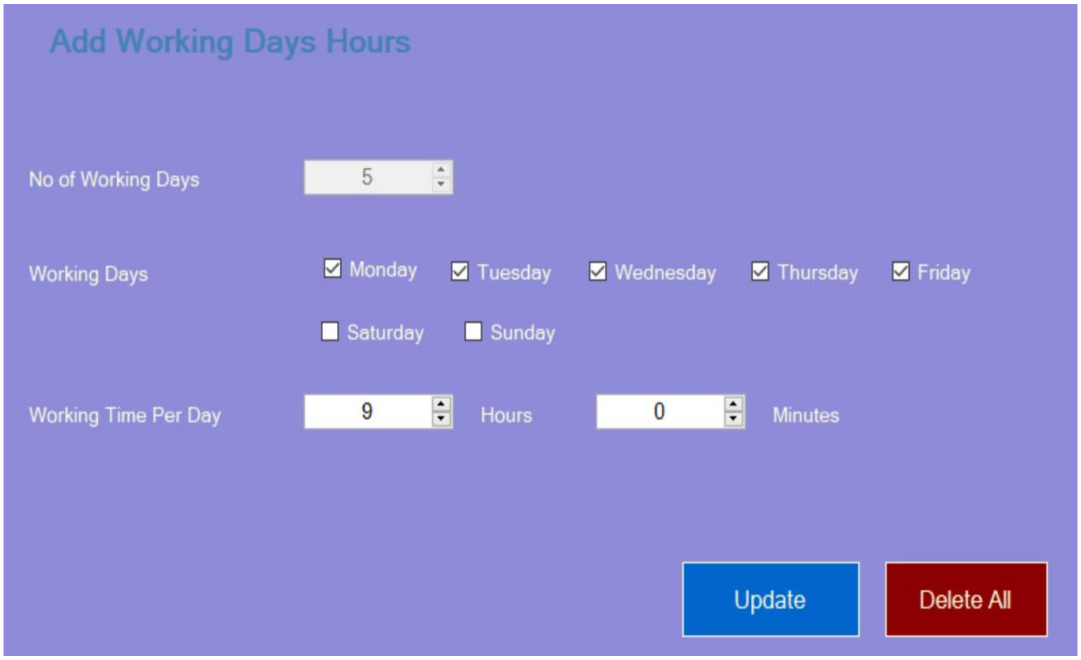
Project Specification

ABC is a leading non-state degree awarding institute. Assume your group is working in the IT division of the ABC institute. You have been asked to develop a **Resource Management application** for managing the resources of the ABC institute. The main functions and features of the system are as follows:

Section 1

This section includes details related to the working days and hours, lecturers, subjects, students, tags, and locations.

- The developed system should include an interface which facilitates the following entries related to the **working days and hours**:
  - Adding, editing, and removing the number of working days per week (Eg: 3)
  - Adding, editing, and removing the working days (Eg: Monday, Tuesday, and Wednesday)
  - Adding the time slots of the timetable. Should facilities the addition of one of the following time slots:
    - One hour time slots (Eg: 13.00 -14.00)
    - Thirty minutes time slots (Eg: 13.30 -14.00)



Add Working Days Hours

No of Working Days: 5

Working Days: ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☐ Saturday ☐ Sunday

Working Time Per Day: 9 Hours 0 Minutes

Fig. 1: Sample interface to display the working days and hours

- The developed system should include an interface which facilitates the following entries related to the **lecturers**:
  - Adding the following lecturer details:
    - Name
    - Employee ID. This should be 6 digit number (Eg: 000150).
    - Faculty (Eg: Computing, Engineering, Business, Humanities & Sciences, etc.)
    - Department
    - Campus/Center (Eg: Malabe, Metro, Matara, Kandy, Kurunagala, and Jaffna)
    - Building (Eg: New building, D-block etc.)
    - Level. The level should be assigned as follows:

Category	Level
Professor	1
Assistant Professor	2
Senior Lecturer(HG)	3
Senior Lecturer	4
Lecturer	5
Assistant Lecturer	6

- ○ Rank. The rank is a combination of the level and employee ID. It is defined as follows: level**.**employee ID (Eg: 2.000150). Accordingly, when deciding on a time slot, from the staff members who have requested for that slot, the staff member with the lowest rank should be allocated that slot.

