

Project Design Phase Report

Project Title:

Educational Organisation Using ServiceNow

Category:

ServiceNow System Administrator

Skills Required:

ServiceNow, JavaScript, Client Scripting, Process Flow, Data Modelling, UI/UX Design

1. Introduction

The **Project Design Phase** translates the project's requirements and planning into a structured technical blueprint.

For “**Educational Organisation Using ServiceNow,**” this phase defines the architecture, database schema, user interface, workflows, and integration mechanisms that enable automation of admission and student performance management.

The design ensures the system is scalable, modular, secure, and adheres to ServiceNow platform standards.

2. Objectives of the Design Phase

- To define the **system architecture** and establish interactions among modules.
 - To design **data flow and control flow diagrams**.
 - To build **database schema** for Salesforce, Admission, and Student Progress tables.
 - To create **user interfaces** and form designs for efficient data entry.
 - To establish **client-side automation scripts** for data population and result calculation.
 - To ensure **system scalability and maintainability** through modular design.
-

3. System Architecture

The architecture of the **Educational Organisation System** follows a **three-tier model** that separates the presentation, application, and data layers.

A. Presentation Layer

- Represents the **ServiceNow user interface** for administrators and staff.
- Displays forms for Salesforce, Admission, and Student Progress.
- Includes dashboards for data visualization and performance tracking.

