Project Plan

University Project Management System

Group Members
Inam karim 04072212031
Shakir Rabbani 04072312040
Aftab Shaban 04072312046

Software Construction (CS-322)

Submitted to:
DR. ONAIZA MAQBOOL

QUAID-I-AZAM UNIVERSITY ISLAMABAD

DEPARTMENT OF COMPUTER SCIENCE

This is to clarify that the semester project entitled as

UNIVERSITY PROJECT MANAGEMENT SYSTEM

Is the Bonafide record of semester project work done by

INAM KARIM

SHAKIR RABBANI (team lead)

AFTAB SHABAN

of BSCS during Spring-2025

_____ _____

Supervisor Head of Department

Declaration

We affirm that this Semester project entitled as University Project Management being submitted in fulfillment for the award of 5th semester of **bachelor's in computer science** is the original work carried out by us.

_____ _____ _____

Inam Karim Shakir Rabbani Aftab Shaban

Preface

The Department of Computer Science requires a management system to streamline the final semester project process. This project will manage project registrations, facilitate communication between students and supervisors, and provide a structured platform for document submissions and feedback.

The intended users of this project documentation are the faculty members and administrative staff overseeing final semester projects, as well as technical personnel responsible for maintaining the system.

Change History

Version 1.0
Defined Project Scope, Objectives and Summary

Version1.1
Project Context, References and Definitions

Version1.2
Project Planning and Risk Management

Version 1.3
Updated the Gantt Chart and Tables

Version 2.0
Draw use case diagram

Version 2.1
Domain model
Use Case text
System Sequence diagram

Version 2.3
Use case text
System sequence diagram
Data Dictionary

Table of Contents

1. Project Overview

a. Case study

The Department of Computer Science wants to develop a project management system. Every semester, the department offers projects to final semester students. These projects are registered by students with supervisors. This project deals with communication between a student and supervisor during the project. It allows all projects to be loaded and supervisors to be defined. Subsequently, throughout the semester, the website can be used for uploading work products (by students) and feedback (by supervisor). The website allows multiple versions of a single work product to be submitted.

b. Project summary

i. Purpose

The purpose of this project is to maintain the complete project information of a department. There are different types of projects that are offered for different programs such as BS, MSc etc. so this project will maintain all types of communication, structured work submission, and feedback mechanisms.

ii. Scope

In this project the system will be able to perform the following functionalities

- Student project registration with supervisors.
- Supervisor assignment and project definition.
- Work product uploads and version control.
- Feedback and grading system.
- Secure login and user roles (students, supervisors, admins).

iii. Objectives

- Enhance project collaboration between students and supervisors.
- Enable easy submission and review of work products.
- Provide structured feedback for iterative improvements.
- Maintain a historical record of project versions.

iv. Assumption and Constraints

- **Schedule**

  The university required this to be functional at the start of the new semester. So, this project must be completed within this semester.

- **Scope**

  The system will enable students to register their projects with assigned supervisors, while supervisors can define projects and provide feedback on student submissions. It will support multiple versions of work product submissions to track progress and revisions. Secure authentication will be implemented for both students and supervisors to ensure data security and access control. it will feature structured messaging and notifications to facilitate clear and organized communication between users.

- **Software**

  For frontend HTML, CSS and JavaScript will be used for this system and for backend PHP will be used in order to make this system fully functional.

v. Project deliverables.

- **Project Plan**

  In this the whole project plan will be included in document form as a summary of the project schedule design in project libre. The project manager will approve the project plan based on the available resources.

- **Analysis Documentation**

  This documentation includes the brief analysis of project plan and deliver to the internal team member for any type of modification in the project plan. A brief instruction will be included in this document that will help to easily trace the project plan part to be updated.

- **Prototype/Low-Fidelity**

  Before implementation of the project a low-fidelity prototype will be discussed with the stakeholder in order to make changes for better experience. This design can easily be modified so there will be no need to change in the high-fidelity.

- **Software Requirement Specification**

  The SRS will describe the goal of the project and the way users will interact with the system. This also includes how the project will be completed. This will be delivered to the client for agreement and after the approval of this the implementation of the project will start.

- **Frond-end of System**

  This will deliver to the end user a high-level fidelity without any backend to ensure better user interaction with the system. A manual for interaction will also be provided as a part of documents in this project deliverable part.

- **Database Connected System.**
  This deliverable project part provides the system with database connectivity.

vi. Schedule Summary

- **Week 1-2 Planning**

The first week is allocated for project planning which includes the following main point, and its work product will be the project plan.

Project Overview
Project Context
Risk Management
Schedule Allocation

Date 3 March 2025 -14 March 2025

- **Week 3-5 Analysis**

The next three weeks are allocated for analysis, and this includes the following main points, and its work product will SRS.

Use case Diagram and Text.
Data Flow Diagram etc.
Software requirement specification.
FTR for analysis

Date 17 March 2025 - 8 April 2025

- **Week 6-9 Design**

Low-Fidelity Design
High-Fidelity Design
Detailed Design

Date 9 April 2025 - 7 May 2025

- **Week 10 - 13 Development**

Database Connectivity
Refining Front-End
Development of Classes

Date 8 May 2025 – 5 June 2025

- **Week 14 Testing**

Reviewing Software
Presentation

Date 6 June 2025- 12 June 2025

2. References

- ieee-documentation-template
- Systems and software engineering — Life cycle processes — Project management ISO/IEC/IEEE 16326:2019(E) Second edition 2019-12
- Project Libre
- Chapter 31-35 Pressman ISO/IEC/IEEE 16326-2019 guideline

3. Definition

- PMP:

  Project management plan is the world's leading project management certification.
- SRS:

  Software requirement specification is usually called a work product of a project plan. This document required a signature of stakeholders to move toward the development phase.
- IDE:

  An integrated development environment (IDE) is an application that facilitates the development of other Applications
- FTR:

  Formal Technical Review is a software quality control activity performed by software engineers.

4. Project Context

a. Process Model

- **Iterative Waterfall Model**

  Iterative Waterfall Model follows a structured, phase-based approach and is simple and easy to use. We are using this model because we want to use a sequential approach for our project. Secondly, we also have less time and a dedicated portion of time for each phase, so we want to complete one whole portion before going to the next one and the feedback loops enable revisiting and improving previous phases if necessary. This ensures both stability and flexibility of the project.

b. Methods, tools and techniques

Using the Iterative Waterfall model, we are using a sequential and iterative approach. When we work on one phase we are not focusing on any other phase and after moving to the next phase if we want, we can go back to the previous phase as well. This method will help us to use the dedicated portion for each phase effectively and revisit and improve previous phase if necessary.

As we are using a iterative waterfall model for the development of this project, we are using a tool which enables us to proceed from one phase to another in a seamless way to meet the project plan.

- **Programming Languages:** JavaScript (for frontend), PHP (for backend)
  JavaScript is used for developing the interactive and dynamic frontend of the application, ensuring a responsive and user-friendly interface. PHP is chosen for backend development as it is well-suited for web applications, providing efficient server-side scripting and database connectivity.
- **Frameworks & Libraries:** React.js (frontend), Laravel (backend)
  React.js allows for modular and reusable component-based UI development, improving maintainability and scalability. Laravel is a PHP framework that simplifies backend development with built-in security features, routing, and database management.
- **Database:** (PostgreSQL)
  PostgreSQL is used for data storage due to its reliability, scalability, and support for complex queries and transactions.
- **Presentation:** (Microsoft PowerPoint)

  At the end of this project, we are required to give presentation of our project and we will definitely use this software widely used for presentation purpose to give our final presentation.
- **Version Control:** (GitHub)
  GitHub is used for version control, enabling collaborative development, tracking changes, and ensuring code integrity.
- **Communication Tools:** Microsoft Teams, Email
  Microsoft Teams is used for team discussions, file sharing, and virtual meetings, while email ensures formal communication and documentation.
- **Development Tools:** Visual Studio Code, PHPStorm, Postman (API testing)
  Visual Studio Code and PHPStorm provide efficient coding environments with debugging and syntax highlighting features. Postman is used for API testing to ensure smooth backend functionality.
- **Project Management Tools:** Project Libre
  Project Libre is used for project scheduling and timeline management. It provides a range of tools and functionalities which we are looking for to complete our project plan.

c. Product Acceptance plan

As we follow the Iterative Water Fall Model in our project, we provide deliverables to our client for review periodically after two weeks. To do this, we use Formal Technical Reviews (FTR), during which our users can dually check our work and provide guidance on the next steps. We will ensure a high-quality product is a part of our product acceptance plan, which calls for holding weekly meetings to review current products and discuss the next steps

The system will undergo rigorous testing before deployment. The acceptance process will include:

1. **Unit Testing:** Developers will test individual components to ensure they work as expected.
2. **Integration Testing:** Verify that different modules work together correctly.
3. **User Acceptance Testing (UAT):** Students and supervisors will test the system to confirm it meets requirements.
4. **Final Approval:** The project will be validated by the department before deployment.

