



College of Engineering

Software Engineering Department

Software configuration management plan

Title: Library Management System

Group Name	ID
1. Hiwot Belay	ETS0812/14
2. Hiwot Tadesse	ETS0814/14
3. Ililli Juhar	ETS0824/14
4. Kaku Temesgen	ETS0839/14
5. Kenan Mengistu	ETS0902/14
6. Kenean Birru	ETS0903/14

Submitted to: Inst. Yimer Amedie
Submission Date: Dec 30, 2025

Table of Contents

1.	Introduction	2
1.1	Purpose.....	2
1.2	Scope and Applicability.....	2
1.3	Organizational Relationships.....	2
1.4	References.....	2
2.	Criteria for Configuration Identification.....	3
2.1	Configuration Items Covered by Configuration Management.....	3
2.2	Configuration Item Categories.....	3
3.	Limitations and Assumptions.....	3
3.1	Assumptions.....	3
3.2	Limitations.....	4
4.	CM Responsibilities and Authorities.....	4
4.1	Roles and Responsibilities.....	4
4.2	Authority Lines.....	5
4.3	Oversight and Accountability.....	5
5.	Project Organization.....	5
5.1	Organizational Structure.....	5
5.2	Configuration Management Interfaces.....	5
6.	Applicable Policies and Procedures.....	6
6.1	Applicable Standards.....	6
6.2	Directives and Control Policies.....	6
6.3	Configuration Management Methods.....	6
6.4	Configuration Identification, Naming, and Versioning Rules.....	6
7.	Planned SCM Activities.....	6
7.1	SCM Activities Overview.....	7
7.2	Scheduling and Milestones.....	7
7.3	Baseline Management.....	7
8.	Change Control Process.....	8
9.	Tools and Resources.....	8
10.	SCMP Maintenance and Control.....	8
11.	Approval.....	8

1. Introduction

1.1 Purpose

The purpose of this Software Configuration Management Plan (SCMP) is to define the processes, responsibilities, and controls required to manage configuration items (CIs) for the Library Management System project. The plan ensures that all project artifacts are systematically identified, versioned, changed, baselined, and audited in a controlled and traceable manner throughout the project lifecycle.

1.2 Scope and Applicability

This SCMP applies to all configuration items produced during the development of the Library Management System, including documentation, source code, data files, test artifacts, and release records. Configuration management activities span all project phases: planning, development, testing, release, and maintenance. The emphasis of this project is on disciplined SCM practices rather than software complexity.

1.3 Organizational Relationships

The project team consists of six members with clearly defined roles. The SCM Manager oversees configuration activities, version control, and baselines, while the Project Manager ensures alignment with schedules and milestones. The Lead Developer and Developer/Integrator implement system functionality, integrate features, and apply approved changes. The QA & Tester executes tests and validates change requests, while the Documentation Lead & Configuration Auditor maintains documentation and conducts audits. The Instructor reviews and approves the final product, including baselines, releases, and audit outcomes.

1.4 References

- IEEE Std 828™-2012, *IEEE Standard for Configuration Management in Systems and Software Engineering*
- Git Documentation, <https://git-scm.com>
- GitHub Documentation, <https://docs.github.com>

2. Criteria for Configuration Identification

2.1 Configuration Items Covered by Configuration Management

All artifacts that contribute to the development, testing, and release of the system are designated as configuration items. These include project documents (SCMP, CI Register, Change Log), source code files (HTML, CSS, JavaScript), data files (JSON), test cases, and release documentation. Each CI is uniquely identified and placed under version control.

2.2 Configuration Item Categories

CIs are categorized as follows:

Category	Instances	Purpose
Documentation	SCMP, CI Register, Change Log, Release Notes	Provides project guidance and traceability
Source code	HTML, CSS, JS files	Implements system functionality
Data Files	books.json, users.json	Stores library data for system operation
Test Artifacts	Test cases, QA reports	Ensures system quality and compliance

3. Limitations and Assumptions

3.1 Assumptions

It is assumed that all team members follow defined SCM procedures, use GitHub consistently, and submit changes through approved workflows. Access to development tools and repositories is assumed to be continuous.

3.2 Limitations

The project is limited to simple web-based functionality. A JSON file replaces a full database for simplicity. The SCM processes rely on consistent participation from all team members and availability of GitHub services.

4. CM Responsibilities and Authorities

4.1 Roles and Responsibilities

Roles	Team Member	Responsibilities
Project Manager	Hiwot Belay	Oversees schedules, milestones, and deliverables; coordinates team activities.
SCM Manager	Kenean Birru	Maintains the SCMP, CI Register, baselines, and change records; ensures configuration control.
Lead Developer	Kenan Mengistu	Implements core system functionality and key modules.
Developer/Integrator	Kaku Temesgen	Develops additional features, integrates modules, and applies approved changes.
QA & Tester	Ililli Juhar	Validates changes, executes test cases, and supports audits.
Documentation Lead	Hiwot Tadesse	Ensures consistency, completeness, and accuracy

		of all project records and documentation.
--	--	---

4.2 Authority Lines

The SCM Manager has authority over version control and baseline approval. The Project Manager approves schedules and milestone completion. The Configuration Auditor reports audit results to both the Project Manager and SCM Manager. The instructor holds final approval authority over all baselines, releases, and audits.

