SE4106 – WEB SYSTEMS AND TECHNOLOGIES PROJECT PROPOSAL

PROJECT SUPERVISION MANAGEMENT SYSTEM

(PSMS)

DEPARTMENT OF SOFTWARE ENGINEERING

FACULTY OF COMPUTING

SABARAGAMUWA UNIVERSITY OF SRI LANKA

MARCH 2025

Approval of Web System & Technologies Project

Group Details :

1. Index No	2. Name with Initials	3. Email	4. Mobile No	Signature of student
21CSE0156	K.S.K.Kumaranayaka	kskkumaranayaka@std.foc.sab.ac.lk	0702001614	
21CSE0157	W.P.I.N.Kumarasinghe	wpinkumarasinghe@std.foc.sab.ac.lk	0765735091	
21CSE0158	L.B.Madhuranga	lbmadhuranga@std.foc.sab.ac.lk	0763546936	

1. **Title of the Project** : Project Supervision Management System **(PSMS)**
2. **Name of the Supervisor** : Dr. U.A.P.Ishanka
3. **Supervisor's Designation** : Doctor
4. **Supervisor's Organization** : Faculty of Computing
   Sabaragamuwa University of Sri Lanka

For officer use only :

Approved/Not approved : Approved

Signature of the Supervisor : 

Date : 24.03.2025

Suggestions if any :

Contents

1. Acknowledgement

We would like to take this opportunity to extend our heartfelt gratitude to **Dr.U.A.P.Ishanka**, our respected supervisor, for her exceptional guidance, support, and mentorship throughout the development of our Project Supervision Management System **(PSMS)**. Her profound knowledge, dedication, and continuous encouragement have been key factors in steering our project toward successful completion. Her valuable advice and constructive feedback have greatly shaped the direction and quality of our work.

We also wish to express our sincere appreciation to **Mrs.W.M.L.S.Abeythunga**, our subject lecturer, for her insightful guidance and support throughout this project. Her expertise and suggestions have been incredibly helpful in deepening our understanding and improving the overall standard of our project.

Furthermore, we are grateful to all the faculty members, staff, and fellow students who have contributed to our project with their thoughtful feedback, assistance, and resources. Their input has been vital and much appreciated.

Finally, we would like to thank our family and friends for their unwavering support, understanding, and encouragement throughout this journey. Their constant belief in us has been a great source of motivation and strength.

Thank you.

Sincerely,
Group 04

2. Introduction & Objective of the Web Systems and Technologies Project

2.1 Introduction

The Project Supervision Management System (PSMS) is a web-based system designed to address the challenges faced by both teachers and students at the Faculty of Computing, Sabaragamuwa University, Sri Lanka. With an increasing number of students and limited teaching staff, teachers often struggle with managing emails, assignment submissions, and feedback, leading to delays and communication issues. This platform aims to streamline these processes and improve efficiency in managing assignments, group work, and teacher-student interactions.

PSMS enables teachers to manage their assigned student groups, track assignment submissions, provide feedback, and ensure deadlines are met. On the student side, the platform facilitates group collaboration, assignment submission and receiving teacher feedback. Additionally, students can easily communicate with their teachers within the system, ensuring that vital information is not lost in an overwhelming flood of e-mails.

The platform includes features like login and authentication for different user roles (Admin, Teacher, and Student), group creation, assignment management, feedback submission, notifications for deadlines, and much more. With an intuitive and accessible design, PSMS is suitable for users of varying technical proficiency.

By implementing this solution, the university aims to simplify administrative tasks, reduce reliance on email communication, and improve overall workflow for both teachers and students.

2.2 Abstraction

The proposed web system for the Project Supervision Management System (PSMS) at the Faculty of Computing, Sabaragamuwa University, aims to streamline communication between lecturers and students. By addressing the challenge of overwhelming email traffic, the system provides an intuitive platform for managing projects, tracking deadlines, and facilitating structured feedback. With features such as proposal submission, real-time notifications, and an integrated feedback system, the platform ensures efficient supervision and timely responses. Designed for accessibility and ease of use, PSMS enhances collaboration, reduces administrative burden, and fosters a more organized academic workflow.

