SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

ONLINE PROCESS AUTOMATION SYSTEM (OPAS)

College Student Management System

Document Standard: IEEE 29148

7376231EC295 **Subiksha Shree S**

Version:1.0 **03-02-2026**

Table of Contents

1. Introduction

1.1 Purpose of the Document

This Software Requirements Specification (SRS) document provides a **comprehensive, unambiguous, and verifiable description** of the requirements for the *Online Process Automation System (OPAS)* implemented in the **college student management domain**. This document is intended to serve as a **contractual and reference document** between stakeholders and the development team and follows practices adopted in corporate software development environments.

1.2 Scope of the Product

The Online Process Automation System is a **centralized, web-based enterprise application** that automates academic and administrative workflows within a college environment. The system eliminates manual, paper-based processes by providing real-time, rule-driven automation for:

- Attendance management
- Leave and sick leave approvals
- On-Duty (OD) approvals
- Academic performance tracking
- Student profile and achievement management
- Task and workflow management

The solution ensures **process consistency, transparency, auditability, and scalability**, aligning with modern enterprise automation platforms.

1.3 Definitions, Acronyms, and Abbreviations

- **OPAS** – Online Process Automation System
- **OD** – On-Duty
- **SGPA** – Semester Grade Point Average
- **CGPA** – Cumulative Grade Point Average
- **RBAC** – Role-Based Access Control
- **CRUD** – Create, Read, Update, Delete

1.4 References

- IEEE Std 29148-2018 – Systems and Software Engineering — Life Cycle Processes — Requirements Engineering
- IEEE Std 830-1998 – Software Requirements Specification

1.5 Overview

This document is organized to provide a **top-down view** of system requirements, progressing from business context to detailed functional and non-functional requirements, as followed in corporate software projects.

---

2. Overall Description

2.1 Product Perspective

The system is designed as a **modular enterprise web application** built using a three-tier architecture:

Presentation Layer: Web UI (HTML, CSS, JavaScript, Bootstrap)

Application Layer: Java (Spring Boot – RESTful Services)

Data Layer: MySQL Relational Database

Each module is loosely coupled and independently maintainable, enabling future scalability and integration.

2.2 Product Functions

- Secure authentication and authorization using RBAC
- Centralized student profile management
- Automated attendance tracking and analytics
- Rule-based leave, sick leave, and OD approval workflows
- Academic performance visualization (SGPA & CGPA)
- Task assignment and monitoring
- Real-time dashboards and reports
- Audit logging and approval traceability

2.3 User Classes and Characteristics

User Role	Responsibilities
Student	Submit leave/OD requests, view dashboard, manage profile
Mentor / Faculty	Attendance marking, approvals, task assignment
Parent	Approve leave and OD requests
Hostel Warden	Approve hosteler leave and sick leave
Head of Department (HOD)	Final OD approval
System Administrator	User management, configuration, reporting

---

2.4 Operating Environment

- Client: Modern web browsers
- Server: Java-compatible OS (Linux / Windows)
- Database: MySQL Server

2.5 Design and Implementation Constraints

- Backend shall be implemented using Java (Spring Boot)
- Frontend shall be implemented using html,css,javascript(Bootstrap)
- Relational database shall be MySQL
- System shall comply with academic data protection policies

2.6 Assumptions and Dependencies

- All users have valid institutional credentials
- Internet connectivity is available
- Approval authorities are pre-mapped to students

---

3. External Interface Requirements

Actors

- Student
- Mentor / Faculty

- Parent
- Hostel Warden
- Head of Department (HOD)
- System Administrator

Use Cases

Student

- Login to system
- View personal dashboard
- View SGPA and CGPA
- View attendance percentage
- Apply for regular leave
- Apply for sick leave
- Apply for On-Duty (OD)
- View leave and OD status
- Update personal profile
- View achievements and company participation

Mentor / Faculty

- Login to system
- Mark student attendance
- View student profiles
- Approve or reject leave requests
- Approve or reject OD requests
- Assign academic tasks
- View task status

Parent

- Login to system
- View student attendance and performance
- Approve or reject leave requests
- Approve or reject OD requests

Hostel Warden

- Login to system
- View hosteler student details
- Approve or reject hosteler leave
- Approve or reject sick leave

Head of Department (HOD)

- Login to system
- Review OD requests
- Approve or reject OD requests

System Administrator

- Manage users and roles
- Configure approval workflows
- Generate system reports
- Maintain system data

Use Case Relationships

- Leave approval includes mentor approval
- Hosteler leave extends parent approval with hostel warden approval
- Sick leave includes attendance validation
- OD approval includes mentor, parent, and HOD approval

Use Case Diagram Representation (Textual UML)

