SWAZITECH

Innovating the Future

PROJECT PLAN

Course Code: CSC 392

Institution: University of Eswatini

Group Members

NonzuzoDlamini — 202203818

KwaneleDlamini — 202201140

SibonginkhosiShongwe — 202201016

SenaneloXaba — 202202526

Submission Date: 13 April 2025

Prepared by SwaziTech

Contents

Project Plan: Student Attendance System 2

1. Introduction 2

Team Members 2

Aims and Objectives 2

Summary 2

2.Client/User 2

User/Client Involvement 2

3. Risks 3

4. Standards, Guidelines, and Procedures 3

5. Organization of the Project 3

Training schedule 4

6. Project Phases (Life Cycle Model: Incremental) 5

7. Requirements Analysis and Design 6

8. Implementation 6

9. Testing 6

10. Resources 6

11. Quality Assurance 6

Organization responsible 6

12. Changes (Configuration Management) 7

Project Schedule 8

Project Plan: Student Attendance System

1. Introduction

The Student Attendance System is a software solution designed to automate the process of recording student attendance and generating detailed reports for educational institutions like high schools. Historically, attendance tracking has relied on manual methods such as paper registers, which are prone to errors, time consuming and difficult to analyse.This project aims to reduce manual paperwork, improve accuracy, and ensure real-time attendance tracking by modernizing the process which will provide an efficient, reliable and user friendly system.

Team Members

- Leader/Manager: DlaminiNonzuzo
- Designer: ShongweSibonginkhosi
- Architect: XabaSenanelo
- Programmer: DlaminiKwanele

Aims and Objectives

- Develop a software to record student attendance accurately
- Generate automated reports (daily, weekly, monthly attendance summaries)
- Reduce administrative workload and improve data accessibility
- Unsure scalability for use in small schools or large institutions like universities

Summary

The project will deliver a web-based application with database backend, allowing teachers to mark attendance via a simple interface and administrators to generate reports. The project is expected to take 12 weeks from initiation to deployment.

2.Client/User

University/School Administration

User/Client Involvement

The primary clients are educational institutions e.g. high schools . Their involvement includes:

- Information: Provided details on current attendance processes, class schedule, student data formats, and reporting needs by week 1.
- Service: Access to a sample student database for testing purposes
- Facilities: Provision of hosting infrastructure and hardware

Clients will review prototypes and provide feedback during the design phase (week 4) and testing phase (week 9)

3. Risks

Risk	Mitigation Strategy
High cost of implementation and maintenance	Using open-source technologies to reduce software costs
Manual override loopholes where attendance can be manipulated	Role-based access control to limit who can make changes
Software bugs that may incorrectly mark absentees	Thorough testing before deployment: unit tests, system tests, user testing
Resistance from teachers and students due to unfamiliarity or scepticism	Training sessions and user manuals to build confidence in the system

4. Standards, Guidelines, and Procedures

1. Agile Scrum Methodology:
Following the the Agile Scrum approach means:
Work will be done in small, manageable parts (called sprints). At the end of each sprint, we'll review what was done and improve for the next one. The team will have regular meetings to track progress and solve problems together.

2. Documentation Standards:
For writing the system requirements (what the system should do), we’ll use the IEEE 830 standard. This helps make sure all the important details are clearly written and easy to follow.

3. Code Reviews & Sprint Retrospectives:
The team will regularly check each other’s code to find and fix mistakes early and improve the overall quality. After each sprint, we’ll hold a retrospective meeting to discuss what went well, what didn’t, and how we can do better in the next sprint.

5. Organization of the Project

- Project Manager: Oversees planning, execution, and monitoring.
- System Analyst: Gathers and analyses user requirements.
- Developers: Design and implement system modules.
- Testers: Responsible for module and integration testing.
- UI/UX Designer: Designs interfaces based on user roles
- Training Coordinator: Trains staff on system usage.
- Software Architect: Designs the system architecture and selects technologies

Training schedule

Stakeholder	Training duration	Scheduled time	Rationale
Project Manager	2 - 3 days	Week 1	Before project kick-off to establish planning and oversight capabilities.
System Analyst	3 - 4 days	Week 2	Early training to prepare for requirements gathering
Developers	5 - 7 days	Week 3	Before Module Development, after design is finalized, to align with tech stack.
Testers	4 - 5 days	Week 4- 5	Before Testing phase, after prototype delivery, to prepare for validation tasks.
UI/UX Designer	3- 4 days	Week 5 - 6	Before System Design, to ensure readiness for wireframing and prototyping.
Training Coordinator	3 - 5 days	Week 7 - 8	After Integration, to learn the system fully before creating materials.
Software Architect	5-7 days	Week 8	Training helps architects analyze and evaluate different solutions