5. Project Planning:

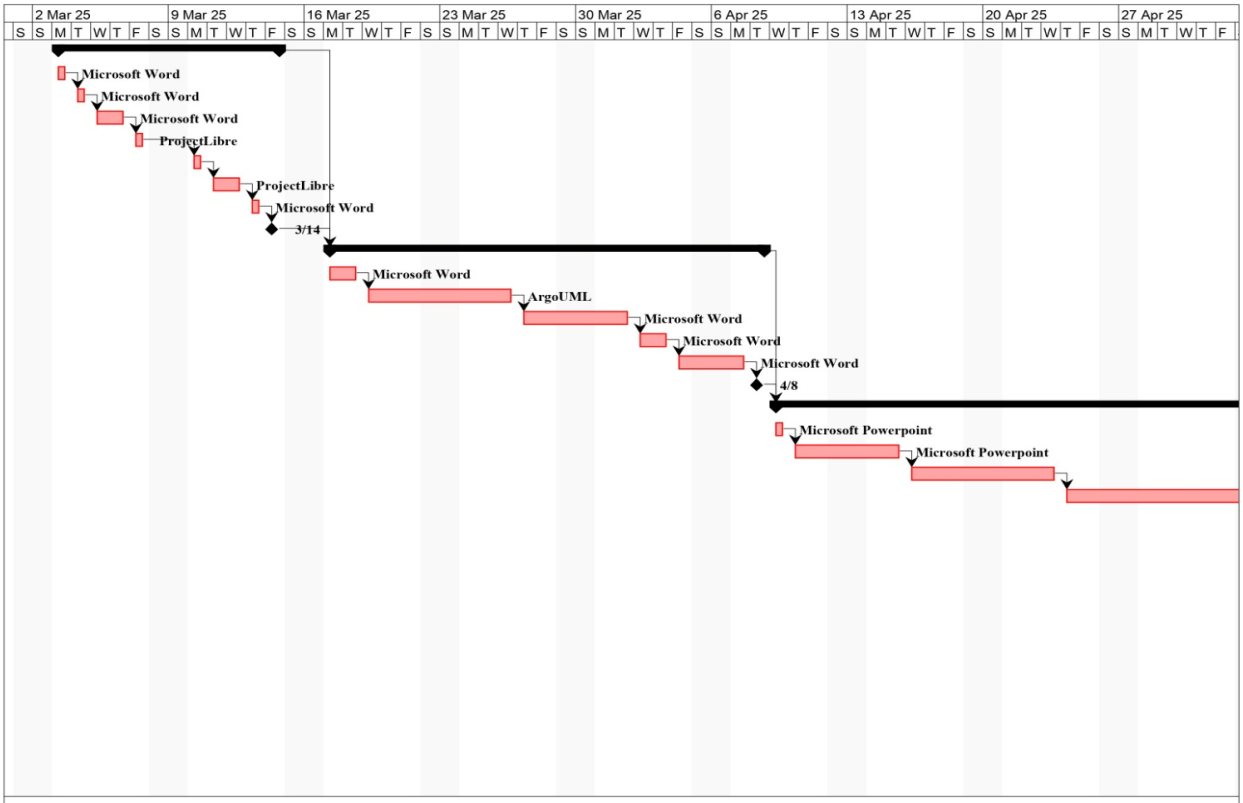
a. Work activities:

The work activities should be represented using a work breakdown structure where an accurate determination of the resources needed and the time frame for each work activity is possible. It contains the timeframe, resources allotted to the project, and all of the tasks required for each project phase.

