Addis Ababa Science and Technology University

College of Engineering

DEPARTMENT OF SOFTWARE ENGINEERING

Software Configuration Management Plan for Mini Project on Student Grade Viewer

	Name	ID Number
1.	Samuel Teshome	ETS1419/14
2.	Tadiyos Dejene	ETS1522/14
3.	Tamrat Demse	ETS1523/14
4.	Tebarek Shemsu	ETS1526/14
5.	Tesfakiael Almaw	ETS1544/14
6.	Tsion Getachew	ETS1581/14
7.	Yohannes Tigistu	ETS1703/14

Dec 2025

1. Introduction

1.1 Purpose

The purpose of this Software Configuration Management Plan (SCMP) is to define and document the configuration management processes that will be applied to the *Student Grade Viewer System*.

The SCMP ensures that changes to software artifacts are systematically managed, approved, implemented, and documented in a controlled manner. By following this plan, the project team aims to maintain the integrity, consistency, and traceability of project deliverables throughout the development lifecycle. The plan is prepared in accordance with **IEEE Std 828-2012**, which provides guidance for effective configuration management in software engineering projects.

1.2 Scope

This Software Configuration Management Plan applies to all phases and artifacts of our Student Grade Viewer System project. The scope of this plan includes configuration management activities related to project documentation, source code, data files, test artifacts, baselines, and release packages.

Specifically, the SCMP governs:

- Identification and classification of configuration items
- Version control and branching strategies using Git and GitHub
- Change control through formal Change Request (CR) procedures
- Baseline creation and maintenance
- Release management and configuration audits

The plan is limited to configuration management practices and does not define detailed software development or testing methodologies, except where they interact with configuration control activities.

1.3 Organizational Relationships

The Student Grade Viewer System is developed by a group of undergraduate Software Engineering students as part of the Software Configuration Management course. The project operates within an academic environment and follows institutional guidelines provided by the Department of Software Engineering here in Addis Ababa Science and Technology University.

Configuration management activities are performed collaboratively by our project team, with specific roles assigned to ensure accountability and separation of responsibilities. Interaction between configuration management, development, and testing activities is facilitated through GitHub, which serves as the central repository and collaboration platform. Decisions related to configuration changes, baselines, and releases are coordinated internally within the team and approved according to defined authority levels.

1.4 References

The following documents and standards are referenced in the preparation and execution of this SCMP:

- IEEE Std 828-2012, *IEEE Standard for Configuration Management in Systems and Software Engineering*
- Software Configuration Management course materials and lecture notes
- Project guidelines and assignment instructions provided by our lecturer

2. Criteria for Identification

2.1 Configuration Item Identification

Configuration Item (CI) identification is the process of selecting and defining the software artifacts that will be placed under configuration management control. For the *Student Grade Viewer System*, configuration items are identified based on their importance to system functionality, project documentation integrity, and the ability to track and control changes effectively.

Any artifact that affects the system's behavior, structure, quality, or documentation is considered a configuration item. Once identified, CIs are uniquely named, versioned, and managed throughout the project lifecycle to ensure traceability and consistency.

2.2 Configuration Items Covered by Configuration Management

The following categories of configuration items are placed under configuration management control in this project:

2.2.1 Documentation Configuration Items

These items define project processes, requirements, and records. They provide guidance and traceability for project activities.

- Software Configuration Management Plan (SCMP)
- Requirements specification document
- Configuration Item Register
- Change Request (CR) forms and Change Log

- Baseline records
- Release notes
- Configuration audit reports

2.2.2 Source Code Configuration Items

These items implement the functional aspects of the Student Grade Viewer System and are managed using Git version control.

- HTML files (e.g., login page, dashboard page)
- CSS files for user interface styling
- JavaScript files for client-side logic
- Folder structure and supporting scripts

2.2.3 Data Configuration Items

These items represent the data used by the system and are required for basic system functionality.

- JSON files containing student records and grade information
- Sample or test data files used for validation

2.2.4 Test Configuration Items

These items are used to verify that the system meets its basic functional requirements.

