

# **CEBU INSTITUTE OF TECHNOLOGY**

## **UNIVERSITY**



**COLLEGE OF COMPUTER STUDIES**

## **Software Requirements Specifications**

*for*

Software Project Management Plan Evaluator

## Change History

Version	Date	Author(s)	Change Description
1.0	2025-09-26	Project Team	Initial draft — outline and TOC
2.0	2025-09-27	Project Team	Added details in Modules
3.0	2025-09-28	Project Team	Added Wireframe in Modules
4.0	2025-09-29	Project Team	Added Non-Functional Requirements

# Table of Contents

<b>Change History</b>	<b>2</b>
<b>Table of Contents</b>	<b>3</b>
<b>1. Introduction</b>	<b>4</b>
1.1. Purpose	4
1.2. Scope	4
1.3. Definitions, Acronyms and Abbreviations	4
1.4. References	4
<b>2. Overall Description</b>	<b>5</b>
2.1. Product perspective	5
2.2. User characteristics	5
2.4. Constraints	5
2.5. Assumptions and dependencies	6
<b>3. Specific Requirements</b>	<b>7</b>
3.1. External interface requirements	7
3.1.1. <i>Hardware interfaces</i>	7
3.1.2. <i>Software interfaces</i>	7
3.1.3. <i>Communications interfaces</i>	7
3.2. Functional requirements	7
<i>Module 1</i>	7
<i>Module 2</i>	
<i>Module 3</i>	
<i>Module 4</i>	8
3.4 Non-functional requirements	8
<i>Performance</i>	8
<i>Security</i>	8
<i>Reliability</i>	8

## 1. Introduction

### 1.1. Purpose

This Software Requirements Specification (SRS) describes the requirements for the **Software Project Management Plan Evaluator** (SPMP Evaluator) a web-based tool designed to automatically evaluate student and professional SPMP documents for structure and format compliance with the **IEEE 1058** standard. The SRS provides the baseline for design, implementation, testing, and validation.

### 1.2. Scope

The **SPMP Evaluator** is a web-based tool designed to automatically analyze, score, and provide structured feedback on Software Project Management Plans (SPMPs) based on IEEE 1058 standards. Its primary purpose is to help students, professors, and project managers ensure compliance, improve document quality, and streamline the evaluation process. The system includes the following core functionalities:

- Authentication and Security
- User-Friendly Web Interface
- Automated Compliance Checking
- Scoring System
- Document Upload and Parsing

### 1.3. Definitions, Acronyms and Abbreviations

- **SPMP** — Software Project Management Plan
- **IEEE 1058** — IEEE standard for Software Project Management Plans
- **Parser** — module that extracts document structure from PDF/DOCX
- **Evaluator** — scoring engine applying rules/weights to parsed output
- **AI** — Artificial Intelligence (used for optional semantic assistance)
- **PM** — Project Manager
- **DB** — Database
- **SRS** — Software Requirements Specification

### 1.4. References

- IEEE Std 1058-1998 (and subsequent revisions) — Software Project Management Plan standard.
- Google Document AI & Google Cloud docs (for proposed parsing integration).
- Apache POI docs (for DOCX parsing).
- PDF.js docs (for PDF extraction).
- Project Title Page and Executive Summary provided by the proponent team (Sept 2025).

## 2. Overall Description

### 2.1. Product perspective

The SPMP Evaluator is a client-server web application. Clients (students, professors, project managers) interact through role-specific web dashboards. The backend performs secure file ingestion, parsing, rules-based and AI-augmented evaluation, scoring, and report generation. Persistent storage holds user data, documents (encrypted), and evaluation logs.

### 2.2. User characteristics

- **Students** — upload SPMPs, view results/feedback, re-upload corrected documents. Low technical expertise assumed.
- **Professors** — can view individual student evaluations, override scores, annotate feedback, download reports. Moderate technical comfort.
- **Project Managers** — aggregate views across teams, benchmark compliance, run batch scoring. Technical comfort varies.
- **System Administrator** — manages users, roles, system settings, and updates rules.

### 2.3. Operating Environment

- **Client:** Modern browsers (Chrome, Firefox, Edge), mobile and desktop responsive.
- **Server:** Hosted on Render (Free Tier) with Node.js backend.
- **Database:** Supabase PostgreSQL (cloud, free tier).
- **File Storage:** Supabase Storage (encrypted at rest).
- **Security:** All communications over HTTPS/TLS.

### 2.4. Constraints

- Initial language support: English only.
- Free-tier cloud services (Supabase, Render, Vercel).
- Must comply with IEEE 1058 structure.
- Budget/time constraints restrict advanced NLP model training; reliance on rule-based checks + third-party parsing/AI APIs.
- Concurrency target: 50–100 simultaneous evaluations without performance degradation.

## **2.5. Assumptions and dependencies**

- Stable internet connection for cloud access.
- Supabase services available within free tier limits.
- Google Document AI and Vertex AI availability (optional).
- Universities provide sample SPMPs for validation.

## 3. Specific Requirements

### 3.1. External interface requirements

#### 3.1.1. Hardware interfaces

- Clients: Any desktop/laptop/smartphone with a modern browser.
- Server: Render-hosted VM with min. 2 vCPU, 2 GB RAM.

#### 3.1.2. Software interfaces

- **Backend:** Springboot and Java (Express).
- **Database:** Supabase PostgreSQ /MySQL
- **File Parsing:** Apache POI for .docx, PDF.js/PDFBox for .pdf, optional Google Document AI.
- **Authentication:** Supabase Auth with JWT-based session tokens.
- **File Storage:** Supabase Storage.

#### 3.1.3. Communications interfaces

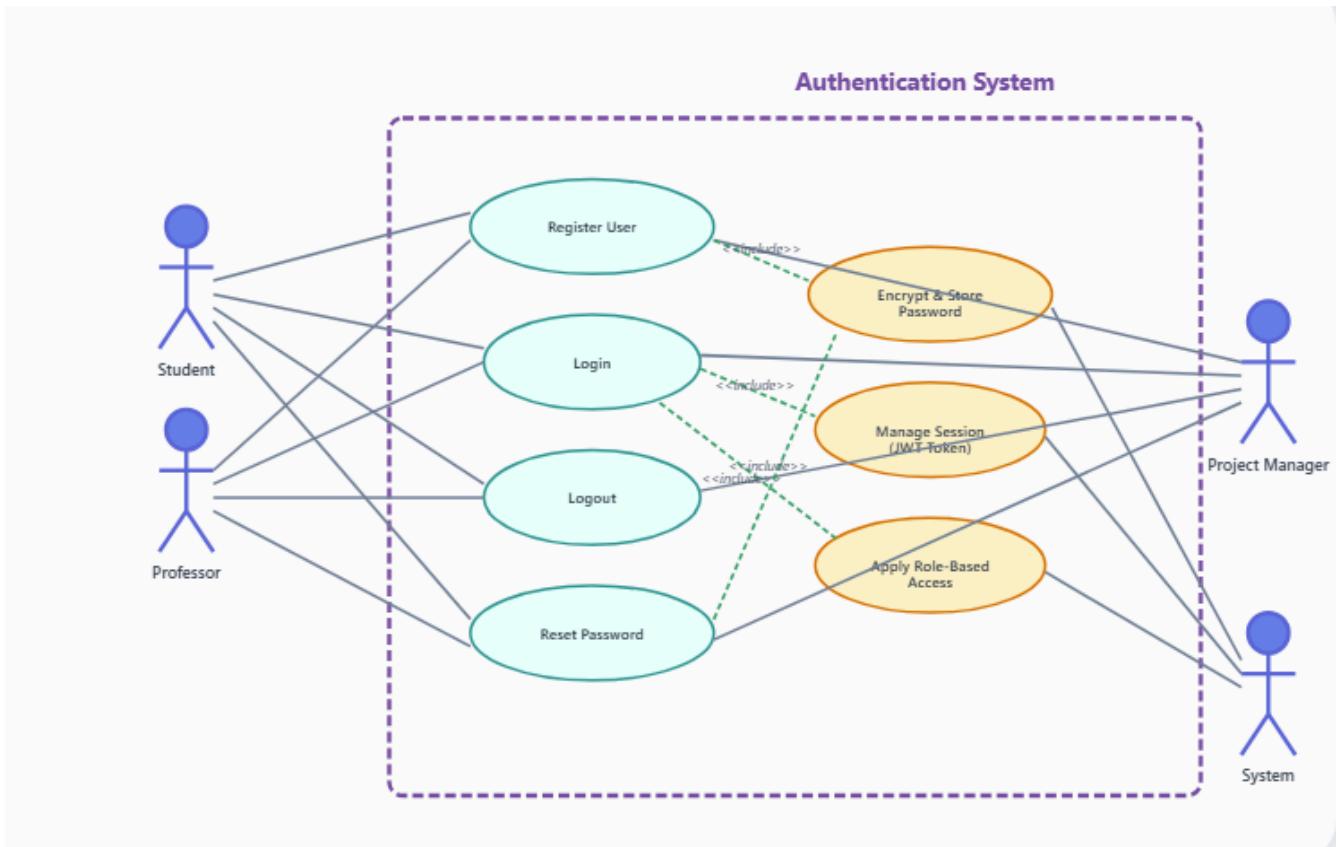
- HTTPS/TLS enforced for all communication.
- REST APIs between frontend and backend.

### 3.2. Functional requirements

#### Module 1

##### 1.1 Basic Login and Security System

- Use Case Diagram



- **Use Case Description**

<b>Use Case 1: Register User</b>		
<b>Name, description</b>	Register User - Allow new users to create an account with unique credentials and assigned role (Student, Professor, or Project Manager). <b>Includes password encryption behavior.</b>	
<b>Actors</b>	Student, Professor, Project Manager, System	
<b>Pre-condition</b>	User has access to registration interface, System is operational, User has valid email address, Database is accessible	
<b>Post-condition</b>	New user account created, Password encrypted and stored securely, User credentials saved in database, User can now login to system	
Main Success Path (primary flow)		
<b>Actor Actions</b>	<b>System Responses</b>	
1. Navigate to registration page 2. Enter required information: - Full name - Email address - Password - Confirm password - Select role (Student/Professor/Manager) 3. Submit registration form	1.1 Display registration form 2.1 Validate input data format 2.2 Check email uniqueness in database <b>2.3 BEHAVIOR: Encrypt password using secure algorithm (bcrypt)</b> <b>2.4 BEHAVIOR: Store encrypted password with salt</b> 2.5 Create new user account with selected role 2.6 Save user credentials to database 2.7 Display success message 2.8 Redirect to login page	
Alternate Path A1	Email Already Exists - User attempts to register with email that's already in system	
<b>Actor Actions</b>	<b>System Responses</b>	
1. Review error message 2. Choose to login or use different email 3. If different email: Re-enter information 4. Resubmit form	1.1 Display "Email already exists" error 2.1 Suggest login option 2.2 Suggest password reset if user forgot 3.1 Clear form for new attempt 4.1 Re-validate new email	
Exception Path E1	Invalid Input Data - Validation fails (weak password, invalid email format, missing required fields)	
<b>Actor Actions</b>	<b>System Responses</b>	
1. Review validation errors 2. Correct invalid information 3. Ensure password meets requirements 4. Resubmit form	1.1 Display specific validation errors 1.2 Highlight problematic fields 1.3 Show password requirements: • At least 8 characters • Include uppercase letter • Include lowercase letter • Include number • Include special character 2.1 Re-validate corrected data	
Scenarios		
Scenario	Post-Conditions	Flow
Successful Student Registration	Student account created with encrypted password, can login immediately	Main Success Path
Duplicate Email Registration	Registration blocked, user directed to login or use different email	Alternate Path A1
Weak Password Attempt	Registration blocked, user prompted to create stronger password	Exception Path E1

## Use Case 2: Login

Name, description	Login - Authenticate user credentials and grant system access. <b>Includes role-based access determination and session management (JWT).</b>
Actors	Student, Professor, Project Manager, System
Pre-condition	User has registered account, System is operational, Database is accessible, User has valid credentials
Post-condition	User authenticated, JWT token generated and stored, Role-based permissions applied, User session established, User redirected to role-appropriate dashboard

Main Success Path (primary flow)

Actor Actions	System Responses
<ol style="list-style-type: none"> <li>1. Navigate to login page</li> <li>2. Enter credentials: <ul style="list-style-type: none"> <li>- Email address</li> <li>- Password</li> </ul> </li> <li>3. (Optional) Check "Remember Me"</li> <li>4. Submit login form</li> </ol>	<ol style="list-style-type: none"> <li>1.1 Display login form</li> <li>2.1 Validate input format</li> <li>2.2 Retrieve user record from database</li> <li>2.3 <b>BEHAVIOR: Verify encrypted password matches</b></li> <li>2.4 <b>BEHAVIOR: Determine user role (Student/Professor/Manager)</b></li> <li>2.5 <b>BEHAVIOR: Generate JWT token with:</b> <ul style="list-style-type: none"> <li>• User ID</li> <li>• Role information</li> <li>• Expiration timestamp</li> <li>• Session ID</li> </ul> </li> <li>2.6 <b>BEHAVIOR: Store session information</b></li> <li>2.7 <b>BEHAVIOR: Apply role-based access controls</b></li> <li>2.8 Set JWT token in secure cookie/header</li> <li>2.9 Log successful login</li> <li>2.10 Redirect to role-specific dashboard</li> </ol>

Alternate Path A1

Invalid Credentials - User enters wrong email or password

Actor Actions	System Responses
<ol style="list-style-type: none"> <li>1. Review error message</li> <li>2. Re-enter correct credentials</li> <li>3. Submit form again</li> <li>4. If password forgotten: Click "Forgot Password"</li> </ol>	<ol style="list-style-type: none"> <li>1.1 Display "Invalid credentials" message</li> <li>1.2 Increment failed login attempt counter</li> <li>1.3 Log failed login attempt</li> <li>1.4 Return to login form</li> <li>1.5 If attempts ≥ 5: Trigger account lockout (Exception E1)</li> </ol>

Exception Path E1

Account Locked - Too many failed login attempts (security measure)

Actor Actions	System Responses
<ol style="list-style-type: none"> <li>1. Read account locked message</li> <li>2. Wait for lockout period (15 minutes)</li> <li>3. Or request password reset</li> <li>4. Attempt login again after cooldown</li> </ol>	<ol style="list-style-type: none"> <li>1.1 Display "Account locked" message</li> <li>1.2 Show lockout duration (15 minutes)</li> <li>1.3 Log security event</li> <li>1.4 Provide password reset option</li> <li>1.5 Automatically unlock after 15 minutes</li> <li>1.6 Reset failed attempt counter</li> </ol>

### Scenarios

Scenario	Post-Conditions	Flow
Successful Professor Login	Professor authenticated, JWT issued, redirected to professor dashboard with appropriate permissions	Main Success Path
Student Wrong Password	Login denied, attempt counted, user prompted to retry	Alternate Path A1
Multiple Failed Attempts	Account locked for 15 minutes, security event logged	Exception Path E1

### Use Case 3: Logout

Name, description	Logout - Terminate user session and clear authentication tokens. <b>Includes session cleanup and token invalidation.</b>
Actors	Student, Professor, Project Manager, System
Pre-condition	User is currently logged in, Active session exists, Valid JWT token present
Post-condition	User session terminated, JWT token invalidated, Session data cleared, User redirected to landing/login page, Logout event logged

Main Success Path (primary flow)

Actor Actions	System Responses
1. Click logout button/menu 2. (Optional) Confirm logout if prompted 3. Acknowledge logout completion	1.1 Receive logout request 1.2 BEHAVIOR: Invalidate JWT token 1.3 BEHAVIOR: Clear session data from server 1.4 BEHAVIOR: Remove token from client (clear cookies/storage) 1.5 BEHAVIOR: Revoke role-based access 1.6 Log logout event with timestamp 1.7 Display logout success message 1.8 Redirect to landing/login page

Alternate Path A1	Session Timeout Logout - Automatic logout due to inactivity
	Actor Actions 1. User inactive for 30 minutes 2. User attempts any action 3. See timeout message 4. Re-authenticate to continue           System Responses 1.1 Detect session expiration 1.2 Automatically invalidate session 1.3 Clear all session data 1.4 Display "Session expired" message 1.5 Redirect to login page 1.6 Log timeout event

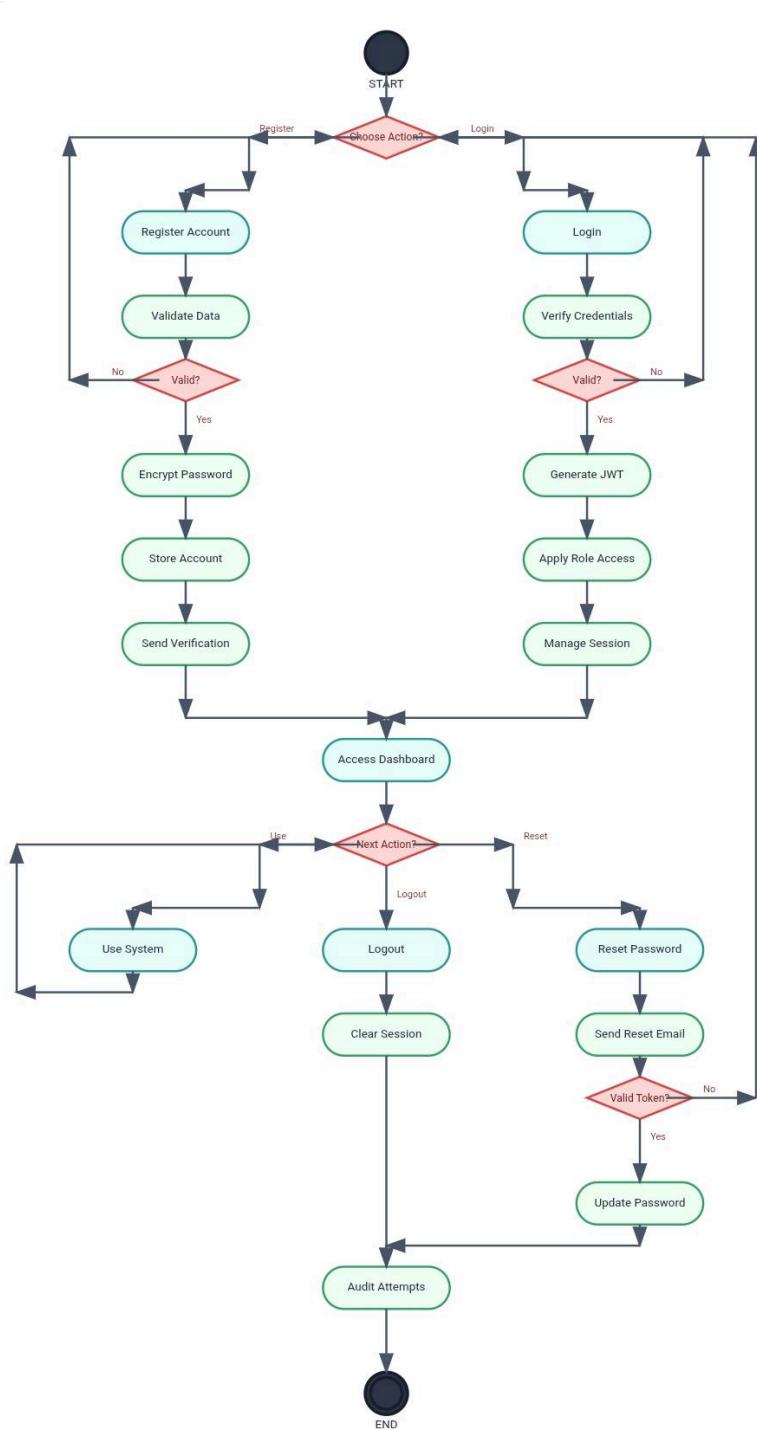
Exception Path E1	Network Failure During Logout - Connection lost while processing logout
	Actor Actions 1. Experience network disconnection 2. Close browser/application 3. Token expires automatically after time limit           System Responses 1.1 Detect network failure 1.2 Queue logout operation 1.3 Clear client-side data immediately 1.4 Session expires on server after timeout period 1.5 Log incomplete logout event

#### Scenarios

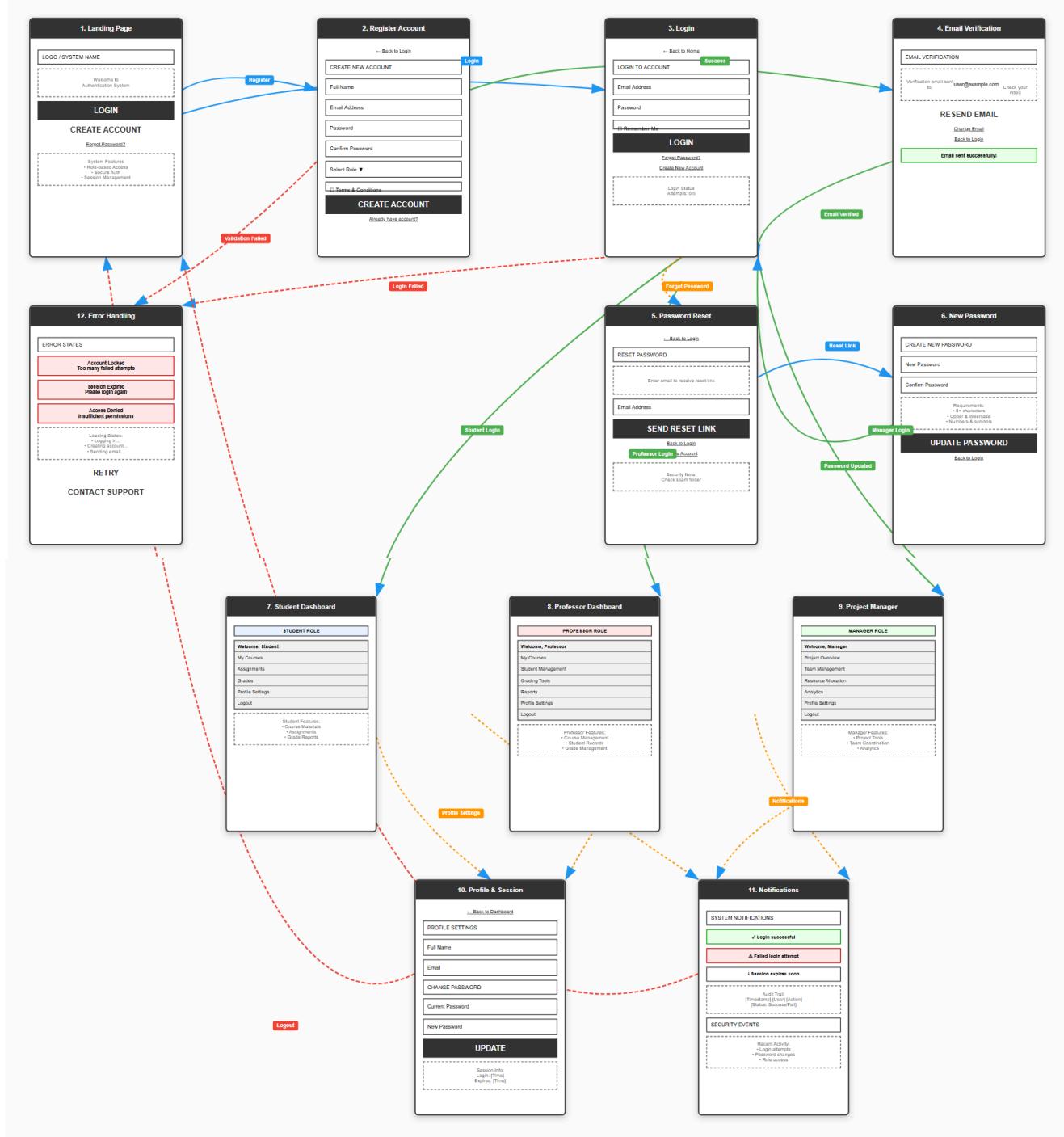
Scenario	Post-Conditions	Flow
Normal User Logout	Session cleanly terminated, user can login again anytime	Main Success Path
Automatic Session Timeout	Session expired due to inactivity, user must re-authenticate	Alternate Path A1
Network Disconnection	Client-side logout completed, server session expires naturally	Exception Path E1

Use Case 4: Reset Password														
Name, description	Reset Password - Allow users to securely reset forgotten password through email verification. <b>Includes password encryption for new password.</b>													
Actors	Student, Professor, Project Manager, System													
Pre-condition	User has registered account, User has access to registered email, Email service is operational, System is operational													
Post-condition	Password reset email sent, Reset token generated and stored, New password encrypted and saved (if completed), User can login with new password													
Main Success Path (primary flow)														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #2e3436; color: white;">Actor Actions</th><th style="background-color: #2e3436; color: white;">System Responses</th></tr> </thead> <tbody> <tr> <td>1. Click "Forgot Password" link 2. Enter registered email address 3. Submit reset request 4. Check email for reset link 5. Click reset link in email 6. Enter new password 7. Confirm new password 8. Submit password change</td><td>1.1 Display password reset request form 2.1 Validate email format 2.2 Check if email exists in database 2.3 Generate secure reset token (expires in 1 hour) 2.4 Store reset token with user record 2.5 Send reset email with secure link 2.6 Display "Check your email" message 3.1 Validate reset token from link 3.2 Check token expiration 3.3 Display new password form 3.4 Validate new password requirements 3.5 <b>BEHAVIOR: Encrypt new password</b> 3.6 <b>BEHAVIOR: Store encrypted password</b> 3.7 Invalidate reset token 3.8 Log password change event 3.9 Display success message 3.10 Redirect to login page</td></tr> </tbody> </table>			Actor Actions	System Responses	1. Click "Forgot Password" link 2. Enter registered email address 3. Submit reset request 4. Check email for reset link 5. Click reset link in email 6. Enter new password 7. Confirm new password 8. Submit password change	1.1 Display password reset request form 2.1 Validate email format 2.2 Check if email exists in database 2.3 Generate secure reset token (expires in 1 hour) 2.4 Store reset token with user record 2.5 Send reset email with secure link 2.6 Display "Check your email" message 3.1 Validate reset token from link 3.2 Check token expiration 3.3 Display new password form 3.4 Validate new password requirements 3.5 <b>BEHAVIOR: Encrypt new password</b> 3.6 <b>BEHAVIOR: Store encrypted password</b> 3.7 Invalidate reset token 3.8 Log password change event 3.9 Display success message 3.10 Redirect to login page								
Actor Actions	System Responses													
1. Click "Forgot Password" link 2. Enter registered email address 3. Submit reset request 4. Check email for reset link 5. Click reset link in email 6. Enter new password 7. Confirm new password 8. Submit password change	1.1 Display password reset request form 2.1 Validate email format 2.2 Check if email exists in database 2.3 Generate secure reset token (expires in 1 hour) 2.4 Store reset token with user record 2.5 Send reset email with secure link 2.6 Display "Check your email" message 3.1 Validate reset token from link 3.2 Check token expiration 3.3 Display new password form 3.4 Validate new password requirements 3.5 <b>BEHAVIOR: Encrypt new password</b> 3.6 <b>BEHAVIOR: Store encrypted password</b> 3.7 Invalidate reset token 3.8 Log password change event 3.9 Display success message 3.10 Redirect to login page													
Alternate Path A1	Email Not Found - User enters email that doesn't exist in system													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #2e3436; color: white;">Actor Actions</th><th style="background-color: #2e3436; color: white;">System Responses</th></tr> </thead> <tbody> <tr> <td>1. See generic success message 2. Wait for email (won't arrive) 3. Verify correct email address 4. Try registration if new user</td><td>1.1 Display generic "Check email" message (security: don't reveal if email exists) 1.2 Log attempted reset for non-existent email 1.3 Do NOT send email 1.4 Protect against email enumeration attack</td></tr> </tbody> </table>			Actor Actions	System Responses	1. See generic success message 2. Wait for email (won't arrive) 3. Verify correct email address 4. Try registration if new user	1.1 Display generic "Check email" message (security: don't reveal if email exists) 1.2 Log attempted reset for non-existent email 1.3 Do NOT send email 1.4 Protect against email enumeration attack								
Actor Actions	System Responses													
1. See generic success message 2. Wait for email (won't arrive) 3. Verify correct email address 4. Try registration if new user	1.1 Display generic "Check email" message (security: don't reveal if email exists) 1.2 Log attempted reset for non-existent email 1.3 Do NOT send email 1.4 Protect against email enumeration attack													
Exception Path E1	Expired or Invalid Reset Token - User clicks old/invalid reset link													
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #2e3436; color: white;">Actor Actions</th><th style="background-color: #2e3436; color: white;">System Responses</th></tr> </thead> <tbody> <tr> <td>1. See token expired message 2. Request new password reset 3. Use new reset link prompt (within 1 hour)</td><td>1.1 Validate reset token 1.2 Check expiration timestamp 1.3 Display "Link expired or invalid" message 1.4 Provide link to request new reset 1.5 Log expired token attempt 1.6 Clear old token from database</td></tr> </tbody> </table>			Actor Actions	System Responses	1. See token expired message 2. Request new password reset 3. Use new reset link prompt (within 1 hour)	1.1 Validate reset token 1.2 Check expiration timestamp 1.3 Display "Link expired or invalid" message 1.4 Provide link to request new reset 1.5 Log expired token attempt 1.6 Clear old token from database								
Actor Actions	System Responses													
1. See token expired message 2. Request new password reset 3. Use new reset link prompt (within 1 hour)	1.1 Validate reset token 1.2 Check expiration timestamp 1.3 Display "Link expired or invalid" message 1.4 Provide link to request new reset 1.5 Log expired token attempt 1.6 Clear old token from database													
<b>Scenarios</b>														
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #2e3436; color: white;">Scenario</th><th style="background-color: #2e3436; color: white;">Post-Conditions</th><th style="background-color: #2e3436; color: white;">Flow</th></tr> </thead> <tbody> <tr> <td>Successful Password Reset</td><td>Password changed and encrypted, user can login with new password</td><td>Main Success Path</td></tr> <tr> <td>Unregistered Email</td><td>No email sent, system doesn't reveal email existence (security)</td><td>Alternate Path A1</td></tr> <tr> <td>Expired Reset Link</td><td>Reset blocked, user must request new link</td><td>Exception Path E1</td></tr> </tbody> </table>			Scenario	Post-Conditions	Flow	Successful Password Reset	Password changed and encrypted, user can login with new password	Main Success Path	Unregistered Email	No email sent, system doesn't reveal email existence (security)	Alternate Path A1	Expired Reset Link	Reset blocked, user must request new link	Exception Path E1
Scenario	Post-Conditions	Flow												
Successful Password Reset	Password changed and encrypted, user can login with new password	Main Success Path												
Unregistered Email	No email sent, system doesn't reveal email existence (security)	Alternate Path A1												
Expired Reset Link	Reset blocked, user must request new link	Exception Path E1												

• *Activity Diagram*



- **Wireframe**

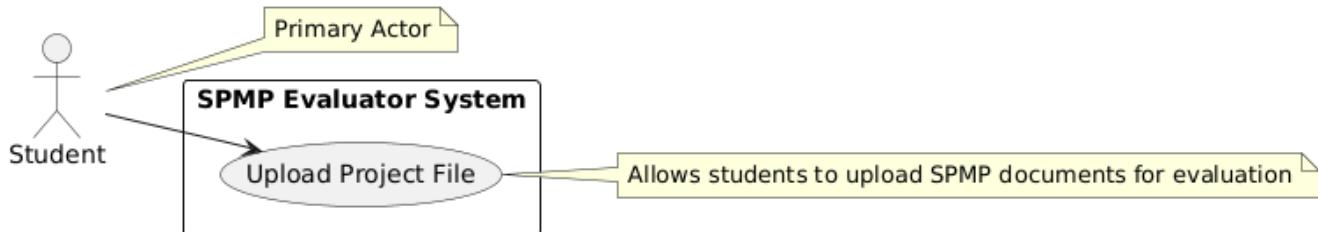


## **Module 2 : Role-Based User Interface Transactions**

---

### **2.1 Student File Upload**

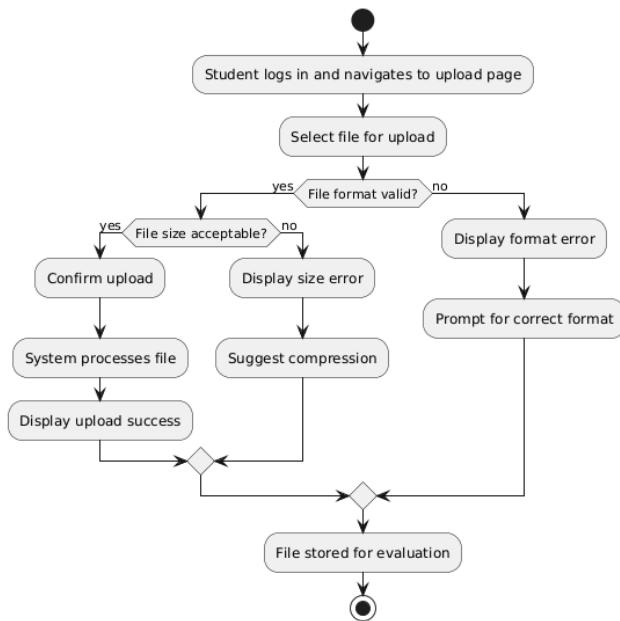
- Use Case Diagram



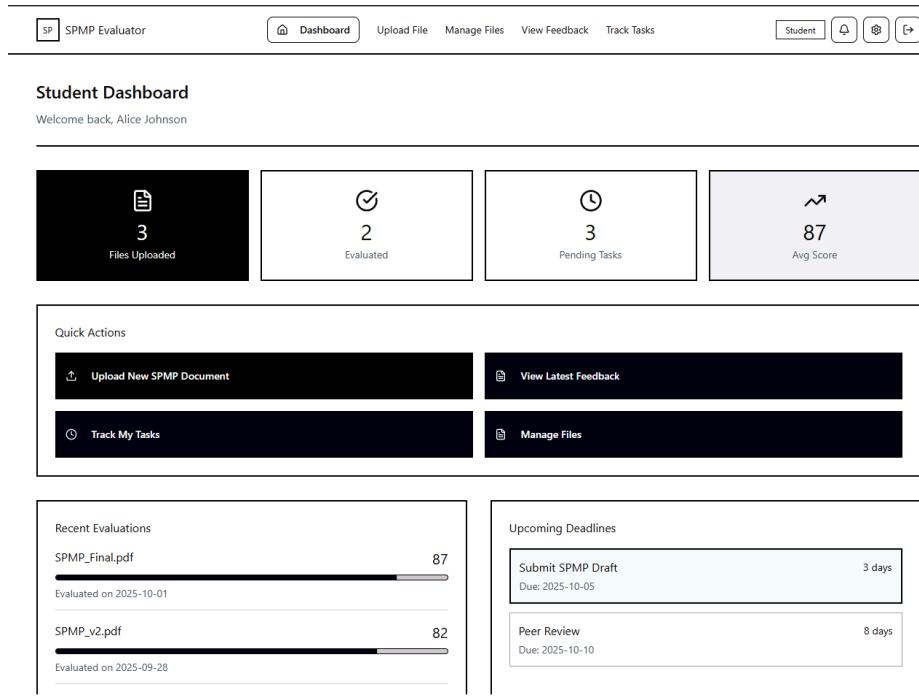
- Use Case Description

Field	Description
Use Case Name	Student File Upload
Primary Actor	Student
Secondary Actors	None
Description	Enables students to upload project-related files (SPMP documents) to the system for evaluation and feedback.
Preconditions	Student is logged in and has a valid SPMP document ready for upload.
Basic Flow	1) Student navigates to the upload section. 2) Student selects the file to upload. 3) System validates file format and size. 4) Student confirms upload. 5) System processes and stores the file. 6) System displays upload confirmation.
Alternative Flows	- Invalid file format: System displays error message and prompts for correct format. - File too large: System rejects upload and suggests compression.
Postconditions	File is stored in the system and available for evaluation.
Exceptions	Network error during upload: System retries or saves draft for later completion.

- *Activity Diagram*



- *Wireframe*



SP SPMP Evaluator

Dashboard Upload File Manage Files View Feedback Track Tasks Student

Home > Dashboard > Upload File

### Upload SPMP Document

Upload your Software Project Management Plan for evaluation

Supported Formats

.PDF .DOCX .DOC .TXT

Maximum file size: 10MB

Drag and drop your file here or [Browse Files](#)

Upload Guidelines

- Ensure your document follows IEEE 1058 standard format
- Include all required sections: Introduction, Organization, Managerial Process, Technical Process, and Supporting Process Plans
- Clear formatting and proper section headers improve AI evaluation accuracy
- You can upload multiple versions for comparison

SP SPMP Evaluator

Dashboard Upload File Manage Files View Feedback Track Tasks Student

Home > Dashboard > Upload File

### Upload SPMP Document

Upload your Software Project Management Plan for evaluation

#### Confirm Upload

Please review the file details before uploading

File Name:	SPMPEvaluator_Proposal_Team02_2025-2026(09-25-2025).pdf
File Size:	288.89 KB
File Type:	application/pdf
Status:	<input checked="" type="checkbox"/> Validated

By uploading this file, you confirm that it contains your SPMP document and is ready for evaluation.

[Confirm and Upload](#) [Cancel](#)

SP SPMP Evaluator

Dashboard Upload File Manage Files View Feedback Track Tasks Student

Home > Dashboard > Upload File

### Upload SPMP Document

Upload your Software Project Management Plan for evaluation

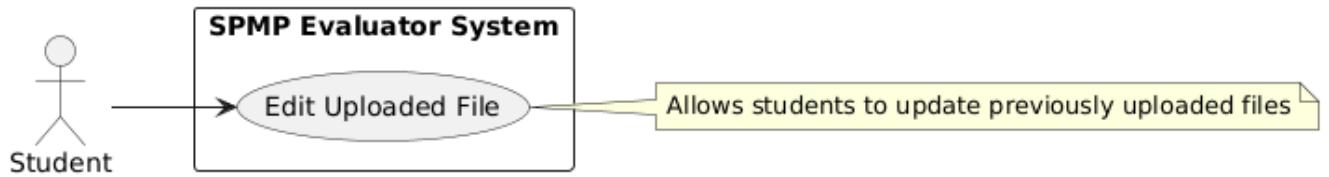
Uploading Document...

Processing: SPMPEvaluator\_Proposal\_Team02\_2025-2026(09-25-2025).pdf 90% complete

Please do not close this window while uploading...

## 2.2 Student File Edit

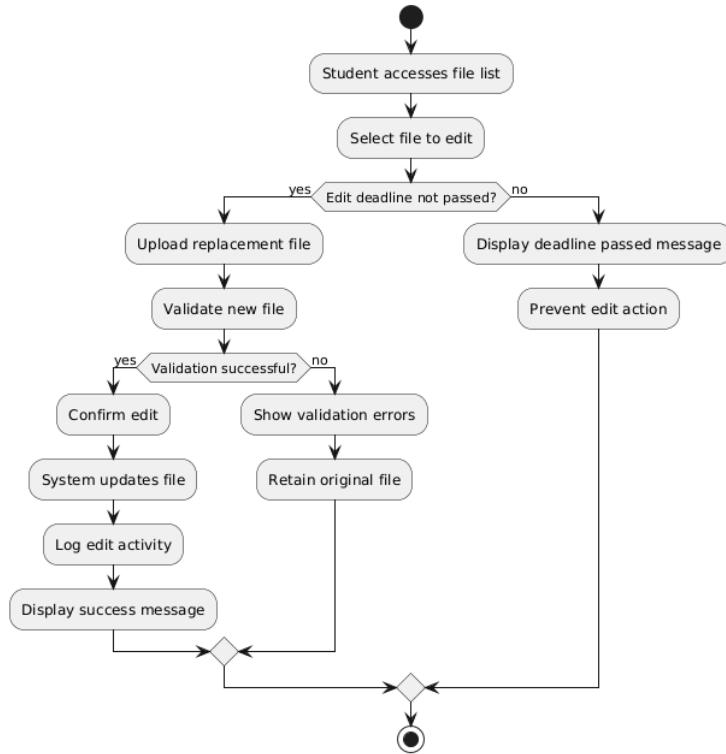
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	Student File Edit
Primary Actor	Student
Secondary Actors	None
Description	Allows students to replace or update previously uploaded project files before evaluation deadlines.
Preconditions	Student is logged in and has previously uploaded a file that can still be edited.
Basic Flow	1) Student views list of uploaded files. 2) Student selects a file to edit. 3) Student uploads replacement file. 4) System validates new file. 5) Student confirms changes. 6) System updates file and logs change.
Alternative Flows	- File past deadline: System prevents edit and shows deadline notice.
Postconditions	Updated file replaces the original and is queued for re-evaluation.
Exceptions	Validation failure: System retains original file and prompts for corrections.

- *Activity Diagram*



- *Wireframe*

The wireframe shows the interface for managing files. At the top, there's a navigation bar with 'SPMP Evaluator' logo, 'Dashboard' (selected), 'Upload File', 'Manage Files' (current page), 'View Feedback', 'Track Tasks', and user icons for 'Student', 'Logout', 'Help', and 'Feedback'. Below the navigation is a breadcrumb trail: 'Home > Dashboard > Manage Files'. On the right side of the header is a 'Upload New' button with an upward arrow icon. The main content area is titled 'Manage Files' and contains a sub-header 'Edit, replace, or remove uploaded documents'. A table lists three files: 'SPMP\_v1.pdf' (Evaluated, 2.4 MB), 'SPMP\_v2.docx' (Processing, 1.8 MB), and 'SPMP\_Final.pdf' (Ready, 3.1 MB). Each row has 'Edit' and 'Delete' icons in the 'Actions' column. Below the table is a 'Replace Existing File' section with a note about preserving evaluation history, a file input field 'Select file to replace...', and a 'Choose File' button.

<input type="checkbox"/>	File Name	Upload Date	Size	Status	Actions
<input type="checkbox"/>	SPMP_v1.pdf	2025-09-28	2.4 MB	Evaluated	
<input type="checkbox"/>	SPMP_v2.docx	2025-09-30	1.8 MB	Processing	
<input type="checkbox"/>	SPMP_Final.pdf	2025-10-01	3.1 MB	Ready	

<input type="checkbox"/>	File Name	Upload Date	Size	Status	Actions
<input type="checkbox"/>	SPMP_v1.pdf	2025-09-28	2.4 MB	Evaluated	
<input type="checkbox"/>	SPMP_v2.docx	2025-09-30	1.8 MB	Processing	
<input type="checkbox"/>	SPMP_Final.pdf	2025-10-01	3.1 MB	Ready	

## 2.3 Student File Removal

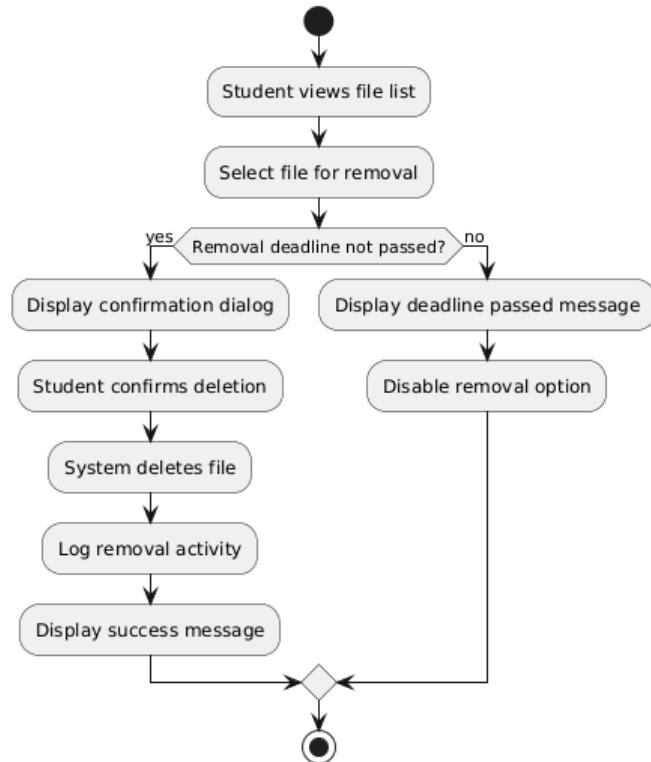
- Use Case Diagram



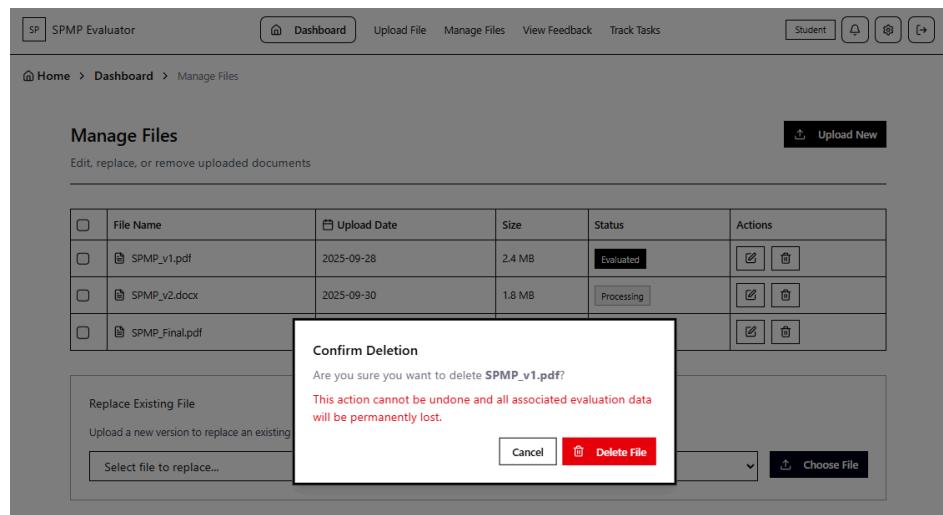
- Use Case Description

Field	Description
Use Case Name	Student File Removal
Primary Actor	Student
Secondary Actors	None
Description	Enables students to delete their uploaded files if they are before the submission deadline.
Preconditions	Student is logged in and has uploaded files that are still within the removal window.
Basic Flow	1) Student views list of uploaded files. 2) Student selects a file for removal. 3) System prompts for confirmation. 4) Student confirms deletion. 5) System removes file and logs action.
Alternative Flows	- File past deadline: System disables removal option.
Postconditions	File is permanently deleted from the system.
Exceptions	Deletion failure: System retains file and shows error message.

- *Activity Diagram*

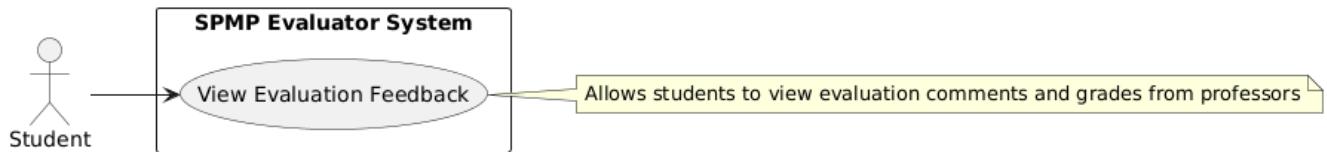


- *Wireframe*



## 2.4 Student View Feedback

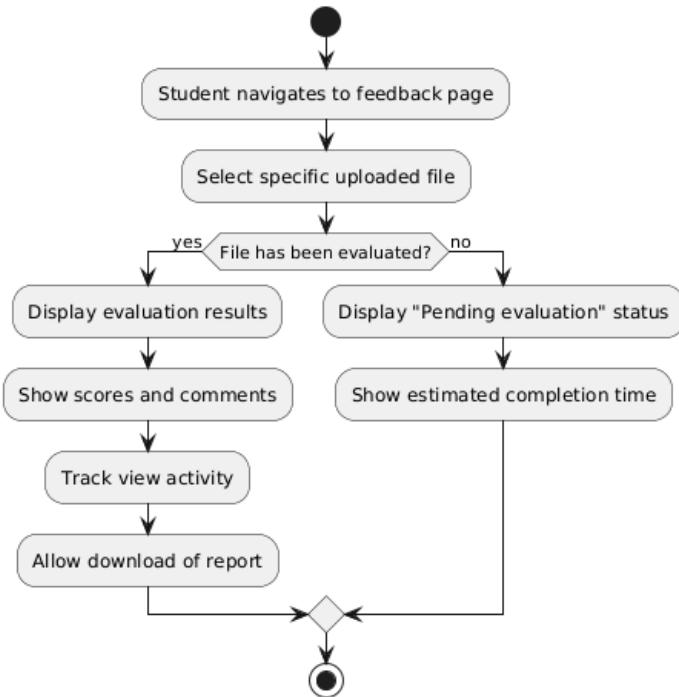
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	Student View Feedback
Primary Actor	Student
Secondary Actors	None
Description	Professor (indirect, provides feedback)
Preconditions	Students can access detailed feedback, comments, and grades provided by professors on their uploaded files.
Basic Flow	Student is logged in and has uploaded files that have been evaluated.
Alternative Flows	1) Student navigates to feedback section. 2) Student selects a specific file. 3) System displays evaluation results. 4) Student reviews comments and scores. 5) System tracks view activity.
Postconditions	- No evaluation yet: System shows “Pending evaluation” status.
Exceptions	System error: Display error message and suggest retry.

- *Activity Diagram*



- *Wireframe*

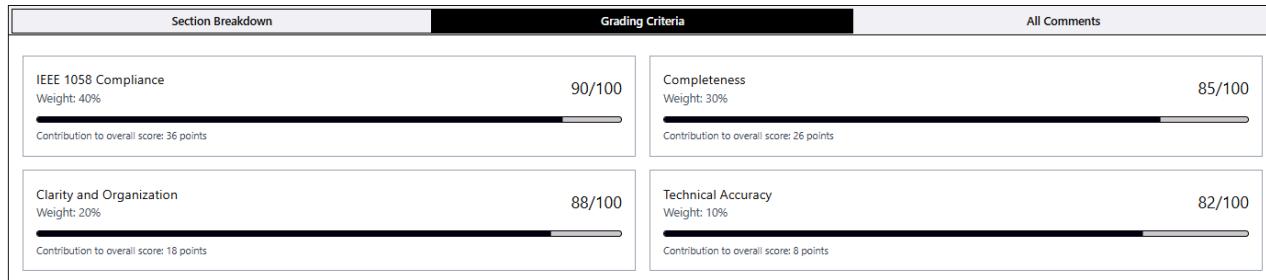
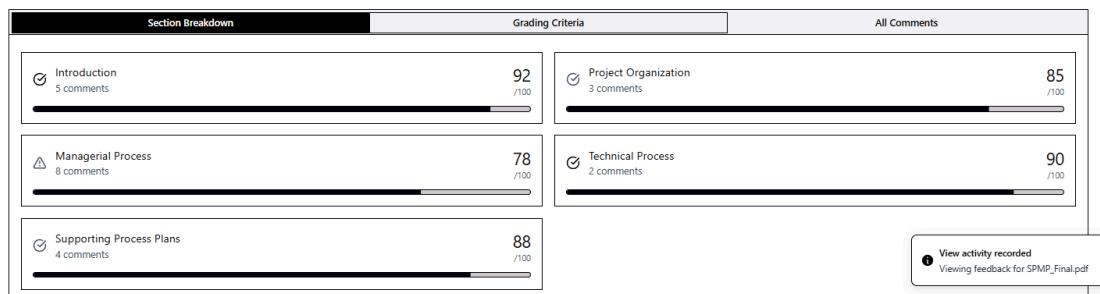
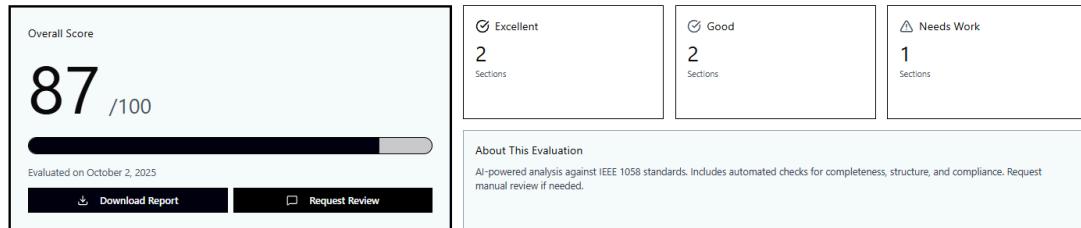
The wireframe shows the 'View Feedback' page. At the top, there is a navigation bar with 'SPMP Evaluator', 'Dashboard' (which is highlighted in blue), 'Upload File', 'Manage Files', 'View Feedback', 'Track Tasks', and user profile icons for 'Student', 'Notifications', 'Settings', and 'Logout'. Below the navigation is a breadcrumb trail: '@ Home > Dashboard > View Feedback'. The main content area is titled 'View Feedback' and contains a sub-instruction 'Select a file to view detailed evaluation results'. Four feedback entries are listed in a grid:

File Name	Score	Details
SPMP_Final.pdf	87 /100	Uploaded on September 28, 2025, Oct 2, Evaluated
SPMP_Draft_v2.pdf	76 /100	Uploaded on September 15, 2025, Sep 20, Evaluated
SPMP_Revised.pdf	Awaiting evaluation	Uploaded on October 1, 2025, Pending Evaluation
SPMP_Update.pdf	Awaiting evaluation	Uploaded on October 3, 2025, Pending Evaluation

[Home](#) > [Dashboard](#) > [View Feedback](#) > SPMP\_Final.pdf

### Evaluation Results

Detailed feedback for SPMP\_Final.pdf

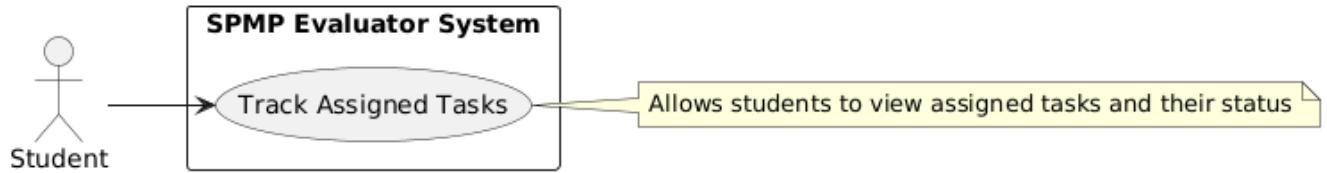


**All Comments**

Section Breakdown	Grading Criteria	All Comments
Introduction	Line 45-52	Excellent project scope definition. Clear objectives and well-defined deliverables align perfectly with IEEE 1058 standards.
Managerial Process	Section 3.2	Risk management plan needs more detail. Consider adding specific mitigation strategies for identified risks.
Technical Process	Section 4.1	Technical approach is well-documented. Development methodology clearly described with appropriate tools and techniques listed.
Supporting Process	Section 5.3	Quality assurance procedures are comprehensive. Consider adding more specific testing metrics and acceptance criteria.

## 2.5 Student Task Tracking

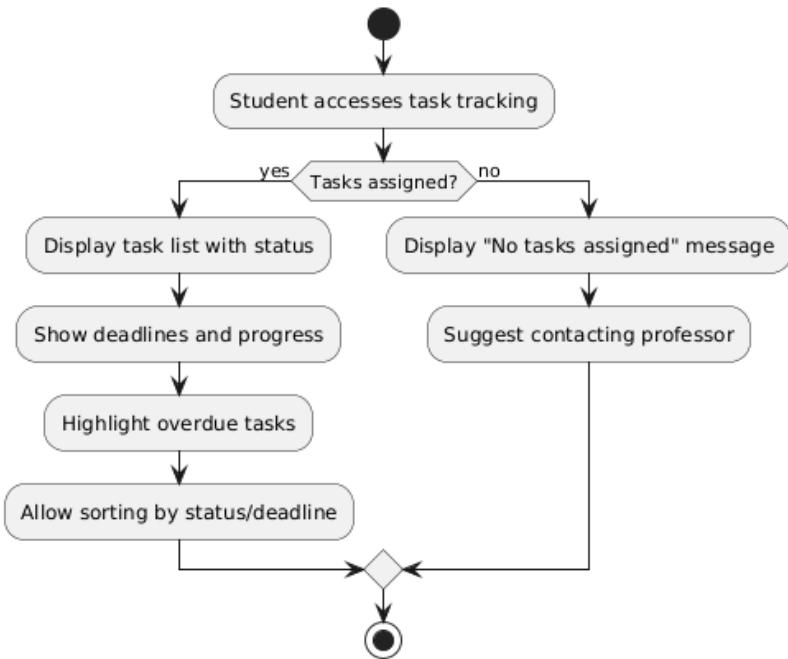
- Use Case Diagram



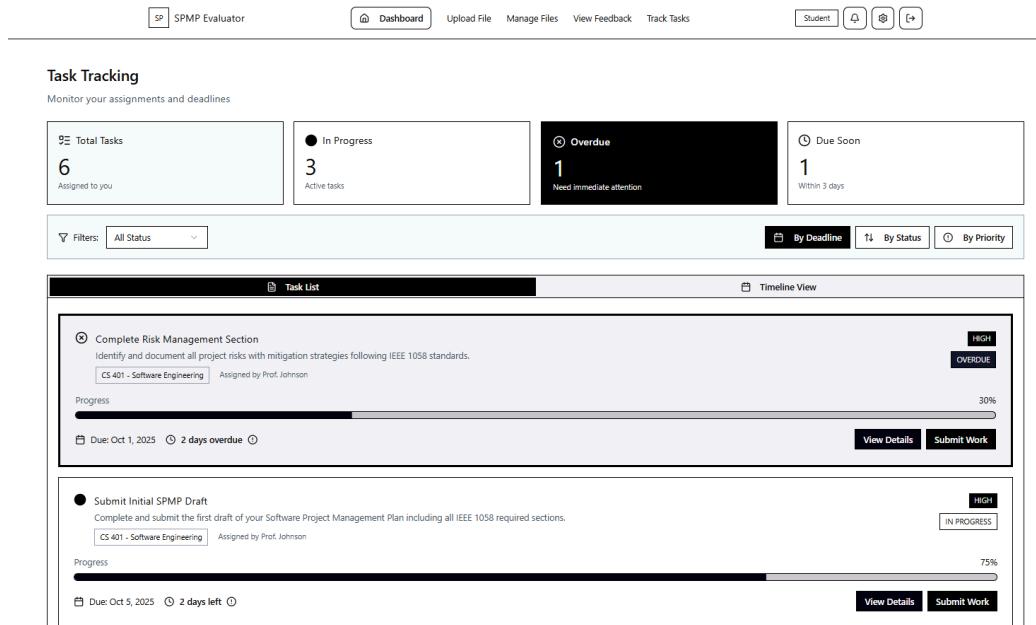
- Use Case Description

Field	Description
Use Case Name	Student Task Tracking
Primary Actor	Student
Secondary Actor	Professor (assigns tasks)
Description	Students can view tasks assigned to them, including deadlines, progress, and completion status.
Preconditions	Professor (assigns tasks)
Basic Flow	1) Student accesses task tracking section. 2) System displays list of assigned tasks. 3) Student views task details and deadlines. 4) Student checks progress indicators. 5) System updates view activity.
Alternative Flows	- No tasks assigned: System shows empty state with helpful message.
Postconditions	Student is aware of current task status and deadlines.
Exceptions	System unavailable: Show cached tasks or offline message.

- *Activity Diagram*

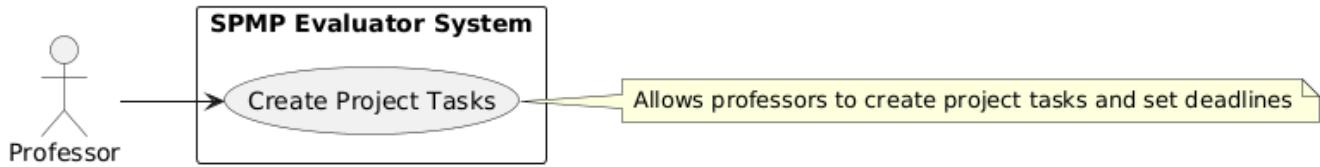


- *Wireframe*



## 2.6 Professor Task Creation

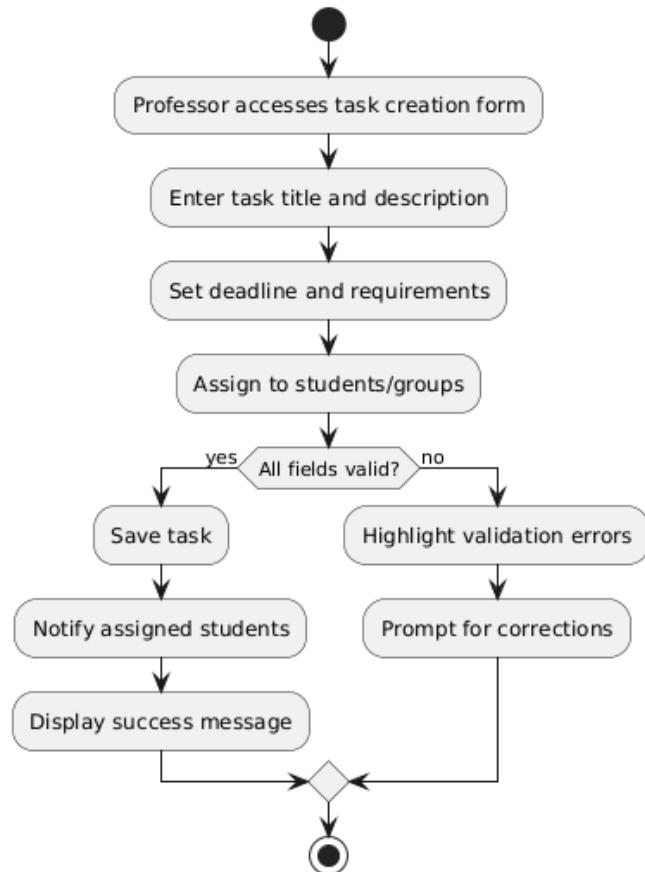
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	Professor Task Creation
Primary Actor	Professor
Secondary Actor	None
Description	Professors can create new project tasks for students, including descriptions, deadlines, and requirements.
Preconditions	Professor is logged in and has permission to create tasks.
Basic Flow	1) Professor navigates to task creation. 2) Professor enters task details. 3) Professor sets deadline and requirements. 4) Professor assigns to specific students/groups. 5) System saves task and notifies students.
Alternative Flows	- Save as draft: Task is saved but not published yet.
Postconditions	Task is created and students are notified.
Exceptions	Validation error: System highlights issues and prevents saving.

- *Activity Diagram*



- *Wireframe*



### Create New Task

Assign tasks to students with deadlines and IEEE 1058 compliance requirements

**Basic Information**

Task Title \*

Course \*  
 Task Type \*

Description \*  
  
0 characters

**Task Requirements**

**Deadline Settings**

Start Date

Due Date \*

Due Time

Late Penalty (% per day)

**Assign Students**

Select All  
 Sec A    Sec B    Sec C

Alice Johnson  
alice@uni.edu  
Section A

Bob Smith  
bob@uni.edu  
Section A

Carol Williams  
carol@uni.edu  
Section A

David Brown

0 of 8 selected

**Grading Settings**

Total Points

Enable AI evaluation

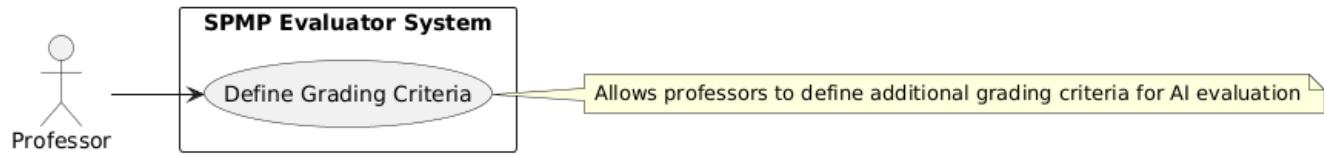
Peer review component

IEEE 1058 rubric

**Action Buttons**

## 2.7 Professor Supplement Grading Criteria

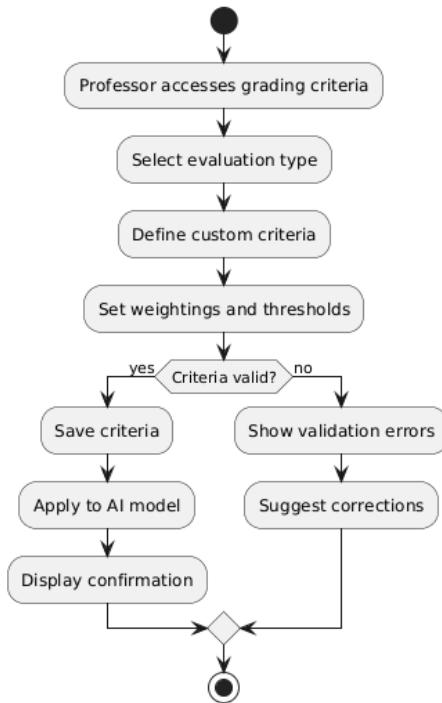
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	Professor Supplement Grading Criteria
Primary Actor	Professor
Secondary Actor	None
Description	Professors can supplement AI-generated evaluations with custom grading rubrics and criteria.
Preconditions	Professor is logged in and has access to grading tools.
Basic Flow	1) Professor selects evaluation criteria section. 2) Professor defines custom criteria. 3) Professor sets weightings for each criterion. 4) Professor saves criteria for use in evaluations. 5) System applies to future evaluations.
Alternative Flows	- Load template: Use pre-defined criteria templates.
Postconditions	Custom criteria are integrated into AI evaluation process.
Exceptions	Invalid criteria: System rejects and prompts for valid inputs.

- *Activity Diagram*



- *Wireframe*

The wireframe displays the 'Supplement Grading Criteria' interface. At the top, there's a navigation bar with links like Home, Dashboard, Create Task, Grading Criteria, Override AI, Update Tasks, Monitor Progress, and Professor. Below the navigation is a breadcrumb trail: Home > Dashboard > Supplement Grading Criteria.

The main area is titled 'Supplement Grading Criteria' and includes a sub-instruction: 'Customize AI evaluation criteria with custom rubrics and weightings'. It features three main input sections:

- Evaluation Type:** A dropdown menu set to 'Complete SPMP Document'.
- Load Template:** A dropdown menu to 'Select a template...' with a 'Load' button.
- Total Weight:** A section showing '100%' with a radio button labeled 'Valid'.

Below these are two rows of 'Custom Criteria' (4 total). Each row contains:

- Criterion Name:** A dropdown menu showing '1 IEEE 1058 Compliance'.
- Description:** A text area containing 'Adherence to IEEE 1058 standard structure and content requirements'.
- Weight:** A slider set at 40%.
- Pass Threshold:** A slider set at 70%.
- Contribution to final score:** A slider set at 40%.
- Minimum score to pass:** A slider set at 70%.

To the right of these rows is a 'Live Preview' section showing a breakdown of the total weight (100%) across four categories: IEEE 1058 Compliance (40%), Completeness (30%), Clarity and Organization (20%), and Technical Accuracy (10%). It also shows a 'Sample Score Calculation' table:

	Score
IEEE 1058 Compliance (90)	36.0
Completeness (85)	25.5
Clarity and Organization (88)	17.6
Technical Accuracy (82)	8.2
<b>Final Score</b>	<b>87.3</b>

A note below the preview states: 'This preview shows how criteria will be weighted in the AI evaluation. The sample calculation demonstrates scoring using typical student performance values.'

At the bottom left is a 'Quick Actions' sidebar with 'Validate Criteria' and 'Export Configuration' buttons. On the right, there are two status indicators: 'Grading criteria accessed' and 'Customize evaluation criteria and weightings'.

Criterion Name *	Weight *	Pass Threshold *
Clarity and Organization	30 %	75 %
Technical Accuracy	20 %	65 %
	10 %	80 %

Criteria	Weight
IEEE 1058 Compliance	40%
Completeness	30%
Clarity and Organization	20%
Technical Accuracy	10%
Total	100%

Sample Score Calculation

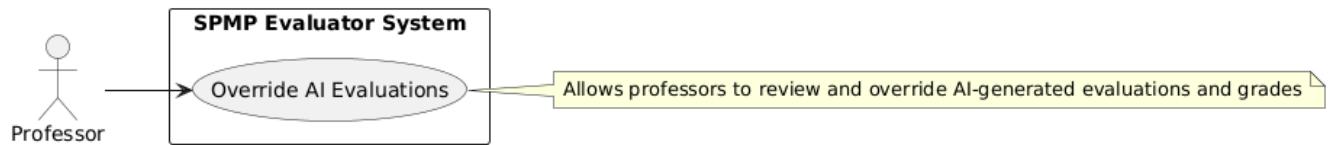
Criteria	Score
IEEE 1058 Compliance (90)	36.0
Completeness (85)	25.5
Clarity and Organization (88)	17.6
Technical Accuracy (82)	8.2
Final Score:	87.3

This preview shows how criteria will be weighted in the AI evaluation. The sample calculation demonstrates scoring using typical student performance values.

Buttons at the bottom: Cancel, Save as Template, Apply Criteria to AI Model

## 2.8 Professor Override AI Results

- Use Case Diagram

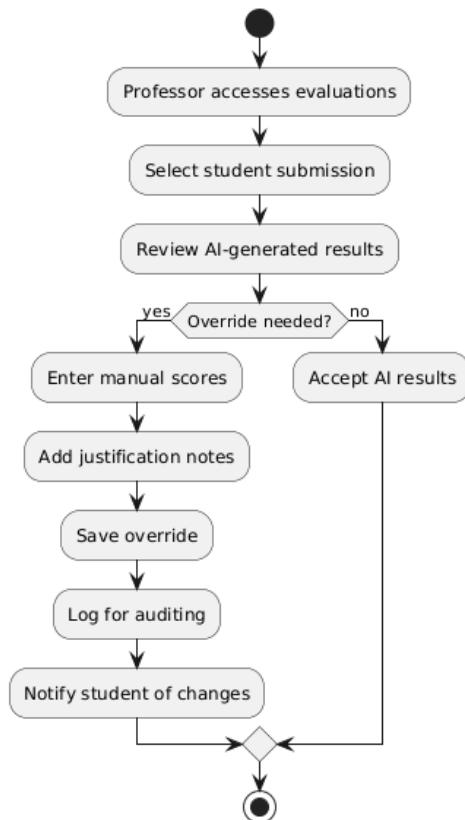


- Use Case Description

Field	Description
Use Case Name	Professor Override AI Results
Primary Actor	Professor
Secondary Actors	None
Description	Professors can review AI-generated evaluations and manually override scores or feedback if necessary.
Preconditions	AI evaluation has been completed for a student's submission.
Basic Flow	1) Professor views AI evaluation results. 2) Professor reviews scores and feedback.

	3) Professor makes manual adjustments. 4) Professor adds justification for override. 5) System saves override and notifies student.
Alternative Flows	- Accept AI results: No changes needed, just confirm.
Postconditions	Overridden results are final and logged for auditing.
Exceptions	Permission denied: Ensure professor has rights for specific student.

- *Activity Diagram*



● **Wireframe**

The wireframe illustrates the user interface for managing student evaluations. At the top, there's a navigation bar with links for SPMP Evaluator, Dashboard, Create Task, Grading Criteria, Override AI, Update Tasks, Monitor Progress, and a Professor section. Below the navigation is a breadcrumb trail: Home > Dashboard > Override AI Evaluation.

### Override AI Evaluation

Review and override AI-generated student evaluations

The main content area displays a list of students with their evaluation status and scores:

- Alice Johnson (alice@uni.edu) - CS 401: EVALUATED
- Bob Smith (bob@uni.edu) - CS 401: OVERRIDDEN (Score: 75.3 /100, Evaluated Sep 30)
- Carol Williams (carol@uni.edu) - CS 401: NO EVALUATION
- David Brown (david@uni.edu) - CS 402: NO EVALUATION
- Eve Davis (eve@uni.edu) - CS 401: NO EVALUATION

A message box at the bottom right indicates: "Evaluation override accessed" and "Review and override AI-generated evaluations".

---

Below this, another instance of the interface shows the details for Bob Smith's evaluation:

### AI Evaluation Review

SPMP\_Draft\_Bob.pdf

Student Information: Name: Bob Smith, Email: bob@uni.edu, Course: CS 401, Submitted: September 25, 2025, Evaluated: September 30, 2025.

AI-Generated Score: 72 /100 (Based on IEEE 1058 compliance criteria)

Final Score: 75.3 /100 (Override applied)

Previous Overrides: Shows a previous override for Prof. Johnson on October 1, 2025.

Section Scores and Grading Criteria table:

Section	AI Score	Total Score
Introduction	75	75
Project Organization	70	70
Managerial Process	68	68
Technical Process	72	85
Supporting Process Plans	75	75

AI-Generated Feedback: Submission shows basic understanding but needs more depth across all sections. Several areas require significant revision to meet IEEE 1058 standards.

The screenshot shows the AI Evaluation Review page for a student named Alice Johnson. At the top, there's a navigation bar with links for SPMP Evaluator, Dashboard, Create Task, Grading Criteria, Override AI, Update Tasks, Monitor Progress, Professor, and other icons.

The main content area starts with "AI Evaluation Review" and the file name "SPMP\_Final\_Alice.pdf". It displays two scores: an "AI-Generated Score" of 87 / 100 and a "Preview Final Score" of 87.3 / 100, noting that the final score includes manual adjustments.

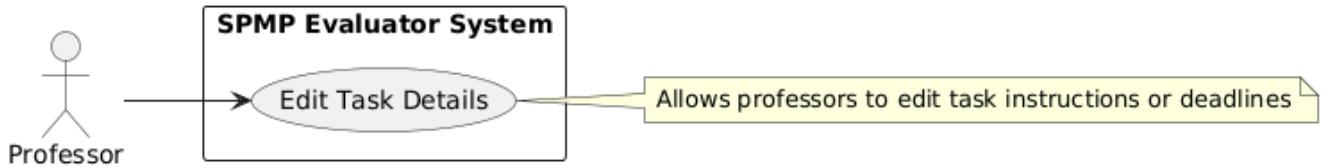
Below this, there are two sections: "Section Scores" and "Grading Criteria". The "Section Scores" section lists five categories with their respective AI and Manual scores: Introduction (AI: 92, Manual: 92), Project Organization (AI: 85, Manual: 85), Managerial Process (AI: 78, Manual: 78), Technical Process (AI: 90, Manual: 90), and Supporting Process Plans (AI: 88, Manual: 88). The "Grading Criteria" section provides feedback for each category.

Under "AI-Generated Feedback", it says: "Overall strong submission that demonstrates good understanding of IEEE 1058 standards. The project scope and technical approach are particularly well developed. Main areas for improvement are in the risk management section and providing more specific metrics for quality assurance." A note at the bottom of this section states: "This justification will be saved in the audit log and included in the student notification."

At the bottom right of the main content area are buttons for "Cancel Override" and "Save Override & Notify Student".

## 2.9 Professor Update Tasks

- Use Case Diagram

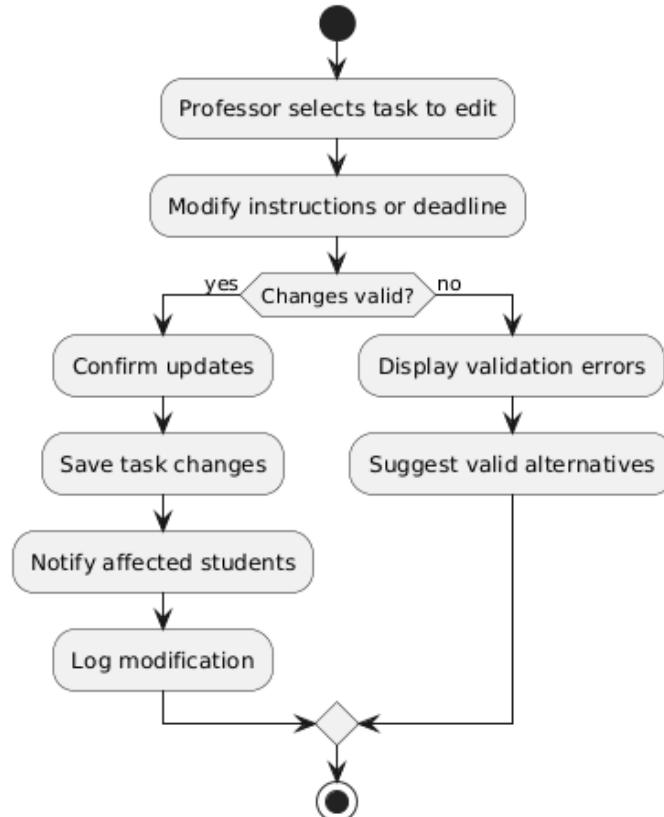


- Use Case Description

Field	Description
-------	-------------

Use Case Name	Professor Task Creation
Primary Actor	Professor
Secondary Actor	Students (affected by changes)
Description	Professors can modify existing task instructions, deadlines, or requirements as needed.
Preconditions	Task exists and professor has edit permissions.
Basic Flow	<ol style="list-style-type: none"> <li>1) Professor selects task to edit.</li> <li>2) Professor modifies instructions/deadlines.</li> <li>3) System validates changes.</li> <li>4) Professor confirms updates.</li> <li>5) System saves and notifies affected students.</li> </ol>
Alternative Flows	- Extend deadline: Automatically notify students of extension.
Postconditions	Task is updated and students are informed of changes.
Exceptions	Invalid deadline: Prevent setting past dates or invalid times.

- *Activity Diagram*



● **Wireframe**

SP SPMP Evaluator      Dashboard      Create Task      Grading Criteria      Override AI      Update Tasks      Monitor Progress      Professor        

Home > Dashboard > Update Tasks

### Update Tasks

Modify existing task instructions, deadlines, or requirements

Search tasks by title or course... All Tasks

Submit Initial SPMP Draft  
CS 401 - Software Engineering  
ACTIVE MODIFIED  
Due: Oct 5, 2025  
12/45 submitted

45 Assigned	12 Submitted	33 Pending
-------------	--------------	------------

Peer Review Assignment  
CS 401 - Software Engineering  
ACTIVE  
Due: Oct 12, 2025  
0/45 submitted

45 Assigned	0 Submitted	45 Pending
-------------	-------------	------------

Final SPMP Submission  
CS 401 - Software Engineering  
ACTIVE  
Due: Oct 15, 2025  
0/45 submitted

45 Assigned	0 Submitted	45 Pending
-------------	-------------	------------

...

SP SPMP Evaluator      Dashboard      Create Task      Grading Criteria      Override AI      Update Tasks      Monitor Progress      Professor        

Home > Dashboard > Update Tasks > Peer Review Assignment

### Update Task

Modify task details and notify students of changes



Current Task Status	Task Information
45 Assigned	Course: CS 401 - Software Engineering Created: October 1, 2025
0 Submitted	
45 Pending	

 Task Details

Task Title \*  
Peer Review Assignment

Description \*  
Review and provide constructive feedback on two peer SPMP submissions. Use the provided rubric.