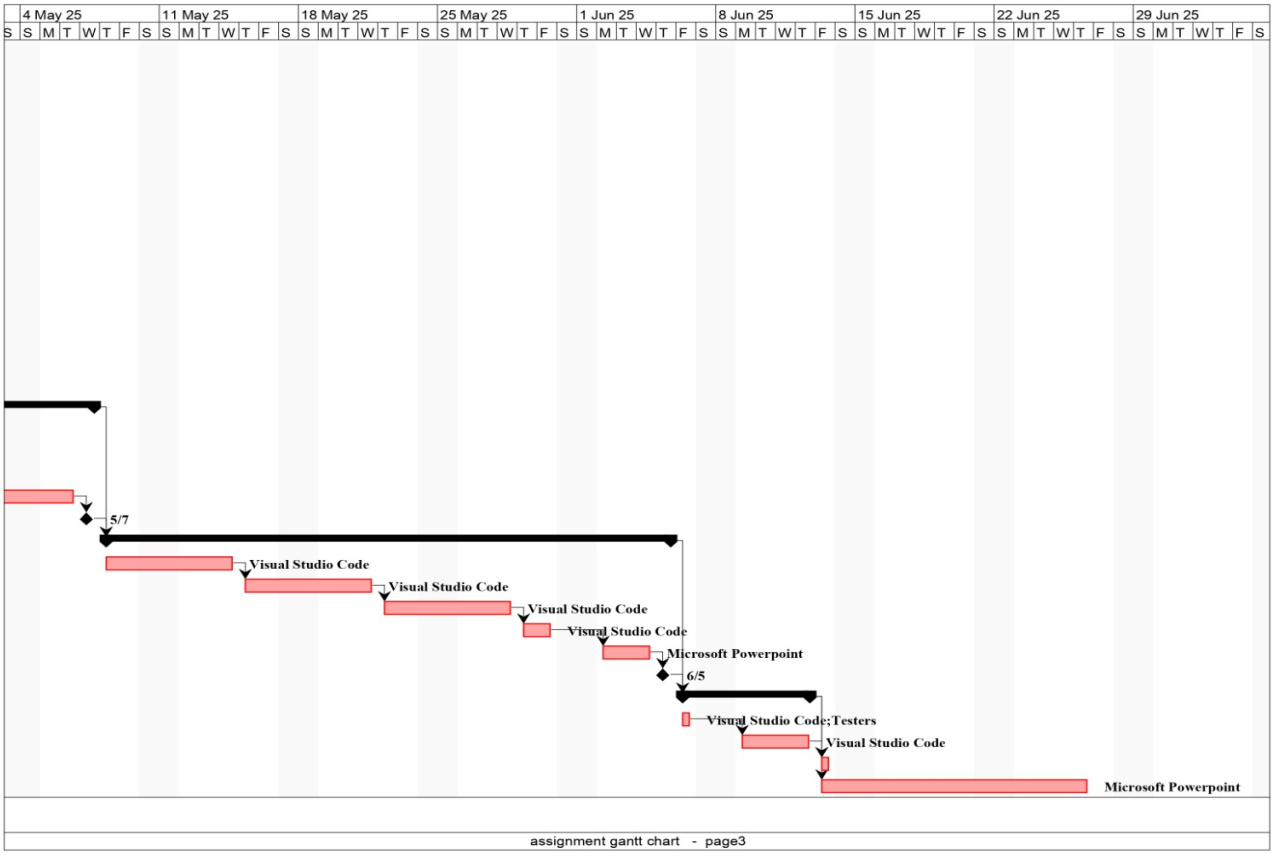
b. Schedule Allocation

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**planning**	**10 days?**	**3/3/25 8:00 AM**	**3/14/25 5:00 PM**		
2		project overview	1 day	3/3/25 8:00 AM	3/3/25 5:00 PM		Microsoft Word
3		Project Frount Mater	1 day?	3/4/25 8:00 AM	3/4/25 5:00 PM	2	Microsoft Word
4		Defining resourses	2 days	3/5/25 8:00 AM	3/6/25 5:00 PM	3	Microsoft Word
5		Budget planning	1 day?	3/7/25 8:00 AM	3/7/25 5:00 PM	4	ProjectLibre
6		Time planning	1 day	3/10/25 8:00 AM	3/10/25 5:00 PM	5	
7		define project context	2 days	3/11/25 8:00 AM	3/12/25 5:00 PM	6	ProjectLibre
8		Risk Management Plan	1 day?	3/13/25 8:00 AM	3/13/25 5:00 PM	7	Microsoft Word
9		Plan Featured	1 day?	3/14/25 8:00 AM	3/14/25 5:00 PM	8	
10		**Analysis**	**17 days?**	**3/17/25 8:00 AM**	**4/8/25 5:00 PM**	**1;9**	
11		Define Usecases	2 days	3/17/25 8:00 AM	3/18/25 5:00 PM		Microsoft Word
12		Develop Analysis Model	6 days	3/19/25 8:00 AM	3/26/25 5:00 PM	11	ArgoUML
13		Develop SRS	4 days	3/27/25 8:00 AM	4/1/25 5:00 PM	12	Microsoft Word
14		Review SRS	2 days	4/2/25 8:00 AM	4/3/25 5:00 PM	13	Microsoft Word
15		Refine SRS	2 days	4/4/25 8:00 AM	4/7/25 5:00 PM	14	Microsoft Word
16		Reviewed and Refined SRS	1 day?	4/8/25 8:00 AM	4/8/25 5:00 PM	15	Microsoft Word
17		**Design**	**21 days?**	**4/9/25 8:00 AM**	**5/7/25 5:00 PM**	**10;16**	
18		Discussing Design Principle	1 day	4/9/25 8:00 AM	4/9/25 5:00 PM		Microsoft Powerpoint
19		Data Design	4 days	4/10/25 8:00 AM	4/15/25 5:00 PM	18	Microsoft Powerpoint
20		Interface Design	6 days	4/16/25 8:00 AM	4/23/25 5:00 PM	19	
21		Detailed Desing	9 days	4/24/25 8:00 AM	5/6/25 5:00 PM	20	
22		Final Product Desing	1 day?	5/7/25 8:00 AM	5/7/25 5:00 PM	21	
23		**Development**	**21 days?**	**5/8/25 8:00 AM**	**6/5/25 5:00 PM**	**17;22**	
24		Database connectivity	5 days	5/8/25 8:00 AM	5/14/25 5:00 PM		Visual Studio Code
25		Refining Front end	5 days	5/15/25 8:00 AM	5/21/25 5:00 PM	24	Visual Studio Code
26		Develop Classses	5 days	5/22/25 8:00 AM	5/28/25 5:00 PM	25	Visual Studio Code
27		Refining Classes	2 days	5/29/25 8:00 AM	5/30/25 5:00 PM	26	Visual Studio Code
28		Tracing classes with desi...	3 days	6/2/25 8:00 AM	6/4/25 5:00 PM	27	Microsoft Powerpoint
29		Working Product	1 day?	6/5/25 8:00 AM	6/5/25 5:00 PM	28	
30		**Testing**	**5 days?**	**6/6/25 8:00 AM**	**6/12/25 5:00 PM**	**23;29**	
31		Reviewing the Develope...	1 day?	6/6/25 8:00 AM	6/6/25 5:00 PM		Visual Studio Code;Testers
32		Removing Errors	4 days	6/9/25 8:00 AM	6/12/25 5:00 PM	31	Visual Studio Code
33		Final Testing Product	1 day	6/13/25 8:00 AM	6/13/25 5:00 PM	30;32	
34		Presentations	10 days	6/13/25 8:00 AM	6/26/25 5:00 PM	30	Microsoft Powerpoint

assignment gantt chart - page1



assignment gantt chart - page2



c. Resource Allocation

	Name	RBS	Type	E-mail Address	Material Label	Initials	Group	Max. Units	Standard Rate
1	Microsoft Word		Material			M			$0.00
2	ProjectLibre		Material			P			$0.00
3	Visual Studio Code		Material			V			$0.00
4	ArgoUML		Material			A			$0.00
5	Microsoft Powerpoint		Material			M			$0.00
6	Testers		Material			T			$0.00

6. Supporting Process Plans

a. Risk Management:

Risk management is crucial to ensure the smooth development and implementation of the Project Management System. The following can be the possible risks:

i. Team members leaving:

- If any member is leaving, who has a significant role in software management, provides him with a more feasible environment or provide him with the desired package if he deserves this.
- Moreover, keep at least 2 employees at the back end so if anyone leaves you must have a backup person to lessen your loss

ii. Loss of Data:

- To avoid data loss, one must have at least 2 backup plans. One in the hard form and one in the soft form. So, if in any case your data is lost one must already know how to cope with it.

iii. Power Breakdown:

- If due to any reason power breakdown occur and your work get deleted unsaved one must have remedies defined for them as generators even work manually we have to switch on them so as to avoid this the whole system should be connected to solar plates so that if power supply get off within less than a second it is connected to solar continuing the work at its best.

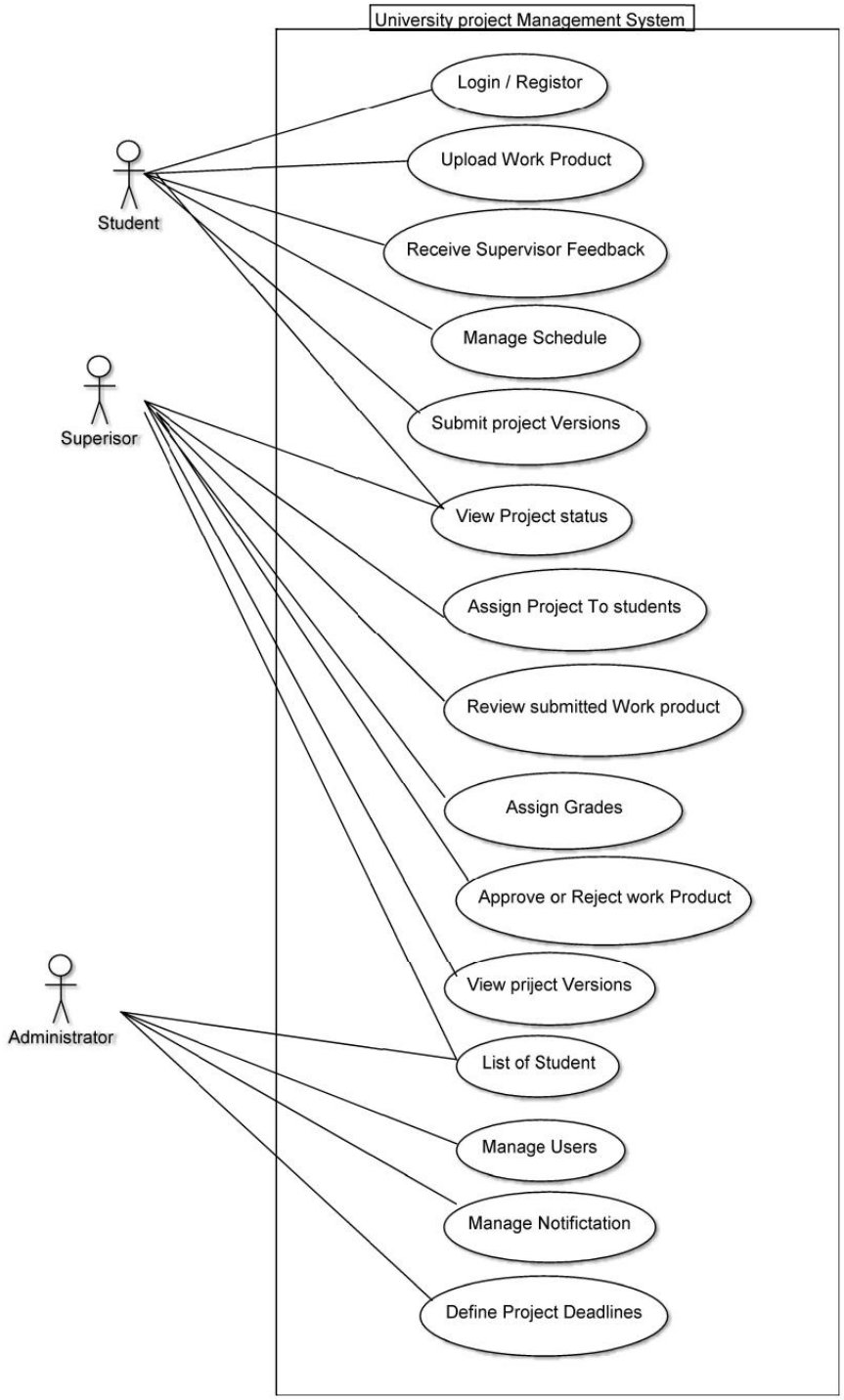
iv. Technical Challenges:

- Integration issues between frontend (React.js) and backend (Laravel) so, conduct regular integration testing and maintain proper documentation.