- Test case documents
- Test scenarios
- Test result records

2.3 Configuration Item Naming Conventions

Each configuration item is assigned a unique and descriptive name to avoid ambiguity and ensure easy identification. Naming conventions are applied consistently across all project artifacts.

- Document names follow the format:
  DocumentName_Version (e.g., SCMP_v1.0)

- Source code files use meaningful names that reflect their functionality (e.g., login.html, grades.js)
- Change Requests are uniquely identified using a standard format (e.g., CR-001, CR-002)
- Baselines and releases are identified using predefined tags (e.g., BL1, BL2, v1.0, v1.1)

2.4 Configuration Item Versioning

Version control is applied to all configuration items to track changes over time. Version numbers are updated whenever a configuration item is modified and approved through the change control process. Minor changes result in minor version increments (e.g., v1.1), while major changes result in major version increments (e.g., v2.0).

2.5 Configuration Item Ownership

Each configuration item is assigned an owner who is responsible for maintaining its accuracy and ensuring compliance with configuration management procedures. Ownership does not imply exclusive access but ensures accountability for updates and reviews.

3. Limitations and Assumptions

3.1 Limitations

This Software Configuration Management Plan is subject to several limitations due to the academic nature and scope of the Student Grade Viewer System project. The project is developed within a fixed academic timeframe, which limits the extent of system functionality and the level of process automation that can be applied. As a result, configuration management activities are implemented in a simplified manner suitable for a mini project rather than a full-scale industrial system.

The system uses simple web technologies and static or JSON-based data storage. Therefore, advanced configuration management practices related to complex databases, continuous integration pipelines, or automated deployment environments are outside the scope of this plan.

3.2 Assumptions

This SCMP is developed under the assumption that all project team members have basic knowledge of Git and GitHub and will follow the defined configuration management procedures throughout the project lifecycle. It is also assumed that the required tools, including GitHub, development environments, and documentation tools, will remain available for the duration of the project.

The plan assumes that the project requirements are relatively stable and that changes will be manageable through the defined Change Request process. Furthermore, it is assumed that all configuration items will be stored and maintained within the designated GitHub repository, ensuring a single, authoritative source for all project artifacts.

4. CM Responsibilities and Authorities

4.1 Role Definition Overview

To ensure effective implementation of configuration management activities, specific roles have been assigned to members of the Student Grade Viewer System project team. These roles define clear responsibilities and authority levels for managing configuration items, controlling changes, maintaining baselines, and ensuring compliance with this Software Configuration Management Plan (SCMP). Assigning explicit roles promotes accountability, coordination, and consistency throughout the project lifecycle.

4.2 Assigned Roles and Responsibilities

4.2.1 Team Leader – Tadiyos Dejene

The Team Leader is responsible for overall coordination of the project and oversight of configuration management activities. This role has final authority over baseline approval, release authorization, and acceptance of major configuration changes. The Team Leader ensures that SCM activities align with project objectives and academic requirements and resolves conflicts related to configuration decisions.

4.2.2 SCM Coordinator – Tamrat Demse

The SCM Coordinator is responsible for planning, implementing, and monitoring all configuration management processes defined in this SCMP. This includes identifying configuration items, maintaining the configuration item register, managing version control practices, and ensuring proper tagging of baselines and releases. The SCM Coordinator also coordinates configuration audits and ensures that SCM documentation remains accurate and up to date.

4.2.3 Lead Developer – Tesfamikael Almaw

The Lead Developer is responsible for guiding the technical implementation of the system while ensuring compliance with configuration management procedures. This role oversees development activities on feature branches, reviews pull requests for technical accuracy, and ensures that code changes follow approved change requests and versioning rules.

4.2.4 Developers – Tebarek Shemsu and Samuel Teshome

Developers are responsible for implementing assigned features and fixes within the Student Grade Viewer System. They work on designated feature branches, commit changes with clear and descriptive messages, and submit pull requests for review and approval. Developers are accountable for ensuring that their contributions adhere to defined SCM procedures and standards.