- ▪ Editing lecturer details
- ▪ Removing lecturers
- ▪ Assigning active hours of lecturers(Some lecturers would not be available in particular days and hours)
- ▪ Viewing added details of lecturers

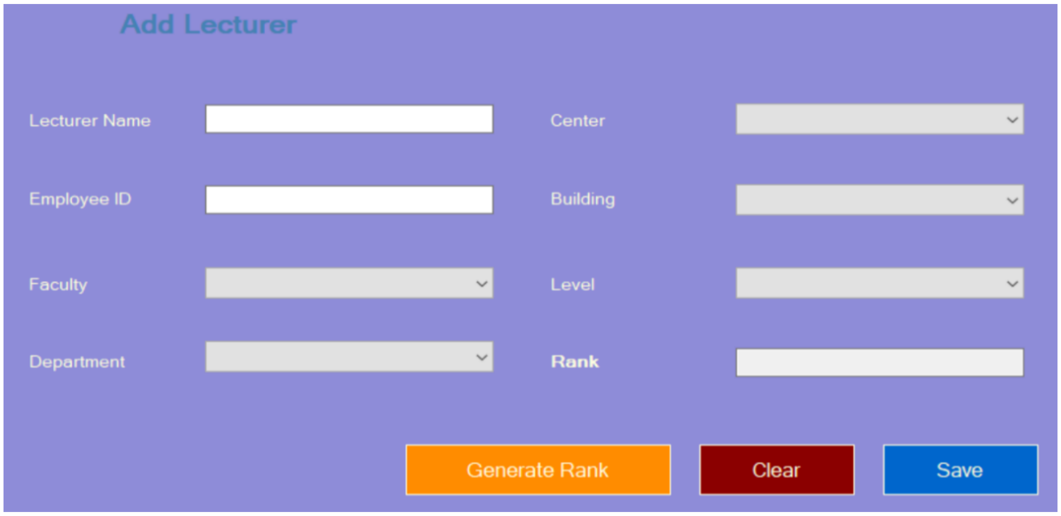
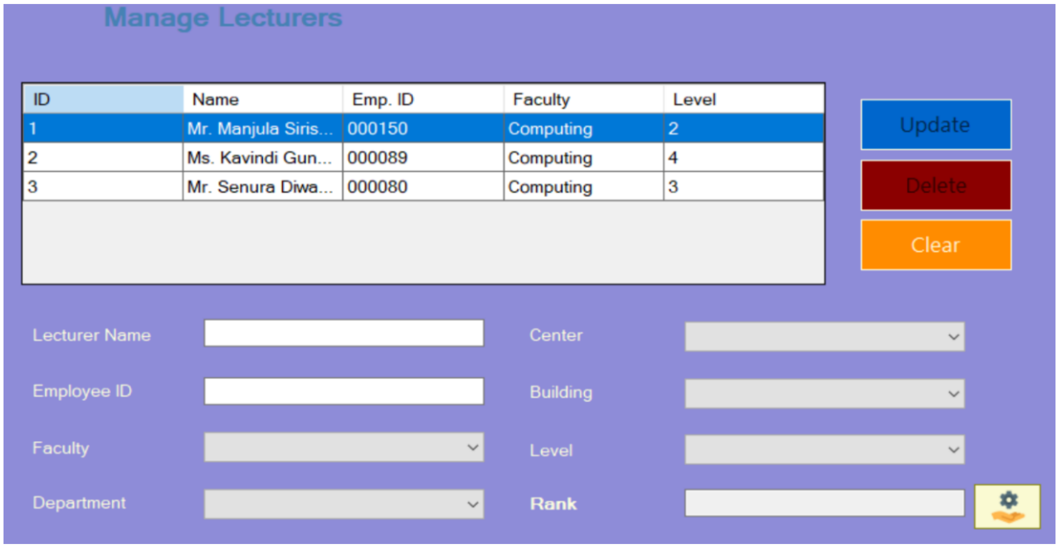