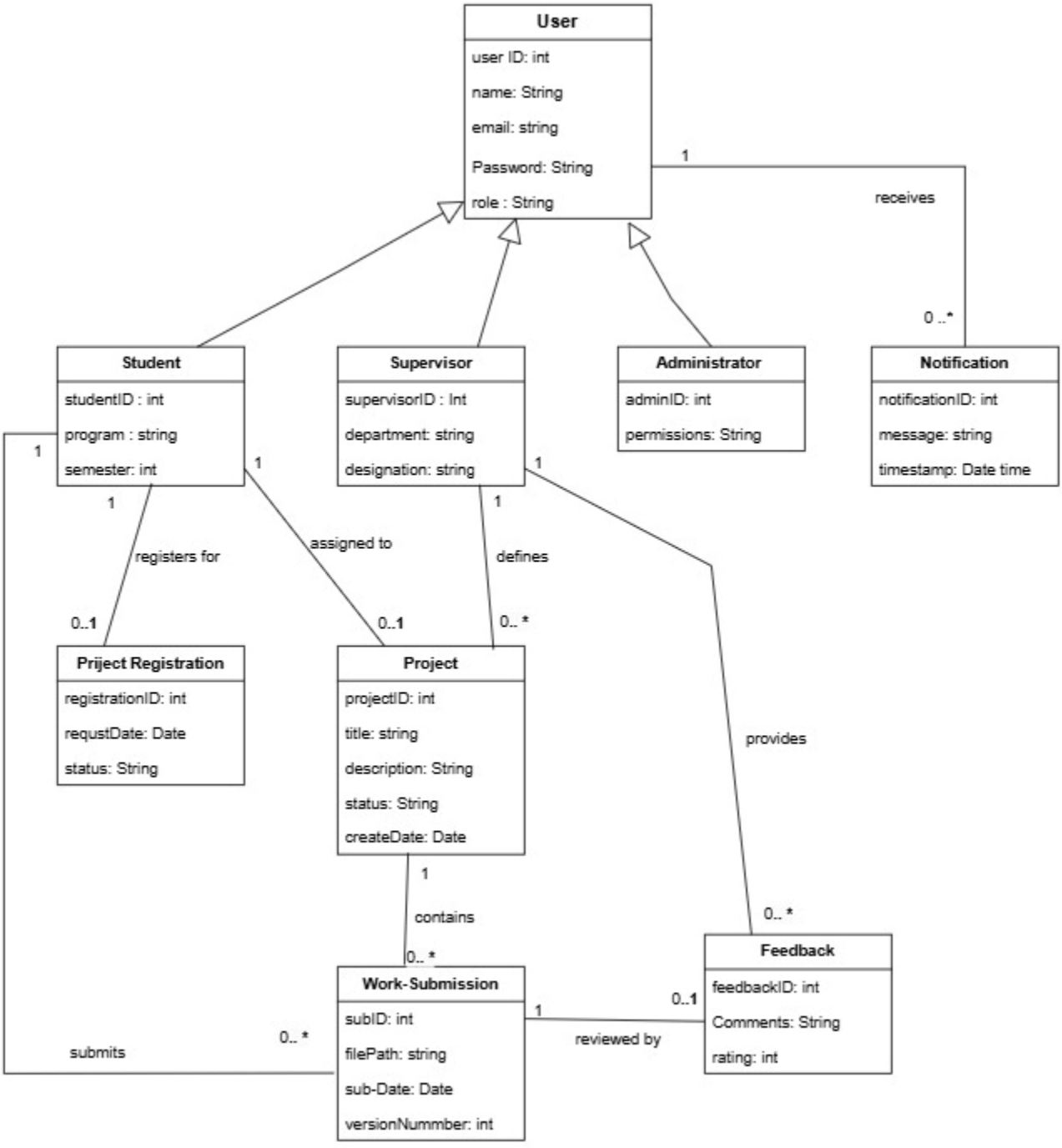
v. Lack of User Adoption:

- Students and supervisors may not use the system as expected so, we can provide user training and detailed documentation to ease adoption.

7. Use case diagram



8. Domain Model



9. Usecase description

1. Login/Register

- **Actor:** Student, Supervisor, Administrator
- **Preconditions:**
  - User must have valid credentials to log in.
- **Inputs:**
  - Username/Email
  - Password
  - Role (Student, Supervisor, Administrator)
- **Main Success Scenario:**
  - The user enters login credentials.
  - The system validates the credentials.
  - If valid, the user is granted access.
- **Alternative Flow:**
  - If the credentials are incorrect, the system displays an error message.
  - If too many failed attempts occur, the account is locked.
- **Outputs:**
  - Success message upon login.
  - Error message if login fails.
- **Postconditions:**
  - The user is logged in successfully.

2. Upload Work Product

- **Actor:** Student
- **Preconditions:**
  - The student must be assigned a project.
- **Inputs:**
  - Work file (PDF, DOCX, ZIP)
  - Student ID
  - Project ID
- **Main Success Scenario:**
  - The student logs in and navigates to the project submission page.
  - The student selects and uploads a work product file.
  - The system validates and stores the file.
  - The system notifies the supervisor.
- **Alternative Flow:**
  - If the file format is incorrect, the system rejects the upload.
- **Outputs:**
  - Confirmation message after successful upload.
- **Postconditions:**
  - The work product is stored and accessible for review.

3. Receive Supervisor Feedback

- **Actor:** Student, Supervisor
- **Preconditions:**
  - A work submission must exist.
- **Inputs:**
  - Submission ID
  - Supervisor Feedback
- **Main Success Scenario:**
  - The supervisor logs in and selects a submission.
  - The supervisor adds feedback.
  - The system stores the feedback and notifies the student.
  - The student logs in to view feedback.
- **Outputs:**
  - Notification to the student.
- **Postconditions:**
  - The feedback is recorded.

4. Manage Schedule

- **Actor:** Student, Supervisor
- **Preconditions:**
  - The project must have deadlines assigned.
- **Inputs:**
  - Project ID
  - Task Schedule
- **Main Success Scenario:**
  - The user logs in and accesses the schedule.
  - The user adds or updates scheduled tasks.
- **Outputs:**
  - Updated project schedule.
- **Postconditions:**
  - The updated schedule is stored.

5. Submit Project Versions

- **Actor:** Student
- **Preconditions:**
  - A project must exist.
- **Inputs:**
  - Updated work file
- **Main Success Scenario:**
  - The student logs in and selects a project.

    - The student uploads a new version.
    - The system increments the version number.
- **Outputs:**
    - Confirmation message.
- **Postconditions:**
    - The submission is stored.

6. View Project Status

- **Actor:** Student
- **Preconditions:**
    - The student must have an assigned project.
- **Inputs:**
    - Project ID
- **Main Success Scenario:**
    - The student logs in and navigates to project status.
    - The system displays project details.
- **Outputs:**
    - Project progress report.
- **Postconditions:**
    - The student views the latest project updates.

7. Assign Project to Students

- **Actor:** Supervisor
- **Preconditions:**
    - Available projects must exist.
- **Inputs:**
    - Student ID
    - Project ID
- **Main Success Scenario:**
    - The supervisor selects a student and assigns a project.
    - The system updates the student's record.
- **Outputs:**
    - Assignment confirmation.
- **Postconditions:**
    - The student is linked to the project.

8. Review Submitted Work Product

- **Actor:** Supervisor
- **Preconditions:**
    - A student must have submitted a work product.
- **Inputs:**

- Submission ID

- **Main Success Scenario:**
  - The supervisor selects a submission.
  - The supervisor reviews and provides feedback.
- **Outputs:**
  - Feedback notification.
- **Postconditions:**
  - Feedback is stored.

9. Assign Grades

- **Actor:** Supervisor
- **Preconditions:**
  - The project must be completed.
- **Inputs:**
  - Submission ID
  - Grade
- **Main Success Scenario:**
  - The supervisor assigns a grade.
  - The system updates the student's record.
- **Outputs:**
  - Grade notification.
- **Postconditions:**
  - The grade is stored.

