ERP System Development SRS

Software Requirements Specification (SRS)

Project: ERP System Development for Academic, Marketing & Finance, and Administration & Human Resource Modules

---

1. Introduction

1.1 Objective

Design, document, and implement a basic ERP system integrating three core modules: Academic, Marketing & Finance, and Administration & Human Resources. The system will support collaborative work, provide a comprehensive SRS, and implement key functionalities for each module.

1.2 Scope

The ERP system will provide a unified platform for managing academic, financial, marketing, administrative, and human resource operations within an organization. The system will ensure secure access, data integrity, and seamless communication between modules.

---

2. Overall Description

2.1 User Classes and Characteristics

- **Admin:** Full access to all modules and system settings.
- **Student:** Access to academic records, dashboards, and relevant notifications.
- **Staff:** Access to assigned modules (e.g., instructors to academic, HR staff to administration).

2.2 Assumptions and Dependencies

- The system will rely on the availability of a relational database and secure authentication mechanisms.
- Third-party API integrations may be required for future expansion.
- System performance depends on the underlying infrastructure and network reliability.

---

3. System Features and Models

3.1 General Features (Common to All Modules)

3.1.1 User Authentication Model

- **Functional Requirements:**
  - Role-based access control (Admin, Student, Staff).
  - Secure login with hashed passwords.
  - Session management and logout functionality.
- **Non-Functional Requirements:**
  - Authentication must comply with security best practices.
  - Sessions should expire after a period of inactivity.
- **Acceptance Criteria:**
  - Only authorized users can access their permitted modules.
  - Passwords are never stored in plain text.

3.1.2 Database Management Model

- **Functional Requirements:**
  - Use of a relational database.
  - Data normalization and integrity enforcement.
  - Backup and recovery features.
- **Non-Functional Requirements:**
  - Database must support ACID transactions.
  - Regular automated backups.
- **Acceptance Criteria:**
  - Data remains consistent and recoverable after failures.

3.1.3 API Integration Model

- **Functional Requirements:**
  - RESTful APIs for inter-module communication.
  - API documentation for third-party integrations.
- **Non-Functional Requirements:**
  - APIs must be secure and follow standard conventions.
- **Acceptance Criteria:**
  - Modules communicate seamlessly via APIs.
  - APIs are documented and accessible.

3.2 Academic Module

3.2.1 Course and Program Management Model

- **Functional Requirements:**
  - Add, modify, and delete courses and programs.
- **Acceptance Criteria:**
  - Admins/instructors can manage course offerings.

3.2.2 Student Performance Tracking Model

- **Functional Requirements:**
  - Record and track grades and attendance.
- **Acceptance Criteria:**
  - Students and instructors can view performance data.

3.2.3 Examination Management Model

- **Functional Requirements:**
  - Schedule exams and publish results.
- **Acceptance Criteria:**
  - Students receive timely notifications of results.

3.2.4 Reporting and Dashboard Model

- **Functional Requirements:**
  - CRUD operations for academic records.
  - Dynamic dashboards for students and instructors.
  - Generate reports (grade transcripts, attendance summaries).
- **Acceptance Criteria:**
  - Users can generate and view relevant reports.

3.3 Marketing & Finance Module

3.3.1 Tuition Fee Management Model

- **Functional Requirements:**
  - Invoice generation and payment tracking.

- **Acceptance Criteria:**
  - Students can view and pay fees; receipts are generated.

3.3.2 Expense and Financial Reporting Model

- **Functional Requirements:**
  - Track expenses and generate financial reports.
- **Acceptance Criteria:**
  - Admins can view monthly financial summaries.

3.3.3 Marketing Campaign Tracking Model

- **Functional Requirements:**
  - Track leads, conversions, and ROI.
- **Acceptance Criteria:**
  - Marketing staff can analyze campaign effectiveness.

3.3.4 Analytics Dashboard Model

- **Functional Requirements:**
  - Analytics dashboard for campaign and financial metrics.
- **Acceptance Criteria:**
  - Users can access visual summaries of key metrics.

3.4 Administration & Human Resource Module

3.4.1 Employee Management Model

- **Functional Requirements:**
  - Recruitment, payroll, and attendance management.
- **Acceptance Criteria:**
  - HR staff can manage employee records and payroll.

3.4.2 Leave and Performance Tracking Model

- **Functional Requirements:**
  - Track staff leave and performance.
- **Acceptance Criteria:**
  - Staff can request leave; managers can approve/reject.

3.4.3 Asset Management Model

- **Functional Requirements:**
  - Manage office inventory and assets.
- **Acceptance Criteria:**
  - Admins can track and update asset records.

3.4.4 HR Dashboard and Notifications Model

- **Functional Requirements:**
  - HR dashboard with leave status and performance analytics.
  - Notifications for critical tasks (e.g., leave approval).
- **Acceptance Criteria:**
  - Users receive timely notifications for HR actions.

---

4. Non-Functional Requirements

- **Performance:** System should respond to user requests within 2 seconds under normal load.
- **Security:** All sensitive data must be encrypted in transit and at rest.
- **Usability:** User interfaces must be intuitive and accessible.
- **Reliability:** System uptime should be at least 99%.
- **Scalability:** System must support growth in users and data volume.

---

5. System Architecture and Design Constraints

- The system must use a relational database.
- Must support RESTful API communication between modules.
- Must enforce role-based access control.
- Must provide backup and recovery mechanisms.
- Design diagrams (use case, class, component) should be included in the technical documentation.

---

6. User Interface Requirements

Clear navigation flows for each user role.

- Screen layouts for dashboards, forms, and reports.
- Consistent design across modules.

---

7. Data Requirements

- Data storage for users, courses, financial records, employees, assets, etc.
- Data formats must be standardized across modules.
- Data flow between modules via APIs.

---

8. External Interface Requirements

- APIs for integration with third-party systems (e.g., payment gateways, external reporting tools).
- Support for standard data exchange formats (e.g., JSON, CSV).

---

9. Acceptance Criteria

- All functional and non-functional requirements are met.
- Each module delivers the specified features and reports.
- System passes user acceptance testing with stakeholders.

End of SRS Document

Merged and Expanded SRS (Standard Structure)

Software Requirements Specification (SRS)

Project: ERP System Development for Academic, Marketing & Finance, and Administration & Human Resource Modules

---

1. Introduction

1.1 Objective / Purpose

Design, document, and implement a basic ERP system integrating three core modules: Academic, Marketing & Finance, and Administration & Human Resources. The system will support collaborative work, provide a comprehensive SRS, and implement key functionalities for each module.

This SRS document specifies the requirements for the development of an ERP system integrating Academic, Marketing & Finance, and Administration & Human Resource modules. The system supports role-based user authentication, secure data handling, and facilitates management and operational functionalities for an academic institution.

1.2 Intended Audience and Stakeholders

- Software Engineering Team
- Project Manager
- Academic and Administrative Staff
- Marketing and Finance Personnel
- End Users (Admins, Students, Staff)

1.3 Scope

The ERP system will provide a unified platform for managing academic, financial, marketing, administrative, and human resource operations within an organization. The system will ensure secure access, data integrity, and seamless communication between modules.

The ERP system will cover three core modules:

- Academic Module for course management and student tracking
- Marketing & Finance Module for fee management and campaign analytics
- Administration & Human Resource Module for employee and asset management

Common functionalities include secure login, role-based access control, API integration, and database management.

1.4 Definitions, Acronyms, and Abbreviations

- ERP: Enterprise Resource Planning
- SRS: Software Requirements Specification
- API: Application Programming Interface
- CRUD: Create, Read, Update, Delete
- ROI: Return on Investment
- HR: Human Resources

---

2. Overall Description

2.1 Product Perspective / User Classes and Characteristics

The ERP system is a standalone solution integrating various institutional functions to improve data consistency, user experience, and operational efficiency.

User Classes:

- **Admin:** Full access to all modules and system settings.
- **Student:** Access to academic records, dashboards, and relevant notifications.
- **Staff:** Access to assigned modules (e.g., instructors to academic, HR staff to administration).

2.2 User Needs

- Secure and role-based access to the system
- Ability to manage academic courses, student records, financial transactions, marketing campaigns, and HR functions
- Data reporting and dashboard views for monitoring and decision-making

2.3 Assumptions and Dependencies

- The system will rely on the availability of a relational database and secure authentication mechanisms.
- Third-party API integrations may be required for future expansion.
- System performance depends on the underlying infrastructure and network reliability.
- Users will access the system via authenticated web clients

  Relational database system such as MySQL or PostgreSQL will be used

- Secure hashing algorithms will be applied for password management
- RESTful APIs will facilitate internal and external communication

---

3. System Features and Requirements

3.1 General Features (Common to All Modules)

3.1.1 User Authentication Model / System

- Role-based access control (Admin, Student, Staff)
- Secure login with hashed passwords
- Session management and logout functionality
- Authentication must comply with security best practices
- Sessions should expire after a period of inactivity
- Only authorized users can access their permitted modules
- Passwords are never stored in plain text

3.1.2 Database Management Model

- Use of a relational database (MySQL/PostgreSQL)
- Data normalization and integrity enforcement
- Backup and recovery features
- Database must support ACID transactions
- Regular automated backups
- Data remains consistent and recoverable after failures

3.1.3 API Integration Model

- RESTful APIs for inter-module communication
- API documentation for third-party integrations
- APIs must be secure and follow standard conventions
- Modules communicate seamlessly via APIs
- APIs are documented and accessible

3.2 Academic Module

- Course and program management (Add, modify, delete courses)

- Student performance tracking (grades, attendance)
- Examination scheduling and results publication
- CRUD operations for academic records
- Dynamic dashboards for students and instructors
- Generate reports (grade transcripts, attendance summaries)
- Students and instructors can view performance data
- Students receive timely notifications of results

3.3 Marketing & Finance Module

- Tuition fee management (invoice generation, payment tracking)
- Expense tracking and financial reporting
- Marketing campaign tracking (leads, conversions, ROI analysis)
- Analytics dashboard for campaign and financial metrics
- Students can view and pay fees; receipts are generated
- Admins can view monthly financial summaries
- Marketing staff can analyze campaign effectiveness
- Users can access visual summaries of key metrics

3.4 Administration & Human Resource Module

- Employee management (recruitment, payroll, attendance)
- Leave and performance tracking for staff
- Asset management (e.g., office inventory)
- HR dashboard with leave status and performance analytics
- Notifications for critical tasks (e.g., leave approval)
- HR staff can manage employee records and payroll
- Staff can request leave; managers can approve/reject
- Admins can track and update asset records
- Users receive timely notifications for HR actions

---

4. Non-Functional Requirements

4.1 Performance

- System should respond to user requests within 2 seconds under normal load
- System shall support concurrent users per role without degradation
- API responses to be within acceptable time limits (e.g., < 2 seconds)

4.2 Security

- All sensitive data must be encrypted in transit and at rest
- Protection against common web vulnerabilities (e.g., SQL injection, XSS)

4.3 Usability

- User interfaces must be intuitive and accessible
- User-friendly interfaces and dashboards tailored to each role
- Responsive design for varying device access

4.4 Reliability & Availability

- System uptime should be at least 99% (target: 99.5%)
- Automated backup and recovery mechanisms

4.5 Maintainability

- Modular design for ease of updates and feature addition
- Well-documented APIs and codebase

4.6 Scalability

- System must support growth in users and data volume

---

5. System Architecture and Design Constraints

- The system must use a relational database
- Must support RESTful API communication between modules
- Must enforce role-based access control
- Must provide backup and recovery mechanisms
- Design diagrams (use case, class, component) should be included in the technical documentation

---

6. User Interface Requirements / External Interface Requirements

6.1 User Interfaces

- Clear navigation flows for each user role
- Authentication screen
- Role-specific dashboards and management screens for each module
- Screen layouts for dashboards, forms, and reports
- Consistent design across modules

6.2 Hardware Interfaces

- Standard web client devices (PC, tablets, smartphones)

6.3 Software Interfaces

- RESTful APIs facilitating communication between modules
- Database interfaces supporting application CRUD operations

6.4 Communications Interfaces

- HTTP/HTTPS protocol for client-server communication

---

7. Data Requirements

- Data storage for users, courses, financial records, employees, assets, etc.
- Data formats must be standardized across modules
- Data flow between modules via APIs

---

8. Other Requirements

8.1 Legal & Regulatory Compliance

- Must comply with data protection regulations relevant to the academic institution location

8.2 Internationalization & Localization

- Support for multiple languages (if required by institution)

8.3 Risks and Mitigation

- Risk: Data loss due to system failure — Mitigation: Regular backups and recovery testing
- Risk: Security breaches — Mitigation: Implement strong authentication and encryption

---

9. External Interface Requirements (continued)

- APIs for integration with third-party systems (e.g., payment gateways, external reporting tools)
- Support for standard data exchange formats (e.g., JSON, CSV)

---

10. Acceptance Criteria

- All functional and non-functional requirements are met
- Each module delivers the specified features and reports
- System passes user acceptance testing with stakeholders

---

11. Appendices

11.1 Glossary

(Include definitions of key terms)

11.2 References

- Software engineering standards applied
- Institutional policy documents
- Large System Development Project PDF

End of Merged SRS Document

This document merges the original SRS and the expanded, standard-structured SRS for the ERP System Development project. All unique details from both have been preserved for completeness.