

# “Lab Student Management System”

## Request for Proposals

Drafted by

Lecturers of Integrated Exercise for Software II

on behalf of the customer

September 2024

## Contents

“Lab Student Management System” .....	1
Request for Proposals .....	1
1 System Description .....	3
1.1 Project Background .....	3
1.1.1 Title of planned system.....	3
1.1.2 Background.....	3
1.2 System Assumptions and Overview.....	3
1.2.1 Assumptions .....	3
1.2.2 Overview.....	3
1.3 Goals.....	6
1.4 Relation with existing workflow .....	6
1.5 Users of the System .....	6
1.5.1 Students .....	6
1.5.2 Instructor (Professor) .....	6
2 Content of Proposal .....	7
2.1 Extent of the Proposal .....	7
2.1.1 Scope of the System.....	7
2.2 System Structure .....	7
2.2.1 Application software .....	7
2.2.2 Hardware .....	8
2.2.3 Network architecture .....	8
2.2.4 Security system .....	8
2.3 Quality and performance assumptions.....	8
3 Requirements for Development .....	8
3.1 Process requirements .....	8
3.1.1 Weekly reports .....	8
3.1.2 Interim demonstrations .....	8
3.2 Development period.....	8

## 1 System Description

### 1.1 Project Background

#### 1.1.1 Title of planned system

“Lab Student Management System”

#### 1.1.2 Background

Faculty members at the University of Aizu instruct graduate and undergraduate students as academic advisors in their laboratories (labs). The students can spend several years in a lab to complete a research project to fulfill their studies. In a lab, an instructor must keep track of the assigned students to schedule and carry out their research activities for their timely completion. The instructor must consider the situation and needs of each student; managing a lab involves much work by the instructor and takes time and effort. Currently, each instructor leverages a mix of spreadsheets, communication tools, and project management tools to plan and track their students.

In this project, scheduling meetings with students and tracking their progress in a mutually understandable form will be realized as a system. The main objective is to reduce the extra workload and complexities caused by a lack of communication.

### 1.2 System Assumptions and Overview

#### 1.2.1 Assumptions

- The system is assumed to be used both on campus and off campus. Students and instructors may use the system from campus workstations, their PCs, or other devices.
- Instructors and students, including both graduate and undergraduate students, are expected to use this system. Some alumni may continue to use the system in the future.
- The system is assumed to manage one lab now. Its extension on hosting multiple labs will be considered in the future.

#### 1.2.2 Overview

- Assignment of students:
  - Undergraduate students are assigned to a lab when an instructor accepts the student and is approved by the University. The basic quota for student assignments is four students per year, but it may vary depending on other factors. In addition to such official assignments, instructors may accept a student to their lab anytime. In that sense, there is no limitation on the number of students that an instructor may accept.
  - When admitted to the graduate school, graduate students, including those enrolled in master's and doctoral programs, are assigned to a lab. The number of graduate students an instructor may accept is not limited.
  - The students in a lab will graduate from (leave) the lab when their research project is

completed and when they complete the corresponding study program. An instructor is responsible for the student until their graduation.

- In some cases, students may change their supervisor and lab during their studies.
- In some cases, students may withdraw from university without graduating.
- Undergraduate and graduate students can enter in April or October and graduate in March or September.
- In addition to students, the instructor can add sub-instructors to co-instruct students as necessary. A sub-instructor is another faculty member.
- Planning and managing projects:
  - The following information is necessary for each student: student ID, name, year/month of entry, planned graduation year/month, program (undergraduate, master's, doctoral), and university email address.
  - An instructor should be able to define projects with the following information: ID, title, description, and links to external services (ex. GitHub repository).
  - The instructor assigns students and sub-instructors, if any, to projects.
  - An instructor should be able to set project milestones with the following information: Date/time, title, status (planned, late, completed), and completion date.
  - An instructor can update the milestone status.
  - An instructor can update the student's information.
- Scheduling regular meetings
  - To instruct the students, the instructor needs to have regular meetings with them. The frequency of the meetings varies among instructors and may also depend on the project.
  - Depending on the instructor's convenience, instructors may choose to meet with students individually or in groups.
  - In principle, meetings are scheduled in units of slots corresponding to the class periods of the university curriculum. A meeting may consist of an arbitrary number of slots, including sub-divisions (ex., half-slots, 2/3 slots).
  - The start and end of regular meetings are usually linked with the academic quarters due to their dependence on course schedules but may vary depending on the instructor's convenience.
  - To schedule regular meetings, it is necessary to avoid the slots occupied by the classes taken by the students. Also, the availability of the instructor's and sub-instructors' slots must be considered. Furthermore, depending on the instructor's policy, each student's conveniences, such as club activities and part-time jobs, may be considered.
  - Considering all the above, the proposal for possible meeting slots should be made