10. Approve or Reject Work Product

- **Actor:** Supervisor
- **Preconditions:**
  - A work submission must exist.
- **Inputs:**
  - Submission ID
  - Approval or Rejection Decision
- **Main Success Scenario:**
  - The supervisor selects a submission.
  - The supervisor approves or rejects it.
- **Outputs:**
  - Notification to the student.
- **Postconditions:**
  - The decision is recorded.

11. View Project Versions

- **Actor:** Student, Supervisor
- **Preconditions:**
  - The student must have submitted at least one version.
- **Inputs:**
  - Project ID
- **Main Success Scenario:**
  - Supervisor requests project versions.
  - System retrieves versions.
  - System displays project versions.
- **Outputs:**
  - List of project versions.

12. List of Students

- **Actor:** Supervisor
- **Preconditions:**
  - Students must be registered.
- **Inputs:**
  - Supervisor ID
- **Main Success Scenario:**
  - Administrator requests student list.
  - System retrieves the list.
  - System displays student details.
- **Outputs:**
  - List of students.

13. Manage Users

- **Actor:** Administrator
- **Preconditions:**
  - Admin privileges required.
- **Inputs:**
  - User details
- **Main Success Scenario:**
  - The admin adds or modifies user accounts.
  - System updates user records.
  - System confirms the update.
- **Outputs:**
  - Confirmation message.

14. Manage Notifications

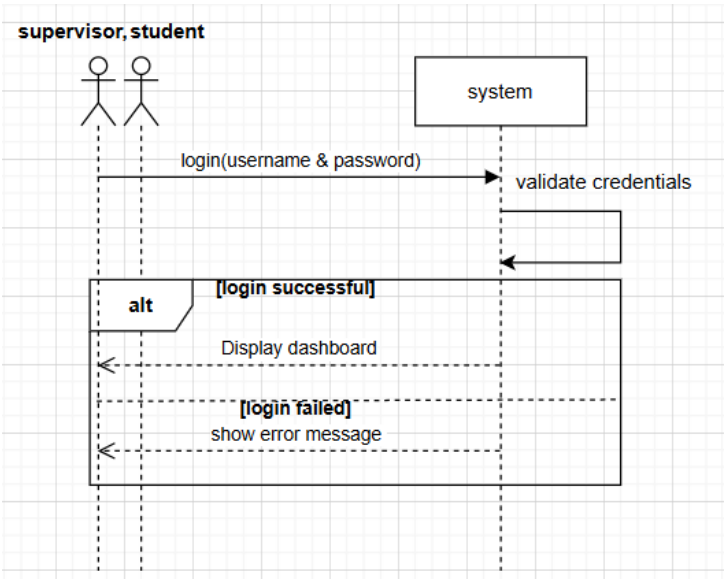
- **Actor:** System
- **Preconditions:**
  - A relevant event occurs.
- **Inputs:**
  - Notification Type
- **Main Success Scenario:**
  - Administrator selects notification settings.
  - Administrator updates notification settings.
  - System stores changes.
  - System displays confirmation
- **Outputs:**
  - Notification sent.

15. Define Project Deadlines

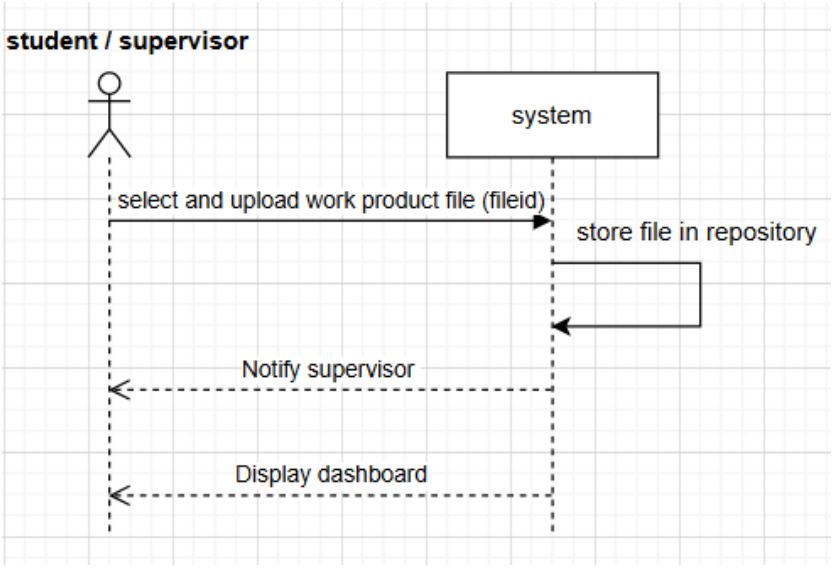
- **Actor:** Supervisor
- **Preconditions:**
  - The project must be registered.
- **Inputs:**
  - Deadline Date
- **Main Success Scenario:**
  - Administrator selects project deadlines.
  - Administrator sets new deadlines.
  - System stores deadlines.
  - System notifies students and supervisors.
- **Outputs:**
  - Deadline notification