3.1 User Interfaces

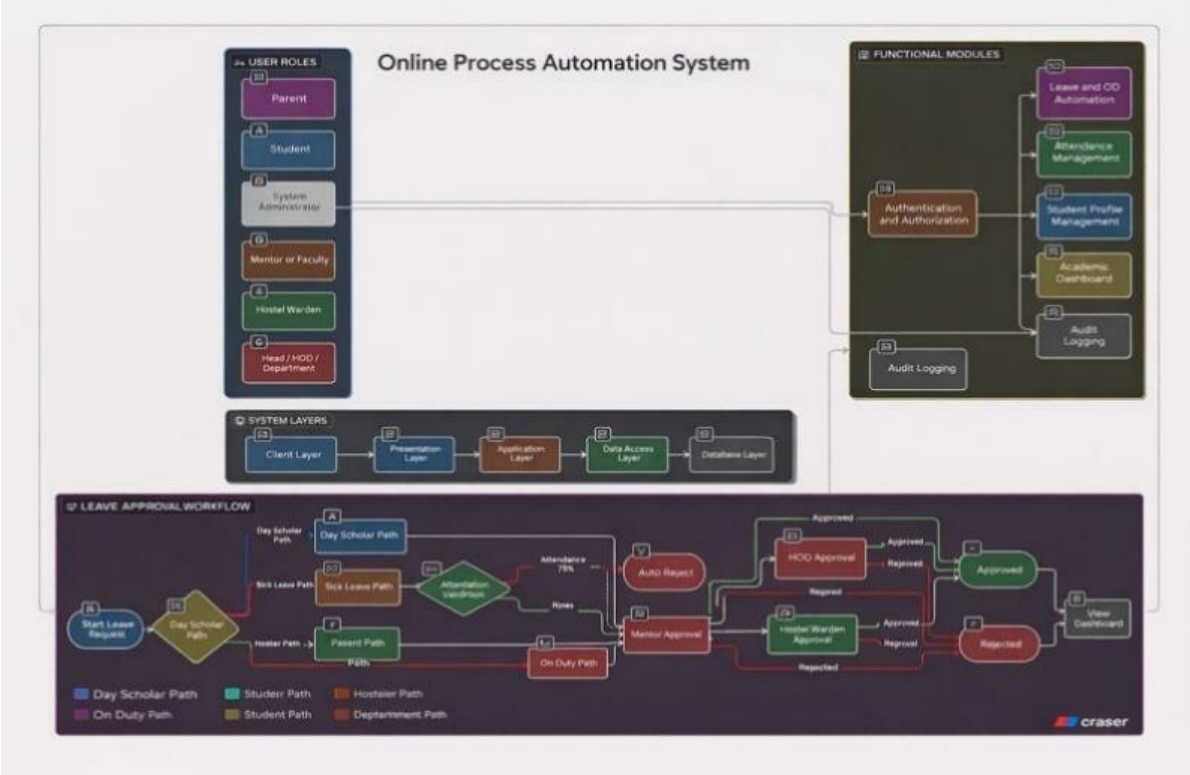
- **Student Dashboard:** Academic metrics, attendance %, leave/OD status, achievements
- **Faculty Dashboard:** Attendance, approvals, task management
- **Parent/Warden Dashboard:** Approval queue and history
- **Admin Console:** Configuration, reports, user management

3.2 Software Interfaces

- RESTful APIs for frontend-backend communication
- MySQL database for persistent storage

3.3 Communication Interfaces

- Secure HTTPS communication
- Session-based authentication



4. Functional Requirements

4.1 Authentication and Authorization

- The system shall support role-based access control.
- The system shall restrict functionality based on user roles.

4.2 Student Profile Management

- The system shall store personal details (email, contact number).
- The system shall store mentor, parent, and warden information.
- The system shall track achievements and company participation.

4.3 Attendance Management

- Faculty shall mark attendance online.
- The system shall compute attendance percentage automatically.

4.4 Leave and Sick Leave Automation

4.4.1 Regular Leave

- **Day Scholar:** Parent → Mentor
- **Hosteler:** Parent → Mentor → Hostel Warden

4.4.2 Sick Leave (Hosteler)

- The system shall validate attendance percentage.
- If attendance < 75%, the system shall automatically decline the request.
- Approved sick leave shall require hostel warden confirmation.

4.5 On-Duty (OD) Approval System

- The system shall capture OD purpose and achievement details.
- Approval flow shall be: Parent → Mentor → HOD.

4.6 Workflow Automation Engine

- The system shall dynamically determine approval paths.

- The system shall maintain an audit trail of approvals.

4.7 Student Academic Dashboard

- The system shall display SGPA for all semesters.
- The system shall calculate and display CGPA.
- The system shall display number of companies attended.
- The system shall display leave and OD history.

---

5. Non-Functional Requirements

5.1 Performance

- System response time shall be less than 3 seconds.

5.2 Security

- Passwords shall be encrypted.
- Sensitive data shall be protected.

5.3 Reliability

- System shall ensure data consistency and availability.

5.4 Scalability

- System shall support future feature expansion.

5.5 Maintainability

- Modular architecture shall allow easy maintenance.

---

6. Data Requirements

- Student
- Attendance
- Leave
- OD
- Approval
- Academic Performance

---

7. System Architecture

7.1 Architectural Overview

The Online Process Automation System follows a **layered enterprise architecture** that separates concerns and improves maintainability, scalability, and security.