- automatically. The instructor should be able to make the final decision.
- The student's course schedule information must be input by either the student or instructor, preferably both.
- The instructor must be able to input the instructor's available slots.
- The sub-instructor must be able to input their available slots.
- Scheduling of spontaneous meetings
  - In addition to regular meetings, instructors can schedule additional meetings with the students. The instructor can view possible slots by specifying the students they must meet with.
- Students and instructors can view the project status
  - Students can understand the project status by viewing the meeting schedule and milestones and be prepared.
  - Instructors can track the progress of the many students with ease.
- Please propose an intuitive and efficient user interaction scheme to support the input of student course schedules.
  - Please propose supportive mechanisms to avoid invalid input.
- Please propose an intuitive and effective visualization of the project schedule to improve mutual understanding and commitment.
- Please propose an effective notification method to remind meeting schedules and milestones.
- The students and instructor can export the meeting schedule for import to other calendar tools and services.
  - The meeting schedule and milestones can be exported in an iCalendar format (ICS file format).

The proposal must include at least the following functions:

- Student Registration
  - Instructors can add students and sub-instructors to the system and update their information.
- Project Planning
  - Instructors can add projects and milestones and update their status.
- Regular Meeting Scheduling
  - Instructors can assign student or group meetings by selecting the recommended candidate slots.
- Project Status
  - Students, sub-instructors, and the instructor can view the meeting schedules, tasks, and

milestone status for the whole project.

In addition to the above, please propose any functions that would improve user experience and ease system management.

- All related data should be stored in a database to ease system management.
  - Using an online database service that requires additional running costs in the future is unacceptable.
- Users are expected to use the system on or outside campus using their own devices.
- It is preferable that any potential user may use the system without installing additional software on their devices, but this is acceptable if the benefits outweigh the costs/effort.
- Libraries, frameworks, and the programming language used to develop the system are not limited. Please propose appropriate combinations to ease operation and maintenance costs.
- When including an online service as part of the solution, the development must be carried out using an account dedicated to the project. It is not acceptable to use a personal account. The account must be handed over to the customer at delivery time.

### 1.3 Goals

- Improve student awareness and commitment to research (lab) activities.
- Improve student-instructor collaboration in carrying out research projects.
- Reduce the workload of the instructor in planning and tracking student meetings.
- Reduce the workload of the instructor in instructing and monitoring students' research.

### 1.4 Relation with existing workflow

The official “Administration System” currently manages students' course schedules. Many calendar tools and services are also often used for everyday scheduling. However, the actual scheduling of meetings and tracking of student progress are performed with spreadsheets. The developed system is expected to replace the work done with spreadsheets.

### 1.5 Users of the System

#### 1.5.1 Students

- Students assigned to the lab: view projects, milestones, and meeting schedules.

#### 1.5.2 Instructor (Professor)

- Instructor of the lab: register students and add/edit projects, milestones, and meeting schedules.