10. System sequence diagram

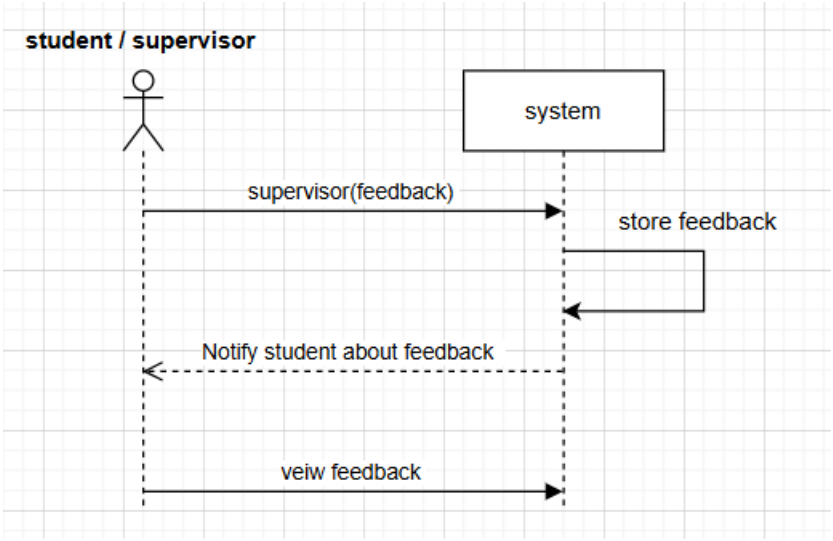
1. Login



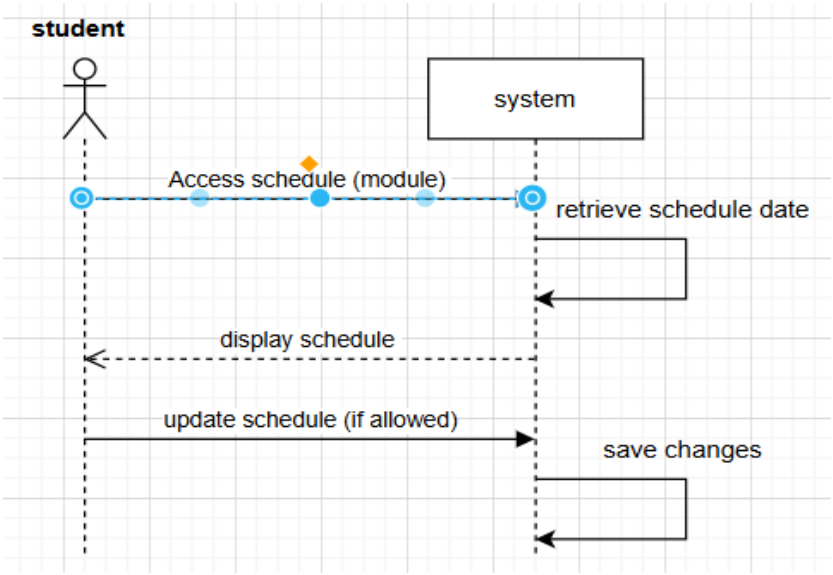
2. Upload Work Submission



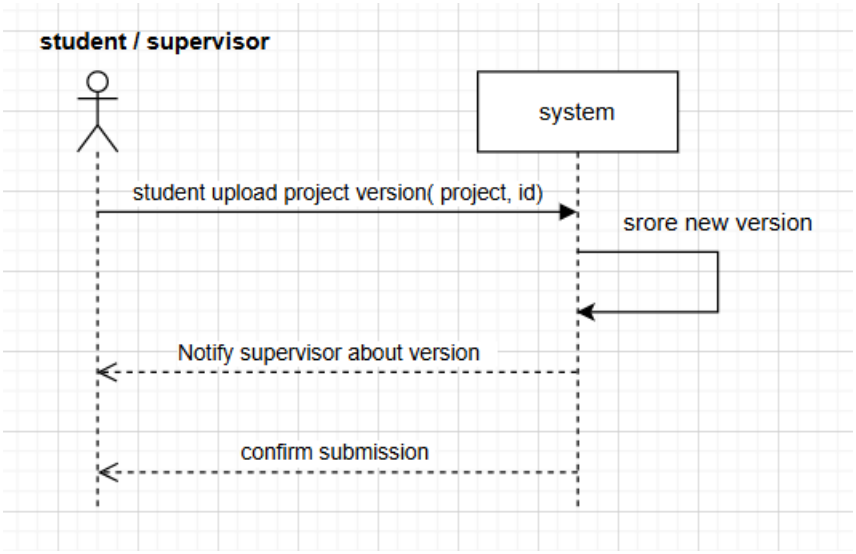
3. Receive Supervisor Feedback



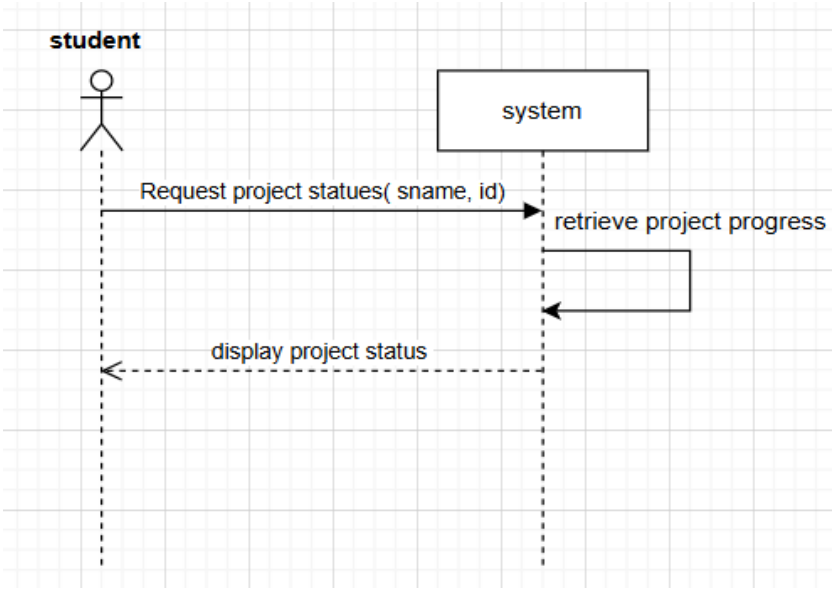
4. Manage Schedule



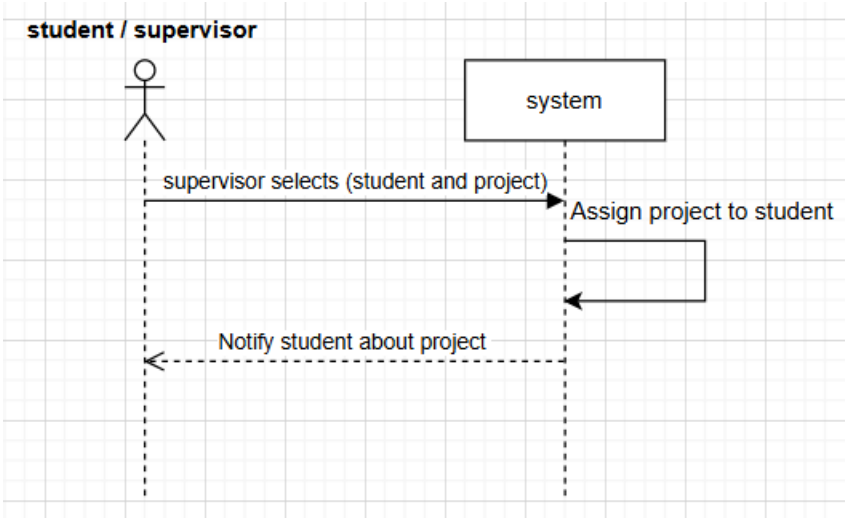
5. Submit Project Versions



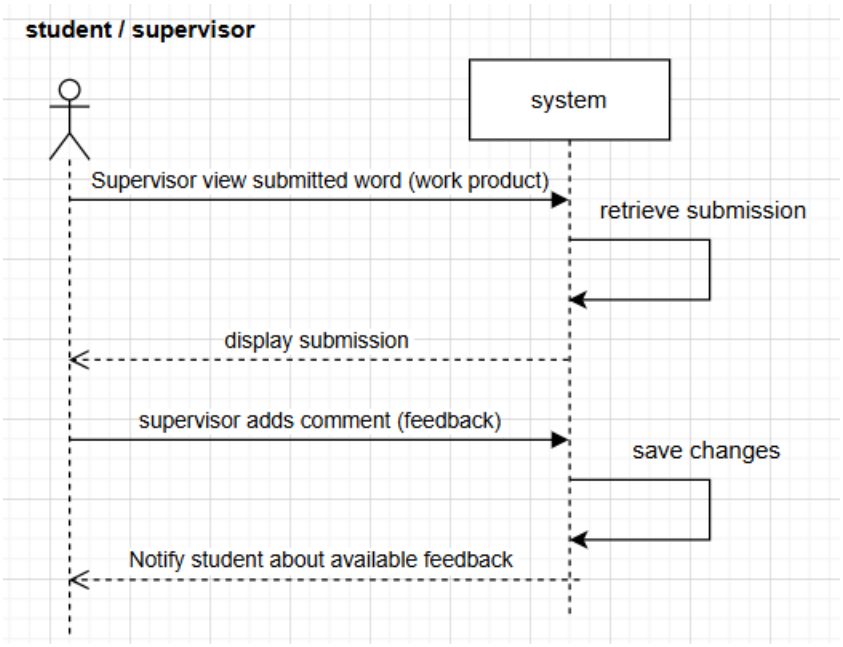
6. View Project Status



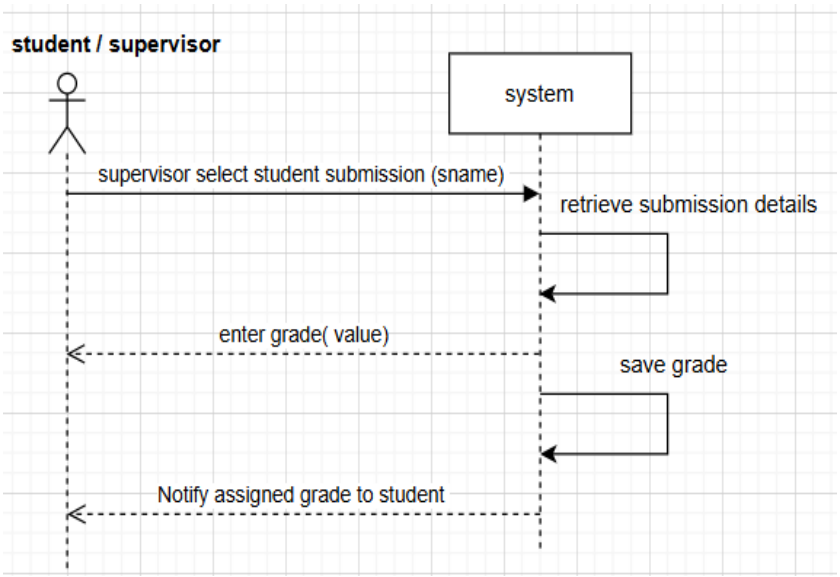
7. Assign Project to Students



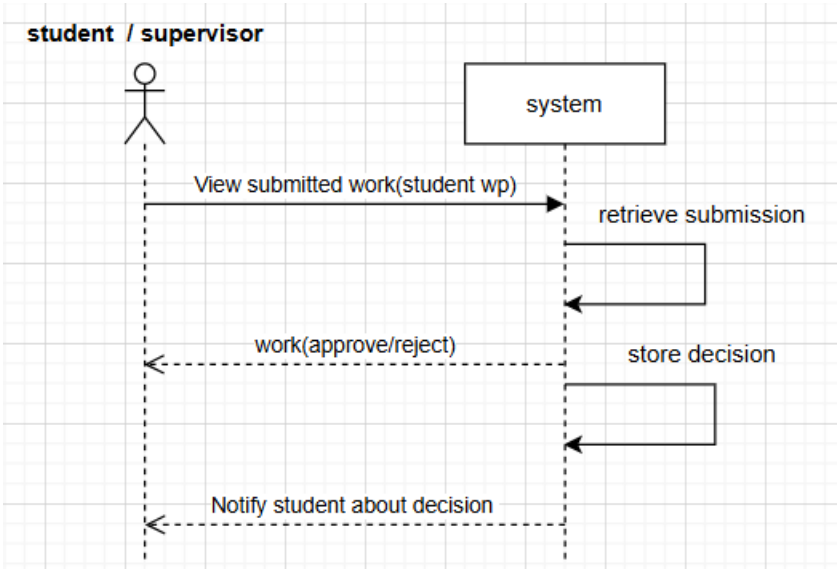
8. Review Submitted Work Product



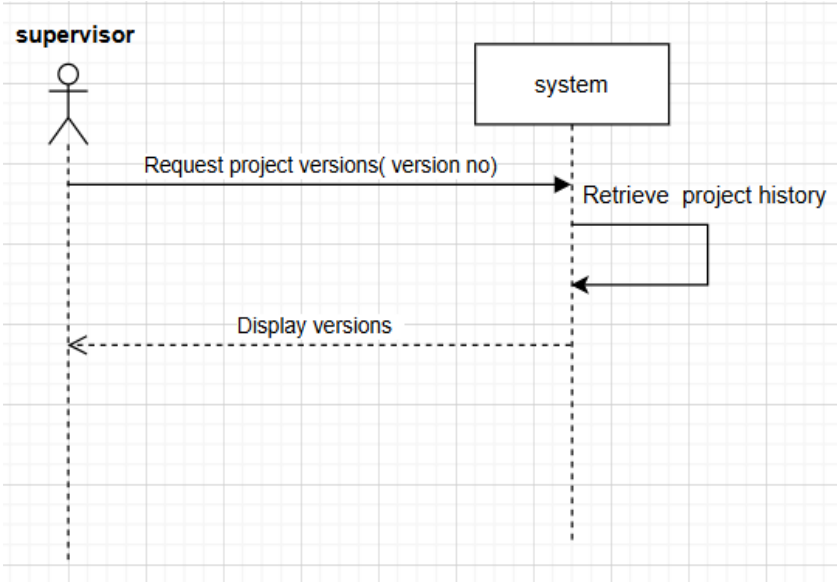
9. Assign Grades



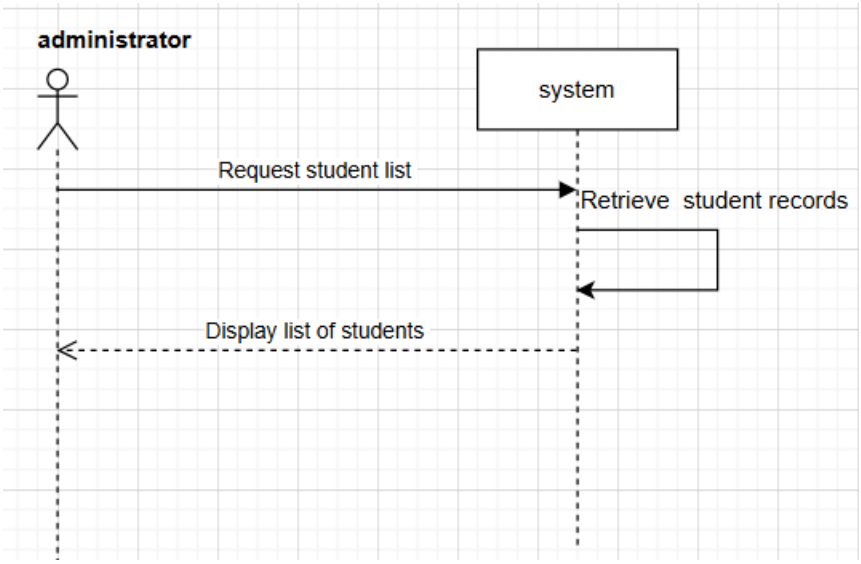
10. Approve or Reject Work Product



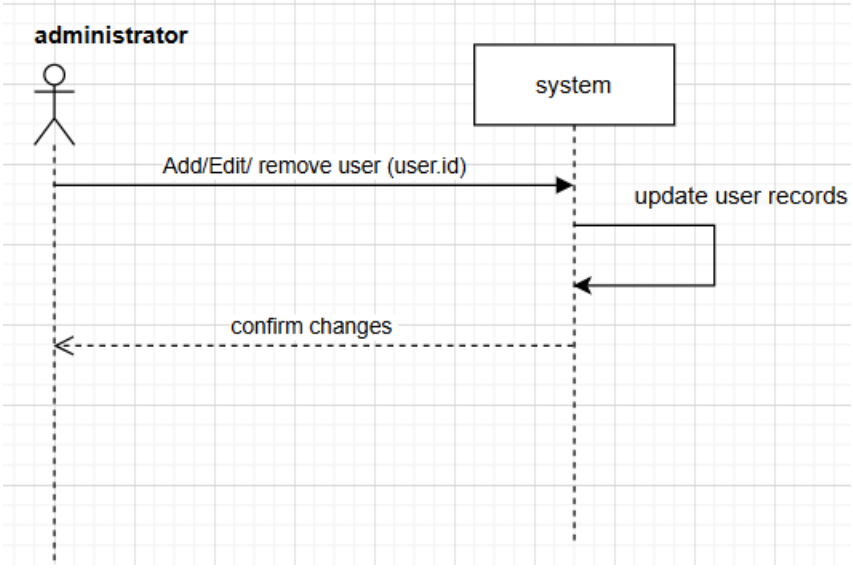
11. View Project Versions



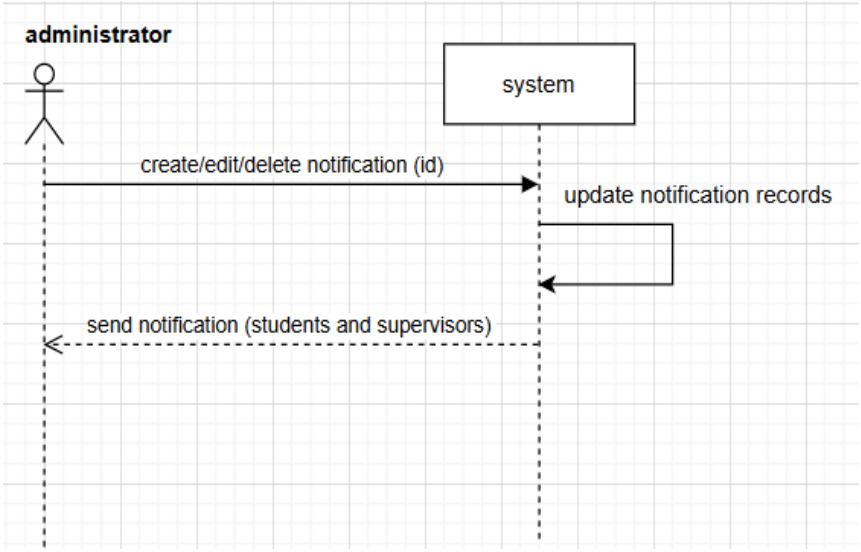
12. List of Students



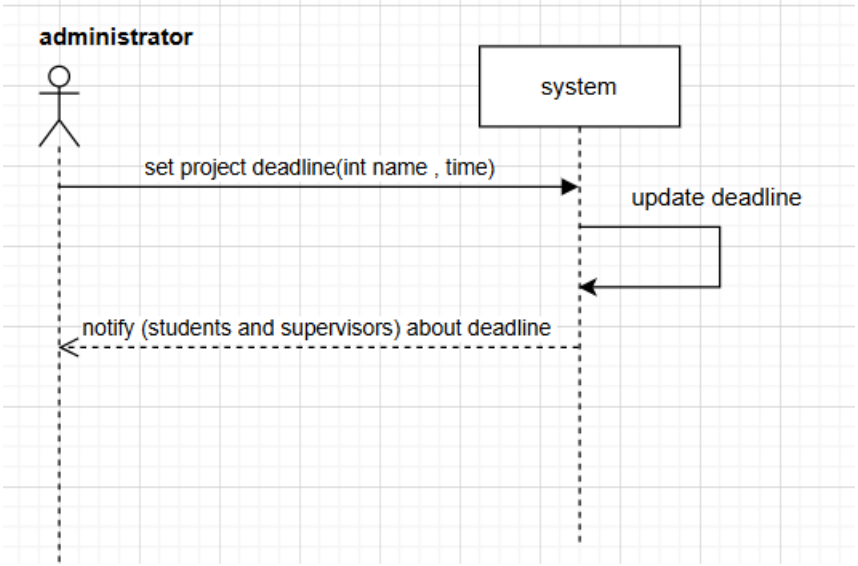
13. Manage Users



14. Manage Notifications



15. Define Project Deadlines



11. Data Dictionary

1. Login

Field Name	Description	Data Type	Length	Conditions
user_id	Unique identifier for each user	INT	10	Primary Key, Auto-increment
username	User's login name	STRING	50	Must be unique
email	User's email address	STRING	100	Must be a valid email format
password	User's encrypted password	STRING	255	Must be hashed

2. Upload Work Product

Field Name	Description	Data Type	Length	Conditions
submission_id	Unique identifier for each submission	INT	10	Primary Key, Auto-increment
student_id	ID of the student submitting the work	INT	10	Foreign Key (References users table)
project_id	ID of the assigned project	INT	10	Foreign Key (References projects table)
file_path	File location	STRING	255	Cannot be NULL
file_type	Type of the uploaded file (PDF, DOCX, ZIP)	STRING	10	Must match predefined formats
submission_date	Date and time of submission	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format
version	Version number of the submission	INT	3	Default: 1, Must be ≥ 1

3. Receive Supervisor Feedback

Field Name	Description	Data Type	Length	Conditions
feedback_id	Unique identifier for feedback	INT	10	Primary Key, Auto-increment
submission_id	ID of the related submission	INT	10	Foreign Key (References submissions table)
supervisor_id	ID of the supervisor providing feedback	INT	10	Foreign Key (References users table)

comments	Feedback comments	STRING	N/A	Cannot be NULL
timestamp	Date and time of feedback submission	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format

4. Manage Schedule

Field Name	Description	Data Type	Length	Conditions
schedule_id	Unique ID for each schedule entry	INT	10	Primary Key, Auto-increment
project_id	ID of the associated project	INT	10	Foreign Key (References projects table)