2.3 Objectives

The main objective of the Project Supervision Management System **(PSMS)** is to develop a user-friendly web platform that facilitates seamless communication between teachers and students at the Faculty of Computing, Sabaragamuwa University of Sri Lanka. This platform aims to streamline assignment management, feedback, and group collaborations while reducing the communication gaps that currently exist due to email overload. The project further aims to achieve the following objectives:

- Provide Organized and Up-to-Date Assignment Information

  Ensure that the PSMS platform offers detailed and up-to-date information regarding student group activities, assignment deadlines, supervisor feedback, and group-specific announcements, catering to the diverse academic requirements of all student batches and departments.

- Encourage Active Teacher-Student Interaction and Collaboration

  Create interactive features such as real-time notifications, assignment submission tracking, and direct feedback options to promote active communication and collaboration between teachers and students, thereby enhancing the learning experience and timely project completion.

- Establish Efficient Communication Channels

  Establish an efficient, centralized communication system within PCMS that allows teachers, students to interact seamlessly. This includes managing assignment-related queries, monitoring group activities, and providing timely responses to students' requests.

- Ensure User-Friendly and Intuitive Design

  Implement intuitive design principles and user interface enhancements to ensure a simple, user-friendly, and responsive web environment that can be easily accessed and navigated by teachers and students of varying technical expertise.

- Incorporate Feedback and Continuous Improvement Mechanisms

  Enable built-in feedback mechanisms and surveys to collect input from both teachers and students regarding their experiences, suggestions, and challenges, thus allowing for continuous improvement of the platform's features and services offered.

3. Analysis

3.1 Technical Feasibility

Technical feasibility of a project or system's involves determining if it can be created and executed with the resources and technology available currently. It entails assessing a range of technical factors to ascertain whether the project can be completed within limitations imposed by budget, time, and technology. Here's the things for evaluating technical feasibility

1. Integration of APIs

Required APIs:

- Email/Notification API for alerts.
- Database API (MySQL, PostgreSQL, or Firebase) for storing assignments, feedback, and user data.
- Authentication API for secure logins.
- APIs for authentication and notifications are widely available and easy to integrate

2. Dependable Smartphone and Internet Penetration Rates

Target Users:

- University students
- Admin
- Lectures

Internet & Device Availability:

- University environments generally have stable internet via WiFi.
- Most students and teachers have access to laptops, desktops, and smartphones.
- Optimize performance for low-speed networks by reducing complex API calls and using compressed assets.

3.2 Compatibility with Different Devices and Operating Systems

Target Devices:

- Desktops, Laptops, Tablets, and Smartphones.

Testing Strategy:

- Browser Testing: Chrome, Firefox, Edge, Safari.
- Mobile Testing: Android (Pixel, Samsung), iOS (iPhone).
- Responsive UI to ensure usability on different screen sizes.

4. App Architecture Scalability:

- Create a scalable architecture with microservices architecture and cloud-based services.
- Use database improvements to boost performance and scalability.

3.3 Operational Feasibility

Operational feasibility evaluates a project or system's usability and practicality in relation to its operational environment. Here's some information of operational feasibility:

1) Support from External Stakeholders and Development Team:

Stakeholders:

- Lecturers: Need an efficient way to review student proposals and deadlines.
- Students: Need timely feedback and an easy submission process.
- University Administration: Supports the system if it improves academic efficiency.

Collaboration Strategies:

- Involve faculty to members early to gather feature requirements.
- Conduct regular feedbacks with lecturers and students.
- Establish a help desk to inform issues to the administrators quickly.

2.) User Acceptance

User Experience Considerations:

- Implement an intuitive UI with clear navigation.
- Provide real-time notifications for assignment deadlines and lecturer responses.
- Include a dashboard for quick access to assignments, groups, and reviews.

User Feedback Integration:

- Conduct beta testing with a small group of students and lecturers.
- Use feedback forms and surveys to refine the system.
- Regularly update features based on real-world usage data.

3.) Training and Assistance

User Training Resources:

- Provide FAQs and step-by-step guides for teachers and students.
- Offer onboarding sessions for faculty members.

Technical Support:

- Offer multi-channel support (email, chat).
- Maintain an FAQ section to address common queries.

4.) Compatibility

Device Compatibility:

- Responsive web design to support desktops, laptops, tablets, and mobile devices.
- Optimized for low-bandwidth usage to accommodate users with slow internet.

Operating System Compatibility:

- Support for Windows, macOS, Linux (web app access).
- Mobile accessibility via Android & iOS browsers.

