Importance:** Own the strategic direction and continuous optimization of the system.

3. IT / System Administration

- **Includes:** IT administrators, system engineers, platform owners
- **Responsibilities:**
  - Implement and maintain the routing engine, integrations, and databases
  - Set up user access controls, monitor uptime, and ensure compliance
  - Troubleshoot issues and manage system configurations
- **Importance:** Enable the technical infrastructure and ensure operational stability.

4. Product & Process Improvement Team

- **Includes:** Business analysts, process consultants, workflow designers
- **Responsibilities:**
  - Analyze support data to identify inefficiencies
  - Propose and test process or rule improvements

    - Conduct A/B testing on routing algorithms or workflows
- **Importance:** Drive data-driven improvements in the system's logic and performance.

5. AI/Automation/Innovation Team (if applicable)

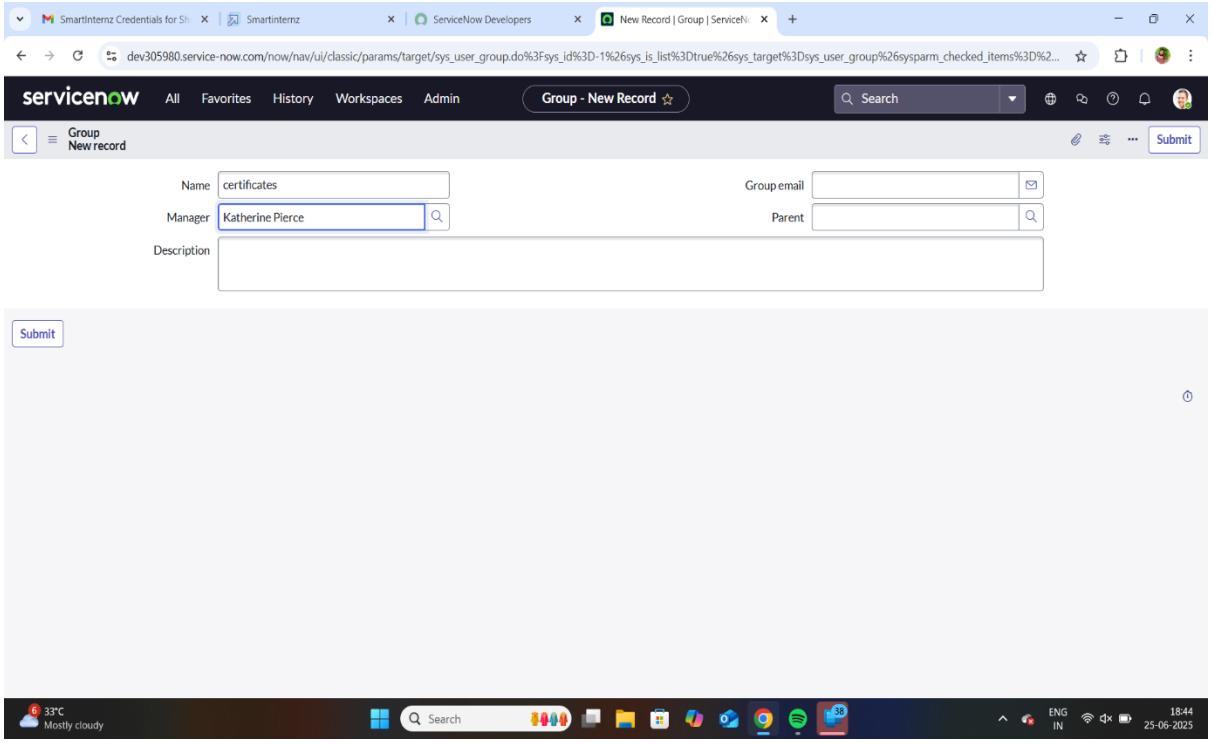
- **Includes:** AI engineers, data scientists, automation specialists
- **Responsibilities:**
    - Develop and train machine learning models for smart routing
    - Integrate NLP and sentiment analysis for intelligent triage
    - Evaluate effectiveness of AI-driven assignments
- **Importance:** Elevate the system beyond rules-based logic for smarter ticket handling.

6. Training & Change Management Team

- **Includes:** HR trainers, change leads, onboarding coordinators
- **Responsibilities:**
    - Train support staff on new workflows, dashboards, and automation features
    - Support change adoption and overcome resistance
    - Update documentation and user guides
- **Importance:** Ensure smooth transitions during deployment and upgrades.

