Software Development Project

Group Project Assignment

BSc Computer Science - SUZA

State University of Zanzibar (SUZA)
School of Computer Communication and Media Studies

Academic Year 2025/2026

Contents

1 Project Overview

1.1 Introduction

This project assignment requires students to work in teams to develop a complete software application following industry-standard software development practices. The project spans the entire semester and covers all phases of the Software Development Life Cycle (SDLC).

1.2 Objectives

By completing this project, students will:

- Apply software engineering principles in a real-world context
- Practice Agile/Scrum methodology with sprint-based development
- Gain hands-on experience with version control (Git/GitHub)
- Develop teamwork and collaboration skills
- Create professional documentation
- Build a portfolio-ready software project

1.3 Team Formation

- Team size: **4-5 members**
- Each team must assign roles: Product Owner, Scrum Master, Developers, QA Lead
- Roles can rotate during the project
- All team members must contribute to coding

2 Project Requirements

2.1 Project Scope

Teams must develop a **web-based application** that solves a real-world problem. Suggested project domains include:

- **E-commerce:** Online store, marketplace, booking system
- **Education:** Learning management, quiz platform, library system
- **Healthcare:** Appointment booking, patient records, pharmacy
- **Finance:** Expense tracker, budgeting app, mobile money integration
- **Agriculture:** Farm management, market prices, crop advisory
- **Tourism:** Hotel booking, tour packages, travel guide
- **Community:** Event management, volunteer coordination, local services

2.2 Technical Requirements

2.2.1 Minimum Features (Must Have)

1. User authentication (registration, login, logout)
2. User roles (at least 2: e.g., admin and regular user)
3. CRUD operations for main entities
4. Search and filter functionality
5. Responsive design (mobile-friendly)
6. Form validation (client and server-side)
7. Database integration

2.2.2 Technology Stack

Teams may choose from the following options:

Frontend:

- HTML5, CSS3, JavaScript
- CSS Framework: Bootstrap, Tailwind CSS, or similar
- Optional: React, Vue.js, or vanilla JavaScript

Backend:

- Node.js with Express.js
- Python with Django or Flask
- PHP with Laravel
- Java with Spring Boot

Database:

- MySQL, PostgreSQL, or SQLite (relational)
- MongoDB (NoSQL - if appropriate)

Version Control:

- Git and GitHub (mandatory)
- GitHub Projects or Trello for task management

3 Project Deliverables

The project is divided into **5 major deliverables**, each submitted at specific milestones throughout the semester.

3.1 Deliverable 1: Project Proposal and Planning

Due: Week 2 | Weight: 10%

Item	Description
Project Charter	Project name, team members, roles, problem statement, objectives, scope
Team Agreement	Communication plan, meeting schedule, conflict resolution, tools to use
GitHub Repository	Created and shared with all members and instructor
Project Board	GitHub Projects or Trello board set up with initial backlog

Submission:

- Project Charter document (PDF)
- GitHub repository link
- Project board link

3.2 Deliverable 2: Requirements Specification

Due: Week 4 | Weight: 15%

Item	Description
User Stories	Minimum 15 user stories with acceptance criteria
Use Cases	3-5 detailed use case descriptions
Functional Requirements	Detailed list of system features
Non-Functional Requirements	Performance, security, usability requirements
MoSCoW Prioritization	Requirements categorized by priority

Submission:

- Software Requirements Specification (SRS) document (PDF)
- User stories added to project board

3.3 Deliverable 3: System Design

Due: Week 6 | Weight: 15%

Item	Description
Architecture Diagram	High-level system architecture showing components
Database Design	Entity-Relationship Diagram (ERD) with all tables
API Design	List of endpoints with methods, parameters, responses
UI/UX Design	Wireframes or mockups for main screens (minimum 5)
Technology Justification	Explanation of technology stack choices

Submission:

- System Design Document (SDD) (PDF)
- Wireframes/Mockups (Figma link or images)
- ERD diagram

3.4 Deliverable 4: Implementation (3 Sprints)

Due: Weeks 7-11 | Weight: 40%

Implementation is divided into 3 sprints, each 2 weeks long.