5.) Capability to Scale:

User Base Growth:

- Take into account variables like rising demand and market expansion when designing the platform to handle future user base growth.
- Communication needs: Make that the platform is scalable enough to accommodate new features, functionalities, and communication channels as well as adapt to users' changing communication demands.

3.4 Economic Feasibility

Sources of Income :

- University Funding – The university may allocate a budget for this platform as an essential academic tool.
- Sponsorships & Partnerships – Collaborate with tech companies, education service providers, or government initiatives supporting digital learning.
- University Research Grants – Seek funding under research or academic technology development programs.

3.5 Other

Procedures

1. Procedure for User Registration (Students & Teachers) :

Request user details (name, role [student/teacher], email, password, department, batch).

Validate input:

- Ensure email is unique.
- Verify password strength.
- If a teacher, check department assignment.

Store user details in the database.

confirmation email with registration credentials (optional)

2. Assignment Submission & Feedback Procedure

Procedure:

1. Students upload a proposal draft.
2. The system stores the submission with a timestamp.
3. Notify the assigned lecturer about the new submission.
4. The teacher reviews the assignment and provides feedback.
5. Students can revise and resubmit based on the feedback.
6. Track all feedback cycles until final submission.

3. Deadline & Notification System Procedure

Procedure:

1. Send reminder notifications to students if the deadline is near.
2. Notify teachers of upcoming review deadlines.

4. Teacher Feedback Management Procedure

Procedure:

1. Allow lecturer to access submitted proposal.
2. Provide an interface for adding comments and feedback.
3. Store feedback in the database and associate it with the submission.
4. Notify students when feedback is available.
5. Track feedback history for improvement tracking.

5.Admin Management Procedure

Procedure:

1. Allow admins to add or remove lecturers and students.
2. Assign students to groups if needed.

3.6 Use Case Diagrams

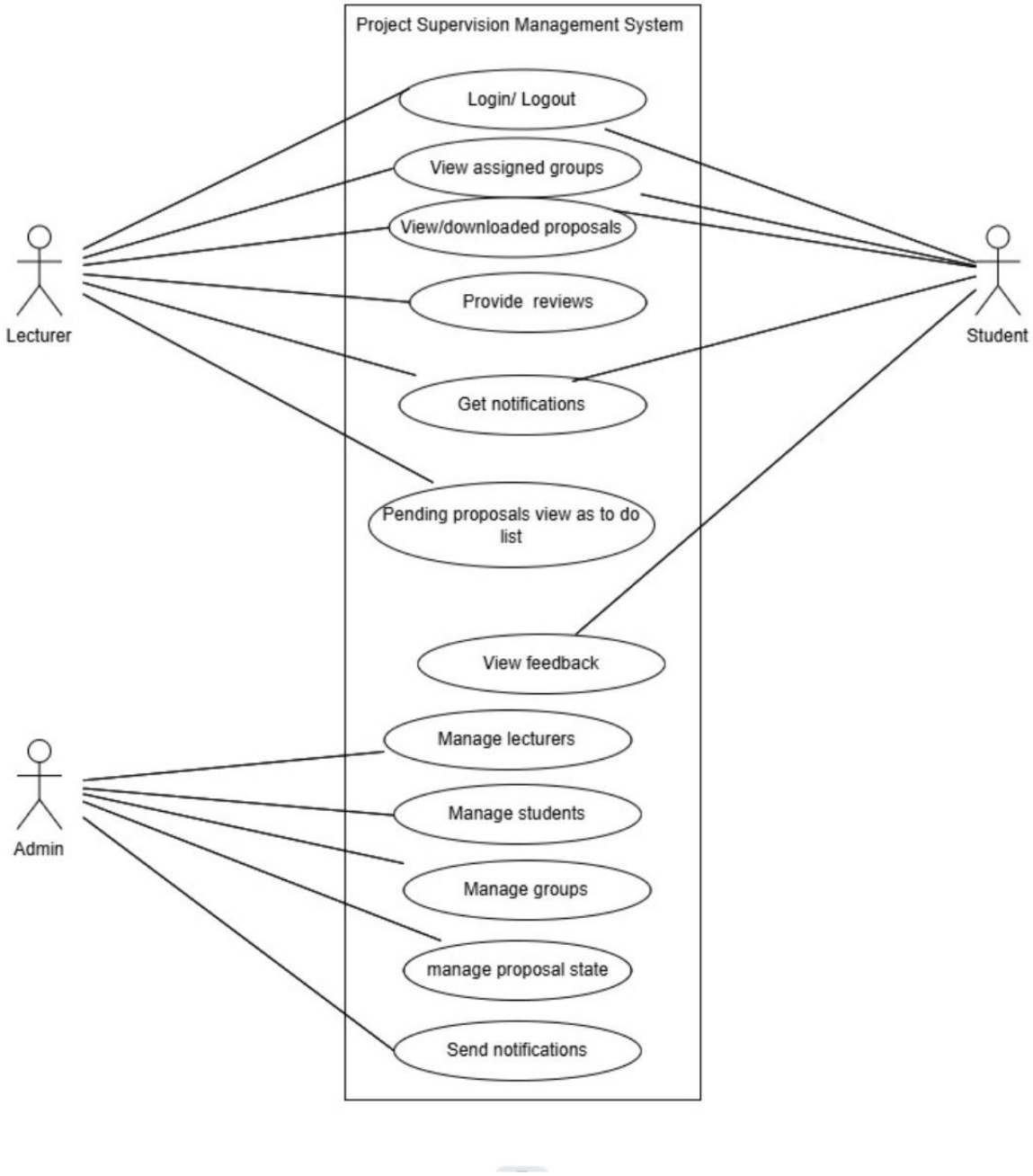

Figure 1 Use Case diagram for Project Supervision Management System

3.7 DFD Diagrams

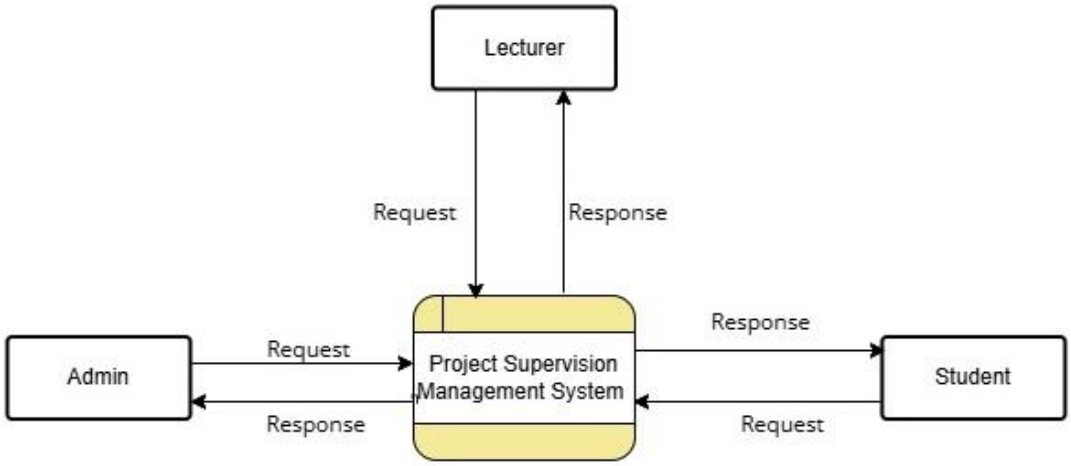

Figure 2 DFD Level 0 Diagram for Project Supervision Management System

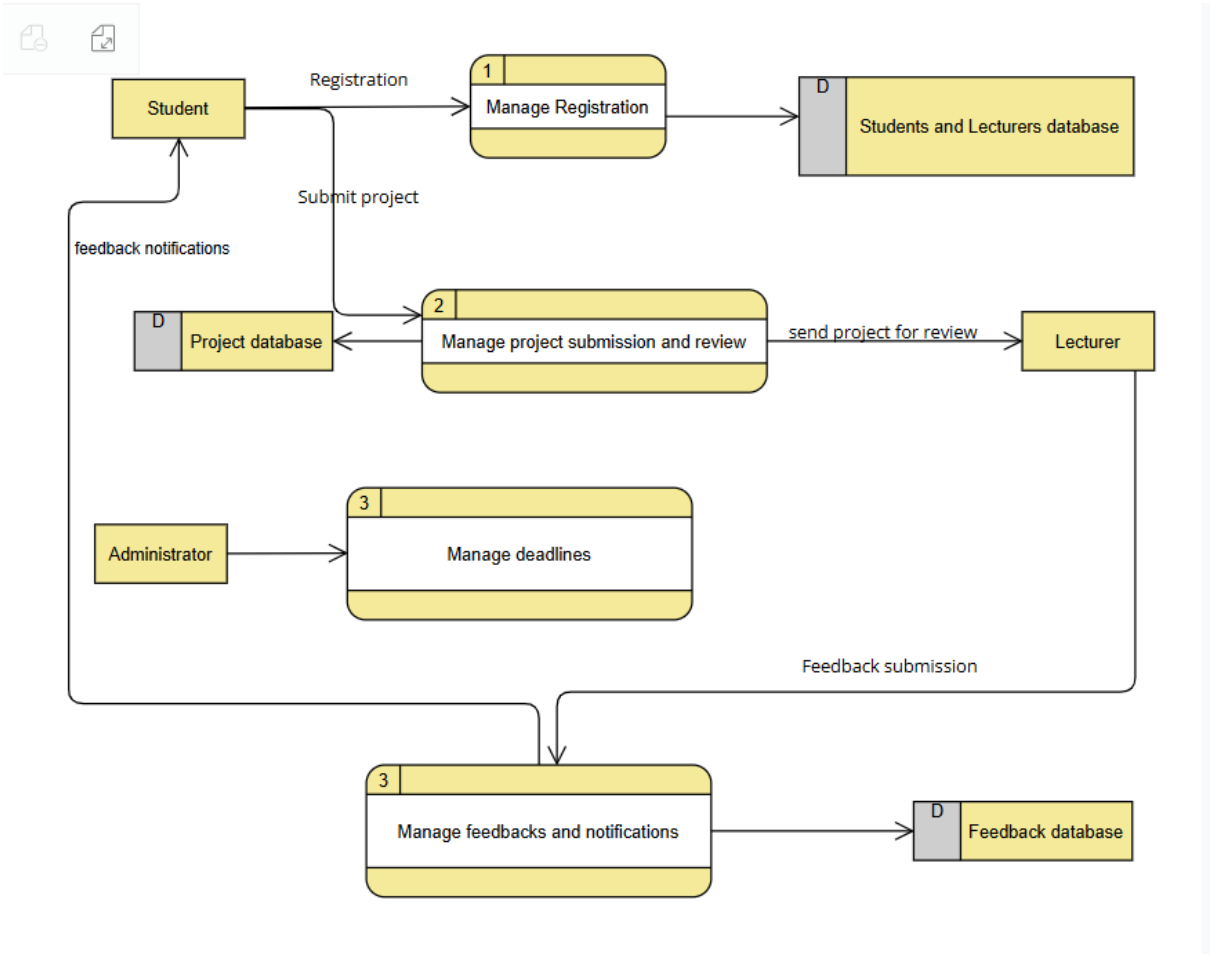

Figure 3 DFD Level 1 DFD Diagram for Project Supervision Management System

3.8 ER Diagrams

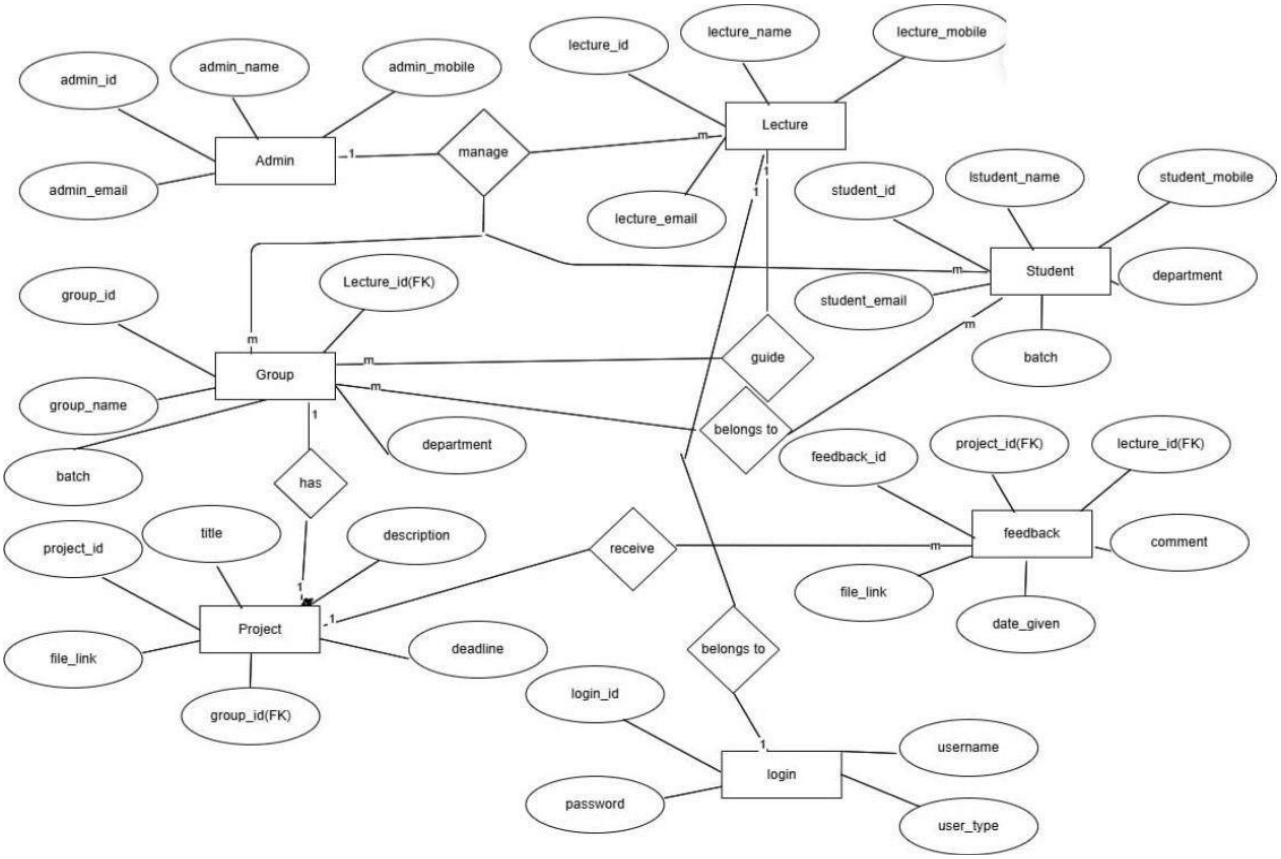

Figure 4 ER Diagram for Project Supervision Management System

4 Hardware and Software Requirement

4.1 Hardware Requirements

Server:

- Processor: Intel Xeon E5 or equivalent
- RAM: 8GB DDR4 ECC
- Storage: 256GB SSD RAID 1

Client Devices:

Desktop Computers:

- Processor: Intel Core i3 or equivalent

- RAM: 4GB DDR4