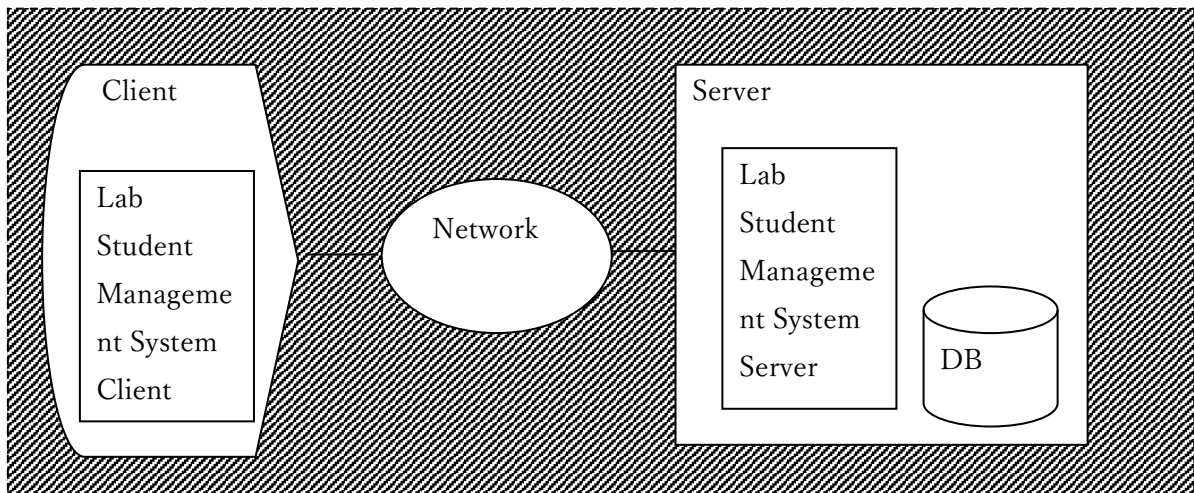
## 2 Content of Proposal

Please clarify any assumptions. Please explain the differences if your proposal does not satisfy the requirements or you have a better one.

### 2.1 Extent of the Proposal

The proposed system's scope is described in the previous section, "System Description." Please propose appropriate functions for the system using the following information as a guide. Note that this is only for general orientation and is not intended as any limitation.

#### 2.1.1 Scope of the System

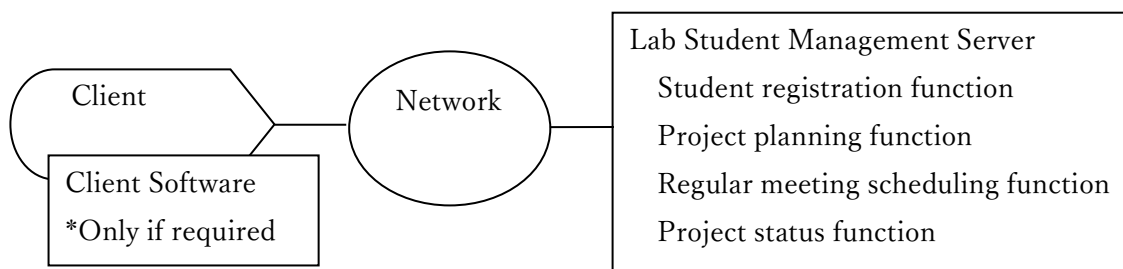


\*The inclusion of the client software is up to the developer's decision.

### 2.2 System Structure

Please devise an adequate system architecture and propose requirements for the following items. The architecture should support the viewing and managing of the corresponding data with reasonable reliability. Preferably, the system should be reasonably cost-effective and support future enhancements.

#### 2.2.1 Application software



### 2.2.2 Hardware

Please propose requirements for the server and client hardware as necessary.

### 2.2.3 Network architecture

Please propose a sufficient network environment for the system.

### 2.2.4 Security system

There are no specific requirements, so please propose a configuration if necessary.

### 2.3 Quality and performance assumptions

Please propose the quality and performance assumptions for the proposed system.

## 3 Requirements for Development

Conditions of system development are as follows.

### 3.1 Process requirements

#### 3.1.1 Weekly reports

Please summarize and report the progress of the development every week.

#### 3.1.2 Interim demonstrations

Please provide at least three product demonstrations to confirm the main design decisions and share project concerns. The dates will be specified separately.

### 3.2 Development period

- Beginning of development : Oct. 16, 2024
- Date of delivery : Jan. 29, 2025 (final class)