

# **Software Requirements Specification**

for

## **Previous Year Question Paper Management System**

**Version 1.0 approved**

Prepared by Sneha Verma, Urvashi, Faiqa

Amity University Punjab

February 13, 2026

# Table of Contents

## 1. Introduction

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions and Abbreviations
- 1.4 Overview
- 1.5 References

## 2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 Assumptions and Dependencies

## 3. Functional Requirements

- 3.1 Functional Requirements for Registration and Authentication
- 3.2 Functional Requirements for User Access Screen
- 3.3 Functional Requirements for Admin Access Screen
- 3.4 Functional Requirements for PYQ Management Module (Editor – Admin Access)
- 3.5 Functional Requirements for Search and Retrieval System
- 3.6 Functional Requirements for Document Viewer
- 3.7 Functional Requirements for Analytics and Reporting
- 3.8 Functional Requirements for Notification System
- 3.9 Functional Requirements for Security
- 3.10 Functional Requirements for API and Integration

## 4. Non-Functional Requirements

- 4.1 Performance Requirements
- 4.2 Security Requirements
- 4.3 Reliability Requirements
- 4.4 Maintainability Requirements
- 4.5 Scalability Requirements
- 4.6 Legal and Compliance Requirements

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the requirements for the **Previous Year Question Paper Management System (PYQPMS)**.

This document specifies the functional and non-functional requirements of the system and serves as a reference for developers, project managers, testers and stakeholders involved in the design, development, deployment and maintenance of the website.

The SRS aims to ensure a clear understanding of the system's functionality and constraints so that the final product meets user needs and is managed efficiently by a trusted organization.

## 1.2 Scope

The Previous Year Question Paper Management System is a web-based application designed to provide centralized access to previous year question papers for various academic courses across Amity University Punjab.

The system will:

- Allow users (students and educators) to view and download previous year question papers.
- Categorize the question papers based on course, subject, year, semester and examination type.
- Be managed and moderated by a trusted organization to ensure authenticity, accuracy and reliability of content for uploading, updating and managing question papers.

The application aims to reduce the difficulty students face in locating reliable question papers and to promote equal access to academic resources.

## 1.3 Definitions and Abbreviations

Term	Definition
SRS	Software Requirements Specification
PYQPMS	Previous Year Question Paper Management System
User	Any individual accessing the system to view or download question papers

<b>Term</b>	<b>Definition</b>
Course	An academic program or subject offered by an institution
Trusted Organization	The authorized body responsible for validating and managing the system
Web Application	A software application accessed through a web browser

## 1.4 Overview

This document is organized to provide a comprehensive description of the system requirements.

The subsequent sections of the SRS include:

- **Overall Description:** Describes the system's context, user characteristics, constraints, and assumptions.
- **Specific Requirements:** Details the functional and non-functional requirements of the system.
- **External Interface Requirements:** Specifies user interfaces, hardware interfaces, and software interfaces.
- **System Features:** Describes key features and their requirements in detail.

This structure follows the IEEE recommended format for Software Requirements Specification documentation.

## 1.5 References

1. IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.
2. IEEE Std 29148-2018, *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*.
3. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill.
4. Sommerville, I., *Software Engineering*, Pearson Education.

## **2. Overall Description**

### **2.1 Product Perspective**

The Previous Year Question Paper Management System (PYQPMS) is a new, standalone web-based application designed to centralize and digitize access to previous year question papers of all the courses of the University. The system is not a modification of any existing software but may integrate with external services such as authentication systems, cloud storage or institutional databases if required in the future.

The application follows a client–server architecture where users access the system through a web browser and the backend manages data storage, retrieval and administrative controls. The system is managed by a trusted organization responsible for validating content and maintaining system integrity.

### **2.2 Product Functions**

The major functions of the system include, but are not limited to, the following:

- Display previous year question papers categorized by course, subject, semester, year and examination type.
- Provide search and filter functionality to allow users to easily locate specific question papers.
- Enable users to view and download question papers in supported formats (e.g., PDF).
- Allow administrators to upload, update, verify and remove question papers.
- Maintain role-based access control for administrators and general users.
- Ensure authenticity and reliability of question papers through organizational moderation.
- Maintain logs and records of uploaded and modified content.