- Storage: 128GB SSD

Mobile Devices:

- Minimum Requirements: iOS 12 or Android 9

Networking Equipment:

- Router: Dual-band wireless router supporting 802.11ac
- Switch: Gigabit Ethernet switch with at least 8 ports
- Cabling: CAT6 Ethernet cables for wired connections

4.2 Software Requirements

Operating System:

- Server: Ubuntu Server 20.04 LTS
- Client Devices:
    - Desktop: Windows 10 or macOS Catalina
    - Mobile: Android 10 or iOS 14

Web Server:

- Apache HTTPS Server 2.4.x

Database Management System:

- MySQL 8.0.x

Programming Languages and Frameworks:

- Backend: Node.js 14.x with Express.js 4.x
- Frontend: React.js 17.x with Bootstrap 4.x

Development Tools:

- Visual Studio Code 1.5.x
- Git 2.x for version control

Security:

- SSL Certificate for HTTPS encryption

- Firewall software for network security

5 Tables and Structure

5.1 Number of Modules

The project supervision management system implements with several of modules but the below 5 modules describes within the system. But in future work we can add more modules.

5.2 Details of Modules

1. Authentication and User Registration:
    - Manages user registration for Admins, lectures, and Students.
    - Handles login/logout and secure user authentication.
    - Ensures role-based access control (RBAC).

2. User Profile Management:
    - Allows users to create and manage profiles.
    - Users can upload profile pictures.
    - Provides privacy settings and role-based configurations.

3. Module for Infrastructure Management:
    - Enables students to create/join groups.
    - Assigns lectures as supervisors to groups.
    - Allows students to submit assignments.
    - lectures can review, provide feedback, and track deadlines.

