
Software Requirements Specification

for

DOCTORAL STUDENT'S PROGRESS TRACKER

Version 1.0

Prepared by

Group Name: AN-10

Bagu Sai Nischal
Yapala Shivananda
NDSL Karthikeya

B220216CS
B220603CS
B221062CS

bagu_b220216cs@nitc.ac.in
yapala_b220603cs@nitc.ac.in
narayanasetty_b221062cs@nitc.ac.in

Team lead: Yapala Shivananda

Course: CS4096D Software Engineering Lab

Date: 19/02/2025

Contents

CONTENTS	II
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 DOCUMENT CONVENTIONS	2
2 OVERALL DESCRIPTION	3
2.1 PRODUCT OVERVIEW	3
2.2 PRODUCT FUNCTIONALITY	3
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	3
2.4 ASSUMPTIONS AND DEPENDENCIES	4
3 SPECIFIC REQUIREMENTS	5
3.1 EXTERNAL INTERFACE REQUIREMENTS	5
3.2 FUNCTIONAL REQUIREMENTS	5
3.3 USE CASE MODEL	6
4 OTHER NON-FUNCTIONAL REQUIREMENTS	10
4.1 PERFORMANCE REQUIREMENTS	10
4.2 SAFETY AND SECURITY REQUIREMENTS	10
4.3 SOFTWARE QUALITY ATTRIBUTES	10

1 Introduction

1.1 Document Purpose

This document outlines functional/non-functional requirements, design constraints, and interface specifications for the Doctoral Progress Tracker (DPT), a web-based platform to manage PhD student milestones, DC meetings, publications, and SWAYAM course approvals.

1.2 Product Scope

The DPT enables:

- Students to log DC meetings, submit exam requests, track publications, and register for SWAYAM courses.
- Supervisors to approve requests and monitor student progress.
- Coordinators to manage comprehensive exams and system-wide configurations

1.3 Intended Audience and Document Overview

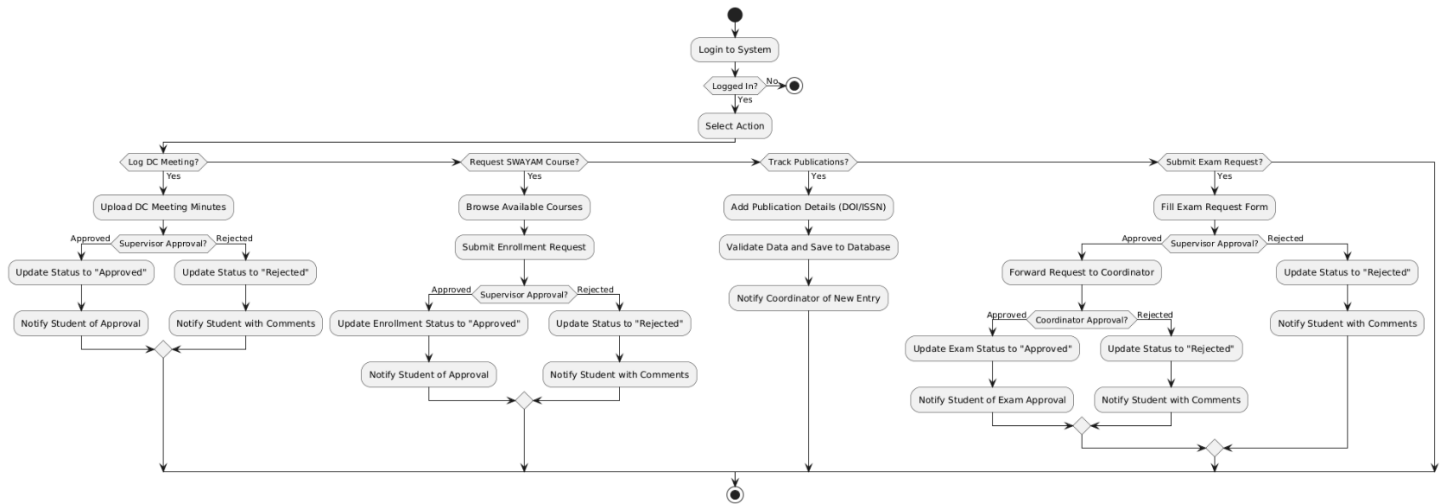
This document is intended for developers, testers, PhD students, faculty supervisors, and academic coordinator and instructors evaluating the project. The document is organized to provide an overview of the system's requirements, its design constraints, functionality, and quality attributes, followed by specific use cases and requirements.

1.4 Document Conventions

This document follows IEEE formatting requirements using Arial font size 11 or 12. Section and subsection titles conform to the template style, with major sections in bold.