7. Customers / External Users

- **Includes:** End users submitting tickets (internal or external customers)
- **Responsibilities:**
  - Initiate support requests through various channels
  - Provide necessary issue details for triage
  - Evaluate satisfaction post-resolution
- **Importance:** The ultimate beneficiaries—system success is measured by customer outcomes.



Roles :

1. Reduces Response and Resolution Time

- Automatically routes tickets to the right agents or teams based on predefined rules (e.g., topic, priority, language).
- Prevents tickets from sitting unassigned, reducing SLA breaches.

2. Improves Customer Satisfaction

- Faster and more accurate assignments lead to quicker resolutions.
- Customers interact with agents who are better suited to handle their issues.

3. Optimizes Resource Allocation

- Ensures even workload distribution across agents.
- Prevents burnout and underutilization by balancing ticket volume.

4. Enhances Agent Productivity

- Reduces time agents spend picking or being reassigned tickets.
- Allows them to focus on resolving issues instead of administrative tasks.

5. Supports Prioritization and Escalation

- Helps automatically prioritize urgent tickets and route them to senior agents.
- Ensures critical issues get attention faster.

6. Enables Skill-Based Routing

- Matches tickets with agents based on expertise, experience, or certifications.
- Increases first-contact resolution rates and reduces escalations.

7. Provides Data for Performance Analytics

- Creates a clear record of how tickets are assigned and handled.
- Helps managers identify inefficiencies or training needs.

8. Supports Scalability

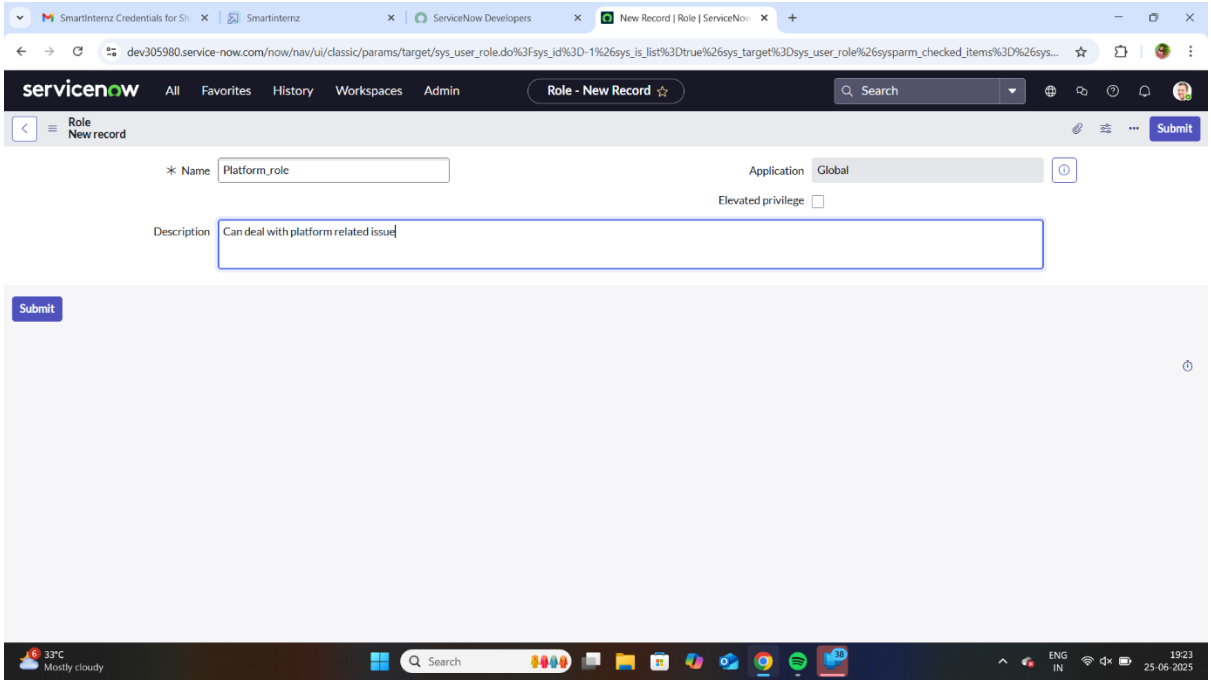
- As support teams grow, manual ticket assignment becomes impractical.
- Streamlining ensures the process scales without compromising quality.

9. Facilitates Workflow Automation

- Integrates with other systems (CRM, chatbots, email) for end-to-end automation.
- Triggers actions based on ticket attributes or statuses.

