



Addis Ababa Science and Technology University

College of Engineering

DEPARTMENT OF SOFTWARE ENGINEERING

**Software Configuration Management Final Project Report for
Mini Project on Student Grade Viewer**

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

Dec 2025

1. Introduction	3
2. Project Purpose and Objectives	3
3. Project Scope	4
4. Project Team and Organizational Structure	5
5. Software Configuration Management Approach	5
5.1 SCM Planning	5
5.2 Configuration Identification	5
6. Version Control and Change Management	6
7. Implemented Change Requests	6
8. Baseline Management	7
8.1 Baseline 1 (BL1)	7
8.2 Baseline 2 (BL2)	7
9. Release Management	7
Release v1.0	7
Release v2.0 (v1.1)	8
10. Testing and Validation	8
11. Configuration Audits	8
12. Limitations and Assumptions	9
13. Lessons Learned	9
14. Conclusion	9
15. References	10

1. Introduction

The *Student Grade Viewer System* is a mini software project developed as part of the **Software Configuration Management (SCM)** course in the Department of Software Engineering at **Addis Ababa Science and Technology University**. The project was designed to demonstrate the practical application of configuration management principles throughout the software development lifecycle.

The system provides a simple web-based interface that allows students to log in, access a dashboard, and view their academic grades. While the functional scope of the system is intentionally limited due to its academic nature, strong emphasis was placed on applying structured SCM practices in alignment with **IEEE Std 828-2012**.

2. Project Purpose and Objectives

The primary purpose of this project was to apply and demonstrate Software Configuration Management concepts in a controlled development environment. Rather than focusing solely on functionality, the project emphasized traceability, version control, change management, baselining, auditing, and release management.

The key objectives of the project were:

- To define and implement a Software Configuration Management Plan (SCMP)
- To identify and manage configuration items systematically
- To control changes using formal Change Requests (CRs)
- To establish and maintain project baselines
- To perform physical and functional configuration audits
- To deliver controlled software releases

3. Project Scope

The scope of the Student Grade Viewer System includes both **software artifacts** and **configuration management activities**. From a functional perspective, the project covers:

- A login interface with client-side validation
- A student dashboard interface
- Login-to-dashboard integration
- A grades page with grade calculation logic

From an SCM perspective, the project includes:

- Configuration identification and classification
- Version control using Git and GitHub
- Change request management
- Baseline and release management
- Configuration audits and documentation

The project does not include backend authentication, database integration, or production deployment, as it is intended strictly for academic demonstration.

4. Project Team and Organizational Structure

The project was developed by a team of undergraduate Software Engineering students, with clearly defined roles to support effective configuration management.

Name	Role
Tadiyos Dejene	Team Leader
Tamrat Demse	SCM Coordinator
Tesfamikael Almaw	Lead Developer
Tebarek Shemsu	Developer
Samuel Teshome	Developer
Tsion Getachew	Tester / Quality Reviewer
Yohannes Togistu	Documentation & Audit Support

This role-based structure ensured accountability, separation of responsibilities, and effective coordination throughout the project lifecycle.

5. Software Configuration Management Approach

5.1 SCM Planning

A comprehensive **Software Configuration Management Plan (SCMP)** was prepared at the beginning of the project in accordance with **IEEE Std 828-2012**. The SCMP defined the procedures for configuration identification, version control, change management, baselining, and audits.

5.2 Configuration Identification

Configuration Items (CIs) were identified based on their impact on system functionality, documentation integrity, and project traceability. Identified CIs included:

- Documentation artifacts (SCMP, CI Register, Change Log, Baseline Records)
- Source code files (HTML, CSS, JavaScript)
- Data files (JSON-based sample data)
- Test artifacts (test cases and test records)

All CIs were uniquely named, versioned, and tracked in the CI Register.

6. Version Control and Change Management

Git and GitHub were used as the primary version control and collaboration tools. A feature-branch-based workflow was adopted to ensure controlled development and integration.

All changes were managed using formal **Change Requests (CRs)**. Each CR followed a defined lifecycle:

1. Submission and approval
2. Implementation on a feature branch
3. Review through pull requests

4. Merge into the `main` branch

This process ensured that no uncontrolled or undocumented changes entered the system.

7. Implemented Change Requests

The following Change Requests were successfully implemented and merged:

- CR-001: Modern login page with separate CSS
- CR-002: Client-side validation for login
- CR-004: Student dashboard after login
- CR-005: Dashboard data retrieval from JSON
- CR-006: UI improvements and visual consistency
- CR-007: Conflict resolution in data retrieval
- CR-008: Login-to-dashboard integration
- CR-009: Grades page with calculation logic
- CR-010: UI improvements and link fixes for grades page

These changes collectively resulted in a fully integrated academic prototype.

8. Baseline Management

8.1 Baseline 1 (BL1)

Baseline 1 captured the initial project setup, including:

- SCM documentation
- Repository structure
- CI Register v1.0

8.2 Baseline 2 (BL2)

Baseline 2 represents the **functional baseline** of the system. It includes:

- Login, dashboard, and grades pages
- Integrated navigation and validation
- Updated CI Register v2.0
- Completed configuration audit

BL2 serves as the reference point for the final release.

9. Release Management

Release v1.0

The first release included:

- Login interface
- Dashboard interface
- Basic validation
- Partial system functionality

Release v2.0 (v1.1)

The final release includes:

- Login-dashboard integration
- Grades page with calculation logic
- UI enhancements
- All approved CRs implemented

Release v2.0 represents the **final deliverable** of the project.

10. Testing and Validation

Testing focused on verifying basic functional requirements:

- Input validation on login
- Correct redirection behavior
- Dashboard access control
- Accurate grade calculation
- Navigation consistency

Test cases were documented, executed, and reviewed prior to baseline and release approval.

11. Configuration Audits

Both **Physical Configuration Audit (PCA)** and **Functional Configuration Audit (FCA)** were conducted.

- The PCA verified consistency between documentation and repository artifacts.
- The FCA confirmed that all approved CRs were implemented and functional.

The audit concluded that the system was compliant with the SCMP and ready for release.

12. Limitations and Assumptions

Due to the academic scope and timeframe, the system uses client-side logic and static data. Advanced features such as backend authentication, databases, and automated deployment were outside the project scope. It was assumed that team members possessed basic Git knowledge and adhered to SCM procedures.

13. Lessons Learned

The project highlighted the importance of SCM in:

- Managing team collaboration

- Maintaining traceability
- Preventing uncontrolled changes
- Supporting structured software evolution

Applying SCM principles significantly improved project organization and quality.

14. Conclusion

The Student Grade Viewer System successfully demonstrates the practical application of Software Configuration Management principles in an academic setting. Through structured planning, disciplined change control, baselining, auditing, and controlled releases, the project achieved its objectives and delivered a stable, traceable, and well-documented software system.

With **Baseline 2** and **Release v2.0** completed, the project satisfies all academic requirements and serves as a strong example of SCM-driven software development.

15. References

- IEEE Std 828-2012, *IEEE Standard for Configuration Management in Systems and Software Engineering*
- Software Configuration Management Course Materials
- Project Assignment Guidelines