REQUIREMENT ANALYSIS

Date	**1 November 2025**
Team ID	**NM2025TMID03580**
Project Name	**Educational Organisation using Service Now**
Maximum Marks	**12 Marks**

Requirement Analysis Phase – Educational Organisation Using ServiceNow

The Requirement Analysis Phase is the foundation of the project development process. It focuses on identifying user needs, understanding functional workflows, defining solution requirements, and determining the technical architecture needed for system implementation on ServiceNow. This phase ensures that the Educational Management System is designed with clear goals, measurable outcomes, and optimal usability for educational institutions.

1. Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) visually represents the movement of data within the system — how information is input, processed, and output between users, processes, and data stores.

Level 0 DFD (Context Diagram)

At Level 0, the system is represented as a single process that interacts with external entities:

Entities:

- **Admin** – Manages admission, student details, and user access.
- **Teacher** – Updates student progress and academic results.
- **Student/Parent** – Views admission status and academic performance.
- **System Database (ServiceNow Tables)** – Stores all records securely.

Process:

- "Educational Management System" receives, processes, and outputs data related to admissions and student performance.

Data Flows:

- Admin inputs admission and student data.
- Teachers update marks and results.
- Students/Parents access progress reports.
- The system stores and retrieves all data through ServiceNow tables.

[Admin] → [Educational Management System] ← [Teacher]

↓ ↑

[Database] ←→ [Student/Parent]

Level 1 DFD (Detailed View)

Processes:

1. **Admission Management**
   - Inputs: Student details, documents, pincode, etc.
   - Outputs: Admission record with generated Admin ID.
   - Data stored in: Admission Table.
2. **Student Record Management**
   - Inputs: Name, class, address, marks.
   - Processing: Auto-population and validation.
   - Data stored in: Salesforce Table.
3. **Progress Tracking**
   - Inputs: Marks from teachers.
   - Processing: Total, Percentage, and Result calculation.
   - Data stored in: Student Progress Table.
4. **Result Reporting**
   - Outputs: Reports for students and parents via UI.
   - Processing: Fetches data from linked tables and displays summary.

Data Stores:

- Salesforce Table
- Admission Table
- Student Progress Table

2. User Stories

User stories describe how different stakeholders interact with the system to achieve their goals. Each story includes a role, action, and expected outcome.

User Story ID	**As a (Role)**	**I want to (Goal/Action)**	**So that I can (Outcome/Benefit)**
US-01	Admin	Create and manage student admission records	Maintain accurate student data for each academic year
US-02	Admin	Auto-generate unique admission numbers	Avoid duplication and maintain data consistency
US-03	Admin	View and update student details	Keep records up-to-date for official purposes
US-04	Teacher	Enter marks and evaluate student performance	Monitor and analyze student academic progress
US-05	Teacher	Automatically calculate total and percentage	Reduce manual calculation errors
US-06	Teacher	Determine “Pass/Fail” results automatically	Simplify evaluation process
US-07	Student	View my academic performance report	Track personal progress and improvement
US-08	Parent	Check admission and result details	Stay informed about child’s academic status
US-09	System	Auto-fill address fields based on pincode	Increase accuracy and data entry speed

User Story ID	As a (Role)	I want to (Goal/Action)	So that I can (Outcome/Benefit)
US-10	System	Disable computed fields for users	Prevent accidental modification of calculated values

3. Solution Requirements

The solution requirements define what the system must do (functional) and how it should behave (non-functional) to meet the organization's needs.

3.1 Functional Requirements

ID	Functional Requirement Description
FR-01	The system shall allow admins to create, read, update, and delete admission records.
FR-02	The system shall generate a unique admission number automatically for each new student.
FR-03	The system shall auto-populate fields (e.g., name, grade, address) from Salesforce when an admission ID is selected.
FR-04	The system shall automatically fetch city, district, and mandal details based on entered pincode.
FR-05	The system shall calculate total marks, percentage, and determine result (Pass/Fail) based on student scores.
FR-06	The system shall disable computed fields to prevent manual edits.
FR-07	The system shall allow teachers to record and update student academic data.
FR-08	The system shall enable parents and students to view results via the portal interface.

ID	Functional Requirement Description
FR-09	The system shall link Admission, Salesforce, and Progress tables for data consistency.
FR-10	The system shall generate reports for administrators regarding admissions and academic performance.

3.2 Non-Functional Requirements

I.0D	Non-Functional Requirement Description
NFR-01	**Performance:** The system should load forms and execute scripts within 3 seconds.
NFR-02	**Reliability:** The system should ensure 99% uptime and consistent workflow operation.
NFR-03	**Scalability:** The platform should handle increasing data volumes and users without degradation.
NFR-04	**Security:** All data should be securely stored in ServiceNow's cloud environment with restricted access.
NFR-05	**Usability:** The interface should be intuitive and accessible for non-technical users.
NFR-06	**Maintainability:** Scripts and workflows should be modular and easily updatable.
NFR-07	**Integration:** The system should seamlessly integrate Salesforce and other data tables.
NFR-08	**Data Accuracy:** Auto-validation should ensure clean, error-free data entry.
NFR-09	**Availability:** The system should be accessible 24/7 via the ServiceNow instance.
NFR-10	**Compliance:** The system should adhere to institutional data protection policies.

4. Technology Stack

4.1 Solution Architecture

The solution architecture defines how the system components interact within the ServiceNow platform.

Architecture Layers:

1. **Presentation Layer (Frontend):**
   - ServiceNow UI Pages, Forms, and Catalog Items.
   - Provides user interfaces for Admins, Teachers, and Students.
   - Client Scripts for auto-population, validation, and field manipulation.
2. **Application Layer (Logic):**
   - Business Rules, Script Includes, and Workflows for automation.
   - Handles data validation, process flow transitions, and logic execution.
3. **Data Layer (Backend):**
   - ServiceNow Tables:
     - **Salesforce Table** – Source of student and school data.
     - **Admission Table** – Handles admission process details.
     - **Student Progress Table** – Stores academic performance records.
4. **Integration Layer:**
   - Connects external data sources (e.g., Salesforce or institutional APIs) to ServiceNow.
   - Ensures synchronization and consistency across data modules.

4.2 Technology Stack Overview

Component	Technology / Tool Used	Purpose
Platform	ServiceNow	Cloud-based platform for workflow automation and data management
Database	ServiceNow Tables (MySQL backend)	Data storage and relational linking
Frontend Scripting	JavaScript (Client Scripts, UI Policies)	Validation, dynamic field control, and automation
Backend Scripting	GlideRecord, Business Rules	Database operations and logic implementation
Integration	REST APIs / ServiceNow Connectors	Integration with Salesforce data
Workflow Automation	Flow Designer / Process Flow	Automating state transitions and notifications
Testing Environment	ServiceNow Developer Instance	For building and testing modules safely
Reporting	ServiceNow Report Builder	Generating student and admission reports
Security	Role-based Access Control (RBAC)	Controls user permissions and protects data integrity

5. Summary – Requirement Analysis Phase

The Requirement Analysis Phase provided a clear understanding of system expectations and design directions.
Through detailed user stories, functional breakdown, and architecture mapping, the analysis ensures the Educational Management System aligns with institutional needs.

This phase concluded that:
✅ The system meets user expectations across administrative, academic, and student levels.

✅ Functional and non-functional requirements are feasible within the ServiceNow environment.
✅ The proposed architecture is scalable, secure, and performance-optimized.

The finalized requirements serve as the blueprint for the subsequent Design and Implementation Phases.