Project Plan

Project title: Student Management System (SMS)

Full name: Nguyen Quang Bao

University: Duy Tan

Email: nguyenquangbao575@gmail.com

Part 1 – Concept & Pre-study

Students and Administrators often struggle with course registration, viewing personal profiles and managing courses, classrooms, and users. These challenges can lead to lack of students in each classroom, disrupt and poorly manage academic.

The Student Management System (SMS) aims to solve these issues by providing a centralized digital platform that streamlines course registration and academic management.

Project Objectives

- Develop a digital platform for course registration and personal profile views.
- Assist administrative management for students, classrooms, courses, and users.

Project Risks

- **Technical Risk:** Difficulties in integrating backend and frontend components, poor task analysis, and potential misconfiguration in database or Docker setup.
- **Schedule Risk:** Balancing school, project, and personal schedule may cause inconsistent progress.
- **External Risk:** Hardware breakdown or power outages.
- **Resource Risk:** Limited access to testing software or devices.
- **Financial Risk:** Minimal, but potential costs if paid hosting or premium tools are required.

Part 2 – Feasibility Analysis

I. Technical Feasibility

- Frontend: The user interface will be built using standard web technologies like HTML, CSS and JavaScript.
- Backend: The backend APIs will be developed using Spring Boot framework with Java on Eclipse or Visual Studio Code.
- Database: I will use MySQL for database management.
- Collaboration tools: I will use Docker to connect MySQL to eclipse, GitHub for upload and version control.
- Testing: Postman for API testing and JUnit for backend unit tests. The API will follow RESTful standards.

⇨ The project is technically feasible with available tools.

II. Economic Feasibility

- Cost efficiency: All tools (Eclipse, MySQL, Docker, GitHub, Visual Studio Code) are free.
- Hardware: Personal computer meet all requirements.
- Deployment: Free or trial cloud hosting options (Azure trial/Vercel/Infinity Free).

⇨ The total development cost is near zero, making it highly economical. Therefore, the project is cost-effective and suitable for individual implementation.

III. Scope Feasibility

- Clearly defined modules: Student (Registration, login, view profile, enroll in courses), Admin (Manage users, courses, classrooms, and view reports).
- Features include registration, personal view profile and academic management.
- Deploy in website.

⇨ The scope is well-defined and manageable within the given timeframe.

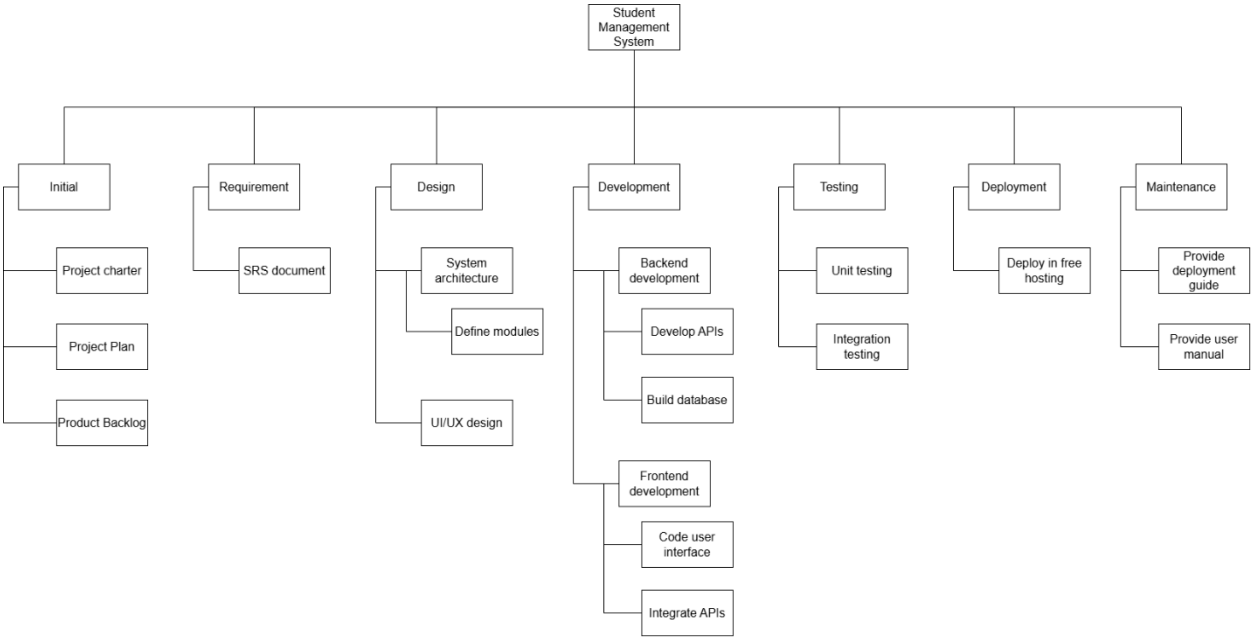
IV. Schedule Feasibility

- School and extracurricular schedules are not fixed so that it's very flexible and easy to adjust my project schedules.

⇨ With flexible schedules, I can easily adjust my project process and finish it.

Part 3 – Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) divides the project into manageable components and tasks.



Part 4 – Project Organization

The project follows a self-managed structure with a single developer handling all roles. Version control is done through GitHub.

Because this system is designed to support students and administrators in registration and academic management. Also, it is my own project to upload to GitHub, improving my technical skills and experience.

Role & Responsibilities:

Project Leader/Developer/Designer/Tester: Quang Bao.

Part 5 – Agile Scrum Application

The Agile Scrum methodology is suitable for this project because it supports continuous feedback, flexibility, and iterative development.

Sample Product Backlog Items:

As a student, I want to view my profile so that I can know my private information.

As a student, I want to register for course so that I can study the course I want.

As a student, I want to view my list of registered courses so that I can know the courses I registered for.

As an administrator, I want to have fully CRUD functional so that I can manage courses, classrooms, students and users.

Sprint length: 3 weeks

Estimated number of Sprints: 3-4 sprints

Sprint 1	Develop interface, login page, personal profile viewing and integrate with existing system
Sprint 2	Implement CRUD for Course, Course registration
Sprint 3	Implement CRUD for Classroom, Student
Sprint 4	Implement CRUD for User + Testing, Fix bugs, Adjust UI

Sprint Review: At the end of each sprint, I note errors and missed information to support the next sprint.

Sprint Retrospective: After each sprint, analyze what went well, what went wrong, and how to improve in the next sprint.