task_name	Name of the scheduled task	STRING	100	Cannot be NULL
start_date	Task start date	DATE	N/A	Must follow 'YYYY-MM-DD' format
end_date	Task end date	DATE	N/A	Must follow 'YYYY-MM-DD' format

5. Submit Project Versions

Field Name	Description	Data Type	Length	Conditions
version_id	Unique ID for each project version	INT	10	Primary Key, Auto-increment
submission_id	ID of the work submission	INT	10	Foreign Key (References submissions table)
version_number	Version number of the submission	INT	3	Default: 1, Must be ≥ 1
upload_date	Date of version submission	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format

6. View Project Status

Field Name	Description	Data Type	Length	Conditions
status_id	Unique ID for each project status entry	INT	10	Primary Key, Auto-increment
project_id	ID of the project	INT	10	Foreign Key (References projects table)
status	Current status of the project	ENUM	N/A	Values: 'In Progress', 'Completed', 'Pending'

7. Assign Project to Students

Field Name	Description	Data Type	Length	Conditions
assignment_id	Unique ID for the project assignment	INT	10	Primary Key, Auto-increment
student_id	ID of the assigned student	INT	10	Foreign Key (References users table)
project_id	ID of the assigned project	INT	10	Foreign Key (References projects table)

8. Review submitted Work Product

Field Name	Description	Data Type	Length	Conditions
review_id	Unique ID for the review entry	INT	10	Primary Key, Auto-increment
submission_id	ID of the submitted work	INT	10	Foreign Key (References submissions table)
supervisor_id	ID of the supervisor reviewing the work	INT	10	Foreign Key (References users table)
student_id	ID of the student who submitted the work	INT	10	Foreign Key (References users table)
review_comments	Supervisor's feedback on the work	STRING	255	Cannot be NULL
review_date	Date and time of review	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format
status	Work product status after review	ENUM	N/A	Values: 'Accepted', 'Needs Improvement', 'Rejected'

9. Assign Grades

Field Name	Description	Data Type	Length	Conditions
grade_id	Unique ID for the grading entry	INT	10	Primary Key, Auto-increment
submission_id	ID of the submitted work	INT	10	Foreign Key (References submissions table)

student_id	ID of the student receiving the grade	INT	10	Foreign Key (References users table)
supervisor_id	ID of the supervisor assigning the grade	INT	10	Foreign Key (References users table)
grade	Assigned grade (Letter or Numeric)	STRING	5	Values: 'A', 'B', 'C', 'D', 'F' or numeric range (0-100)
grading_comments	Supervisor's comments on the grade	STRING	255	Optional field
grading_date	Date and time of grading	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format

10.Approve or Reject Work Product

Field Name	Description	Data Type	Length	Conditions