4.3 Oversight and Accountability

The SCM Manager ensures compliance with SCM processes, while the Project Manager oversees team accountability. All team members report on progress and deviations, maintaining proper documentation and adhering to established workflows.

5. Project Organization

5.1 Organizational Structure

The project team operates in a single-team structure with clearly assigned responsibilities. Coordination occurs through weekly meetings, GitHub repository usage, and shared documentation.

5.2 Configuration Management Interfaces

The project uses several interfaces to manage and control configuration items effectively:

- **GitHub Repository** – Central platform for version control, branching, merging, and maintaining baselines.
- **Pull Requests (PRs)** – Mechanism for reviewing and approving code changes before integration into the main branch.

- **Issue / Change Request Tracker** – Records proposed modifications, tracks approvals, and ensures traceability of all changes.
- **Documentation Reviews** – Formal review and approval process for project documents, ensuring accuracy and completeness.
- **Release Management Interface** – Publishes release notes and tagged versions for controlled distribution and audit purposes.

6. Applicable Policies and Procedures

6.1 Applicable Standards

This plan conforms to IEEE 828–2012. Git and GitHub best practices are applied for version control and collaboration.

6.2 Directives and Control Policies

All changes must be traceable to an approved Change Request (CR). Direct commits to the main branch are restricted. Naming and versioning rules must be followed for all CIs.

6.3 Configuration Management Methods

SCM activities include configuration identification, version control, change control, baseline management, release management, and configuration auditing.

6.4 Configuration Identification, Naming, and Versioning Rules

Each CI is assigned a unique identifier and descriptive name. Semantic versioning (v1.0, v1.1) is used. Git branching includes a main branch and feature or documentation branches. Pull requests are mandatory for integration.

7. Planned Activities

7.1 SCM Activities Overview

Planned activities include CI identification, repository setup, branching, change request processing, baseline creation, releases, and audits.

7.2 Scheduling and Milestones

Software configuration management activities are aligned with defined project milestones to ensure controlled progress and traceability. Baseline 1 (BL1) is established after the initial planning phase and includes approved project documentation such as the Software Configuration Management Plan (SCMP), the Configuration Item (CI) Register, initial requirements, and the initialized GitHub repository structure. Approval of BL1 is granted by the SCM Manager, with confirmation from the Project Manager, indicating readiness to proceed to development.

Baseline 2 (BL2) is created after completion of core development and implementation of approved Change Requests. This baseline captures the stable working prototype, including source code, UI components, updated data files, and revised documentation. BL2 is approved by the SCM Manager in coordination with the QA & Tester, confirming functional stability.

Following BL2, Release v1.0 is produced as the initial working system and includes packaged source code and release notes. Release v1.1 is prepared after incorporating approved refinements or fixes. Final releases are reviewed internally by the project team and submitted as the final project deliverables for instructor evaluation.

7.3 Baseline Management

Baselines represent formally approved snapshots of selected configuration items at specific points in the project lifecycle. Once established, a baseline serves as a reference for development, change control, and audit activities. Each baseline is uniquely identified and tagged in the GitHub repository to ensure traceability and reproducibility. A baseline record is maintained for every baseline, documenting the baseline identifier, date, included configuration items, and approval authority. Changes to baselined items are permitted only through the approved change control process, ensuring the integrity and stability of controlled artifacts throughout the project.

8. Change Control Process

All changes are initiated through a Change Request (CR). CRs are reviewed by the SCM Manager and relevant team members acting as a lightweight Change Control Board (CCB). Approved changes are implemented in branches, reviewed, merged, and documented to maintain traceability.

9. Tools and Resources

The Library Management System project uses a small set of widely available tools to support software configuration management activities. Git and GitHub are used for version control, branching, change tracking, baseline tagging, and release management. GitHub also serves as the central repository for all configuration items and provides support for pull requests and change request tracking.

Visual Studio Code is used as the primary development environment for writing and editing source code files. Google Docs is used for preparing and maintaining project documentation, including the SCMP, CI Register, Change Log, and audit reports. Web browsers are used for testing and validating system functionality. These tools collectively support traceability, collaboration, and controlled management of project artifacts.

10. SCMP Maintenance and Control

The SCMP is a controlled configuration item. It is reviewed at major milestones or when process changes occur. Updates follow the same change control process as other CIs, ensuring version history and auditability.

11. Approval

This Software Configuration Management Plan (SCMP) has been prepared and reviewed to support the Library Management System project and to meet the requirements of the Software Configuration Management course.

Prepared by:

SCM Manager / Project Team

Reviewed and Approved by:

Project Team

Final Review:

Course Instructor

Version: v1.0

Date: December 2025