4.2.5 Tester / Quality Reviewer – Tsion Getachew

The Tester is responsible for verifying that implemented changes meet the defined system requirements and that approved change requests are correctly implemented. This role executes test cases, documents test results, and supports functional configuration audits. The Tester provides validation feedback prior to merging changes into the main branch and release preparation.

4.2.6 Documentation & Audit Support – Yohannes Togistu

The Documentation and Audit Support role is responsible for preparing and maintaining project documentation related to configuration management. This includes assisting with CI registers, change logs, baseline records, release notes, and configuration audit reports. This role supports both physical and functional configuration audits by ensuring documentation accuracy and completeness.

4.3 Authority Lines and Decision Control

Authority for configuration management decisions is structured to ensure clear accountability. The Team Leader holds final approval authority for baselines, releases, and major configuration changes. The SCM Coordinator has operational authority to enforce SCM procedures, manage configuration records, and coordinate audits. Developers, testers, and documentation support staff operate under these authorities and must obtain approval through the Change Request process before implementing or finalizing changes.

5. Project Organization

5.1 Organizational Structure

The Student Grade Viewer System project follows a small, team-based organizational structure suitable for an academic software development environment. The project team consists of defined roles including Team Leader, SCM Coordinator, Developers, Tester, and Documentation Support. Each role has clearly assigned responsibilities to ensure effective coordination and accountability.

5.2 Configuration Management Interfaces

Configuration management interfaces exist between development, testing, documentation, and release activities. These interfaces are managed using Git and GitHub, which serve as the central platform for version control, collaboration, and communication. Feature branches, pull requests, and issue tracking are used to coordinate interactions between team members and ensure controlled integration of changes into the main branch.

6. Applicable Policies and Procedures

6.1 Standards and Guidelines

This project follows the principles outlined in **IEEE Std 828-2012** for software configuration management. Internal project guidelines and course instructions provided by our lecturer are also followed to ensure compliance with academic requirements.

7. Planned Activities

7.1 Schedule and Milestones

The configuration management activities for the Student Grade Viewer System are planned over a roughly one-month period, aligned with the academic project timeline. The activities are organized into sequential phases to ensure systematic control and traceability of project artifacts. This is a rough depiction of what amount of time the project development would take.

Roughly During **Week 1**, the project repository is created, the folder structure is established, and initial project documentation, including the Software Configuration Management Plan and Configuration Item Register, is prepared. Baseline 1 is established at the end of this week to capture the initial project setup.

During **Week 2**, development activities are carried out on designated feature branches. Core system components such as the login interface and grade viewing functionality are implemented. Configuration management activities focus on version control, commit management, and pull request reviews.

During **Week 3**, formal Change Requests are submitted, evaluated, and approved. Approved changes are implemented, tested, and documented. Baseline 2 is created to reflect the updated and stabilized version of the system after change implementation.

During **Week 4**, release management and configuration auditing activities are performed. Two project releases are prepared and documented. Physical and Functional Configuration Audits are conducted to verify consistency between documentation, configuration items, and system functionality. Final project documentation is reviewed and prepared for submission.

7.2 Resources

The project utilizes GitHub as the primary configuration management and collaboration platform. Development is performed using lightweight tools such as Visual Studio Code, while documentation is prepared using standard word processing tools. Human resources consist of the assigned project team members, each performing defined configuration management, development, testing, and documentation responsibilities.

8. CMP Maintenance

8.1 Review and Update Mechanism

This Software Configuration Management Plan is treated as a controlled document and is subject to review and updates as the project evolves. Changes to the SCMP are proposed through the defined change control process and must be approved by the Team Leader before implementation.

8.2 Version Control of the SCMP

All approved versions of the SCMP are maintained within the project's GitHub repository under configuration control. Updates are documented with version numbers and change descriptions to ensure traceability and consistency throughout the project lifecycle.