

# Software Requirements Specification

## 1. Introduction

### 1.1 Purpose

Managing internal workflows, approvals, and hierarchical communication through manual processes is inefficient and error-prone. The Internal Workflow ERP Portal digitizes and centralizes this workflow, allowing employees, group heads, heads, and admins to collaborate in a traceable, secure system. Features include role-based access, application flow, notice upload with auto-translation and summarization, and LDAP-based authentication.

### 1.2 Intended Audience and Reading Suggestions

This document is intended for developers, project supervisors, testers, and organization stakeholders. It outlines both functional and non-functional system requirements.

### 1.3 Project Scope

This ERP portal enables:

- Role-based login (Admin, Head, Group Head, Employee, Staff)
- A structured application forwarding system (Employee -> Group Head -> Head)
- File uploads (notices), with auto Hindi title translation and message summarization
- A dashboard for library access, flash alerts, and notices
- Centralized authentication via LDAP
- System logging and transparent application tracking

## 2. Overall Description

### 2.1 Product Perspective

The portal replaces internal paper-based or email-based communication. Every application follows a defined route, with timestamps and status logs.

# Software Requirements Specification

## 2.2 User Classes and Characteristics

- Admin: Manages users, books, notices, flash alerts, and staff hierarchies.
- Head: Approves/rejects forwarded applications; can post notices.
- Group Head: Forwards/rejects applications, posts notices.
- Employee: Sends applications to GH, views responses, notices, alerts, library.
- Staff: Limited view/post access depending on Admin settings.

Hierarchy: 1 Head -> 2 Group Heads -> 6 Employees

## 2.3 Product Functions

- LDAP login with JWT sessions
- Dashboard with role-specific access
- Application inbox with tracking
- Hindi title translation for notices
- Summarized content
- Flash alert system
- Library access (admin managed)

## 2.4 Operating Environment

- Frontend: React + Tailwind CSS
- Backend: Node.js + Express
- Database: MongoDB
- Auth: LDAP + JWT
- Works on all OS with a modern browser

## 2.5 Design

## **Software Requirements Specification**

- Modular frontend with protected routing
- Real-time inbox tracking
- Admin dashboard and role hierarchy
- Centralized notice board

### **3. System Features**

#### 3.1 Description and Priority

- LDAP Login: High
- Role-based Routing: High
- Application Flow: High
- Auto Translation: High
- Summarization: Medium
- Flash Alerts: Medium
- Library Management: Medium

#### 3.2 Functional Requirements

- Admin: manage users, books, alerts, notices
- Employees: send apps, view replies
- Group Heads: forward/reject/send apps
- Head: approve/reject final apps
- Shared access to notices, alerts, library
- Notices auto-translated and summarized

### **4. Other Nonfunctional Requirements**

#### 4.1 Performance Requirements

## **Software Requirements Specification**

- Response time < 2s
- Async processing for translation/summarization

### **4.2 Security Requirements**

- LDAP authentication
- JWT for sessions
- Role-based access enforcement

### **4.3 Software Quality Attributes**

- Modular, testable backend
- Responsive frontend
- Logs maintained
- Easy maintenance and extension

### **4.4 Business Rules**

- One Head -> 2 GH -> 6 Employees enforced
- Strict workflow routing
- Admin-only user control
- Notices processed with translation/summarization

## **5. Other Requirements**

- Support for future modules like attendance, leave
- Built-in audit trail
- Admin analytics dashboard