2 Overall Description

2.1 Product Overview



2.2 Product Functionality

- **DC Meeting Management:** Log dates, upload minutes, and track approvals.
- **Exam Management:** Submit/approve comprehensive exam requests and results.
- **Publication Tracking:** Maintain a database of research papers linked to student profiles.
- **SWAYAM Registration:** Request/enroll in courses with supervisor approval.
- **Role-Based Dashboards:** Custom interfaces for students, supervisors, and coordinators.

2.3 Design and Implementation Constraints

- **Frontend:** HTML/CSS/JavaScript with Bootstrap for responsive UI.
- **DBMS:** MySQL for structured data storage.
- **Security:** Role-based access control (RBAC) and encrypted credentials.
- **Standards:** COMET design method and UML modeling language will be utilized for design.

2.4 Assumptions and Dependencies

- The system assumes reliable internet connectivity.
- User authentication assumes standard web security protocols.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **Login/Registration:** Role-specific entry (Student/Supervisor/Coordinator).
- **Dashboard:** Summary of pending tasks, upcoming DC meetings, and milestones.
- **DC Meetings:** Form to upload minutes, discussion summaries, and approval status.
- **SWAYAM Courses:** List of available courses with "Request Enrollment" options.

3.1.2 Hardware Interfaces

The system interfaces with PCs, laptops and desktops which support:

- **Processor:** Minimum of Intel core i3 processors.
- **Disk Storage:** 25GB of disk space
- **RAM:** 2GB RAM minimum. 4GB RAM preferred.

3.1.3 Software Interfaces

- **Database:** Store student profiles, DC meeting logs, and publication records.
- **Frontend:** Using HTML, JavaScript and CSS for the user interface.

3.2 Functional Requirements

Student Functions

User Login

- Users shall log in with their unique email and password, which are authenticated against the database.

DC Meeting Logging:

- Students upload meeting minutes (PDF/DOCX).

Exam Request Submission:

- Students attach coursework documents for supervisor approval.

SWAYAM Enrollment:

- Coordinators view/approve course requests from students.

Publication Indexing:

- Auto-fetch journal impact factors via ISSN/DOI lookup.

