✅ Mini Project: Problem Management System in ServiceNow

Description:

○ Develop the Problem Management lifecycle and stages.

○ Configure forms, manage field conditions, and introduce custom modifications.

✅ Project Objective

To design and implement a Problem Management System in ServiceNow that helps identify, investigate, and eliminate the root causes of incidents. The system includes a structured lifecycle, form customizations, validations, automated workflows, and seamless integration with knowledge management for long-term solutions.

🧩 Step 1: Use Global Application Scope

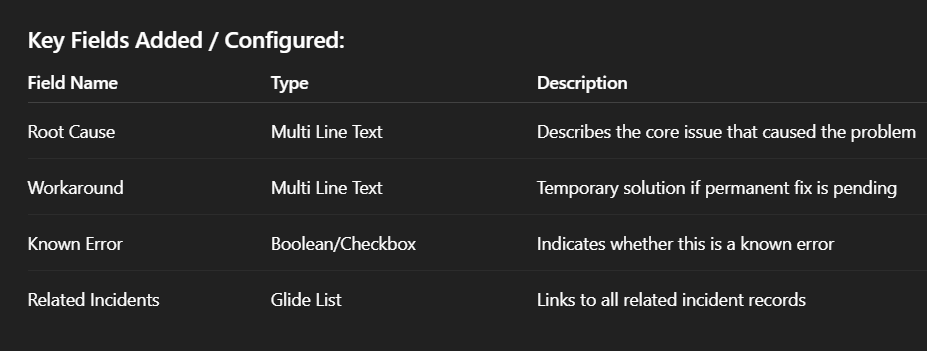
Since Problem Management is part of the Global scope in ServiceNow:

* No need to create a new scoped application
* Work directly with the existing problem table and its related modules

🗃️ Step 2: Configure Fields & Form Layout

Table Used: problem

Key Fields Added / Configured



📝 Step 3: Implement UI Policies

✅ UI Policy 1 – Show Workaround when State = Investigating

* Table: Problem
* Condition: State = Investigating
* Action: Workaround → Visible = ✅

✅ UI Policy 2 – Make Root Cause and Known Error Editable when State = Resolved

* Condition: State = Resolved
* Action: Root Cause & Known Error → Read-Only = ❌

✅ UI Policy 3 – Lock all custom fields when State = Closed

* Condition: State = Closed
* Action: All custom fields → Read-Only = ✅

**⚙️ Step 4: Implement Business Rules**

✅ **Business Rule – Enforce Root Cause on Resolution**

* Name: Validate Root Cause
* Table: Problem
* When: Before → Insert/Update
* Condition: State = Resolved AND Root Cause is empty
* Result: Show error message and prevent submission

✅ **Business Rule – Auto Assign Manager**

* Name: Auto Assign Problem Manager
* Table: Problem
* When: Before Insert
* Condition: Priority = 1 or 2
* Action: Set Assigned To = Default Manager

✅ **Business Rule – Auto Update State When Workaround Is Entered**

* Name: Set State on Workaround
* Table: Problem
* When: Before Update
* Condition: Workaround is not empty AND State = Investigating
* Result: Set State = Workaround Defined (custom value if implemented)

**🔁 Step 5: Configure Problem Lifecycle Workflow**

Designed a custom state model to reflect the full problem lifecycle using **Workflow Editor or Flow Designer**.

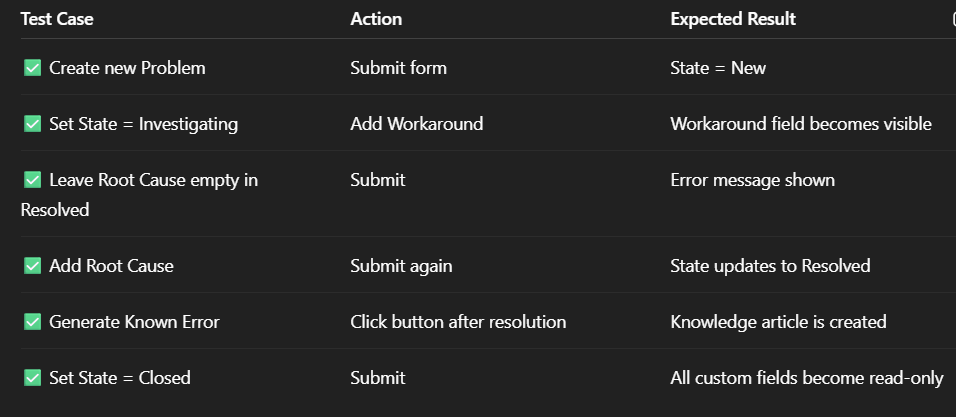
**Lifecycle Stages:**

* **New** – Problem logged
* **Investigating** – Under analysis
* **Workaround Defined** – Temporary fix in place
* **Resolved** – Root cause eliminated
* **Closed** – Problem verified and completed

**Additional Logic:**

* Enable "Generate Known Error Article" on resolution
* Record links to KB articles for future reference
* Enable visibility rules based on state

🧪 Step 6: Testing the Flow



**✅ Status**

✔️ **Completed**  
🛠️ **Easily extendable with CMDB or Incident linkage**  
📬 **Notifications, validations, and KB integration verified**

**👩‍💻 Author**

**Munagapati Bhavana**  
B.Tech, GITAM University, Bangalore  
📌 GitHub: [MunagapatiBhavana](https://github.com/MunagapatiBhavana)