### **2.3 User Classes and Characteristics**

The system identifies the following user classes:

#### **Students**

- Primary users of the system.
- Access question papers for exam preparation.
- Require a simple, intuitive and responsive user interface.

#### **Administrators**

- Authorized personnel from the trusted organization.
- Responsible for uploading, validating, updating and managing question papers.
- Require secure login, content management tools and system monitoring features.
- Possess technical proficiency to manage system operations.

## 2.4 Operating Environment

The system will operate in the following environment:

**Client Side:** Web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge or Safari and supported on desktops, laptops, tablets and mobile devices.

**Server Side:** Web server hosted on a cloud server. Database server for storing question papers and metadata.

**Software Environment:** Operating systems such as Windows, Linux or macOS for servers. Standard web technologies (HTML, CSS, JavaScript) and backend frameworks.

## 2.5 Design and Implementation Constraints

The system is subject to the following constraints:

- The application must comply with standard web security practices to protect data integrity.
- Only authorized administrators from the trusted organization are allowed to manage content.
- Question papers must be stored in commonly supported digital formats.
- The system must support scalability to accommodate an increasing number of users and documents.
- Legal and copyright restrictions related to academic question papers must be considered.

---

## 2.6 Assumptions and Dependencies

### Assumptions

- Users have access to the internet and a compatible web browser.
- The trusted organization provides accurate and verified question papers.
- Users possess basic knowledge of navigating web-based applications.

- Institutions permit the use and distribution of previous year question papers.

### Dependencies

- Availability and reliability of web hosting and database services.
- Dependence on third-party libraries or frameworks for web development.
- Continuous support and maintenance by the managing organization.
- System performance depends on network connectivity and server uptime.

## 3. Functional requirements

<b>Purpose</b>	A description of the functional requirements and its reasons
<b>Inputs</b>	What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input.
<b>Processing</b>	Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations
<b>Outputs</b>	The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error message produced as output.

### Template for functional requirements

#### 3.1. Functional Requirements for Registration and Authentication

Purpose	This feature ensures secure user registration and authentication to allow authorized access to the system based on assigned roles.
Inputs	User details such as name, university email ID, course, password and role entered through web-based forms. Login credentials include registered email ID and password.
Processing	The system validates input formats, verifies user credentials against stored records, encrypts passwords, and enforces role-based access control.
Outputs	Successful registration or login confirmation with redirection to the respective dashboard or error messages such as “Invalid Credentials”.

### 3.2. Functional Requirements for User access screen

Purpose	This module allows students to access, browse, search and download previous year question papers.
Inputs	Authenticated user session along with selected filters such as course, subject, semester, year and examination type entered through the interface.
Processing	The system verifies session validity, retrieves relevant records from the database and applies selected filters and search criteria.
Outputs	Display of categorized question papers with metadata and download options or a “No Results Found” message if no matching records exist.

### 3.3. Functional Requirements for Admin access screen

Purpose	This feature provides authorized administrators with access to manage system content and monitor activities.
Inputs	Administrator login credentials and management actions such as upload, edit, delete or verify.
Processing	The system validates administrative privileges, executes approved actions and records activities in system logs.
Outputs	Admin dashboard interface with confirmation of successful operations or appropriate error messages for unauthorized access.

### 3.4. Functional Requirements for PYQ management module Editor (access for admin)

Purpose	This module enables administrators to upload, update, verify and remove question papers in a controlled environment.
Inputs	Question paper files in approved formats (e.g., PDF) along with metadata including course, subject, semester, year and examination type.
Processing	The system validates file format and size, checks for duplication or corruption and securely stores the file and associated metadata in the database.
Outputs	Confirmation message for successful upload or modification, updated document listings, or error messages for invalid submissions.

### 3.5. Functional Requirements for Search and retrieval system

Purpose	This feature enables users to efficiently locate specific question papers using search and filter options.
Inputs	Search keywords and selected filtering parameters entered through the search interface.
Processing	The system performs database queries based on input criteria, applies filtering and sorting mechanisms and retrieves relevant records.
Outputs	Display of matching question papers with preview and download options or a notification if no matching results are found.