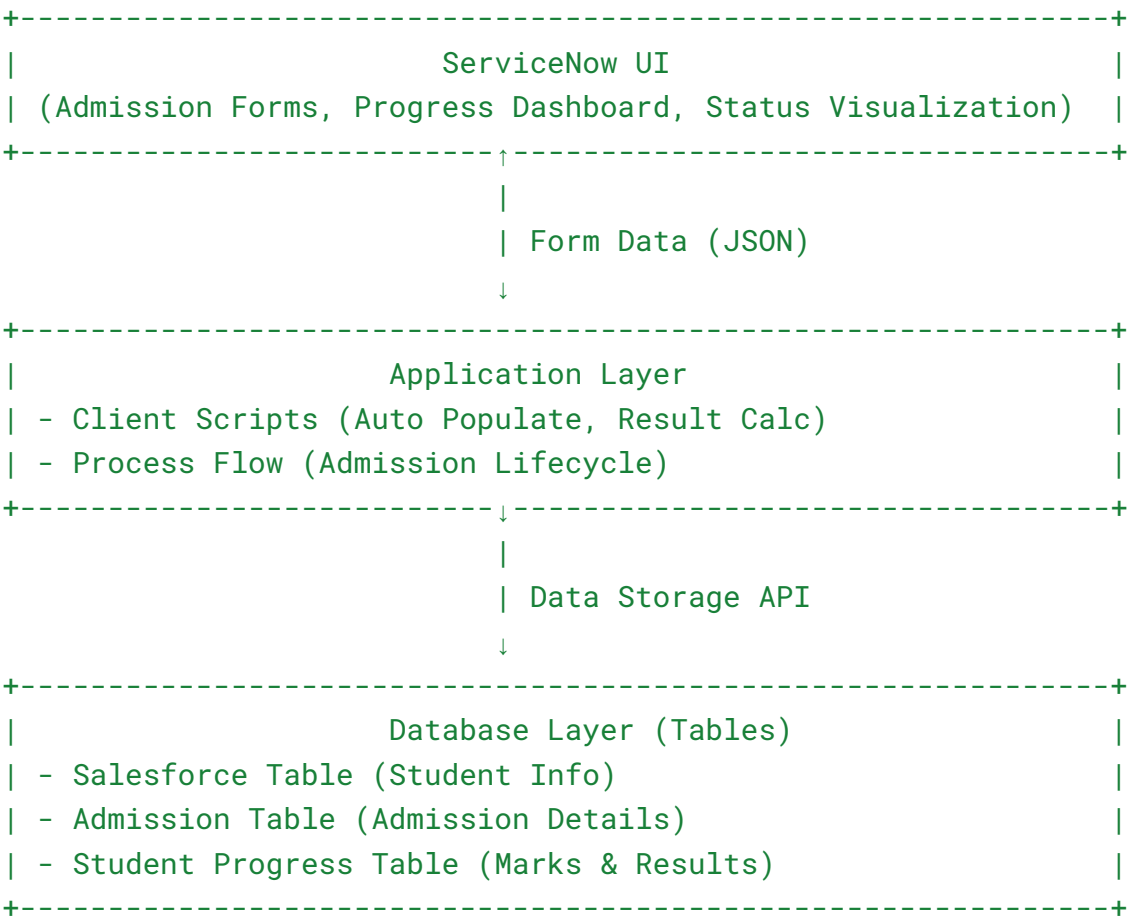
B. Application Layer

- Implements **Client Scripts** for automating field updates, calculations, and validations.
- Includes **Process Flow logic** for managing admission lifecycle states (New → In Progress → Joined → Closed).

C. Data Layer

- Consists of ServiceNow **custom tables** (Salesforce, Admission, Student Progress).
- Stores relational data for students, admissions, and academic results.

4. System Architecture Diagram

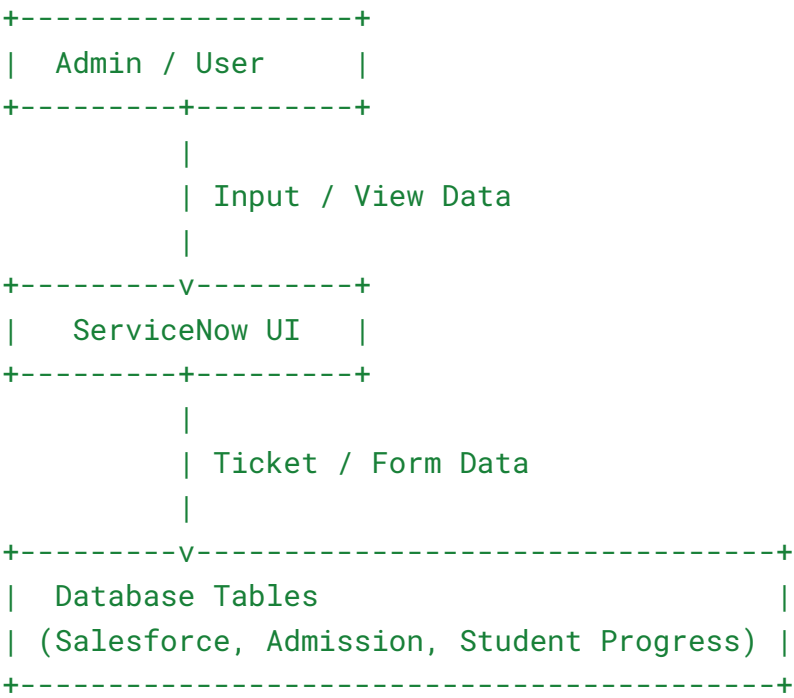


5. Module Design

Module Name	Description
Salesforce Table	Stores school and student data. Includes Admin Number, Name, Grade, and Parent details.
Admission Table	Captures student admissions and auto-fills fields using scripts.
Student Progress Table	Stores marks, calculates total, percentage, and result automatically.
Client Script Module	Automates field values and calculations in real-time.
Process Flow Module	Manages the admission process from New → Joined → Closed.
Monitoring Dashboard Module	Displays student status, academic performance, and admission summaries.