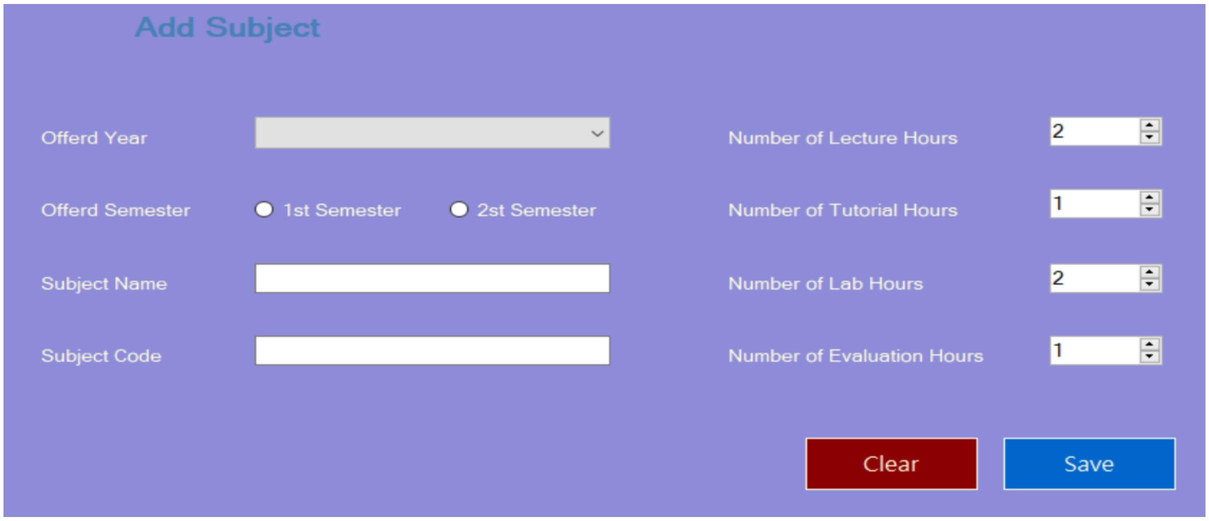
Fig. 2: Sample interface to display adding lectures



ID	Name	Emp. ID	Faculty	Level
1	Mr. Manjula Siris...	000150	Computing	2
2	Ms. Kavindi Gun...	000089	Computing	4
3	Mr. Senura Diwa...	000080	Computing	3

Fig. 3: Sample interface to managing lectures

- The developed system should include an interface which facilitates the following entries related to the **subjects**:
  - Adding the following details related to the subjects:
    - ○ Offered year
    - ○ Offered semester
    - ○ Subject name
    - ○ Subject code
    - ○ Number of lecture hours (Eg: 02)
    - ○ Number of tutorial hours (Eg: 01)
    - ○ Number of lab hours (Eg: 00)
    - ○ Number of evaluation hours (Eg: 02)
  - Editing subject details
  - Removing subjects
  - Viewing added details of subjects



Add Subject

Offerd Year

Offerd Semester ○ 1st Semester ○ 2st Semester

Subject Name

Subject Code

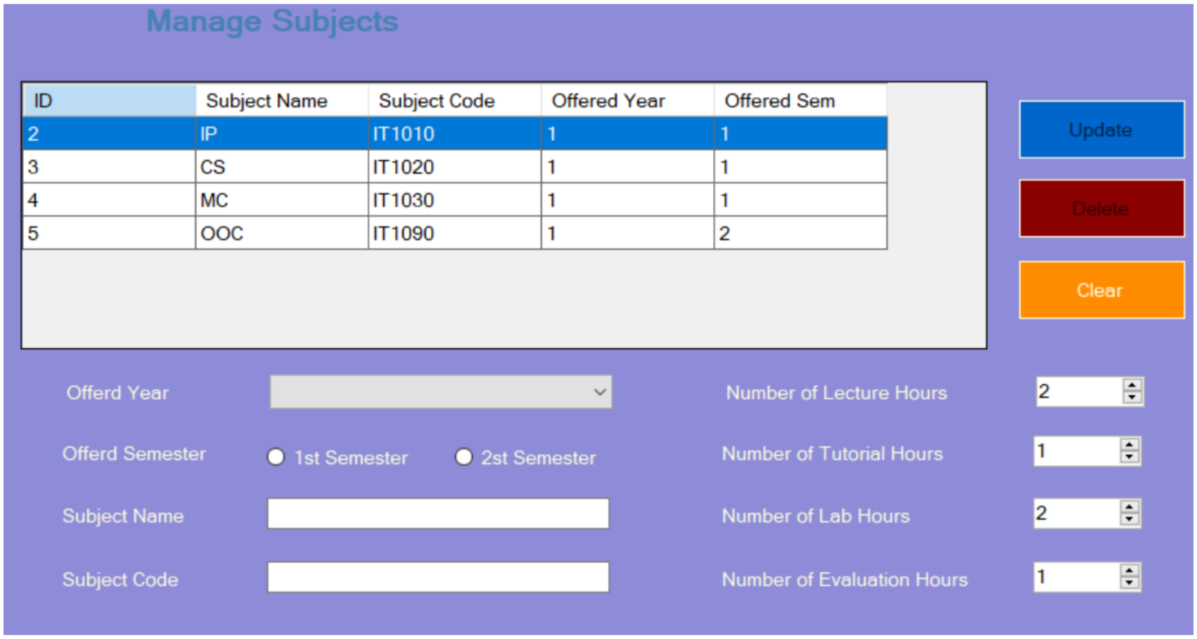
Number of Lecture Hours 2

Number of Tutorial Hours 1

Number of Lab Hours 2

Number of Evaluation Hours 1

Fig. 4: Sample interface for adding subjects



Manage Subjects

ID	Subject Name	Subject Code	Offered Year	Offered Sem
2	IP	IT1010	1	1
3	CS	IT1020	1	1
4	MC	IT1030	1	1
5	OOC	IT1090	1	2