3.4.1 Sprint 1: Core Features (Week 7-8)

Weight: 15%

Item	Description
Project Setup	Development environment, folder structure, database
User Authentication	Registration, login, logout, password hashing
Basic UI	Navigation, layout, responsive design
Core Entity CRUD	Create, Read, Update, Delete for main entity
Sprint Documentation	Sprint planning, daily standups log, retrospective

3.4.2 Sprint 2: Additional Features (Week 9-10)

Weight: 15%

Item	Description
User Roles	Admin panel, role-based access control
Search & Filter	Search functionality, filtering, sorting
Additional Features	2-3 more features from requirements
Form Validation	Client-side and server-side validation
Sprint Documentation	Sprint planning, daily standups log, retrospective

3.4.3 Sprint 3: Polish and Testing (Week 11)

Weight: 10%

Item	Description
Bug Fixes	Address all known issues
UI Polish	Consistent styling, error messages, loading states
Testing	Unit tests (minimum 10), integration tests (minimum 3)
Documentation	Code comments, README file
Sprint Documentation	Sprint planning, retrospective, final review

Sprint Submissions (each sprint):

- Working code pushed to GitHub
- Sprint planning document
- Sprint retrospective document
- Demo video (2-3 minutes showing features)

3.5 Deliverable 5: Final Submission and Presentation

Due: Week 12 | Weight: 20%

Item	Description
Complete Application	Fully functional, deployed application
Source Code	Clean, well-commented code on GitHub
README File	Installation instructions, features, screenshots
Test Report	Test cases, coverage report, bug report
User Manual	How to use the application (PDF)
Project Report	Complete project documentation (PDF)
Presentation	15-minute team presentation with demo

Final Submission:

- GitHub repository (complete with all code)
- Deployed application URL (Vercel, Netlify, Heroku, etc.)
- Project Report (PDF)
- Presentation slides (PDF)
- Individual reflection (1 page per team member)

4 Grading Criteria

4.1 Deliverable Weights

Deliverable	Weight	Due Date
Deliverable 1: Project Proposal	10%	Week 2
Deliverable 2: Requirements	15%	Week 4
Deliverable 3: System Design	15%	Week 6
Deliverable 4: Implementation	40%	Weeks 7-11
Deliverable 5: Final Submission	20%	Week 12
Total	**100%**	

4.2 Evaluation Rubric

Each deliverable is evaluated based on:

Criteria	Description
Completeness (30%)	All required items are present and complete
Quality (30%)	Work is well-done, professional, and follows best practices
Functionality (20%)	Features work correctly without major bugs
Documentation (10%)	Clear, well-organized documentation
Teamwork (10%)	Evidence of collaboration, equal contribution

4.3 Individual Assessment

In addition to group grades, individual performance is assessed through:

- Git commit history (quality and frequency)
- Peer evaluation (team members rate each other)
- Individual reflection document
- Participation in presentations and demos

5 Submission Guidelines

5.1 Document Format

- All documents must be submitted as PDF
- Use the provided templates where available
- Include team name and member names on all documents
- Follow academic writing standards

5.2 Code Submission

- All code must be on GitHub
- Repository must have a clear README.md
- Use meaningful commit messages
- Create branches for features (not all work on main)
- No sensitive data (passwords, API keys) in repository

5.3 Late Submission Policy

- 10% penalty per day for late submissions
- Maximum 3 days late accepted
- No submissions accepted after 3 days without prior approval

6 Academic Integrity

- All work must be original
- External code/libraries must be cited
- Plagiarism will result in zero marks
- AI tools can be used for learning but not for direct submission
- Code will be checked for similarity

7 Support and Resources

7.1 Office Hours

- Instructor office hours: [To be announced]
- Lab sessions: [To be announced]
- Online support via course platform

7.2 Recommended Resources

- Course lecture materials
- Template documents provided
- GitHub documentation
- Stack Overflow (for technical questions)
- MDN Web Docs (for web development)

8 Project Timeline Summary

Week	Activity	Deliverable
1	Team formation, project selection	-
2	Project planning	Deliverable 1
3	Requirements gathering	-
4	Requirements analysis	Deliverable 2
5	System design	-
6	Design review	Deliverable 3
7-8	Sprint 1 development	Sprint 1 Demo
9-10	Sprint 2 development	Sprint 2 Demo
11	Sprint 3 (testing & polish)	Sprint 3 Demo
12	Final presentation	Deliverable 5

9 Checklist for Students

9.1 Before Starting

- ☐ Form a team of 4-5 members
- ☐ Choose a project idea
- ☐ Create GitHub accounts (all members)
- ☐ Set up communication channels (WhatsApp, Discord, etc.)

9.2 During Development

- ☐ Commit code regularly (at least daily during sprints)
- ☐ Attend all team meetings
- ☐ Update project board with task progress
- ☐ Document issues and decisions
- ☐ Test features before marking as complete

9.3 Before Each Submission

- ☐ Review all requirements for the deliverable
- ☐ Proofread all documents
- ☐ Ensure code compiles and runs
- ☐ Get all team members to review
- ☐ Submit before deadline

Good luck with your project! Remember: the goal is to learn and build something you're proud of.