decision_id	Unique ID for approval/rejection	INT	10	Primary Key, Auto-increment
submission_id	ID of the work submission	INT	10	Foreign Key (References submissions table)
supervisor_id	ID of the reviewing supervisor	INT	10	Foreign Key (References users table)
decision	Approval or rejection status	ENUM	N/A	Values: 'Approved', 'Rejected'
comments	Supervisor comments for decision	STRING	N/A	Optional field
decision_date	Date and time of approval/rejection	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format

11.View Project Versions

Field Name	Description	Data Type	Length	Conditions
version_id	Unique ID for project version	INT	10	Primary Key, Auto-increment
submission_id	ID of the work submission	INT	10	Foreign Key (References submissions table)
version_number	Version number	INT	3	Default: 1, Must be ≥ 1
file_path	Location of the uploaded version	STRING	255	Cannot be NULL
upload_date	Date and time of	DATETIME	N/A	Must follow 'YYYY-

	version submission			MM-DD HH:MM:SS' format

12.List of Students

Field Name	Description	Data Type	Length	Conditions
student_id	Unique ID for the student	INT	10	Primary Key, Auto-increment
name	Student's full name	STRING	100	Cannot be NULL
email	Student's email address	STRING	100	Must be a valid email format
registration_number	Student's university registration number	STRING	20	Must be unique
assigned_project_id	ID of the assigned project	INT	10	Foreign Key (References projects table)

13.Manage Users

Field Name	Description	Data Type	Length	Conditions
user_id	Unique identifier for each user	INT	10	Primary Key, Auto-increment
username	User's login name	STRING	50	Must be unique
email	User's email address	STRING	100	Must be a valid email format
password	User's encrypted password	STRING	255	Must be hashed
role	User role (Student, Supervisor, Admin)	ENUM	N/A	Values: 'Student', 'Supervisor', 'Admin'
status	Account status	ENUM	N/A	Values: 'Active', 'Inactive'

14.Manage Notifications

Field Name	Description	Data Type	Length	Conditions
notification_id	Unique ID for notification	INT	10	Primary Key, Auto-increment
user_id	ID of the recipient	INT	10	Foreign Key (References users table)
message	Notification content	STRING	N/A	Cannot be NULL
status	Notification status	ENUM	N/A	Values: 'Unread', 'Read'
timestamp	Date and time of notification	DATETIME	N/A	Must follow 'YYYY-MM-DD HH:MM:SS' format

15.Define Project Deadlines

Field Name	Description	Data Type	Length	Conditions
deadline_id	Unique ID for project deadline	INT	10	Primary Key, Auto-increment
project_id	ID of the assigned project	INT	10	Foreign Key (References projects table)
deadline_date	Submission deadline	DATE	N/A	Must follow 'YYYY-MM-DD' format
description	Details about the deadline	STRING	255	Optional field