4. Notification & Deadline Management:
    - Sends automatic notifications for assignment deadlines, feedback, and meetings.
    - Allows teachers to send reminders to students.
    - Displays alerts for pending work.

5. Admin & Content Management:
    - Admin can add/remove lectures and students.

- Manages group assignments and access control.
- Facilitates announcement publication (e.g., deadlines, university notices)..

5.3 Data Structure

User

u_id	u_name	u_mobile	u_email	u_status
string	string	string	string	string

Group

g_id	g_name	g_leader	lec_id	g_type
int	string	int	int	string

Feedback

f_id	project_id	g_id	feedback
int	int	int	string

Project

project_id	g_id	u_id	status	feedback
int	int	string	string	string

6 Proposed System

6.1 Functional Requirements

User Management:

- Students and lecturers should be able to register for an account using a unique email and password.
- Admins should be able to manage user access and validate supervisor and student accounts.
- Students should be able to view project groups.
- Lecturers should be able to view and manage the group/individual projects and student groups under their supervision.
- Students should be able to edit their submissions.
- Assign group projects and individual projects for lecturers.
- Lecturers can see pending project proposals by to do list.

Project Submission & Management:

- Students should be able to submit project proposals, progress reports, and final reports.
- The system should support PDF (site will be notified that it allows PDF format only) file formats for submission.
- Lecturers should be able to review, comment on, and approve or reject project submissions.
- The system should allow supervisors to track project progress for each group.

Feedback & Communication:

- Lecturers should be able to provide feedback on submitted work directly within the system.
- Students should receive notifications when feedback is provided.
- A messaging system should allow students and lecturers to communicate within the platform.

Deadline & Notification Management:

- The system should track project submission deadlines for different groups.
- Students and lecturers should receive automated reminders for upcoming deadlines.
- Admins should be able to configure deadline policies and extensions.

Supervision & Task Tracking:

- Lecturers should be able to view student groups and track their completion.
- Students should be able to mark tasks as completed and request reviews.
- A progress dashboard should show the status of all ongoing projects.

System Administration:

- Admins should be able to add, remove, or update user roles (Student, Lecturer, Admin).
- Admins should be able to create and manage project categories and submission guidelines.
- Backup and restore functionality should be available for project data

Additional Features

- Mobile responsiveness to allow students and lecturers to access the system on different devices

6.2 Non – Functional Requirements

1.Security

- Encrypted Data Storage: Ensure that all student, lecturer, and project information is securely stored using encryption.
- Secure Authentication: Implement password hashing and role-based access control (RBAC) to restrict access based on user roles.
- Session Management: Enforce session management policies to prevent unauthorized access and session hijacking.
- Compliance with GDPR & Data Privacy: Follow General Data Protection Regulation (GDPR) and other relevant data privacy guidelines to ensure compliance.
- Regular Security Updates: Apply security patches and updates regularly to address vulnerabilities and enhance system security.

2.Performance

- Fast Response Times: The system should ensure page loads under 2 seconds under normal conditions.
- High Concurrent Users: It should support at least 100 simultaneous users without slowdowns.
- Optimized Database Queries: Efficient retrieval of project and user data.
- Scalable Architecture: Designed to handle an increasing number of students and lecturers.

3.Usability

- User-Friendly Interface: The system should have a simple, user-friendly interface for both students and lecturers.
- Calendar Integration: Students should be able to view proposal submission dates and deadlines in a calendar image.
- Intuitive Navigation: Provide intuitive navigation to allow users to easily access their projects, deadlines, and feedback.
- Mobile-Friendly Design: Ensure a responsive design that supports access from smartphones and tablets.
- User Assistance: Include help guides, tooltips, and FAQs to assist users in understanding system features.

4.Availability

- Uptime Requirement: The system should maintain 99.9% uptime, ensuring minimal downtime during academic hours.
- Automated Backups: Implement automated backups to protect data and allow recovery in case of system failures.
- Redundant Hosting: Use a redundant hosting infrastructure to ensure reliability and prevent service disruptions.

- Scheduled Maintenance: Communicate scheduled maintenance in advance to users to minimize inconvenience.

5.Reliability

- The system should function correctly and consistently, with minimal errors and crashes.
- It should support continuous project tracking without data loss.
- Regular system health checks and performance monitoring should be conducted to detect and resolve issues proactively.