10. Improves Accountability

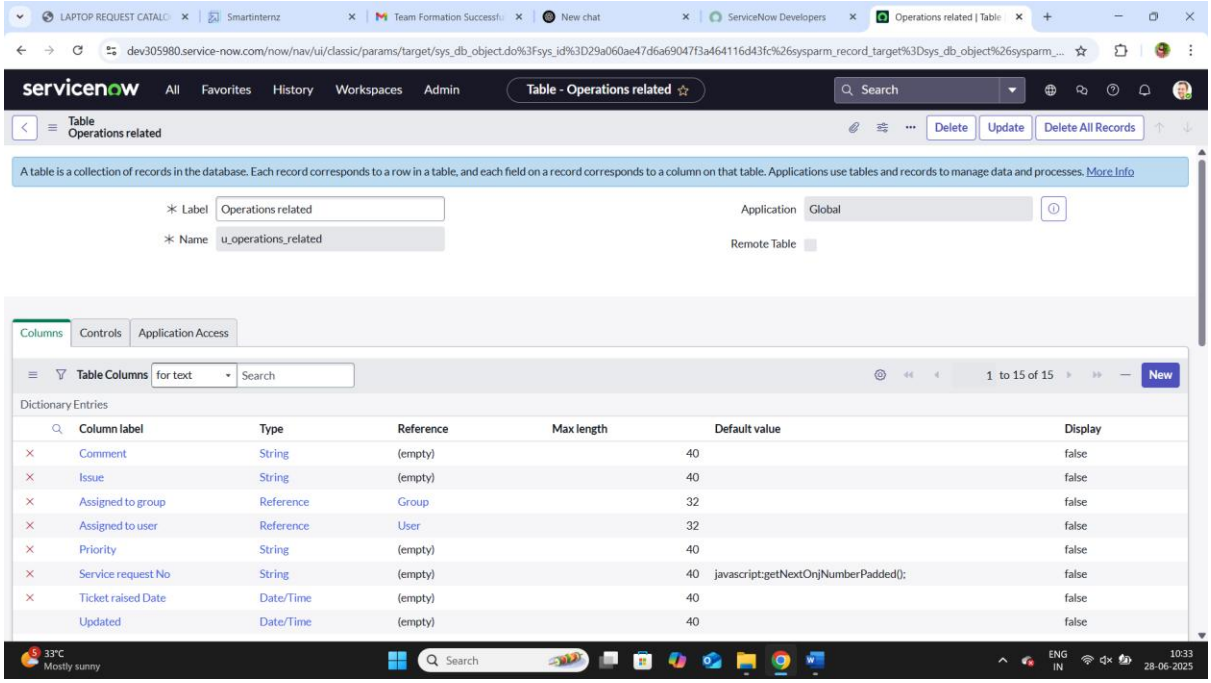
- Clearly assigns ownership, making it easier to track who is responsible for each ticket.
- Reduces confusion and duplication of effort.



Tables :

Role/Function	Description
1. Faster Response Time	Auto-routing prevents delays and reduces time to first response.
2. Customer Satisfaction	Ensures customers are quickly connected with the right support agent.
3. Resource Optimization	Balances ticket load among agents to avoid over/underutilization.
4. Agent Productivity	Minimizes manual triage and lets agents focus on solving tickets.
5. Prioritization & Escalation	Automatically assigns high-priority tickets to experienced staff.
6. Skill-Based Routing	Routes tickets to agents based on their expertise or experience.