[Client Layer]

- Student Portal
- Faculty Portal
- Parent / Warden Portal

↓

[Presentation Layer]

- HTML, CSS, JavaScript, Bootstrap

↓

[Application Layer]

- Java (Spring Boot)
- REST Controllers
- Business Services
- Workflow Automation Engine

↓

[Data Access Layer]

- JPA / Hibernate

↓

[Database Layer]

- MySQL Relational Database

7.2 Logical Architecture

- Controller Layer: Handles incoming requests
- Service Layer: Business logic and approval rules
- Repository Layer: Database access
- Workflow Engine: Approval automation

7.3 Deployment Architecture

User Browser

↓ HTTPS

Web Server (Spring Boot Application)

↓ JDBC

MySQL Database Server

7.4 Security Architecture

- Role-Based Access Control (RBAC)
- Encrypted credentials
- Secure session handling

8.Performance Requirements

The Online Process Automation System (OPAS) shall satisfy the following performance requirements to support real-time operation in a college environment:

- The system shall support simultaneous access by multiple users, including students, faculty, parents, hostel wardens, HODs, and administrators, without performance degradation.
- The system shall provide a response time of less than 3 seconds for standard operations such as login, dashboard loading, leave/OD submission, and approval actions under normal load conditions.
- Attendance marking and approval workflow actions shall be processed in real time, ensuring immediate status updates across all relevant user dashboards.
- The system shall handle peak usage periods (e.g., attendance hours, semester result publication, placement seasons) while maintaining stable performance.
- Database operations related to:
    - SGPA and CGPA calculation
    - Attendance percentage computation
    - Leave, sick leave, and OD status retrieval
    - shall be optimized to ensure fast data access and minimal latency.

- The system shall be scalable to accommodate an increase in the number of students, academic records, and approval requests without requiring architectural redesign.

9.Testing Requirements

The Online Process Automation System shall undergo systematic testing to ensure reliability, correctness, performance, and compliance with institutional requirements.

9.1 Unit Testing

- Each functional module, including authentication, attendance management, leave management, OD approval, and workflow automation, shall be tested independently.
- Business rules such as attendance-based sick leave rejection and role-based approval routing shall be validated at the unit level.

9.2 Integration Testing

- Integration testing shall verify seamless interaction between the frontend, backend services, and the database.
- Multi-level approval workflows involving mentor, parent, hostel warden, and HOD shall be tested end-to-end.

9.3 System Testing

- The complete system shall be tested against all specified functional and non-functional requirements.
- Real-time scenarios such as leave application submission, approval updates, and dashboard refresh shall be validated.

9.4 Performance Testing

- Load testing shall be conducted to evaluate system behavior under concurrent user access.
- Response time, throughput, and system stability shall be measured during peak usage conditions.

9.5 Security Testing

- Authentication and authorization mechanisms shall be tested to ensure only authorized users can access system functionalities.
- Sensitive student and academic data shall be tested for protection against unauthorized access.

9.6 User Acceptance Testing (UAT)

- The system shall be validated by representative end users (students and faculty) to ensure it meets academic and administrative requirements.
- Feedback obtained during UAT shall be incorporated before final deployment

10. Compliance and Quality Attributes

- Auditability
- Traceability
- Data Integrity
- Process Transparency

---

11. Future Enhancements

- Mobile application support
- Notification services
- Analytics dashboards
- ERP integration

---

12. Conclusion

This SRS defines an **enterprise-grade Online Process Automation System** that automates critical academic workflows in a college environment. The system adheres to industry-standard practices, ensuring reliability, scalability, and real-world applicability while aligning with modern corporate software development methodologies.

13.Project Timeline

Phase No.	Phase Name	Activities Covered	Start Date	End Date
Phase 1	Requirement Analysis & Planning	Requirement gathering, problem definition, feasibility analysis	01-03-2026	02-03-2026
Phase 2	System Design	SRS preparation, UML diagrams, architecture design	03-03-2026	05-03-2026
Phase 3	Database Design	ER diagram, table design, schema creation	06-03-2026	07-03-2026

Phase No.	Phase Name	Activities Covered	Start Date	End Date
Phase 4	Backend Development	Java (Spring Boot), business logic, workflow automation	08-03-2026	11-03-2026
Phase 5	Frontend Development	UI design, dashboards, form integration	12-03-2026	13-03-2026
Phase 6	Testing & Validation	Unit testing, integration testing, bug fixing	14-03-2026	14-03-2026
Phase 7	Documentation & Final Review	Report finalization, review, submission preparation	15-03-2026	15-03-2026

14.Revision

Version	Description	Key Features
Initial Version (v1.0)	Basic online automation of college processes	Student profile management, attendance tracking, simple leave application, role-based login, manual approval flow
Updated Version (v2.0)	Enhanced enterprise-level process automation system	Rule-based leave & sick leave automation, attendance-based auto rejection, multi-level OD approval, real-time dashboard (SGPA, CGPA, attendance, approvals), parent & warden integration, workflow engine

15.Signature

Project Owner: