Term Academic Year: 2025/2026

Course Coordinator: Dr. Rahma Ahmed

Software Project Management Plan (SPMP)

Project Title: [MaKSAd]
[Volunteerism and Community Events Management System in KSA]

Team Name & Members:

MaKSAd

Lama Aljalal
Khawlah Abaalkhail
Ghala Alghamdi
Reemas Aleid
Haya Alwuhaibi
Jood Al Turki
Aljawharah Alghudiyan

Advisor: [Maha Nasser Asiri]
Version: [1.0]
Date: [18-10-2025]

Table of Contents

Note: Click on the page name to open the corresponding section

1. Project Overview

1.1 Purpose, Scope, and Objectives

Purpose

This project aims to develop a digital platform that connects community organizations and volunteers in the Kingdom Saudi Arabia. The system will allow organizers to create and manage community events or volunteer activities, while volunteers can register, participate, and track their contributions. In addition, the platform will provide a feature to record and calculate volunteer hours, helping participants monitor their efforts and achievements. The platform will be designed to be user-friendly, accessible, and efficient in promoting community engagement and volunteerism.

Scope of the Project

Includes:

• User registration and login (Organizers & Volunteers).
• Event creation, management, and registration.
• Volunteer participation tracking (attendance, hours).
• Basic reporting for organizers (number of participants, hours logged).

Excludes (Out of Scope):

• Online payment systems for events.
• Integration with external government platforms (e.g., Absher).

Objectives

• Encourage efficient management of community events and volunteer activities.

• Provide volunteers with an easy way to discover, register, and track their participation.

• Facilitate event management, attendance tracking, and report generation for organizers.

• Encourage a culture of community service that supports the social development objectives of Saudi Vision 2030.

• Improve transparency by keeping clear records of participation and volunteer hours.

• Support inclusivity by making opportunities widely available across different regions and groups.

1.2 Assumptions, Constraints, and Risks

Assumptions

- All team members have the necessary personal hardware (laptops) and stable internet access to support development and collaboration.
- The chosen open-source tools (NetBeans, MySQL Workbench, JDBC, GitHub, Figma) will be compatible and sufficient for all project needs.
- Community organizations in KSA will be willing to adopt and use a new platform for managing volunteer activities.
- The team will be able to complete the core features within the tight academic timeline.

Constraints

- **Time:** The project must be completed within the **1st Semester-2025/2026** academic timeline, with the final review due by **December 13**.
- **Scope:** The platform will **exclude** online payment systems for events and integration with external government platforms (e.g., Absher).
- **Tools:** The development is limited to the specified tools and technologies: **NetBeans for Java, MySQL Workbench, JDBC, Google Drive, GitHub, and Figma**.

Top Risks

1. **Time Constraints:** Not completing feature implementation within the rigid semester timeline (Nov 29 - Dec 6).
2. **Scalability Issues:** Difficulty managing performance as the number of users (volunteers and organizers) grows rapidly.
3. **Technical Integration:** Compatibility issues when integrating the multiple planned tools and technologies.
4. **Technical Setbacks:** Handling unexpected bugs, errors, and technical setbacks during the Development and Testing phases.

1.3 Project Deliverables

Deliverable	Format	Expected Submission Date
Project Proposal	PDF	18/9/2025
SPMP	PDF	18-10-2025
Project status report	PDF	23-10-2025
SRS	PDF	6-11-2025
SDS	PDF	20-11-2025
STS	PDF	4-12-2025
Presentation	PDF	14-12-2025

1.4 Schedule Summary

Phase	Activities and Deliverables	Target Week	Target Date	Description and Output
Project Definition	Define the project idea and assign team roles.	Week 3	Sept 13 2025	Establish overall vision, team structure, and initial objectives.
Planning Phase	Conduct preliminary research and outline objectives, scope, and details.	Week 4	Sept 20 2025	Project Proposal submission including title, objectives, scope, and team details.
Project Management	Draft and finalize the Software Project Management Plan (SPMP).	Week 8	Oct 18 2025	SPMP deliverable with timeline, risk plan, and tracking approach.

Status Report 1	Review progress and list updates on initial development tasks.	Week 9	Oct 25 2025	Status Report 1 summarizing team progress.
Software Requirement Specification (SRS)	Develop UI/UX wireframes and database schema, document requirements of the system.	Week 11	Nov 8 2025	SRS defining functional and non-functional requirements.
Software Design Description (SDD)	Implement core design features and prepare design documentation.	Week 13	Nov 22 2025	SDD deliverable and initial code submission showing system design progress.
Status Report 2	Evaluate development progress and highlight remaining tasks or issues.	Week 13	Nov 22 2025	Status Report 2 submission with updated overview of work completed.
Software Test Specification (STS)	Conduct testing, debugging, and finalize testing documentation.	Week 15	Dec 6 2025	STS deliverable with verification and validation results.
Final Project Code	Finalize, implement, and document the system code and database.	Week 16	Dec 13 2025	Final completed code submission including documentation.
Final Review & Demo	Perform final integration and deliver the project presentation and demo.	Week 16–17	Dec 18 2025	Final presentation and demo demonstrating major components and design decisions.

1.5 References

- IEEE 1058, IEEE Standard for Software Project Management Plans. IEEE, 1998.
- I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson Education, 2015.
- R. S. Pressman and B. Maxim, Software Engineering: A Practitioner's Approach, 8th ed. New York, NY, USA: McGraw-Hill, 2015.

1.6 Definitions and Acronyms

Acronyms	**Definitions**
MaKSAd	The project title, derived from the Arabic word"مقصد "meaning purpose.

	It represents a volunteering management system in Saudi Arabia.
SPMP	Software Project Management Plan – a document describing how the project will be managed.
SRS	Software Requirements Specification – defines the system's requirements.
SDS	Software Design Specification – describes the design and architecture of the system
STS	Software Test Specification – explains how the system will be tested.
DB	Database – used to store project data.
API	API: Application Programming Interface – allows communication between system components.
Figma	Design tool used to create the system's UI.
UI	Software Test Specification – explains how the system will be tested.

2. Project Organization

2.1 External Interfaces

The primary external stakeholder is the **Course Instructor (Ms.** Maha Nasser Asiri**)**, who serves as the project advisor, providing guidance, evaluating deliverables, and approving key milestones.
Other external interfaces include:

- **Community organizations (clients)**: potential real-world stakeholders who will provide feedback on the usability of the volunteer event management functions.
- **Volunteer users**: considered external end-users whose needs shape the design and testing stages.

External Communication Flow:

- Instructor → Team Leader → Project Members
- Organizations / Users → Team Leader → Relevant Technical Leads

This ensures all communication passes through a clear, structured channel to maintain consistency and accountability.

2.2 Internal Structure

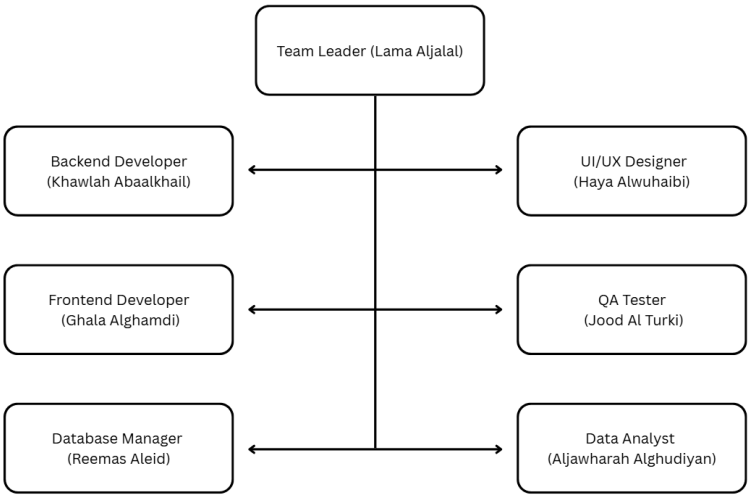
The **MaKSAd** team follows a **functional flat structure**, ensuring flexibility and shared responsibility. Each member leads a specific project area but collaborates across phases.

Internal Communication Tools:

- **WhatsApp** – for daily updates and quick coordination
- **Google Drive** – for shared documentation and deliverables

- **GitHub** – for version control and collaborative development
- **Weekly meetings** – to review milestones, task completion, and blockers

Organizational Chart:



2.3 Roles and Responsibilities

Each member's role and responsibilities are defined as follows:

Member	Role	Responsibilities
Lama Tariq Aljalal	Team Leader & System Designer	Oversee project execution, ensure integration between modules, manage deadlines, and communicate with the instructor.
Khawlah Majid Abaalkhail	Backend Developer	Develop server-side logic, manage database connections, ensure data security and system functionality.
Ghala Ahmed Alghamdi	Frontend Developer	Implement the user interface, ensure responsive design, and integrate front-end with backend services.
Reemas Tariq Aleid	Database Manager	Design and manage database schema, handle data integrity, and create optimized SQL queries.
Haya Sami Alwuhaibi	UI/UX Designer	Create intuitive layouts and user flows that enhance usability for both volunteers and organizers.
Jood Hani Al Turki	QA Tester	Test features, identify bugs, document defects, and ensure product quality before delivery.
Aljawharah Abdulhameed Alghudiyan	Data Analyst	Analyze volunteer participation data, generate reports, and support evaluation metrics.

The **Team Leader** supervises the workflow, while each lead ensures that their respective components meet quality and schedule standards.

3. Managerial Process Plans

3.1 Estimates & Staffing

	A	B	C	D
1	Project Phase	Estimated Duration	Team Members Working	Role Allocation Example
2	Planning (Oct 18)	1 week	All 7 members	Team Leader (Lama), Data Analyst (Aljawharah)
3	Design (Nov 22)	5 weeks	3 members	Team Leader & System Designer (Lama), UI/UX designer (Haya)
4	Development (Nov 29)	1 week	2 members	Backend Developer (Khawlah), Frontend Developer (Ghala)
5	Testing (Dec 6)	1 week	2 members	QA Tester (Jood), Database Manager (Reemas)
6	Final Review (Dec 13)	1 week	All 7 members	All members for Presentation and Documentation

3.2 Work Plan

Work Activity	Resources	Duration	Deliverables	Acceptance Criteria	Predecessor	Successor
Requirements	Laptops, Internet, Google Drive	1 week	SRS Document	All requirements clearly defined and approved by advisor	--	Design
Design	Figma, MySQL Workbench, Google Drive	1 week	SDD Document	Design reviewed and validated by team	Requirements	Coding
Coding	NetBeans (Java), MySQL, JDBC, GitHub	1 week	Working prototype with full functionality	Code runs without errors and integrates all main features	Design	Testing
Testing	JUnit, Manual Testing tools, GitHub Issues	1 week	STS Document	All test cases pass successfully	Coding	--

3.3 Project Tracking Plan

Progress Monitoring

Progress will be tracked through regular conversations and check-ins. Each member will record their completed tasks and any issues they encounter as they go. Recurring deadlines for each major deliverable will help the team stay on schedule and allow us to spot delays early. These updates will make it easy to see overall progress and what still needs to be done.

Change Approval

If any change to the project is proposed, it will first be reviewed to understand why it is needed and whether it benefits the project. Only if the team agrees on the change that it will be implemented. Afterwards, we will check its impact to make sure it actually improves the project and does not create new issues.

Metrics

Progress will be measured using simple metrics to keep the project organized. These include % of tasks completed, number of bugs fixed, schedule dates, and a list of deliverables. These metrics will guide the project in case of any delays and help evaluate performance over time.

3.4 Risk Management Plan

3.4.1. Risk: Time Constraints

- **Likelihood: Medium**
- **Impact: High**
- **Mitigation Strategy: Prioritize core features (MVP) and defer non-critical additions. The Team Leader must strictly monitor the Development milestone due on November 29.**
- **Responsible Role: Team Leader (Lama Tariq Aljalal)**

3.4.2. Risk: Technical Integration/Bugs

- **Likelihood: Medium**
- **Impact: Medium**
- **Mitigation Strategy: Standardize the development environment using NetBeans, MySQL Workbench, and JDBC early. The QA Tester will manage bug fixes during the Testing phase (Dec 6) and use GitHub for issue tracking.**
- **Responsible Role: QA Tester (Jood Hani Al Turki)**

3.4.3. Risk: Scalability Issues

- **Likelihood: Medium**
- **Impact: Medium**
- **Mitigation Strategy: Implement best practices for database schema during the Design phase (Nov 22). The Backend Developer and Database Manager must focus on optimized and efficient code/queries.**
- **Responsible Role: Database Manager (Reemas Tariq Aleid) / Backend Developer (Khawlah Majid Abaalkhail)**

3.4.4. Risk: Documentation and Final Review Quality

- **Likelihood: Low**
- **Impact: Low**
- **Mitigation Strategy: The Team Leader must oversee documentation and the presentation for the Final Review (Dec 13). Google Drive will be utilized for collaborative and continuous documentation.**
- **Responsible Role: Team Leader (Lama Tariq Aljalal)**

3.5 Closeout Plan

The project will officially end after the final code submission in Week 16 and the presentation and demo in Week 17. At this stage, the MaKSAd platform will have been fully developed, tested, and integrated. The system will provide a working volunteering platform and all planned deliverables will have been submitted.

All final files and documents will be organized and stored in a shared location to keep the project accessible. The project will be officially closed once the supervisor reviews and approves the submission.

4. Technical Process Plans

4.1 Process Model

As a team, we adopted the Agile Process Model for the development of **MaKSAd.**

This model was selected because it supports flexibility, iterative development, and continuous improvement throughout the semester. It allows the team to divide the work into smaller increments, receive feedback regularly, and make adjustments when necessary.

Each increment focuses on a specific functional part of the system, such as user registration, event management, and volunteer tracking, which helps ensure steady progress, maintain quality, and minimize risks related to time constraints or technical challenges.

4.2 Methods, Tools, and Techniques

The development process relies on a combination of programming tools, design platforms, and collaboration techniques to ensure efficiency and consistency.

Category	Tools / Methodes	Purpose
Programming Language	Java (NetBeans IDE)	Main development environment for backend and interface logic.
Database	MySQL Workbench with JDBC	Data storage, connection setup, and query management.
Version Control	GitHub	Code versioning, collaboration.
Design & Prototyping	Figma	Designing the user interface and preparing system wireframes.
Documentation & Sharing	Google Drive	Storing shared reports and project files.
Testing	Manual and functional testing	Ensuring feature accuracy, performance, and user satisfaction.

4.3 Infrastructure

The project development environment consists of the following setup:

- **Hardware:** Personal laptops (8 GB RAM minimum) with stable internet connectivity to support development and collaboration.
- **Software** Environment:
  - NetBeans IDE for Java development.
  - MySQL Workbench for database operations.
  - GitHub for source control and collaboration.
  - Figma for interface design and layout.
  - Testing Environment: Localhost setup using JDC to connect Java applications with the MySQL database.

All team members maintain a consistent environment to prevent compatibility issues and ensure seamless integration and testing.

4.4 Product Acceptance

The project will be accepted when the following conditions are met:

- All functional requirements are fully implemented and verified through testing.
- The system performs the intended tasks without errors, including user registration, event creation, volunteer tracking.
- The user interface is functional, responsive, and user-friendly.
- The project advisor, Ms. Maha Nasser Asiri, approves the final system after the presentation and demo.

Once these criteria are satisfied, the final version of **MaKSAd** will be considered complete and ready for submission.

5. Supporting Process Plans

5.1 Documentation Plan

All project documentation, including the Proposal, SPMP, SRS, SDS, STS, and Final Report, will be stored and managed in a shared Google Drive folder to ensure accessibility and consistency across the team. Each document will be created using Microsoft Word and exported as PDF before submission.
The Supporting Processes Lead, **Khawlah Majid Abaalkhail**, is responsible for organizing, reviewing, and maintaining the documentation to align with IEEE standards and the course requirements.
Before submission, each deliverable will be reviewed by the Team Leader and QA Tester to verify quality, completeness, and adherence to project objectives.

5.2 Quality Assurance

Quality assurance aims to ensure that all deliverables meet the project requirements and maintain a high standard of accuracy, reliability, and usability.
The QA process will include:

- Conducting peer reviews for both code and documents.
- Performing functional and integration testing on major modules, such as registration, event creation, and volunteer tracking.
- Holding regular review meetings after each milestone to assess progress and identify improvement areas.

All QA results and testing feedback will be documented in Google Drive and GitHub for traceability.

Detected issues will be tracked, discussed during weekly meetings, and resolved before final submission.

5.3 Configuration Management

The project will use **GitHub** as the main platform for version control and collaborative development.

Each module will be developed on a separate branch and merged into the main branch after review and successful testing. Commit messages will follow a consistent naming format (e.g., "update: registration form validation") to maintain clarity.

Major versions will be tagged according to key phases such as **v1.0 (Design)**, **v2.0 (Development)**, and **v3.0 (Testing)**.