### 3.6. Functional Requirements for Document Viewer

Purpose	This feature provides an integrated environment for users to preview exam papers online without the need to download files to a local device.
Inputs	This feature provides an integrated environment for users to preview exam papers online without the need to download files to a local device.
Processing	The system retrieves the PDF file from the cloud storage; the viewer renders the content in the browser; validity checks are performed to ensure the file is not corrupted.
Outputs	Output consists of a high-resolution rendering of the question paper; provides error messages if the file fails to load due to network or server issues.

### 3.7. Functional Requirements for Analytics and Reporting

Purpose	This module provides administrators with a statistical overview of system performance, including which subjects or years are most frequently accessed.
Inputs	Admin selects parameters such as "Academic Year" or "Department"; raw click data and download timestamps are pulled from the database.
Processing	The system aggregates total hits per document; identifies the top 10 most popular papers; calculates upload trends over the last TBD months.
Outputs	Output is a graphical dashboard containing bar charts or pie charts; provides a downloadable summary report in CSV or PDF format.

### 3.8. Functional Requirements for Notification System

Purpose	This system ensures that users stay updated on the status of their uploads or the availability of newly requested papers.
Inputs	System events such as "New Paper Uploaded" or "Request Fulfilled"; user email address or profile ID.
Processing	The system triggers a message generation engine; identifies the relevant user group; schedules the delivery of the notification.
Outputs	An in-app alert message displayed at the top of the user's screen; an automated email confirmation for successful registrations or uploads.

### 3.9. Functional Requirements for Security

Purpose	This requirement provides data protection by ensuring only verified users can access the system and only admins can modify sensitive data.
Inputs	User ID and access code; role-based tokens (e.g., Student, Staff, or Admin); file type validation during upload.
Processing	The system compares credentials against the central server; validates the session timeout period; rejects files that do not meet the .pdf or .jpg extension requirements.
Outputs	Granting access to specific dashboard menus based on user role; "Access Denied" or "Invalid Password" error messages.

### 3.10. Functional Requirements for API and Integration

Purpose	This allows the PYQPMS to integrate with external systems, such as institutional Google accounts for Single Sign-On (SSO).
Inputs	Token requests from external authentication services; unique document IDs requested by external departmental links.
Processing	The system validates the external token; maps the external user identity to the local database; generates a secure, time-limited link for document sharing.
Outputs	Successful login redirection to the home screen; generation of a "Share" URL that can be copied to the clipboard for student use.

## **4. Non-Functional Requirements**

### **4.1 Performance Requirements**

- The system shall load the homepage within 3 seconds under normal network conditions.
- Search results shall be displayed within 2–4 seconds for typical queries.
- The system shall support at least 500 concurrent users without performance degradation.
- File downloads (e.g., PDF question papers) shall begin within 5 seconds of request.
- The system shall be optimized for minimal server response time and efficient database queries.

### **4.2 Security Requirements**

- The system shall implement secure user authentication and authorization mechanisms.
- Role-based access control (RBAC) shall restrict administrative functions to authorized personnel only.
- All sensitive data transmissions shall use HTTPS (SSL/TLS encryption).
- The system shall prevent unauthorized uploads, deletions or modifications of question papers.
- The system shall protect against common web vulnerabilities such as SQL injection, cross-site scripting (XSS) and cross-site request forgery (CSRF).
- Regular data backups shall be performed to prevent data loss.

### **4.3 Reliability Requirements**

- The system shall ensure 99% uptime excluding scheduled maintenance.
- Automatic error handling mechanisms shall prevent system crashes.
- The system shall maintain data integrity during uploads, downloads and updates.
- Backup and recovery mechanisms shall allow restoration of data within a defined recovery time (e.g., 24 hours).

### **4.4 Maintainability Requirements**

- The system shall be modular in design to allow future enhancements.
- Source code shall follow standardized coding conventions and documentation practices.
- The system shall allow easy updates of question papers and system features.
- Logs shall be maintained for monitoring system activities and troubleshooting.

#### **4.5 Scalability Requirements**

- The system shall be scalable to accommodate increasing numbers of users and uploaded question papers.
- The architecture shall support future integration with external systems such as university authentication systems or cloud storage services.

#### **4.6 Legal and Compliance Requirements**

- The system shall ensure that uploaded question papers comply with university copyright and data policies.
- User data shall be handled according to applicable data protection regulations.