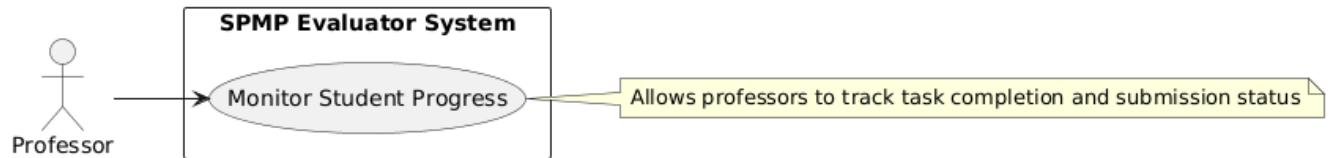
 Deadline Settings

Due Date \*  
10/12/2025

Due Time \*  
11:59 PM

## 2.10 Professor Monitor Student Progress

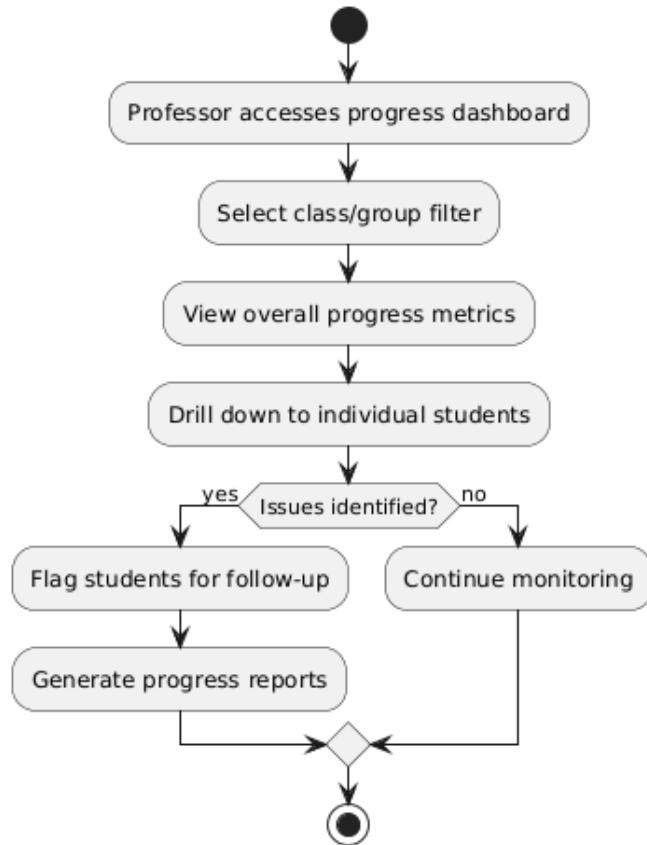
- Use Case Diagram



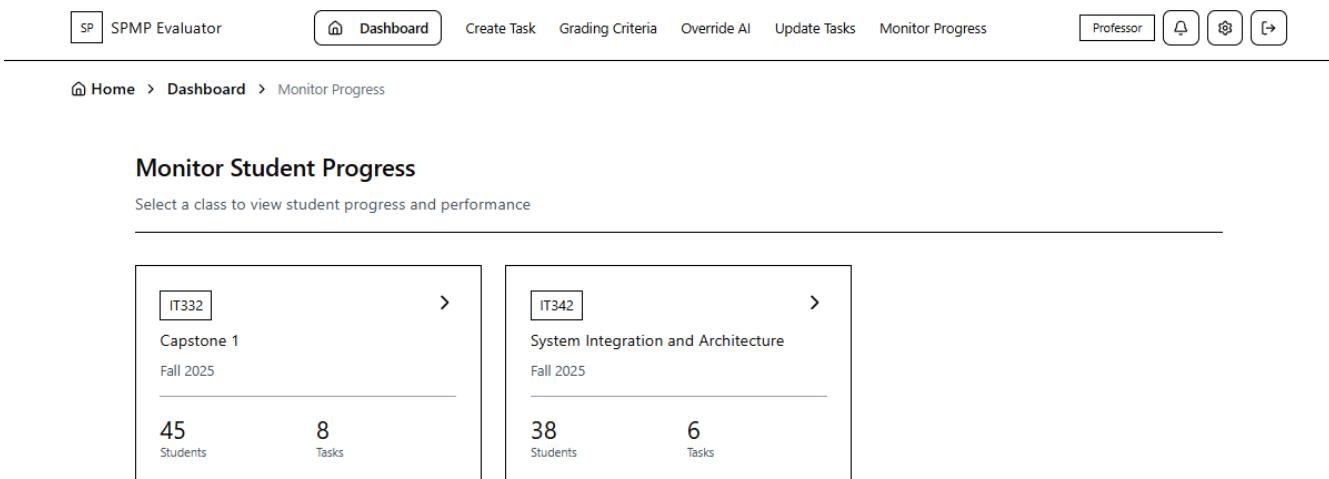
- Use Case Description

Field	Description
Use Case Name	Professor Monitor Student Progress
Primary Actor	Professor
Secondary Actors	Students (being monitored)
Description	Professors can track overall student progress, submission status, and identify students needing assistance.
Preconditions	Professor has access to student progress data.
Basic Flow	1) Professor accesses progress monitoring. 2) Professor selects class or group. 3) System displays progress dashboard. 4) Professor reviews individual student status. 5) Professor identifies issues or follows up.
Alternative Flows	- Filter by status: View only overdue or incomplete tasks.
Postconditions	Professor has current view of student progress.
Exceptions	Data unavailable: Show last known status or error message.

- *Activity Diagram*



- *Wireframe*



The dashboard displays the following key metrics:

- Total Students: 6
- Submitted: 33
- Pending: 8
- Overdue: 7
- Flagged: 2
- Class Avg: 77

Completion Rate: 69% of all tasks.

Student Status Distribution:

Status	Count
Excellent	2
On Track	2
At Risk	1
Critical	1

Search bar: Search students by name or email... | All Status | Show flagged only

Student Progress (6) section shows details for Alice Johnson and Bob Smith.

Alice Johnson (alice@uni.edu): Overall Progress 88%.

Bob Smith (bob@uni.edu): Overall Progress 75%.

Alice Johnson (alice@uni.edu)

Status: EXCELLENT (Exceeding expectations)

Average Score: 88 /100

Attendance: 95 %

Participation: 90 %

Task Summary:

Status	Count
Submitted	7
Pending	1
Overdue	0

Recent Activity:

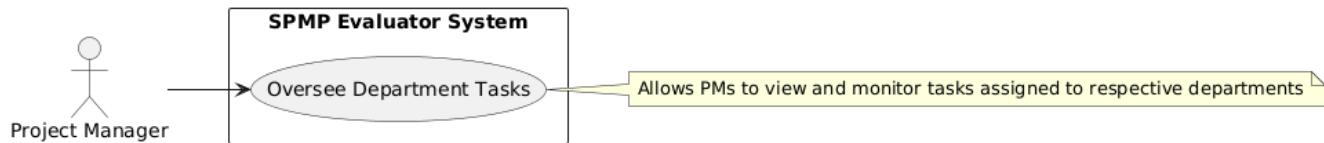
- Last submission: 2025-10-06 Quality Assurance Plan
- Enrolled: September 1, 2025

Task Details:

Task	Score	Status
SPMP Initial Draft	92	SUBMITTED
Risk Analysis Document	88	SUBMITTED
Project Schedule	85	SUBMITTED
Quality Assurance Plan	90	SUBMITTED
Resource Allocation		PENDING

## 2.11 PM Task Oversight

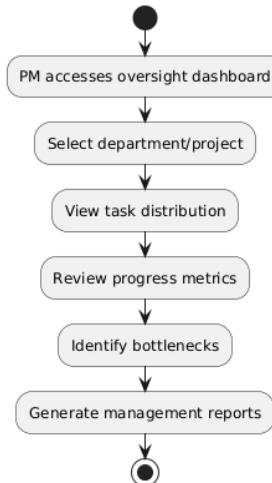
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	PM Task Oversight
Primary Actor	Project Manager
Secondary Actors	Professors, Students (indirect)
Description	Project managers can view and monitor all tasks assigned across their departments for oversight.
Preconditions	PM is logged in and has oversight permissions.
Basic Flow	1) PM accesses task oversight dashboard. 2) PM selects department or project view. 3) System displays task status overview. 4) PM reviews progress and resource allocation. 5) PM generates oversight reports.
Alternative Flows	- Department filter: Focus on specific areas.
Postconditions	PM has comprehensive view of departmental task status.
Exceptions	Access denied: Ensure PM has rights for specific departments.

- Activity Diagram



- *Wireframe*

The wireframe illustrates two main dashboards:

### Project Manager Dashboard

This dashboard provides a high-level overview of the Computer Science Department. It features five key metrics in cards:

- Active Tasks: 63
- Professors: 19
- Students: 378
- Evaluations: 1,247
- Avg Score: 82%

Below these cards is a section titled "Quick Actions" containing four buttons:

- View Task Oversight
- Manage Roles & Access
- Monitor Progress
- Generate AI Reports

Further down, there are three departmental summaries:

Computer Science	Software Engineering	Information Technology
Professors: 8	Professors: 5	Professors: 6
Students: 156	Students: 98	Students: 124
Tasks: 24	Tasks: 18	Tasks: 21
Completion: 78%	Completion: 82%	Completion: 65%

Two side panels provide real-time updates:

- System Alerts:** 2 departments below 70% completion rate. Includes a "Review" button.
- Recent Activity:** New user created: Prof. Anderson (2 hours ago), 127 evaluations completed today (Today), CS Department reached 78% completion (Yesterday).

### Task Oversight Dashboard

This dashboard monitors task distribution, progress, and resource allocation. It includes a navigation bar and a "Generate Report" button.

At the top, there are two view options: "Department View" (selected) and "Project View".

Below are four summary cards:

Total Tasks	Students	Professors	Avg Completion
56	337	19	77%

Filtering options include "Select Department" (set to Information Technology) and "Filter by Status" (set to All Courses).

The main content area displays departmental metrics for Information Technology:

Students	Professors	Total Tasks	Completed	Completion
83	6	14	9	72%

It also shows a breakdown of tasks by status:

Status	Count
Completed	9 tasks
Pending	3 tasks
Overdue	2 tasks

The screenshot displays the Task Oversight Dashboard of the SPMP Evaluator system. At the top, there's a navigation bar with links for SPMP Evaluator, Dashboard, Task Oversight, Progress Monitor, AI Reports, User Management, Override Evaluations, Project Manager, and several icons for settings and help. Below the navigation is a breadcrumb trail: Home > Dashboard > Task Oversight.

The main area is titled "Task Oversight Dashboard" and includes a subtitle "Monitor task distribution, progress, and resource allocation". A "Generate Report" button is located in the top right corner. Below this, there are two tabs: "Department View" and "Project View", with "Project View" being active.

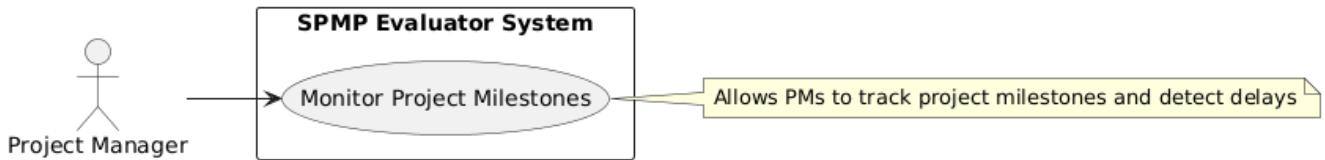
The dashboard features a search bar and filters for "All Projects" and "All Status". It lists four projects:

- E-Commerce Platform Redesign**: Status ON TRACK, Team Alpha, 5 students, 6/8 tasks completed, due Oct 15, 2025, updated 2 hours ago. Includes a "View Project Details" button.
- Hospital Management System**: Status AT RISK, Team Beta, 5 students, 4/8 tasks completed, due Oct 15, 2025, updated 1 day ago.
- Smart Parking Solution**: Status CRITICAL, Team Gamma, 4 students, 3/6 tasks completed, due Oct 12, 2025, updated 3 days ago. Includes a "View Project Details" button.
- Educational Portal Integration**: Status ON TRACK, Team Delta, 5 students, 4/6 tasks completed, due Oct 12, 2025, updated 5 hours ago.

At the bottom, a "Project Status Summary" section shows the count of projects in three categories: On Track (2), At Risk (1), and Critical (1).

## 2.12 PM Progress Monitoring

- Use Case Diagram

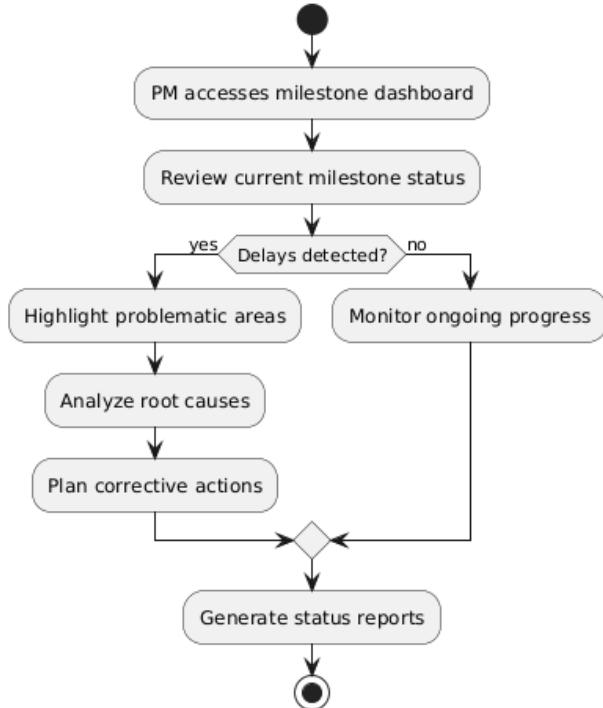


- Use Case Description

Field	Description
Use Case Name	PM Progress Monitoring
Primary Actor	Project Manager

Secondary Actors	Team members (indirect)
Description	Project managers track overall project milestones, timelines, and detect potential delays or issues.
Preconditions	PM has access to project progress data.
Basic Flow	<ol style="list-style-type: none"> <li>1) PM accesses milestone tracking.</li> <li>2) PM reviews current milestone status.</li> <li>3) System highlights any delays or issues.</li> <li>4) PM analyzes trends and patterns.</li> <li>5) PM takes corrective actions if needed.</li> </ol>
Alternative Flows	- Alert setup: Configure notifications for milestone changes.
Postconditions	PM is informed of project health and potential issues.
Exceptions	Data sync issues: Show cached data or error state.

- *Activity Diagram*



• Wireframe

The wireframe displays a dashboard for project monitoring. At the top, there's a navigation bar with tabs: SPMP Evaluator (selected), Dashboard, Task Oversight, Progress Monitor, AI Reports, User Management, and Override Evaluations. To the right of the tabs are buttons for Project Manager, Notifications, Help, and Logout.

The main area is titled "Milestone Progress Monitoring" and includes a subtitle "Track project timelines, milestones, and identify potential delays". Below this is a summary section with six boxes showing counts: Total Milestones (8), Completed (2), Delayed (4), Active Issues (3), Healthy (2), and Critical (1).

A dropdown menu shows "All Projects". Below it are three tabs: Overview (selected), Timeline, and Issues & Actions.

The dashboard features four project cards:

- E-Commerce Platform Redesign:** Current Phase: Development (6/8 milestones). Status: HEALTHY. End Date: Oct 15. Days Left: 4 days.
- Hospital Management System:** Current Phase: Testing (4/8 milestones). Status: WARNING. End Date: Oct 15. Days Left: 4 days.
- Smart Parking Solution:** Current Phase: Integration (3/6 milestones). Status: CRITICAL. End Date: Oct 12. Days Left: 1 days. A "View Details" button is present.
- Educational Portal Integration:** Current Phase: Testing (4/6 milestones). Status: HEALTHY. End Date: Oct 12. Days Left: 1 days.

At the bottom, there's a "Progress Trends" section with three cards: Completion Rate (25%, +12% from last week), Avg. Delay (2.3 days, -0.5 days from last week), and Issue Resolution (73%, +8% from last week).

SPMP Evaluator    Dashboard    Task Oversight    Progress Monitor    AI Reports    User Management    Override Evaluations    Project Manager   

Home > Dashboard > Progress Monitoring

### Milestone Progress Monitoring

Track project timelines, milestones, and identify potential delays

8 Total Milestones	2 Completed	4 Delayed	3 Action Issues	2 Healthy	1 Critical
--------------------	-------------	-----------	-----------------	-----------	------------

All Projects

Overview    **Timeline**    Issues & Actions

#### Milestone Timeline

- Requirements Analysis**  
 Completes system requirements documentation  
 Due: Sep 8    Done: Sep 7  
 E-commerce Platform Redesign    Team Alpha    COMPLETED  
 Progress: 100%
- Design Specification**  
 UI/UX design and architecture planning  
 Due: Sep 15    Done: Sep 14  
 E-commerce Platform Redesign    Team Alpha    COMPLETED  
 Progress: 100%  
 Dependencies: 1 milestone(s)
- Core Development**  
 Implement core system functionality  
 Due: Oct 1  
 E-commerce Platform Redesign    Team Alpha    ON-TRACK  
 Progress: 85%  
 Dependencies: 1 milestone(s)
- Integration Testing**  
 Completes system integration tests  
 Due: Oct 8  
 E-commerce Platform Redesign    Team Alpha    IN-PROGRESS  
 Progress: 45%  
 Dependencies: 1 milestone(s)
- System Architecture**  
 Database and API design  
 Due: Sep 22    Done: Sep 25  
 Hospital Management System    Team Beta    DELAYED  
 Progress: 100%
- Module Integration**  
 Integrates all system modules  
 Due: Oct 5  
 Hospital Management System    Team Beta    CRITICAL  
 Progress: 35%  
 Dependencies: 1 milestone(s)

### Milestone Progress Monitoring

Track project timelines, milestones, and identify potential delays

8 Total Milestones	2 Completed	4 Delayed	3 Action Issues	2 Healthy	1 Critical
--------------------	-------------	-----------	-----------------	-----------	------------

All Projects

Overview    Timeline    **Issues & Actions**

#### Active Issues (3)

- Hospital Management System - Module Integration**  
 Integration testing delayed due to incomplete module development  
 Impact: May affect final delivery deadline  
 Detected: Oct 5 • Assigned to Team Beta Lead  
 Plan Action
- Smart Parking Solution - Integration Layer**  
 Third-party API integration blocked by vendor delays  
 Impact: Critical path affected, 3-day delay expected  
 Detected: Oct 8  
 Plan Action
- E-commerce Platform Redesign - Integration Testing**  
 Limited test environment availability  
 Impact: Testing phase may extend by 1-2 days  
 Detected: Oct 7 • Assigned to Team Alpha Lead  
 Plan Action

#### Corrective Actions

- In-Progress** Hospital Management System  
 Allocate additional developer resources to complete module integration  
 Responsible: Team Beta Lead  
 Deadline: Oct 12, 2025
- Planned** Smart Parking Solution  
 Implement fallback integration strategy using alternative API  
 Responsible: Team Gamma Lead  
 Deadline: Oct 11, 2025
- Completed** E-commerce Platform Redesign  
 Schedule extended test environment access hours  
 Responsible: Team Alpha Lead  
 Deadline: Oct 9, 2025

SP SPMP Evaluator    Dashboard    Task Oversight    Progress Monitor    AI Reports    User Management    Override Evaluations    Project Manager   

---

8  
Total Milestones

2  
Completed

4  
Delayed

3  
Active Issues

2  
Healthy

1  
Critical

---

All Projects

---

OverviewTimelineIssues & Actions

---

#### Milestone Timeline

Requirements Analysis  
Complete system requirements documentation

E-Commerce Platform Redesign | Team Alpha | COMPLETED

Progress

100%

Due: Sep 8  
 Done: Sep 7

Design Specification  
UI/UX design and architecture planning

E-Commerce Platform Redesign | Team Alpha | COMPLETED

Progress

100%

Dependencies: 1 milestone(s)

Due: Sep 15  
 Done: Sep 14

Core Development  
Implement core system functionality

E-Commerce Platform Redesign | Team Alpha | ON-TRACK

Progress

85%

Dependencies: 1 milestone(s)

Due: Oct 1

Integration Testing  
Complete system integration tests

E-Commerce Platform Redesign | Team Alpha | AT-RISK

Progress

45%

Dependencies: 1 milestone(s)

Due: Oct 8

System Architecture  
Database and API design

Hospital Management System | Team Beta | DELAYED

Progress

100%

Due: Sep 22  
 Done: Sep 25

Module Integration  
Integrate all system modules

Hospital Management System | Team Beta | CRITICAL

Progress

35%

Dependencies: 1 milestone(s)

Due: Oct 5

API Development  
Build RESTful API endpoints

Smart Parking Solution | Team Gamma | DELAYED

Progress

100%

Due: Sep 28

Page 48 of 84

## 2.13 PM AI-Generated Reports

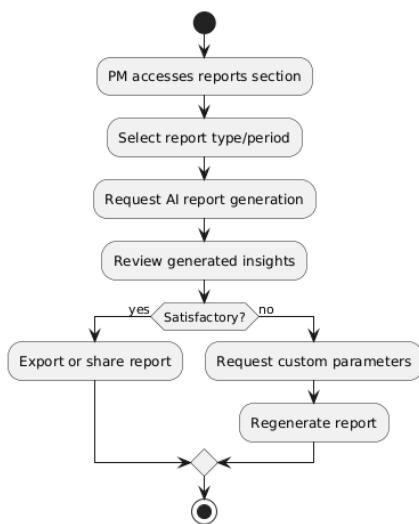
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	PM AI-Generated Reports
Primary Actor	Project Manager
Secondary Actors	AI System (generates reports)
Description	Project managers can access AI-generated summaries, analytics, and performance evaluations for projects.
Preconditions	AI evaluations have been completed and reports are available.
Basic Flow	<ol style="list-style-type: none"> <li>1) PM navigates to reports section.</li> <li>2) PM selects report type or time period.</li> <li>3) System generates or retrieves AI report.</li> <li>4) PM reviews analytics and insights.</li> <li>5) PM exports report if needed.</li> </ol>
Alternative Flows	- Custom reports: Specify parameters for tailored reports.
Postconditions	PM has access to comprehensive project insights.
Exceptions	Report generation failure: Retry or use manual fallback.

- Activity Diagram



● **Wireframe**

The wireframe illustrates the user interface for generating and viewing AI-generated reports. It includes a header bar with navigation links and a footer bar at the bottom.

**Top Header:** SPMP Evaluator, Dashboard, Task Oversight, Progress Monitor, AI Reports, User Management, Override Evaluations, Project Manager, and three icons (refresh, search, add).

**Breadcrumb:** Home > Dashboard > AI-Generated Reports

**Section Headers:** AI-Generated Reports (with a 'Custom Report' button) and Generated Reports, AI Insights, Performance Analytics.

**Report Generation Form:** Generate New Report, Report Type (Project Summary), Time Period (Current Month), Course Filter (All Courses), and a 'Generate AI Report' button.

**Report Preview Grid:** Displays four report cards:

- October 2025 Project Overview:** Generated: October 11, 2025. Contains sections for AI Insights Generated (12 insights, AI-powered analysis complete) and a large empty box.
- Q4 Performance Analysis:** Generated: October 8, 2025. Contains sections for AI Insights Generated (18 insights, AI-powered analysis complete) and a large empty box.
- Semester Trend Report:** Generated: October 5, 2025. Contains sections for AI Insights Generated (15 insights, AI-powered analysis complete) and a large empty box.
- SPMP Quality Evaluation:** Generated: October 1, 2025. Contains sections for AI Insights Generated (10 insights, AI-powered analysis complete) and a large empty box.

**Bottom Header:** SPMP Evaluator, Dashboard, Task Oversight, Progress Monitor, AI Reports, User Management, Override Evaluations, Project Manager, and three icons (refresh, search, add).

**Section Headers:** AI-Generated Reports (with a 'Custom Report' button) and Generated Reports, AI Insights, Performance Analytics.

**Performance Metrics Dashboard:** Shows three summary cards: Avg Completion (74%), Active Students (83), and Quality Score (82).

**Performance Metrics Dashboard:** Shows six detailed performance metrics with progress bars and percentage changes from last period:

- Overall Completion Rate: 74% (+8% from last period)
- On-Time Delivery Rate: 68% (-5% from last period)
- Quality Score Average: 82% (+12% from last period)
- Team Collaboration Index: 71% (+3% from last period)
- Documentation Compliance: 88% (+15% from last period)
- Risk Mitigation Rate: 65% (-3% from last period)

The screenshot displays the SPMP Evaluator software interface. At the top, there is a navigation bar with links for SPMP Evaluator, Dashboard, Task Oversight, Progress Monitor, AI Reports, User Management, and Override Evaluations. To the right of these are buttons for Project Manager, a bell icon, a user icon, and a search icon. Below the navigation bar, the URL shows Home > Dashboard > AI-Generated Reports.

The main content area is titled "AI-Generated Reports" with a subtitle "Access AI-powered summaries, analytics, and performance evaluations". There is a "Custom Report" button with a gear icon.

A "Generate New Report" section contains fields for "Report Type" (Project Summary), "Time Period" (Current Month), and "Course Filter" (All Courses). A large "Generate AI Report" button is present below these fields.

Below this section are three tabs: "Generated Reports" (selected), "AI Insights" (highlighted in black), and "Performance Analytics".

The "AI Insights" tab displays a card titled "AI-Generated Insights (5)". It contains two sections:

- Strong Overall Project Performance**: Labeled with a checkmark icon, "PERFORMANCE", and "POSITIVE".

IT332 teams are demonstrating excellent progress with 78% average completion rate, exceeding semester benchmarks.

**Recommendation:** Continue current methodology and consider documenting best practices for future cohorts.

**Impact:** High positive impact on student learning outcomes  
**Affected:** E-Commerce Platform, Hospital Management System
- Integration Phase Delays Detected**: Labeled with a warning icon, "TIMELINE", and "WARNING".

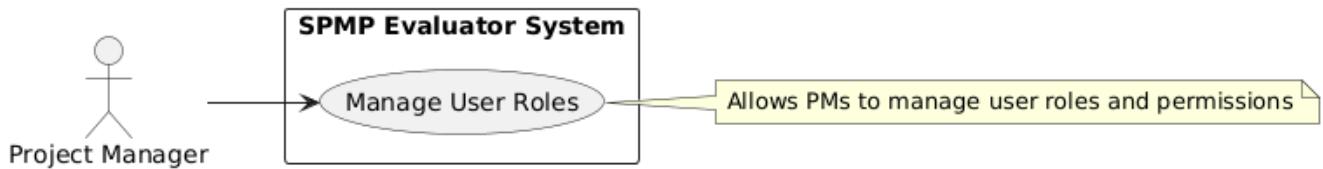
IT342 teams showing consistent 2-3 day delays in integration testing phases, potentially affecting final deliverables.

**Recommendation:** Allocate additional lab time for integration testing and consider peer review sessions.

**Impact:** Medium risk to on-time project completion  
**Affected:** Smart Parking Solution, Educational Portal

## 2.14 PM Role and Access Management

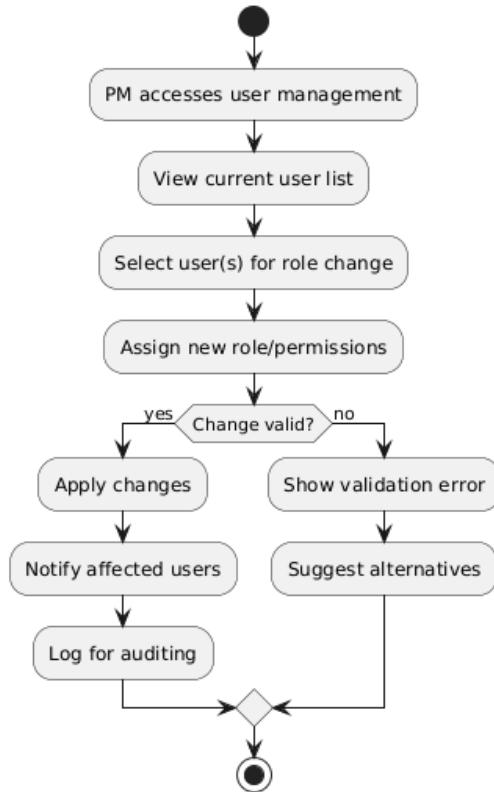
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	PM Role and Access Management
Primary Actor	Project Manager
Secondary Actors	System Administrator (potential conflicts)
Description	Project managers can assign, modify, or revoke user roles and permissions within their scope.
Preconditions	PM has administrative privileges for role management.
Basic Flow	1) PM accesses user management section. 2) PM views current user roles. 3) PM makes role assignments or changes. 4) System validates permissions. 5) PM confirms changes and notifies users.
Alternative Flows	- Bulk changes: Apply role changes to multiple users.
Postconditions	User roles and permissions are updated.
Exceptions	Permission conflict: Prevent changes that violate security policies.

- *Activity Diagram*



- *Wireframe*

The wireframe shows the 'Role & Access Management' page. At the top, there's a navigation bar with 'SPMP Evaluator', 'Dashboard', 'Task Oversight', 'Progress Monitor', 'AI Reports', 'User Management', 'Override Evaluations', and a 'Project Manager' section with icons for 'Edit', 'Delete', and 'Add'. Below the navigation is a breadcrumb trail: 'Home > Dashboard > Role & Access Management'. The main header is 'Role & Access Management' with a subtitle 'Manage user roles, permissions, and access control'. On the right, there's a button for 'Add User'. Below the header are four summary boxes: 'Total Users' (7), 'Professors' (2), 'Students' (4), and 'Active' (6). A 'User Management' tab is active, showing a table of users with columns: Name, Email, Role, Department, Status, and Actions. The table lists eight users: Dr. Maria Santos, Dr. Robert Chen, Alice Johnson, Bob Smith, Carol Williams, David Martinez, and Eve Davis. Each row includes checkboxes for selection and buttons for 'Edit' and 'Delete'.

	Name	Email	Role	Department	Status	Actions
<input type="checkbox"/>	Dr. Maria Santos	maria.santos@uni.edu	PROFESSOR	Information Technology	ACTIVE	<input type="button"/> <input type="button"/>
<input type="checkbox"/>	Dr. Robert Chen	robert.chen@uni.edu	PROFESSOR	Information Technology	ACTIVE	<input type="button"/> <input type="button"/>
<input type="checkbox"/>	Alice Johnson	alice.johnson@student.uni.edu	STUDENT	Information Technology	ACTIVE	<input type="button"/> <input type="button"/>
<input type="checkbox"/>	Bob Smith	bob.smith@student.uni.edu	STUDENT	Information Technology	ACTIVE	<input type="button"/> <input type="button"/>
<input type="checkbox"/>	Carol Williams	carol.williams@student.uni.edu	STUDENT	Information Technology	ACTIVE	<input type="button"/> <input type="button"/>
<input type="checkbox"/>	David Martinez	david.martinez@uni.edu	PM	Information Technology	ACTIVE	<input type="button"/> <input type="button"/>
<input type="checkbox"/>	Eve Davis	eve.davis@student.uni.edu	STUDENT	Information Technology	SUSPENDED	<input type="button"/> <input type="button"/>

The screenshot shows the 'User Management' tab selected in the top navigation bar. A modal dialog box titled 'Edit User' is open, prompting the user to 'Modify user role and permissions'. Inside the dialog, the user's name is set to 'Dr. Maria Santos', email to 'maria.santos@uni.edu', role to 'Professor', status to 'Active', and department to 'Information Technology'. There is a 'Save Changes' button at the bottom right. To the right of the dialog, a table lists users with columns for Status (ACTIVE or SUSPENDED) and Actions (two checkboxes).

Status	Actions
ACTIVE	<input type="checkbox"/> <input type="checkbox"/>
SUSPENDED	<input type="checkbox"/> <input type="checkbox"/>

The screenshot shows the 'User Management' tab selected. A modal dialog box titled 'Confirm Changes' displays a warning message: '⚠ Action: Delete user' and 'Affected Users: 1'. It also contains a note: '⌚ Users will be notified of these changes via email'. At the bottom are two buttons: a light blue 'Cancel' button and a dark blue 'Delete User' button.

Home > Dashboard > Role & Access Management

### Role & Access Management

Manage user roles, permissions, and access control

Add User

7 Total Users	2 Professors	4 Students	6 Active
------------------	-----------------	---------------	-------------

User Management    Permission Matrix

#### Permission Matrix

Content Permissions

Permission	Student	Professor	PM
Upload Files Upload SPMP documents	✓	✓	✓
Edit Files Edit uploaded documents	✓	✓	—
View Feedback View AI and professor feedback	✓	✓	✓
Track Tasks View and track assigned tasks	✓	✓	✓

Grading Permissions

Permission	Student	Professor	PM
Create Tasks Create and assign tasks	—	✓	✓
Grade Submissions Review and grade student work	—	✓	✓
Override AI Override AI evaluations	—	✓	✓
Update Tasks Modify existing tasks	—	✓	✓

### Role & Access Management

Manage user roles, permissions, and access control

Add User

7 Total Users	2 Professors	4 Students	6 Active
------------------	-----------------	---------------	-------------

User Management    Permission Matrix

#### Permission Matrix

Content Permissions

Permission	Student	Professor	PM
Upload Files Upload SPMP documents	✓	✓	✓
Edit Files Edit uploaded documents	✓	—	—
View Feedback View AI and professor feedback	✓	✓	✓
Track Tasks View and track assigned tasks	✓	✓	✓

Add New User

Create a new user account with role assignment

Full Name:

Email Address:

Role:

Department:

Send welcome email:

Add User

## 2.15 PM Override Project Evaluations

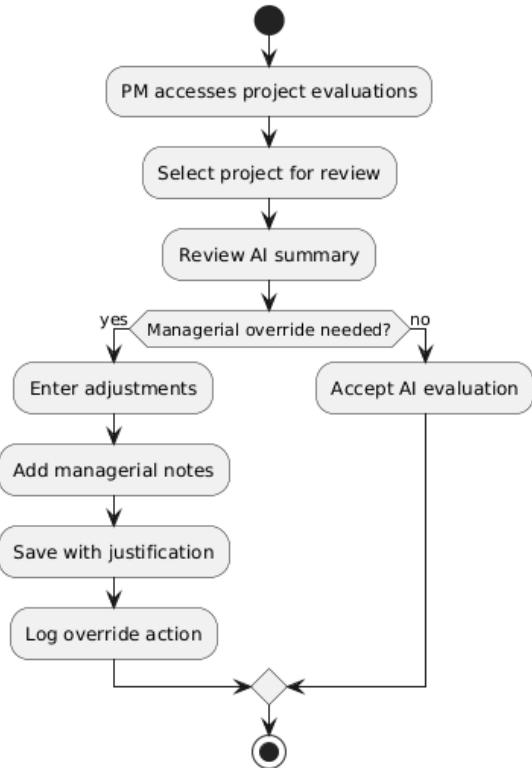
- Use Case Diagram



- Use Case Description

Field	Description
Use Case Name	PM Override Project Evaluations
Primary Actor	Project Manager
Secondary Actors	None
Description	Project managers can review AI-generated project evaluation summaries and make managerial adjustments or annotations.
Preconditions	Project evaluations have been completed by AI system.
Basic Flow	1) PM accesses project evaluations. 2) PM reviews AI-generated summaries. 3) PM makes managerial adjustments. 4) PM adds annotations or notes. 5) System saves override with justification.
Alternative Flows	- Accept as-is: No changes needed.
Postconditions	Overridden evaluation is final and documented.
Exceptions	Insufficient permissions: Restrict to appropriate managerial level.

- *Activity Diagram*



- *Wireframe*

The wireframe shows the 'Override Project Evaluations' interface. At the top, there are four cards displaying project details:

- Pending Review: 1 Pending Review
- Accepted: 1 Accepted
- Overridden: 1 Overridden
- Avg Score: 83

Below these are two filter sections: 'Filter by Status' (All Status) and 'Filter by Course' (All Courses). The main content area displays three projects with their AI scores and override status:

- E-commerce Platform - SPMMP**: AI Score 82, PENDING REVIEW (IT332), Oct 8, 2025. Includes a 'Review Evaluation' button.
- Smart Parking Solution - SPMMP**: AI Score 76 → Final Score 80, OVERIDDEN (IT342), Oct 9, 2025. Includes a 'Review Evaluation' button.
- Hospital Management System - SPMMP**: AI Score 88, ACCEPTED (IT332), Oct 10, 2025. Includes a 'Review Evaluation' button.

Home > Dashboard > Override Project Evaluations

PENDING REVIEW IT332

E-Commerce Platform - SPMP

A, Alice Johnson Submitted October 8, 2025

AI Score 82

Accept as Final

Section Scores    Override History    Annotations

Project Overview Score 18 / 20

AI Evaluation Clear project scope and objectives. Well-defined success criteria.  
AI Score: 18 / 20

Override Score

Organization Structure Score 15 / 20

Risk Management Score 14 / 20

Schedule & Milestones Score 17 / 20

Quality Assurance Score 18 / 20

Override History

No overrides recorded for this evaluation

Home > Dashboard > Override Project Evaluations

PENDING REVIEW IT332

E-Commerce Platform - SPMP

A, Alice Johnson Submitted October 8, 2025

AI Score 82

Accept as Final

Section Scores    Override History    Annotations

Annotations & Notes

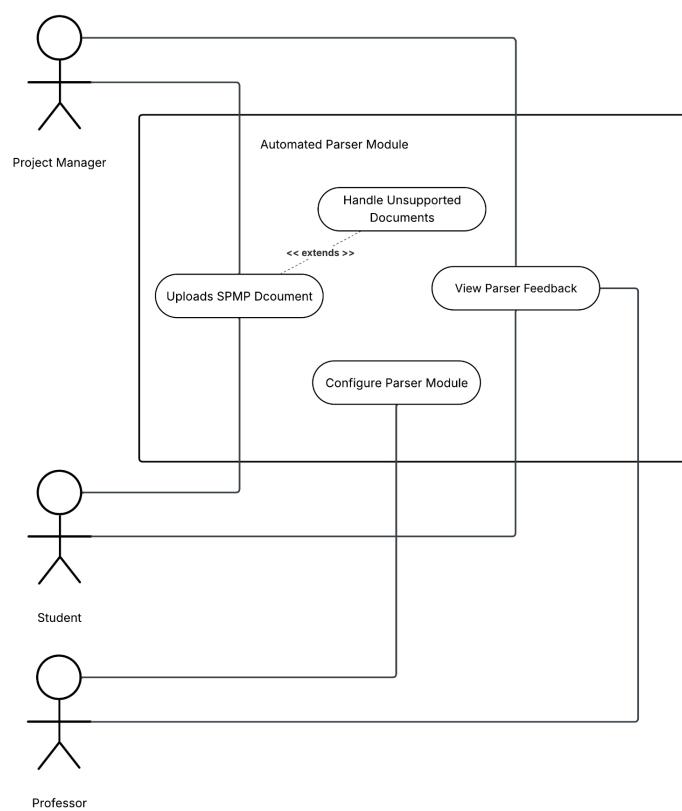
Add Note

No annotations added to this evaluation

## Module 3

### 3.1 Automated Parser Module

- Use Case Diagram



- *Use Case Description*

Use Case ID	UC-3.1
Use Case Name	Upload SPMP Document
Actors	Student, Project Manager
Description	This use case allows a Student or Project Manager to upload an SPMP (Software Project Management Plan) document for automated compliance evaluation against the IEEE 1058 standard. The system validates the uploaded file, preprocesses it, and forwards it to the parser module for clause detection and AI-based compliance analysis. The process extends to 'Handle Unsupported Files' when an invalid or unsupported file type is uploaded.
Preconditions	<ul style="list-style-type: none"><li>- The user must be authenticated.</li><li>- The parser module must be configured by the Professor.</li><li>- The SPMP document must be in a supported format (PDF or DOCX).</li></ul>
Postconditions	<ul style="list-style-type: none"><li>- The SPMP document is successfully uploaded and stored.</li><li>- The system preprocesses the document and applies AI-based clause evaluation.</li><li>- Structured feedback is generated and stored for viewing.</li></ul>

Basic Flow	<ol style="list-style-type: none"><li>1. Student or Project Manager initiates the upload of an SPMP document.</li><li>2. The system validates the file type and format.</li><li>3. The system stores the document temporarily.</li><li>4. The system forwards the document to the parser module.</li><li>5. The parser module is automatically set up for evaluation.</li><li>6. The system preprocesses the document (text extraction and cleanup).</li><li>7. The system detects IEEE 1058 clauses.</li><li>8. The system applies AI compliance evaluation and generates structured feedback.</li><li>9. The system stores the results and notifies the user of a successful upload.</li></ol>
Alternative Flow	<b>Handle Unsupported Files:</b> If the uploaded file format is not supported, the system shows an error message and prompts the user to re-upload a valid document.
Extension Points	Handle Unsupported Files (invalid file upload).
Special Requirements	<ul style="list-style-type: none"><li>- Internet connection required for parser and AI model execution.</li><li>- File upload size must not exceed system limits.</li><li>- Compliance evaluation accuracy depends on parser configuration set by the Professor.</li></ul>

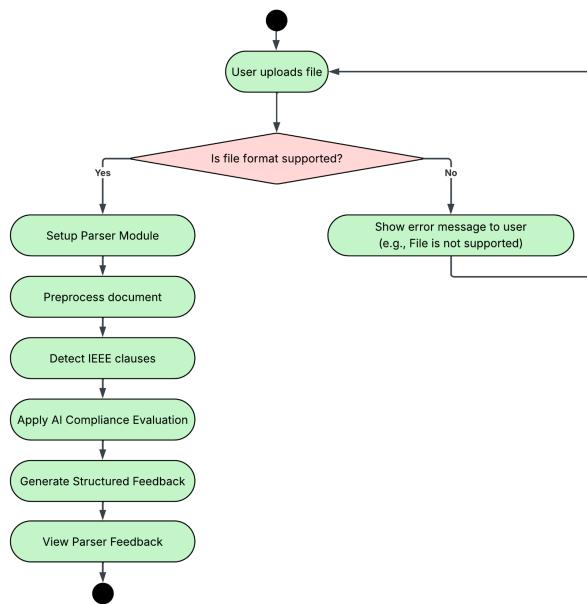
Use Case ID	UC-3.2
Use Case Name	Configure Parser Module
Actors	Professor
Description	This use case allows the Professor to configure the Automated Parser Module by adjusting clause weights, defining rule mappings, and setting evaluation parameters to align with the IEEE 1058 standard. The configuration ensures accurate and explainable compliance assessments.
Preconditions	<ul style="list-style-type: none"><li>- The Professor must be authenticated.</li><li>- The parser module must be initialized.</li><li>- System access permissions must allow configuration changes.</li></ul>
Postconditions	<ul style="list-style-type: none"><li>- Parser configuration (rules, clause weights, and criteria) is saved successfully.</li><li>- The updated configuration is applied to future SPMP evaluations.</li></ul>
Basic Flow	<ol style="list-style-type: none"><li>1. The Professor navigates to the parser configuration page.</li><li>2. The system displays current parser settings and evaluation parameters.</li><li>3. The Professor adjusts rule weights, clause mappings, or AI evaluation criteria.</li><li>4. The Professor saves the configuration changes.</li><li>5. The system validates and stores the updated configuration.</li><li>6. The system confirms successful configuration update.</li></ol>
Alternative Flow	<b>**Invalid Configuration:**</b> If a rule weight or clause definition is invalid, the system notifies the Professor and prompts correction before saving.
Extension Points	None

Special Requirements	<ul style="list-style-type: none"><li>- Configuration changes require administrative privileges.</li><li>- System must maintain version history of configuration updates for traceability.</li><li>- Only the Professor role has permission to modify parser rules.</li></ul>
----------------------	---

Use Case ID	UC-3.3
Use Case Name	View Parser Feedback
Actors	Student, Project Manager, Professor
Description	This use case allows authorized users to view the structured parser feedback generated from the evaluation of uploaded SPMP documents. The feedback includes compliance scores, missing IEEE 1058 clauses, and AI-generated recommendations for improvement.
Preconditions	<ul style="list-style-type: none"><li>- The SPMP document must have been successfully uploaded and processed.</li><li>- The parser feedback must be generated and stored in the system.</li><li>- The user must be authenticated with the appropriate role.</li></ul>
Postconditions	<ul style="list-style-type: none"><li>- The user successfully views the structured compliance feedback.</li><li>- The system logs access activity for auditing and version tracking.</li></ul>
Basic Flow	<ol style="list-style-type: none"><li>1. The user navigates to the feedback or results section.</li><li>2. The system retrieves available parser feedback records.</li><li>3. The user selects a document to view results.</li><li>4. The system displays compliance metrics, missing clauses, and suggestions.</li><li>5. The user may download or print the feedback report.</li></ol>

Alternative Flow	**No Feedback Available:** If the system has no feedback record for the selected document, an informational message is displayed (e.g., 'No results available yet').
Extension Points	None
Special Requirements	<ul style="list-style-type: none"> <li>- Access control must ensure that users only see feedback relevant to their role.</li> <li>- The feedback interface should support exporting reports in PDF or CSV format.</li> <li>- All viewed results are logged for accountability.</li> </ul>

- *Activity Diagram*



- *Wireframe*

The wireframe illustrates the 'Document Upload & Submission' page of the IEEE 1058 SPMP Compliance System. At the top, there is a header bar with the title 'IEEE 1058 SPMP Compliance System' and a subtitle 'Automated Document Analysis & Compliance Evaluation'. Below the header, there are three navigation tabs: 'Student Interface' (purple), 'Project Manager' (white), and 'Professor Interface' (light blue). The main content area is titled 'Document Upload & Submission'. It features a large dashed rectangular area for document upload, containing a yellow folder icon and the text 'Drop your SPMP document here'. Below this, it specifies 'Supported formats: PDF, DOCX (Max 50MB)' and a 'Choose File' button. Underneath the upload area, there are three input fields: 'Document Title' (placeholder 'Enter document title'), 'Project Name' (placeholder 'Enter project name'), and 'Additional Notes' (placeholder 'Any specific requirements or notes...'). At the bottom of the form are two buttons: 'Submit for Analysis' (purple) and 'Save as Draft' (grey).

## Analysis Results & Feedback

**Overall Compliance Score**

**85%**



**Processing Status**

Analysis Complete

Processed on: 2025-01-15 14:30  
Processing time: 8.5 seconds

**Detailed Feedback**

✓ **Section 1.1 - Project Overview**  
Complete and well-structured. Meets IEEE 1058 requirements.  
Confidence: 95%

✗ **Section 2.3 - Risk Management**  
Missing required subsections: Risk Assessment Matrix, Mitigation Strategies  
Confidence: 92% | [View Standard](#)

⚠ **Section 4.1 - Budget Allocation**  
Present but lacks detailed breakdown. Consider adding cost categories.  
Confidence: 87%

[Download Report](#) [Request Override](#) [Submit Revision](#)

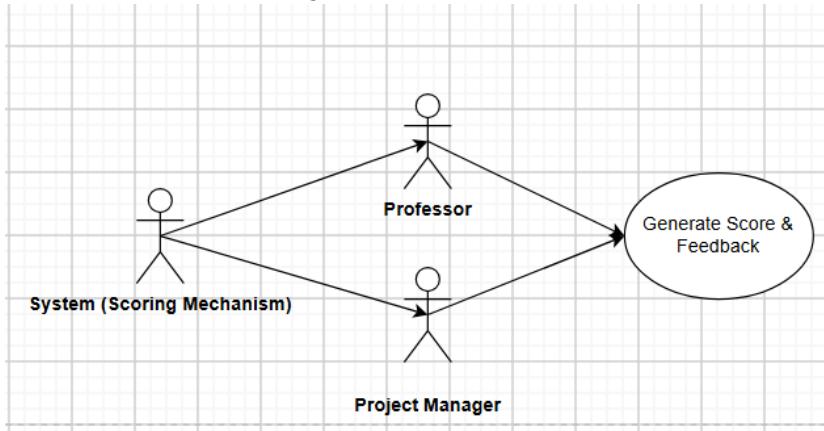
## Submission History

Document	Submitted	Score	Status	Actions
SPMP_v2.1_Final.pdf	2025-01-15	85%	Approved	<a href="#">View</a>
SPMP_v2.0_Draft.docx	2025-01-12	72%	Needs Revision	<a href="#">View</a>

## Module 4

### 4.1 Generate Score & Feedback

- Use Case Diagram

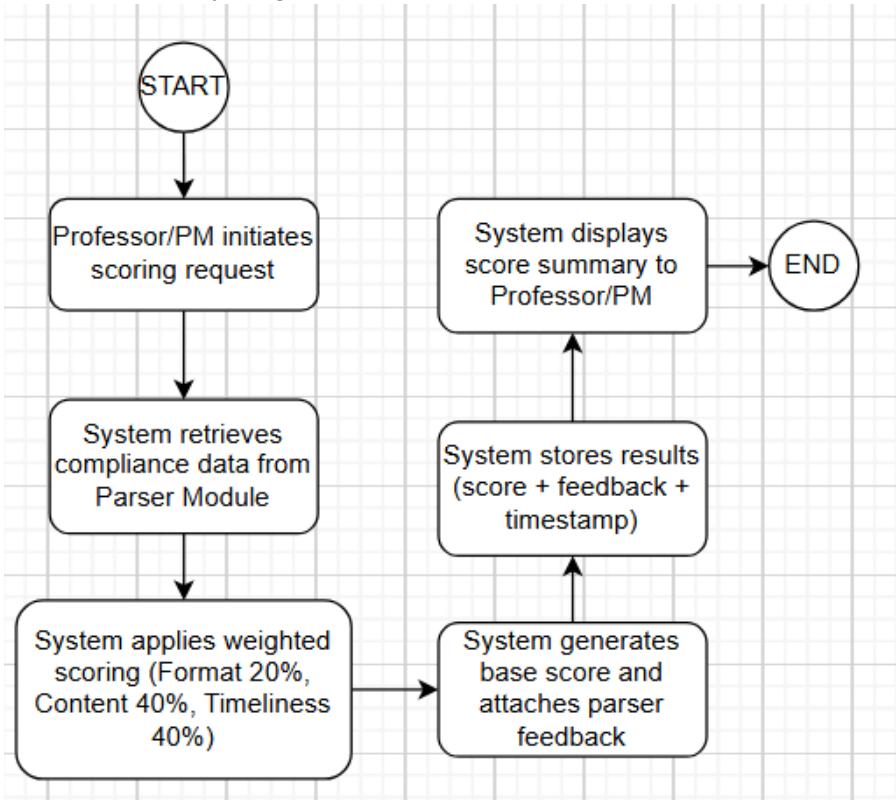


- Use Case Description

Section	Details
Use Case ID	4.1
Use Case Name	Generate Score & Feedback
Primary Actors	System, Professor, Project Manager
Secondary Actor(s)	Parser Module (provides compliance data)
Description	The system generates compliance scores based on data received from the Automated Parser Module. It applies predefined weighted criteria (Format 20%, Content Relevance 40%, Timeliness 40%) and attaches parser feedback for transparency. Professors and Project Managers can trigger and review results.
Preconditions	<ul style="list-style-type: none"><li>- User (Professor or PM) is authenticated.</li><li>- Parser data is successfully received.</li><li>- Scoring weights are configured in the system.</li></ul>

<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. User initiates score generation.</li> <li>2. System retrieves compliance data.</li> <li>3. Applies weighted scoring criteria.</li> <li>4. Generates score and attaches parser feedback.</li> <li>5. Stores result with timestamp and audit log.</li> <li>6. Displays generated score to the user.</li> </ol>
<b>Alternative Flows</b>	<p>AF1: <i>Missing Data Handling</i> – If parser output is incomplete, the system performs partial scoring and flags missing parts.</p> <p>AF2: <i>Batch Scoring (Project Manager)</i> – For 50+ submissions, the system queues and processes sequentially.</p>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- Scores and feedback are stored securely.</li> <li>- Reports are accessible to authorized users only.</li> </ul>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>- Invalid parser data triggers an error and prompts reprocessing.</li> <li>- System timeout logs failure if processing exceeds 5 seconds.</li> </ul>

- *Activity Diagram*



- *Wireframe*

## SPMP Evaluator - Scoring Dashboard

SELECT DOCUMENT:

[Browse](#) [Upload](#)

WEIGHTING CRITERIA:

Format: 20%

Content: 40%

Timeliness: 40%

[Generate Score & Feedback](#) [Cancel](#)

RESULTS

Format Compliance:	18 / 20	<div style="width: 90%;"></div>
Content Relevance:	36 / 40	<div style="width: 90%;"></div>
Timeliness:	38 / 40	<div style="width: 95%;"></div>

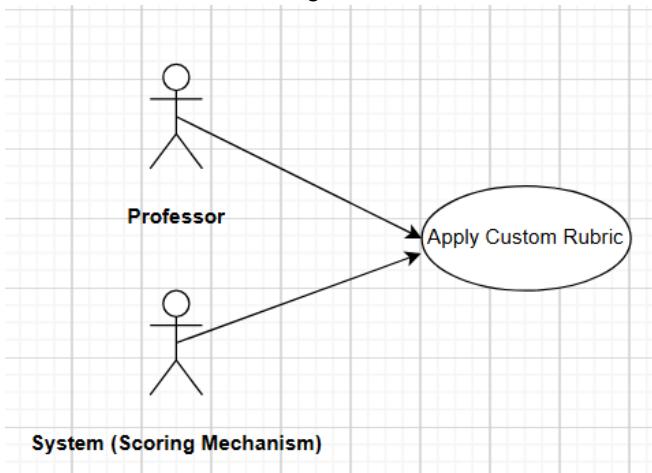
**TOTAL SCORE: 92%**

★ Parser Feedback:  
"Document follows IEEE 1058 structure but lacks detail in Resource Plan section."

[Save Report](#) [Export PDF](#) [View Full Report](#)

## 4.2 Apply Custom Rubric

- Use Case Diagram

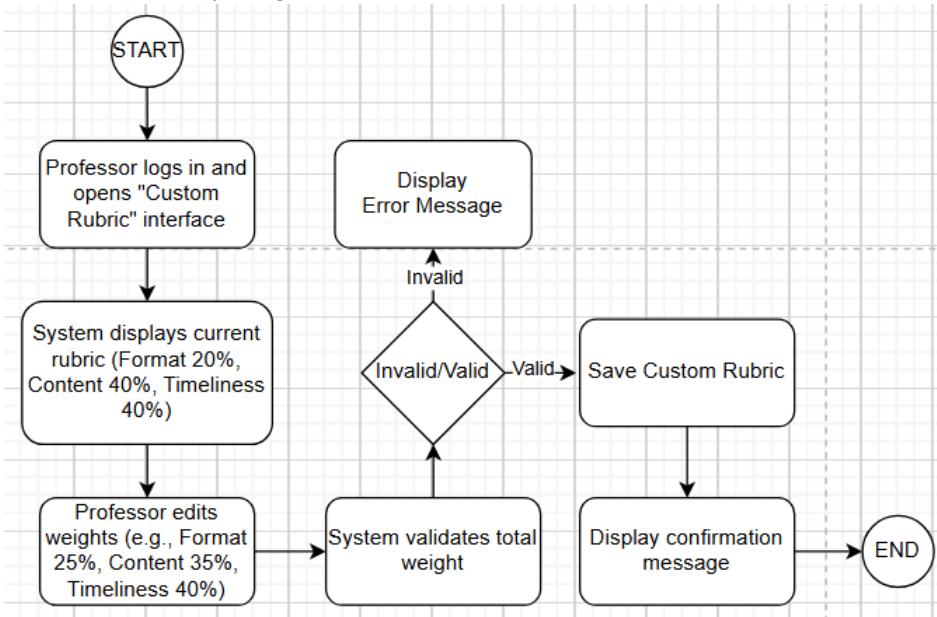


- Use Case Description

Section	Details
Use Case ID	4.2
Use Case Name	Apply Custom Rubric
Primary Actor	Professor
Secondary Actor(s)	System
Description	The professor defines or modifies a grading rubric by assigning custom weights to evaluation criteria (e.g., Format, Content, Timeliness). The system validates that the total equals 100% and applies the custom rubric to future score computations. This allows professors to align AI-assisted scoring with their preferred grading approach while maintaining IEEE 1058 structure.
Preconditions	- Professor is authenticated and authorized. - Parser data exists in the system. - Default rubric (Format 20%, Content 40%, Timeliness 40%) is already configured.

<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>Professor opens the “Custom Rubric” configuration interface.</li> <li>Enters or adjusts percentage weights for each scoring criterion.</li> <li>System validates that total = 100%.</li> <li>System saves rubric and confirms successful update.</li> <li>Future scoring sessions use the new rubric values.</li> </ol>
<b>Alternative Flows</b>	<p>AF1: <i>Invalid Total Weight</i> → If weights ≠ 100%, system displays error and prompts correction.</p> <p>AF2: <i>Revert to Default</i> → Professor clicks “Reset to Default” to restore system’s default rubric.</p>
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>- Custom rubric is saved in database and applied to all future evaluations.</li> <li>- Audit trail records the rubric change (date, time, and user).</li> </ul>
<b>Exceptions</b>	<ul style="list-style-type: none"> <li>- Network or save failure → system logs error and retains old rubric.</li> <li>- Unauthorized access → system denies access and returns to dashboard.</li> </ul>

• *Activity Diagram*



- *Wireframe*

SPMP Evaluator - Custom Rubric Settings

CURRENT RUBRIC:

Format:	25	%
Content Relevance:	35	%
Timeliness:	40	%

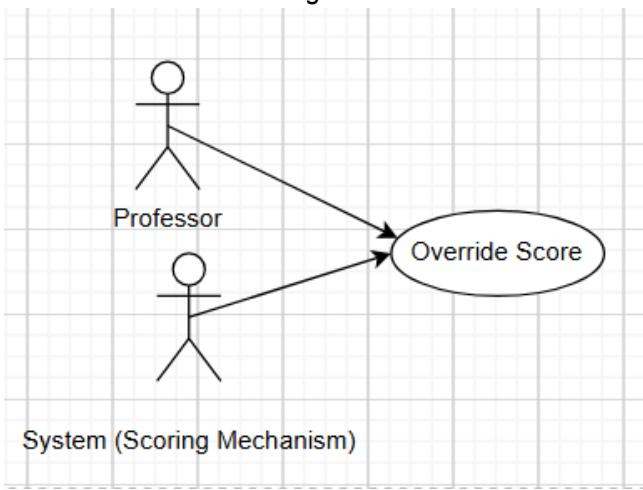
**TOTAL: 100%**

**Save Custom Rubric**   **Reset to Default**

**Validation:**  
"Rubric successfully updated."

#### 4.3 Override Score

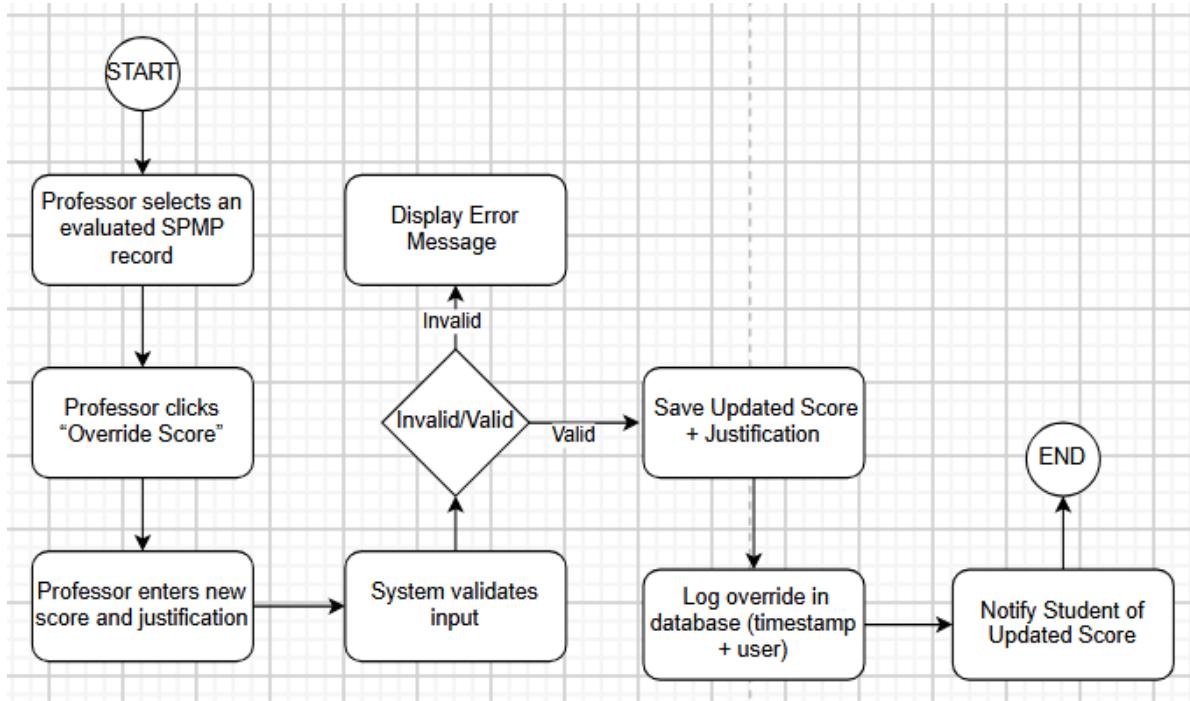
- *Use Case Diagram*



- *Use Case Description*

Section	Details
<b>Use Case ID</b>	4.3
<b>Use Case Name</b>	Override Score
<b>Primary Actor</b>	Professor
<b>Secondary Actor(s)</b>	System
<b>Description</b>	This use case allows professors to manually override an AI-generated SPMP evaluation score. The override may be done to correct discrepancies or apply subjective assessment criteria (e.g., originality, research depth). The system ensures the action is justified, logs all overrides with timestamp and justification, and updates the official record.
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- Professor is authenticated and authorized.</li><li>- A score already exists for the selected SPMP submission.</li><li>- The override justification form is available.</li></ul>
<b>Basic Flow</b>	<ol style="list-style-type: none"><li>1. Professor opens a student's evaluation report.</li><li>2. Reviews AI-generated score and feedback.</li><li>3. Clicks "<b>Override Score</b>" button.</li><li>4. Inputs new score and justification comment.</li><li>5. System validates input (score range, required comment).</li><li>6. System saves updated score with justification and timestamp.</li><li>7. System notifies the student of the updated grade.</li></ol>
<b>Alternative Flows</b>	<p>AF1: <i>Invalid Score Range</i> – If input score is outside 0–100%, the system shows a validation error and requests correction.</p> <p>AF2: <i>Missing Justification</i> – System disallows submission until justification comment is provided.</p>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- The new score replaces the AI-generated one.</li><li>- Justification and timestamp logged for transparency.</li><li>- Updated report visible to student and professor.</li></ul>
<b>Exceptions</b>	<ul style="list-style-type: none"><li>- Network failure → System saves override locally and retries synchronization.</li><li>- Unauthorized access → System denies the override attempt and logs it.</li></ul>

- *Activity Diagram*



- *Wireframe*

**SPMP Evaluator - Override Score**

Student: John Cruz - Group A

Original Score: **86%** AI-generated

Feedback: "Missing resource allocation details."

Override Score: **90**

Justification:

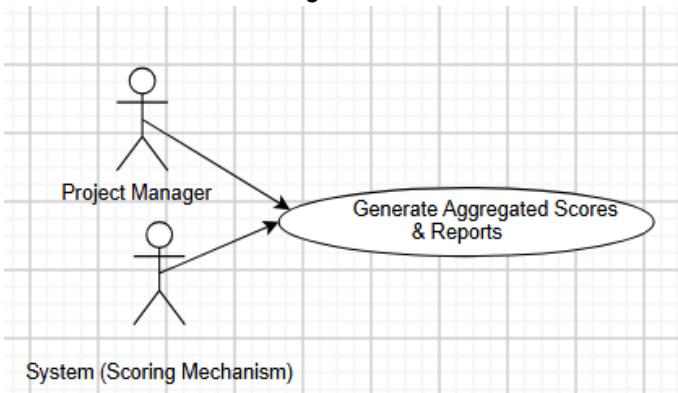
Professor adjusted score due to additional documentation provided.

**Save Override**    **Cancel**

**System Message:**  
Override saved successfully.  
Students will be notified of the change.

#### 4.4 Generate Aggregated Scores & Reports

- *Use Case Diagram*

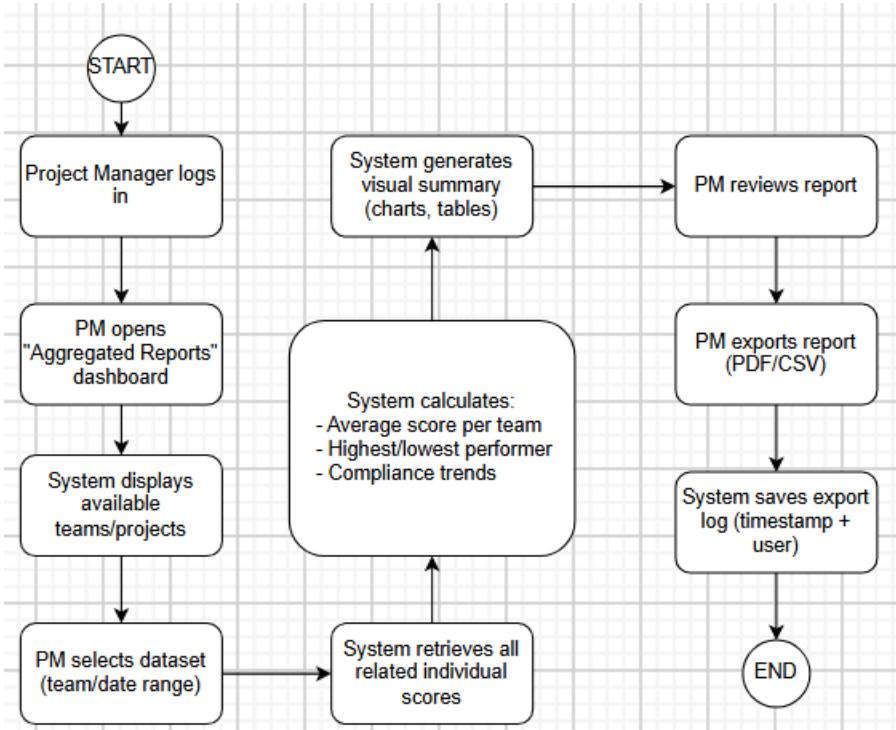


- Use Case Description

Section	Details
<b>Use Case ID</b>	4.4
<b>Use Case Name</b>	Generate Aggregated Scores & Reports
<b>Primary Actor</b>	Project Manager
<b>Secondary Actor(s)</b>	System
<b>Description</b>	The Project Manager can initiate batch scoring or view aggregated results of multiple SPMP submissions. The system calculates team or group averages, identifies compliance trends, and generates visual or tabular performance reports. Reports may emphasize timeliness, consistency, and IEEE 1058 compliance levels.
<b>Preconditions</b>	- Project Manager is authenticated. - Individual SPMP scores are already available in the database. - Parser and scoring modules (UC 4.1) have been executed successfully.
<b>Basic Flow</b>	1. Project Manager opens the “Aggregated Reports” dashboard. 2. Selects batch, team, or date range for analysis. 3. System retrieves all corresponding SPMP scores and feedback. 4. System calculates aggregated metrics (mean, min, max, standard deviation). 5. Generates report with visual charts or summary table. 6. System displays and allows export (PDF/CSV).
<b>Alternative Flows</b>	AF1: <i>No Available Data</i> – System displays “No records found” message and suggests re-running individual scoring. AF2: <i>Partial Data</i> – System calculates using available scores and flags incomplete submissions. AF3: <i>Custom Filters</i> – Project Manager applies filters (group, project type, timeframe) before generating report.
<b>Postconditions</b>	- Aggregated report generated and saved for future reference. - Reports viewable/exportable only by authorized users.

<b>Exceptions</b>	- Database connection lost → System notifies user and aborts aggregation. - File export failure → System retries or provides manual download option.
-------------------	---

- *Activity Diagram*



● Wireframe

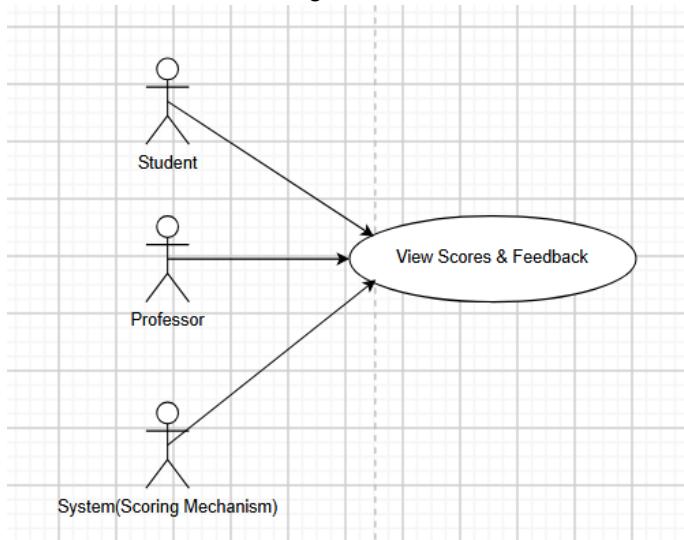
SPMP Evaluator - Aggregated Reports Dashboard

The wireframe illustrates the layout of the dashboard:

- Header:** SPMP Evaluator - Aggregated Reports Dashboard
- Filtering:** Filter by: Team ▾, Date Range ▾, Project Type ▾
- Action Buttons:** + Generate Report, Export PDF, Export CSV
- RESULTS Section:**
  - Team A - Average Score: 88%
  - Team B - Average Score: 91%
  - Team C - Average Score: 84%
- Compliance Trend Chart:** A horizontal bar chart showing compliance levels across three categories.
- Timeliness Summary:** On-time (92%) vs Late (8%)
- Remarks:** "Team B showed best adherence to IEEE 1058"
- Bottom Buttons:** View Individual Scores, Download Full Report

#### 4.5 View Scores & Feedback

- Use Case Diagram

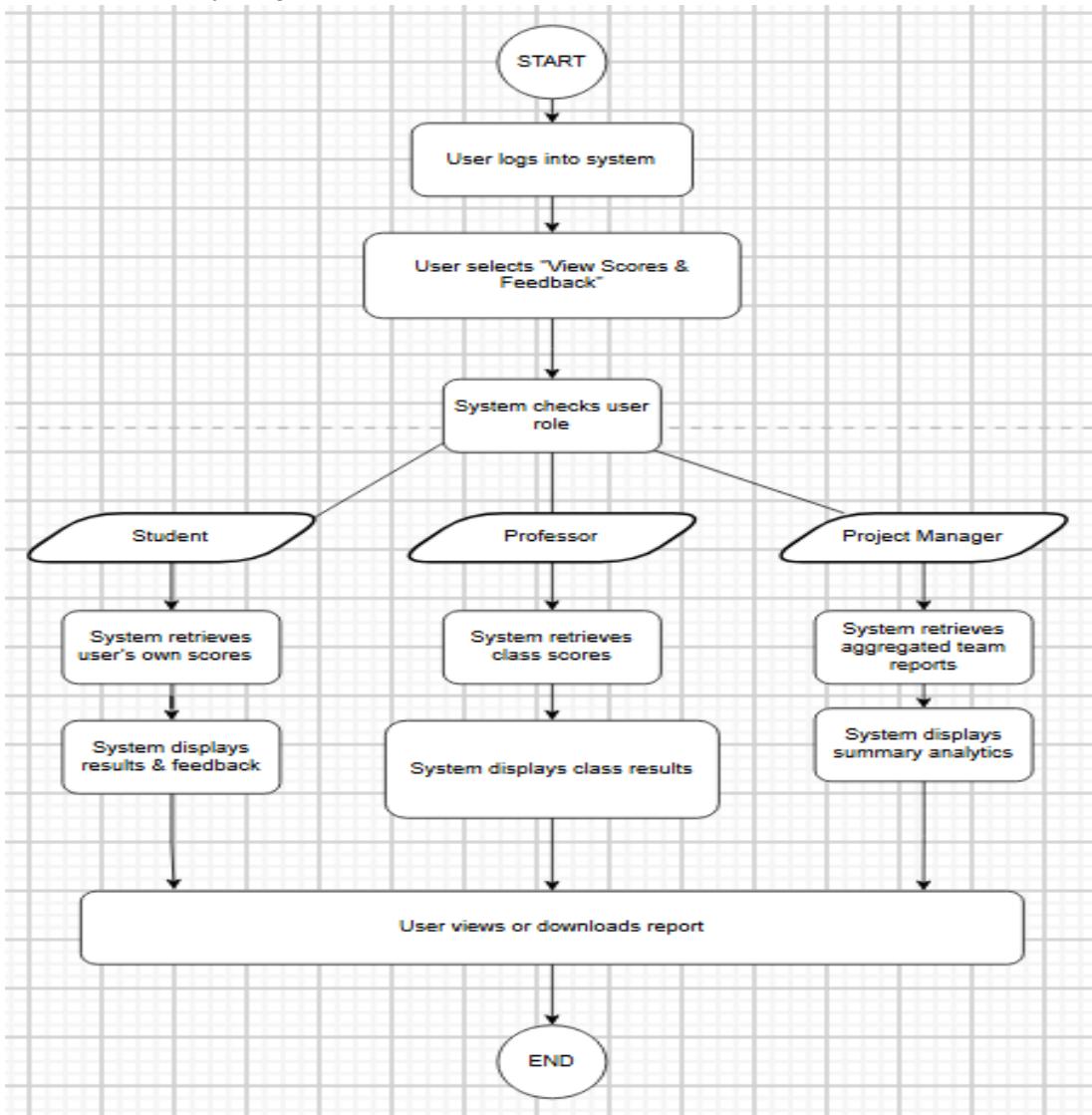


- Use Case Description

Section	Details
Use Case ID	4.5
Use Case Name	View Scores & Feedback
Primary Actors	Student, Professor, Project Manager
Secondary Actor(s)	System
Description	This use case allows authenticated users to view scoring results and feedback. Students can view only their own scores and feedback. Professors can view all their students' evaluations for grading purposes, while Project Managers can access aggregated results for performance tracking. The system ensures proper access control based on user roles.
Preconditions	- User is authenticated and authorized. - Scoring and feedback data are available in the system. - Role-based permissions are configured.

<b>Basic Flow</b>	1. User logs in and opens the “View Scores & Feedback” page. 2. System identifies user role. 3. For Students: system retrieves and displays only their own SPMP score and feedback. 4. For Professors: system retrieves all student scores under their supervision. 5. For Project Managers: system retrieves team-level or batch-level scores. 6. User views feedback, downloads report, or exports data if allowed.
<b>Alternative Flows</b>	AF1: <i>No Data Found</i> – System displays “No results available.” AF2: <i>Filter Applied</i> – User filters results by date, project, or team. AF3: <i>Unauthorized Access</i> – User tries to view restricted data → system denies and logs attempt.
<b>Postconditions</b>	- Feedback and scores displayed securely. - User may export or download report if permitted. - All access attempts logged for auditing.
<b>Exceptions</b>	- Database or server error → System displays “Unable to load data.” - Session timeout → User is logged out automatically.

- *Activity Diagram*



- *Wireframe*

The wireframe shows a user interface for managing software project management plans. It includes a header with a welcome message and user role, a search/filter bar, and a results section. The results section is divided into sections for Students, Professors, and PMs, each displaying relevant data such as SPMP titles, scores, feedback, and compliance charts.

Welcome, [User Name] (Role: Student / Professor / PM)

Filter: | Prittor ▾ Date | Team ▾ | Search

————— RESULTS ————

For Students:

SPMP Title: "Team A Documentation"  
Score: 92%  
Feedback: "Excellent adherence to IEEE 1058,  
but lacks testing plan details."

For Professors:

[Student] [Score] [Remarks] [Date Evaluated]

John Cruz	92%	Passed	09/15/25
Maria DelaC	87%	Needs Rev	09/15/25

For PMs

Team A Avg: 90% Team B Avg: 86% Team C: 93%  
Compliance Chart:

### **3.4 Non-functional requirements (VERANO)**

#### **Performance (High)**

- *The login process should complete quickly ( $\leq 2$  seconds) under normal conditions.*
- *Transactions (uploads, task creation, monitoring, report generation) must complete within 3 seconds under normal load.*
- *File analysis must finish in under 10 seconds for a standard-length report (<50 pages).*
- *Score generation must complete within 5 seconds per submission under normal load.*

## **Security (High)**

---

- *The system must enforce encryption, secure communication (SSL/TLS), and safe password handling to prevent breaches.*
- *Enforce strict role-based access control for all features.*
- *All uploaded files must be sanitized, encrypted in transit and at rest, and isolated from training datasets.*
- *Only authorized users (Professors and PMs) can view, modify, or override scores.*

## **Reliability (High)**

---

- *The system should remain consistently available (99.9% uptime) for authentication services.*
- *Ensure 99.9% uptime for role-specific operations and AI services.*

## **Accuracy & Precision (High)**

---

- *The parser must correctly identify formatting and compliance issues with at least 85–90% accuracy, and avoid false positives.*

## **Consistency (High)**

---

- *Given the same input file, the parser must return the same result every time (determinism).*
- *Identical inputs must always yield the same score unless deliberately overridden.*

## **Usability (Medium)**

---

- *The login interface should be simple enough for first-time users to log in without confusion.*
- *Tailor each role's UI to their responsibilities for clarity and simplicity.*

## **Auditability (Medium)**

---

- *All login attempts (successful or failed) must be logged for monitoring and security reviews.*
- *Log all role actions (uploads, overrides, assignments, AI report generation) with timestamps.*
- *Every evaluation must be logged with input metadata, compliance results, and timestamps.*

## **Explainability (Medium)**

---

- *Feedback must clearly indicate which standard was violated and why.*

## **Scalability (Medium)**

---

- *The parser should support at least 50–100 concurrent file evaluations without degrading accuracy or speed.*
- *The system must support batch scoring (e.g., 50+ submissions in one run) without degradation.*

## **Maintainability (Medium)**

---

- *Standards-checking rules (IEEE, APA, etc.) must be easily updatable without retraining the entire AI model.*
- *Role-specific scoring criteria must be configurable without rewriting the entire module.*