6.Maintainability

- The system should be modular and easy to upgrade when new features are introduced.
- The codebase should be well-documented, making future modifications easier.
- Admins should have simple tools to manage users, projects, and system settings.

7.Scalability

- The system should be designed to handle increasing numbers of students, lecturers, and project submissions.
- It should support cloud-based deployment to allow dynamic resource allocation.
- Efficient database indexing and query optimization should be implemented to maintain performance as data grows.

8. Compatibility & Integration

- The system should be compatible with modern browsers (Chrome, Firefox, Edge, Safari).
- It should integrate with email services for notifications.

6.3 Methodology

For the development of this system, both the Waterfall and Agile methodologies were considered. However, Agile was chosen due to its flexibility, iterative approach, and ability to incorporate continuous user feedback.

1. Flexibility and User Input: Students and teachers may have changing needs, and Agile allows for modifications based on real-time feedback.
2. Faster Delivery: Agile enables releasing features in small sprints, allowing early user interaction and refinement.
3. Handling Unexpected Challenges: Any technical difficulties or requirement changes can be easily managed with Agile's adaptability.

Key Advantages of Agile for This System:

- User-Centric Design: Iterative development ensures that the system aligns with student and teacher needs.
- Quick Delivery & Adaptability: Short sprints ensure a working version is delivered fast, with improvements based on feedback.
- Effective Change Management: Allows adjustments in response to unexpected technical or functional changes.

7.Modules Split-up and Gantt Chart References

Activity	Time duration (weekly)														
	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Project planning															
Requirements analysis & Definition															
Learning technique															
Documentation															
Design															
Development															
Testing & correction															
Implementation															

Figure 5 Gantt chart

Cost Analysis

Development Cost:

Technology Stack:

- Database hosting and maintenance: $20-$50 per month
- Automation tools: $10-$50 per month (some services are free)

Prototyping:

- Costs for developing prototypes: $10-$100

Third-party Integrations:

- Expenses for analytics programs, email services, etc.: $10-$40 per month

Infrastructure Costs:

Hosting:

- Cloud or dedicated hosting services: $50-$100 per month

Domain Name:

- Cost of purchasing a domain: $10-$80 per month

Licensing and Legal Cost:

Software Licenses:

- Licensing fees for proprietary software: Student package

Total Cost Estimate:

- $100 + per month

8.References

https://ieeexplore.ieee.org/Xplore/home.jsp

https://www.pmi.org

https://www.geeksforgeeks.org

https://www.sab.ac.lk

9.Conclusion

The proposed Project Supervision Management System for Sabaragamuwa University represents a collaborative effort aimed at streamlining communication between students and supervisors while enhancing the efficiency of project management. This journey, from initial conceptualization to a structured project proposal, has been shaped by the invaluable guidance of our esteemed supervisor, Dr. U.A.P.Ishanka, whose expertise has been instrumental in refining our vision. We also extend our sincere

gratitude to Mrs.W.M.L.S.Abeythunga for her insightful contributions that have enriched our understanding of project supervision dynamics.

The system aims to address the challenges faced by students and faculty members in managing project-related communication, tracking deadlines, and providing timely feedback. By offering an intuitive and structured platform, the Project Supervision Management System will empower supervisors to efficiently manage student groups, monitor assignment progress, and provide guidance with ease.

Our analysis of technical, operational, and economic feasibility has provided a strong foundation for development, ensuring that the system can be implemented effectively within the available resources, technology, and budget. The hardware and software requirements have been carefully outlined to ensure seamless performance, security, and scalability.

The functional and non-functional requirements reflect our commitment to delivering a reliable, user-friendly, and responsive platform that caters to the evolving needs of both students and supervisors. By leveraging an Agile development methodology, we aim to incorporate continuous user feedback, deliver features incrementally, and proactively address challenges that may arise during implementation.

As we move into the development phase, we recognize the importance of effective project management and collaboration. The outlined module breakdown and project timeline (Gantt chart) will serve as a roadmap, ensuring that milestones are achieved efficiently and the system is delivered within the stipulated timeframe.

In conclusion, the Project Supervision Management System is a testament to our dedication to improving academic project management at Sabaragamuwa University. We look forward to successfully implementing this solution and making a significant impact on both student and faculty experiences. We express our gratitude to all who have supported us throughout this journey and are excited about the opportunities that lie ahead.

Sincerely,

Group 04