This approach ensures proper tracking of changes, avoids conflicts, and support organized teamwork.

5.4 Problem Resolution

A structured approach will be followed to report and resolve issues identified during development or testing.

All bugs or enhancement requests will be recorded in the **GitHub Issues Tracker** and assigned a priority level:

- **High:** Critical issues affecting main system functionality.
- **Medium:** Moderate issues that need to be fixed before the final review.
- **Low:** Minor or cosmetic adjustments.

The Supporting Processes Lead will monitor the issue tracker and coordinate with the QA Tester to ensure timely resolution.
Resolved issues will be verified and closed after testing confirmation.

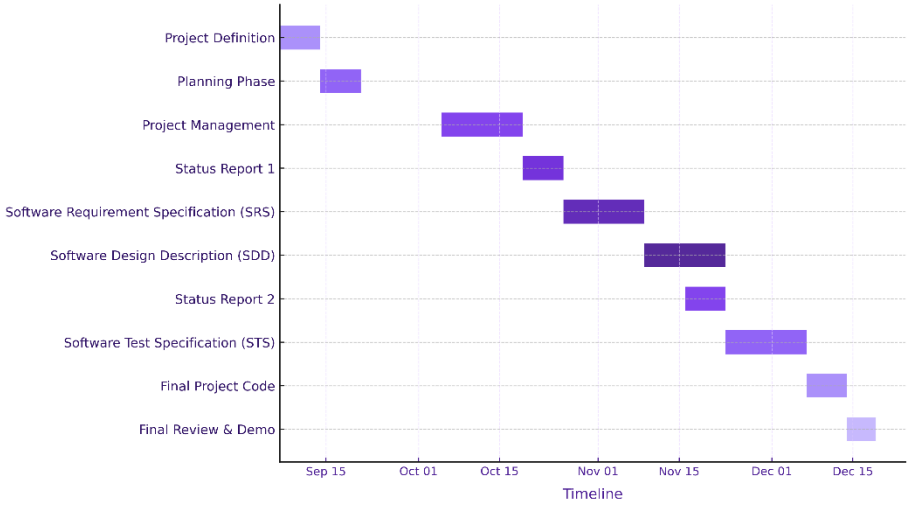
6. Appendices

6.1 Glossary

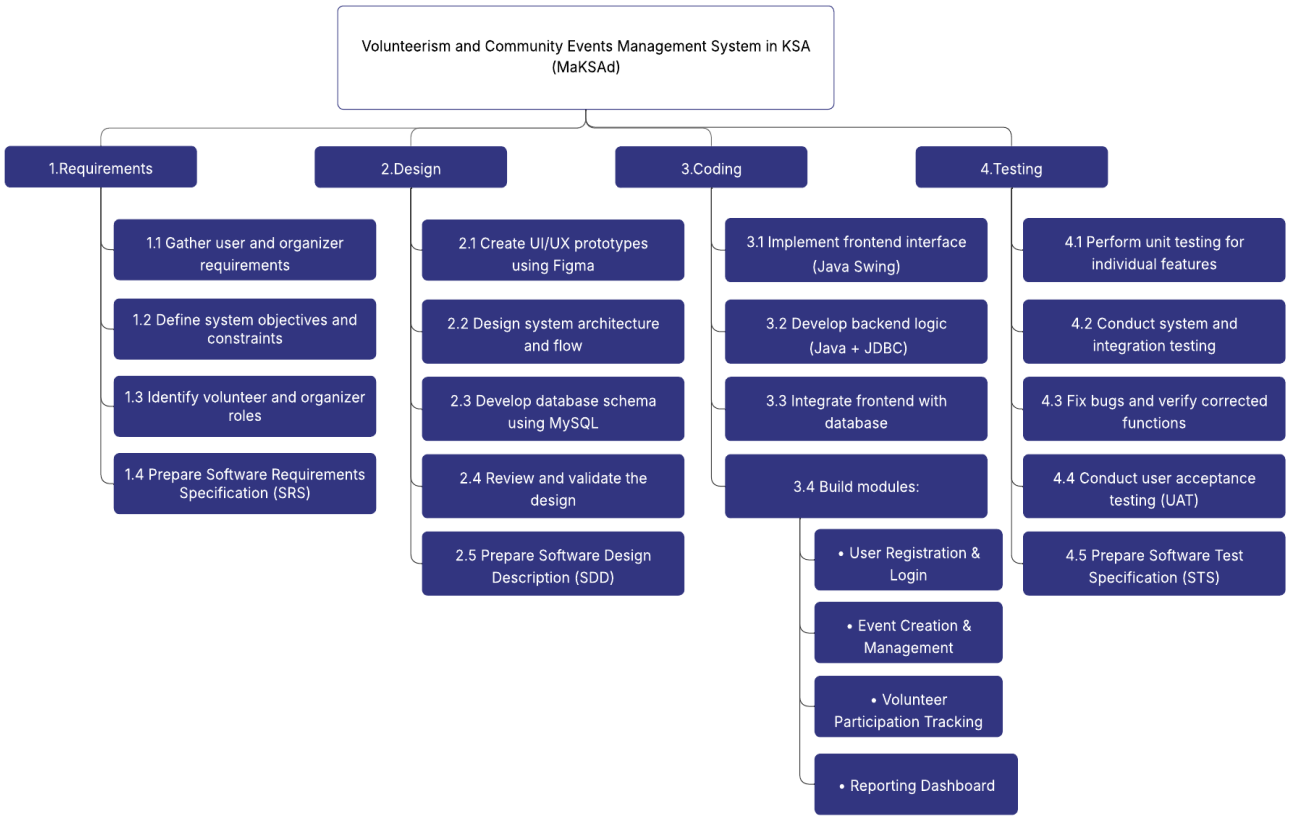
Organizer	A person that creates and manages the volunteering events.
Volunteer	A registered user who participates in the events.
Event	A volunteering activity created by an organizer.
Dashboard	The control panel for the organizer that displays reports and statistics.
Participation Tracking	The system's feature that records and monitors volunteers' hours
Certificate Generator	A tool that automatically generates participation certificates for a volunteer.