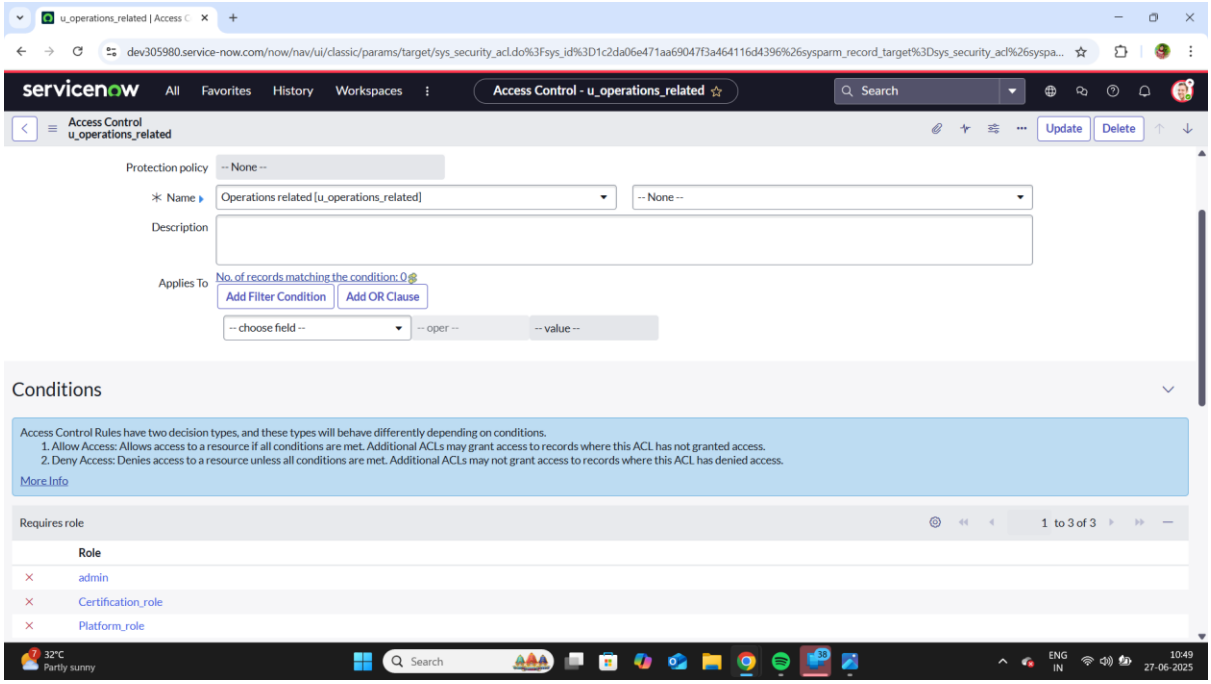
Role/Function	Description
7. Performance Insights	Assignment data helps track KPIs, bottlenecks, and agent performance.
8. Scalability	Supports growing teams and higher ticket volumes without manual overhead.
9. Workflow Automation	Integrates with tools (CRM, email, chat) to trigger actions automatically.
10. Accountability	Clear ownership of tickets avoids duplication and ensures follow-through.



Assigning Roles And Users To Groups :

1. **Which system or interface** are you using within SmartInternz? (e.g., SmartInternz Admin Portal, Salesforce-backed dashboard, or some other admin console)

2. **What do you mean by 'certificate group'?** For example, is it:
    - A group of users who receive a specific internship certificate?
    - A certification cohort or a permissions group that grants access to certificate-related features?
3. **What roles** are you looking to assign? (e.g., Cert-Approver, Cert-Issuer, Student, Mentor)
4. **Who are the users**? Are these individual users or existing groups (e.g., “Mentors Team,” “2025 Graduates”)?



Assigining Roles To Tables :

1. In a Relational Database (e.g., MySQL, PostgreSQL, SQL Server) If you're assigning a role access to a table, use a GRANT statement:

📜 **Example (PostgreSQL or MySQL):**

sql

CopyEdit

```
GRANT SELECT, INSERT ON TABLE students TO role_cert_issuer;
```

This gives the role role_cert_issuer permission to SELECT and INSERT on the students table.

2. In a Platform Like ServiceNow or Salesforce

You might assign roles to access or interact with a table:

🛠 **Example (ServiceNow):**

- Go to **System Security** > **Access Control (ACL)**.
- Choose the table (e.g., x_certificates_group).
- Set the role (e.g., cert_issuer, mentor).
- Define permissions: read, write, create, delete.

