Project Design Phase
Solution Architecture

Date	7-11-2025
Team ID	NM2025TMID09117
Project Name	Educational Organisation Using ServiceNow
Maximum Marks	4 Marks

Solution Architecture

Goals of the architecture

The Problem-Design-Solution Architecture template serves as the technical blueprint for approved initiatives, bridging the conceptual solution with the practical, technical build within the ServiceNow environment. This stage is crucial because it defines the necessary platform modules, required integrations, data structures, and overall system configuration, ensuring the solution is scalable, maintainable, and aligned with the institution's enterprise architecture standards.

The Problem (Context and Goal)

This section briefly summarizes the approved problem statement and the desired outcome, ensuring the architectural work remains focused.

- **Focus:** A concise restatement of the validated user need (e.g., "Reduce the time faculty members spend on manual student leave request approvals from 4 days to less than 24 hours"). This ensures the architecture is designed to meet this specific performance goal.

The Design (Functional Requirements)

This defines the user experience and functional requirements, outlining what the solution must *do* from the user's perspective.

- **Focus:** The core features and end-to-end process flow (e.g., "The solution must allow students to submit a leave request via a mobile-friendly portal, automatically route the request to the correct department head based on student record data, and provide email notifications at every status change.").

The Solution Architecture (Technical Blueprint)

This is the technical core, detailing the specific components and how they fit together on the platform.

- **Core Modules & Components:** Identifies the foundational ServiceNow applications (e.g., utilize **HR Service Delivery (HRSD)** Case Management for tracking, **Employee Center Pro** for the student interface, and **Service Catalog** for the intake form).
- **Data Model:** Defines the necessary tables and fields. For example, creating a custom u_student_leave_request table that links to the existing **User** table (for the student and faculty approver) and the **CMDB** (if linking to campus services).
- **Workflow Automation:** Specifies the engine driving the process (e.g., leverage **Flow Designer** to create the multi-stage approval flow, including conditional routing based on leave duration or department).
- **Integration Points:** Identifies necessary connections to external campus systems (e.g., requires a **ServiceNow IntegrationHub** spoke to connect to the Student Information System (SIS) via REST API for student status verification and post-approval record update).
- **Security & Governance:** Details access control (ACLs) and role management (e.g., ensure only users with the u_faculty_approver role can view and approve requests).

By clearly defining the Solution Architecture, the organization minimizes technical debt, accelerates development, and ensures the new application integrates seamlessly with the broader ServiceNow enterprise platform.