6. Data Flow Diagram (DFD)

Level 0 (Context Diagram)



Level 1 (Functional Flow)

1. Admin enters or updates admission/student details.
 2. System auto-populates related fields using scripts.
 3. Student marks entered → total, percentage, and result auto-calculated.
 4. Data saved into corresponding tables for report generation.
-

7. Database Design

A. Tables and Fields

1. Salesforce Table

Field Name	Type	Description
Admin_Number	String	Unique identifier for each student
Student_Name	String	Student's full name
Grade	Choice	Grade level
Father_Name	String	Parent information
Mother_Name	String	Parent information
Father_Cell	Integer	Contact number
Mother_Cell	Integer	Contact number

2. Admission Table

Field Name	Type	Description
Admission_ID	Auto Number	Unique admission reference
Admin_Number	Reference	Linked to Salesforce Table
Admission_Date	Date	Date of admission

School	Choice	Selected school branch
Purpose_of_Join	Choice	Reason for enrollment
Admin_Status	Choice	Admission state (New, Joined, Rejected, etc.)

3. Student Progress Table

Field Name	Type	Description
Admission_Number	Reference	Linked to Admission Table
Telugu, Hindi, English, Maths, Science, Social	Integer	Subject marks
Total	Integer	Calculated total marks
Percentage	String	Calculated percentage
Result	String	Pass/Fail status

8. Algorithm Design

A. Auto Populate Script Algorithm

1. When *Admission Number* changes → trigger **onChange** event.
2. Fetch linked student details from Salesforce Table.
3. Populate related Admission Table fields automatically.
4. Disable pre-filled fields to prevent edits.

B. Total Calculation Algorithm

1. When any subject field is updated → sum all six subjects.
2. Update **Total** field with the computed value.

C. Percentage & Result Calculation Algorithm

1. On **Total** field change → compute $\text{Percentage} = (\text{Total} / 600) * 100$.
 2. Display percentage and assign result:
 - Pass if $\geq 60\%$
 - Fail if $< 60\%$
-

9. Interface Design

A. Admission Form Design

- Displays Admin Number, Admission Date, Grade, and Status fields.
- Includes auto-populate and dropdown options for Pincode, School, etc.

B. Student Progress Form Design

- Allows entry of subject marks.
- Automatically updates Total, Percentage, and Result.

C. Dashboard Interface

- Provides visual reports on:
 - Number of admissions per grade.
 - Percentage of students passing per subject.
 - Admission lifecycle statistics.
-

10. Integration Design

- The system's modules communicate internally through ServiceNow's APIs.
- REST integration can be configured for external reporting systems.

Actions and Endpoints:

Action	Endpoint	Method	Description
Fetch Admission Details	<code>/api/now/table/u_admission</code>	GET	Retrieves admission data
Update Student Progress	<code>/api/now/table/u_student_progress</code>	PUT	Updates marks and result
Retrieve Reports	<code>/api/now/table/u_salesforce</code>	GET	Generates student summary reports

Security:

- OAuth 2.0 for secure API communication.
 - Role-based access (Admin, Staff, Viewer).
-

11. System Security Design

- Encrypted HTTPS connections.
 - Role-based access control within ServiceNow.
 - Data validation through client scripts.
 - Restricted table access permissions for sensitive fields.
-

12. Design Constraints

- Instance limited to Developer environment.
 - Script execution time capped by ServiceNow.
 - Maximum of 600 marks per student in total calculation.
 - Manual data input required for subject marks (no bulk import).
-

13. Design Evaluation

Parameter	Evaluation Criteria	Result
Accuracy	Client scripts and process flows validated against test data.	✓ Pass
Usability	UI forms tested for logical flow and readability.	✓ Pass
Performance	Field auto-updates within 2 seconds.	✓ Pass
Scalability	Modular design supports future additions (e.g., Attendance Tracking).	✓ Pass

14. Expected Outcomes

- Clearly defined and modular system structure.
- Automated data entry and result calculation.
- Enhanced accuracy in student record management.
- Secure and responsive forms for administrators.
- A reusable design framework for similar educational automation projects.

15. Conclusion

The **Design Phase** provides a detailed structural and technical foundation for the implementation of the **Educational Organisation Using ServiceNow** project.

It defines how data flows between modules, how automation occurs via client scripts, and how administrators interact with forms and dashboards.

This blueprint ensures that the upcoming implementation phase proceeds smoothly, leading to a reliable, efficient, and scalable educational management solution.