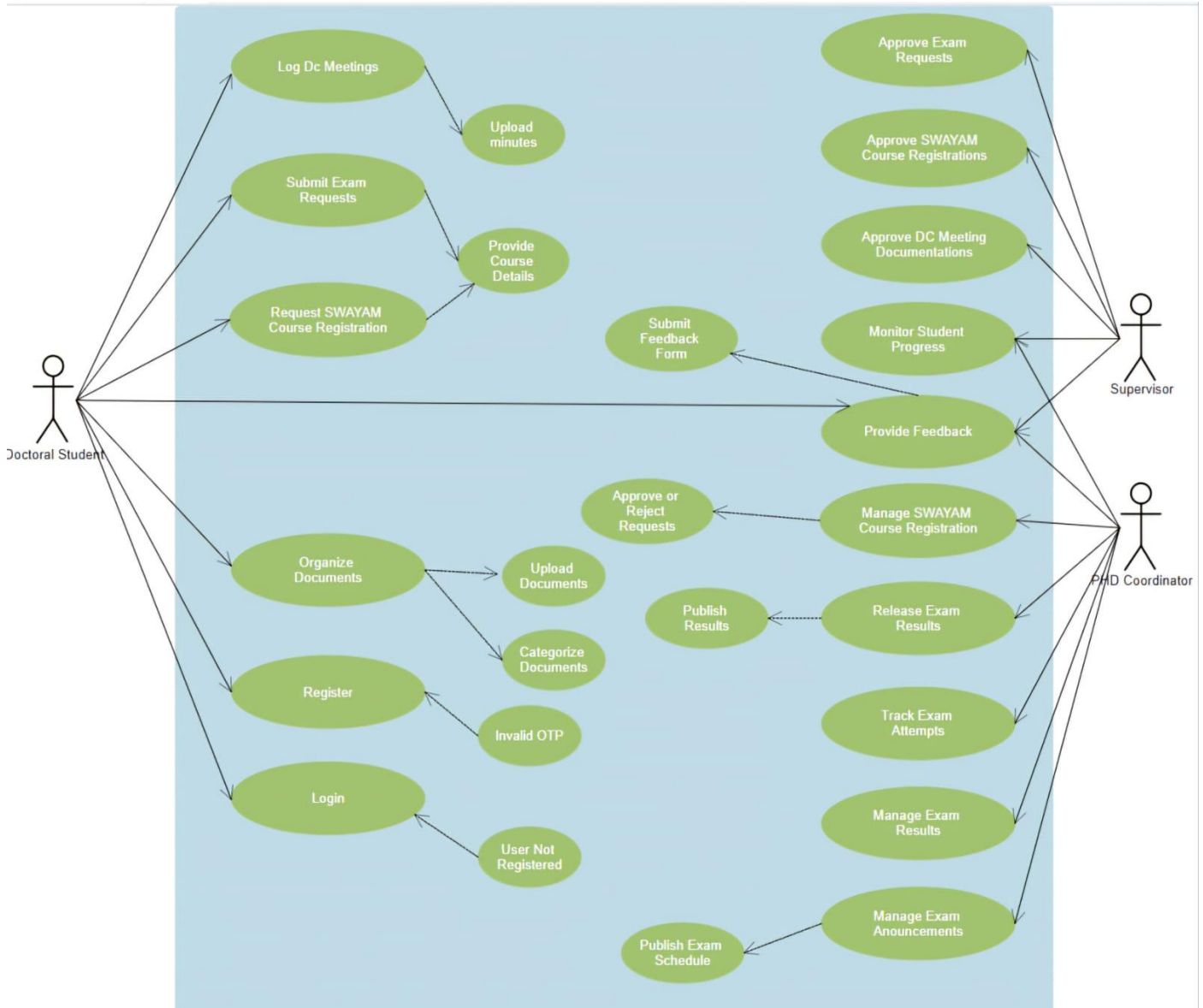
Supervisor Functions

- Approve/reject DC meeting documentation.
- Validate SWAYAM course requests.
- Monitor student progress via visual dashboard.

Coordinator Functions

- Schedule comprehensive exams
- Publish exam results
- Configure academic timelines

3.3 Use Case Model



Actors

1. Doctoral Student: The primary user of the system who performs tasks such as registering, logging in, organizing documents, submitting requests, and providing feedback.
2. Supervisor: A faculty member responsible for monitoring the student's progress, approving requests, and providing feedback.
3. PhD Coordinator: An administrative role responsible for managing course registrations, exam results, and other approvals.

Use Case

Doctoral Student's Tasks

1. Log DC Meetings:
 - The student logs details of Doctoral Committee (DC) meetings.
 - They can also upload meeting minutes for documentation.
2. Submit Exam Requests:
 - Students submit requests to appear for exams.
3. Request SWAYAM Course Registration:
 - Students request registration for courses offered through SWAYAM (an online learning platform).
4. Provide Course Details:
 - Students share details about courses they are taking.
5. Submit Feedback Form:
 - Students provide feedback about courses or the program.
6. Organize Documents:
 - Students upload and categorize documents required for their academic progress.
 - The system handles scenarios like invalid OTPs or unregistered users during login/registration.
7. Register/Login:
 - Students register on the platform and log in to access its features.
 - If an OTP is invalid or the user is not registered, appropriate error handling occurs.

Supervisor's Tasks

1. Approve Exam Requests:
 - Supervisors review and approve requests submitted by students to appear for exams.
2. Approve SWAYAM Course Registrations:
 - Supervisors approve students' requests for SWAYAM course registrations.
3. Approve DC Meeting Documentation:
 - Supervisors review and approve DC meeting minutes uploaded by students.
4. Monitor Student Progress:
 - Supervisors track the academic progress of their assigned students.
5. Provide Feedback:
 - Supervisors give feedback to students based on their progress or submissions.

PhD Coordinator's Tasks

1. Approve or Reject Requests:
 - The coordinator reviews and decides on various student requests (e.g., documents or registrations).
2. Manage SWAYAM Course Registration:
 - The coordinator oversees all SWAYAM course registrations for PhD students.
3. Publish Results:
 - The coordinator publishes exam results after approval.
4. Release Exam Results:
 - Results are officially released by the coordinator after processing.
5. Track Exam Attempts:
 - The coordinator tracks how many times students have attempted specific exams.
6. Manage Exam Results and Announcements:
 - The coordinator manages exam-related announcements and results publication.
7. Publish Exam Schedule:
 - The coordinator publishes schedules for upcoming exams.

4 Other Non-functional Requirements

4.1 Performance Requirements

- **Response Time:** The system shall display shows within 3 - 5 seconds of the user request.
- **Scalability:** The application shall handle up to 250 concurrent users without performance degradation.

4.2 Safety and Security Requirements

- **Data Protection:** User passwords shall be securely hashed and stored into the database. This will be achieved using hashing tool called bcrypt. AES-256 encryption for document uploads
- **Role based Access Control:** Users with different roles will have access to various functions.

4.3 Software Quality Attributes

4.3.1 Reliability

- The system shall provide reliable access to data, with a target uptime of 99.5%.

4.3.2 Usability

- The interface shall be intuitive, with minimal training required for end-users. Consistent navigation and error-handling mechanisms shall be implemented.