6. Project Phases (Life Cycle Model: Incremental)

Phase	**Key milestones**	Duration
Phase 1: Requirements gathering	-Approval from SRS -Document functional and non-functional requirements -Obtain stakeholder approval on SRS	1 week
Phase 2: System design	-Create wireframes/ mock-ups for UI/UX -design database schema -Finalize and review System Design Document -Develop high level architecture	1 week
Phase 3: Module development	-Set up development environment and version control-Develop core modules: authentication and attendance logging -Implement secondary features: student portal, reporting -Deliver a working prototype or MVP	3 weeks
Phase 4: Integration	-Integrate all modules into a single codebase - Integrate frontend (UI)with backend -Establish and test API connections or data flows -Resolve integration bugs and ensure system stability -Fix critical bugs identified during testing -Obtain user acceptance testing (UAT) approval - Demonstrate end to end flow	1 week
Phase 5: Testing	-Perform functional testing (validate features against requirements) -Conduct non-functional testing (e.g., performance, security) -Fix critical bugs identified during testing -Obtain user acceptance testing (UAT) approval	1 week
Phase 6: Deployment and testing	-Deploy the application to production servers or app stores e.g. school's cloud -Develop training materials (e.g., user guides, tutorials) -Conduct training sessions for end-users and support teams -Hand over documentation and provide initial post-deployment support -Go live and user experience	1 week

7. Requirements Analysis and Design

•Techniques: Use cases, ER diagrams, flowcharts
•Tools: Visual Paradigm, Lucidchart, Figma for design.
•Deliverables: SRS document, design documents.

8. Implementation

- Backend: Java
- Frontend: ReactJS or HTML/CSS with Bootstrap
- Database: MySQL/PostgreSQL
- Frameworks: Django/Express.js
- Version control: Git + GitHub

9. Testing

- Test Environment: Virtual server
- Unit tests: Written by developers
- Integration tests: Handled by QA team
- Tools: Selenium, Postman

10. Resources

- Hardware: Laptops, Local Server/Cloud VM
- Software: IDE (VS Code), DBMS, GitHub, Testing tools
- Human resources: 6-person team

11. Quality Assurance

Quality assurance will ensure that the attendance tracking system meets predefined quality standards in terms of functionality, reliability, usability, and performance.

Organization responsible

- Internal QA Team: led by the programmer(KwaneleDlamini), with oversight from the project manager (NonzuzoDlamini)
- Client Validation: The client will participate in user acceptance testing to confirm the system meets their expectations.

Procedures:

- Code reviews
- Testing standards
- Requirement traceability
- Performance metrics
- Usability checks **Quality Aspects**:

- Functional quality: all features (attendance entry, report generation) work as specified
- Reliability: system uptime of 99.9% during testing
- Security: Basic user authentication and data encryption (e.g. HTTPS) implemented
- Maintainability: Code is well documented and modular **Documentation**:

Verification:
Milestones (e.g. design approval, test completion) will be verified through client sign-off and test reports

12. Changes (Configuration Management)

- Change requests submission:
  - Any team member/client can submit a change request via email, detailing the change, its justification and potential impact
  - Deadline: Request must be submitted at least 3 days before the next weekly meeting
- Evaluation:
  - The project manager assess the change's impact on scope, timeline and budget
  - Timeframe: Evaluation completed within 2 days of submission
- Approval:
  - The project manager approves or rejects the change, consulting the client if it affects deliverables
  - Approved changes are documented
- Implementation:
  - The lead programmer assigns tasks to incorporate the change updating the project schedule if needed

Project Schedule

Gantt Chart Table

Task	Duration (Days)	Start date	End date	Cost per day(SZL)	Total cost(SZL)
Requirements Gathering	7	2025-04-01	2025-04-07	80	560
System Design	7	2025-04-08	2025-04-14	150	1050
Module Development	21	2025-04-15	2025-05-05	180	3780
Integration	7	2025-05-06	2025-05-12	150	1050
Testing	7	2025-05-13	2025-05-19	100	700
Deployment And Training	7	2025-05-20	2025-05-36	100	700