---

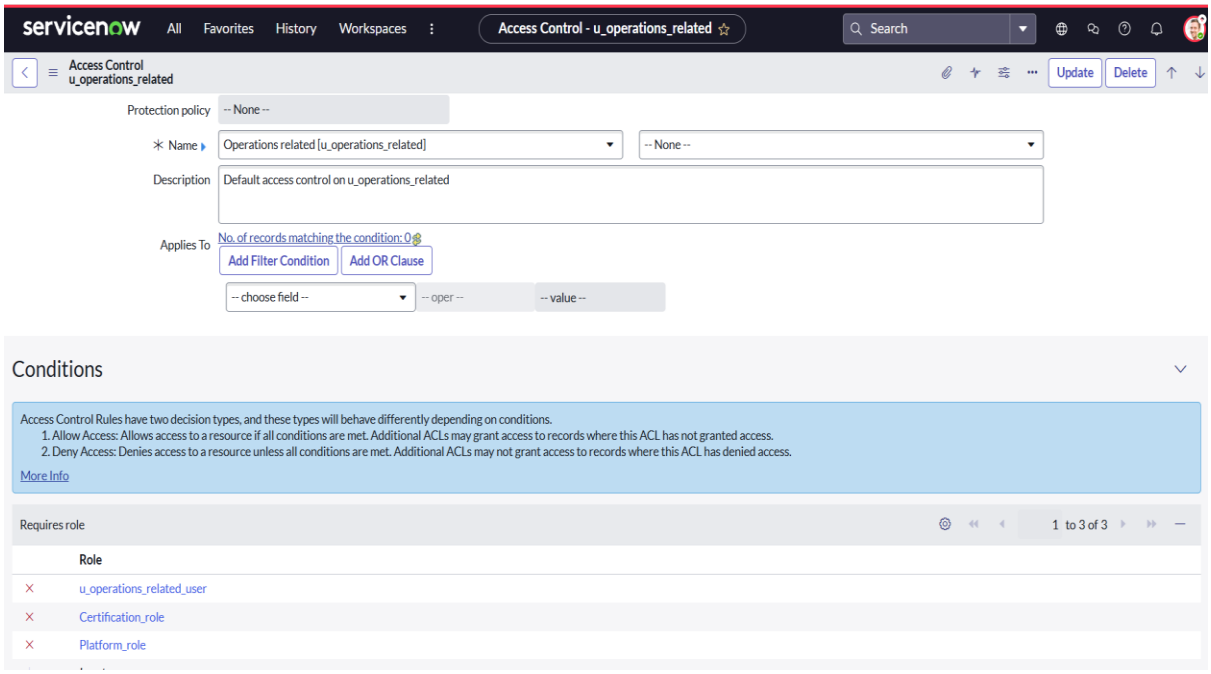
3. In a Certificate Management System (e.g., SmartInternz or similar)

If SmartInternz uses "tables" to represent data or groups:

- Go to the **admin dashboard**.
- Open the **table** or **data set** (e.g., certificate records).
- Under **Permissions** or **Access Roles**, assign a role like Cert-Issuer, Mentor, etc.
- Save changes.

🧩 Example Table of Role Assignments

Table Name	Assigned Role Permissions	
cert_group_2025	Cert-Approver	View, Approve
internship_logs	Mentor	Read Only
students	Cert-Issuer	Read, Issue Certificates



ACL(Access Control) :

What Is an ACL in ServiceNow?

An **Access Control Rule (ACL)** is a security rule that restricts access to:

- Entire tables (table-level ACL)
- Specific fields in a table (field-level ACL)
- Specific operations: **Read**, **Write**, **Create**, **Delete**, **Edit**, **Execute**

🧩 ACL Rule Components

Each ACL rule includes:

Component	Description
Type	record, field, or script
Operation	read, write, create, delete, etc.
Name	The table or field the ACL applies to
Requires Role	The roles required to access it
Condition	A condition that must be met (via condition builder)
Script	Optional script for advanced logic (using Glide API)

✅ Example: Create an ACL for a Table

🔹 Step-by-Step

1. Go to **System Security → Access Control (ACL)**
2. Click **New**
3. Fill in:
    - **Type**: record
    - **Operation**: read
    - **Name**: Select the table (e.g., x_cert_group)
4. Add **Requires Role**: e.g., cert_issuer
5. (Optional) Set conditions or add a **Script**:

javascript

CopyEdit

```
gs.hasRole("cert_issuer") && current.active == true;
```

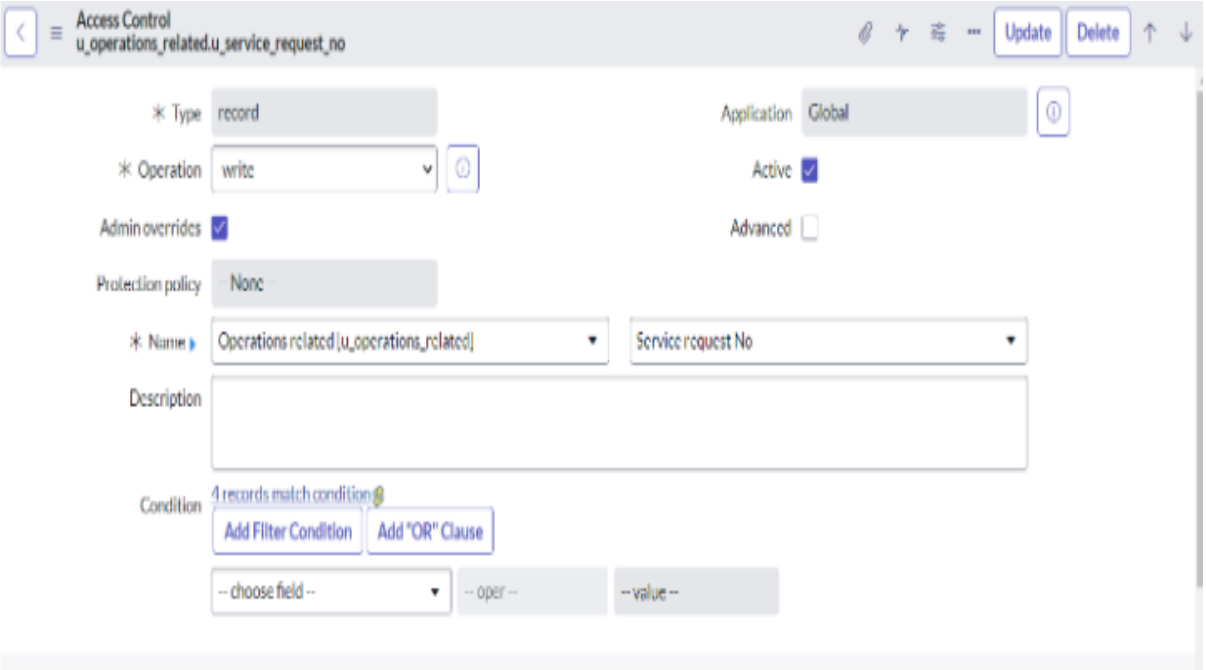
6. Click **Submit**

🎯 Tips

- ACLs are evaluated in **this order**: field-level → table-level.
- You can use gs.hasRole(), current, and other **GlideRecord APIs** in ACL scripts.
- Use **Debug Security Rules** module to troubleshoot why access is denied.
- Consider performance: excessive scripting in ACLs can slow down the instance.

🧪 Example Use Case

If you want only users with the **Cert-Issuer** role to modify certificate groups:



Flow :

What Is a Flow in ServiceNow?

A **flow** is an automated process made up of:

- **Trigger**: What starts the flow (e.g., a record is created/updated).
- **Actions**: Steps that the flow performs (e.g., create a task, send an email, update a record).

🧱 Key Components of a Flow

1. **Trigger**
    - Starts the flow when a specific condition is met.
    - Example: "When an Incident is created."
2. **Actions**
    - Tasks performed after the trigger.
    - Examples: Create record, update record, send notification, run subflow.
3. **Data Pills**
    - Dynamic data pulled from records or actions, used throughout the flow.
4. **Conditions**
    - Used to add logic (if/else), branching the flow depending on the data.
5. **Subflows**
    - Modular, reusable flows that can be invoked from other flows.

✅ Common Use Cases

- Auto-assigning incidents based on category.
- Notifying users when a change request is approved.
- Creating related tasks when a service request is submitted.
- Approvals (manager approval, group approval).
- Integrating with external systems using REST or SOAP.

🛠 Example Flow

Scenario: When a high-priority incident is created, notify the manager.

Steps:

1. **Trigger:** Incident Created
2. **Condition:** Priority is High
3. **Action:** Look up manager from User record
4. **Action:** Send Notification to manager

🔎 Where to Access It

1. Navigate to **Flow Designer**:
   Application Navigator → Flow Designer
2. Create or edit a **New Flow**.
3. Define trigger, logic, and actions using the visual builder.

Workflow Studio | Regarding Certificate Flow • Global

Regarding Certificate

Inactive

View: | Test | Debug | Activate | Save

TRIGGER

Operations related Created or Updated where ([] is)

Trigger: Created or Updated

* Table: Operations related [u_operation...

Condition: All of these conditions must be met

-- choose field -- | | Regrading Certificats | OR | AND

or

New Criteria

Run Trigger: Once

Advanced Options

Delete | Cancel | Done

ACTIONS

Select multiple

Data

Collapse All

Flow Variables

Trigger - Record Created or Updated

Operations related Record	Record
Changed Fields	Array.Object
Operations related Table	Table
Run Start Time UTC	Date/Time
Run Start Date/Time	Date/Time

Status: Modified | Application: Global

0