Report	File that summarizes volunteer activity, attendance, and hours.
Login	The authentication process that allows users to access their accounts.
User Registration	The process of creating a new account for volunteers or organizers.

6.2 Gantt chart



6.3 WBS diagram



Evaluation

To be completed by the instructor or supervisor.

3. Project SPMP (18 points / 3 marks) – Week 8

Section	Excellent (Full Points)	Good (75%)	Fair (50%)	Poor (25% or below)	Points
1. Project Overview (3 pts)	Clear purpose, welldefined scope, objectives measurable, risks realistic, deliverables & schedule complete.	Purpose/scope adequate, some objectives unclear, limited risks, minor gaps in deliverables/schedule.	Vague or incomplete scope, objectives weak, little risk identification.	Missing or irrelevant.	/3
2. Project Organization (2 pts)	Well-structured team org, clear roles/responsibilities, external interfaces defined.	Team roles mostly clear, some vagueness in structure or interfaces.	Roles incomplete or unclear, poor organization.	Missing or irrelevant.	/2
3. Managerial Process Plans (5 pts)	Estimates realistic, WBS & schedule clear, tracking plan defined, risks detailed with mitigation, closeout plan solid.	Most elements present but some weak (e.g., vague WBS or shallow risk analysis).	Several parts incomplete or poorly justified.	Very minimal or missing.	/5
4. Technical Process Plans (3 pts)	Process model justified, tools/methods listed, infrastructure described, acceptance criteria realistic.	Adequate but some parts generic or weak justification.	Vague process model, missing tools/infrastructure details.	Very poor or absent.	/3
5. Supporting Process Plans (3 pts)	Documentation plan complete, QA standards clear, version control explained, bug tracking method defined.	Most parts covered but not in detail.	Minimal coverage (e.g., only documentation).	Missing or irrelevant.	/3
6. Additional Plans (1 pt)	At least one plan (training, security, maintenance) included.	Mentioned but vague.	Minimal or irrelevant.	Not included.	/1
7. Overall Quality (1 pt)	Well-written, wellstructured, free of major grammar/formatting issues, follows template fully.	Clear but some formatting/clarity issues.	Hard to follow, weak formatting.	Poorly presented or incomplete.	/1

Note: If a student is listed as a member at the beginning of the SPMP **but not mentioned within the tasks, roles, or responsibilities throughout the document**, that student **will not receive the same score as active contributors**.

- Marks for that student may be reduced to reflect lack of contribution.
- Instructors should verify **role allocation tables, WBS assignments, and responsibilities** for evidence of individual involvement