Offerd Year

Offerd Semester ○ 1st Semester ○ 2st Semester

Subject Name

Subject Code

Number of Lecture Hours 2

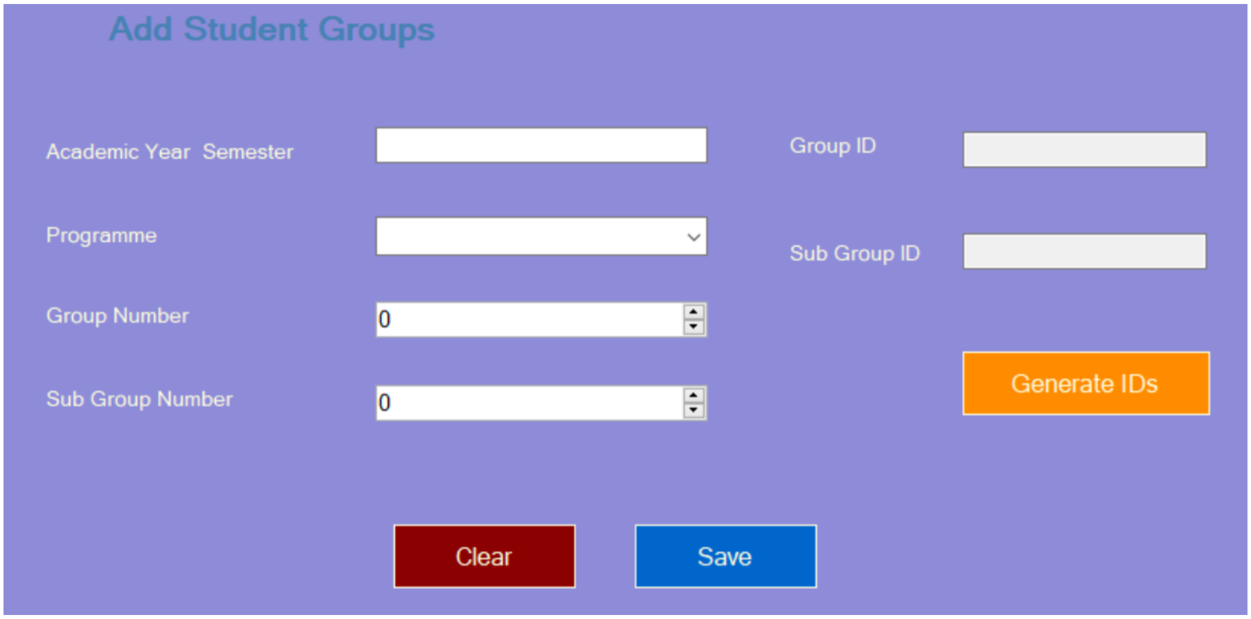
Number of Tutorial Hours 1

Number of Lab Hours 2

Number of Evaluation Hours 1

Fig. 5: Sample interface for managing subjects

- The developed system should include an interface which facilitates the following entries related to the **students**:
  - Adding the academic year and semester (Eg: Y1.S1, Y1.S2, Y2.S1, Y2.S2, Y3.S1, Y3.S2, Y4.S1, and Y4.S2)
  - Editing the academic year and semester
  - Removing the academic year and semester
  - Adding the programme (Eg :IT/CSSE/CSE/IM)
  - Editing the programme (Eg :IT/CSSE/CSE/IM)
  - Removing the programme (Eg :IT/CSSE/CSE/IM)
  - Adding group numbers (Eg: 01, 02, 03 etc.)
  - Editing group numbers
  - Removing group numbers
  - Generating group IDs. Group ID is defined as follows:
    - Year.semester.programme.group number (Eg: Y1.S1.IT.01)
  - Removing generated group IDs
  - Adding sub-group numbers (Eg: 1, 2, 3 etc.)
  - Editing sub-group numbers
  - Removing sub-group numbers
  - Generating sub-group IDs. Sub-group ID is defined as follows:
    - Year.semester.programme.group number.sub-group number (Eg: Y1.S1.IT.01.1)
  - Removing generated sub-group IDs
  - Viewing added details of students



Add Student Groups

Academic Year Semester

Programme

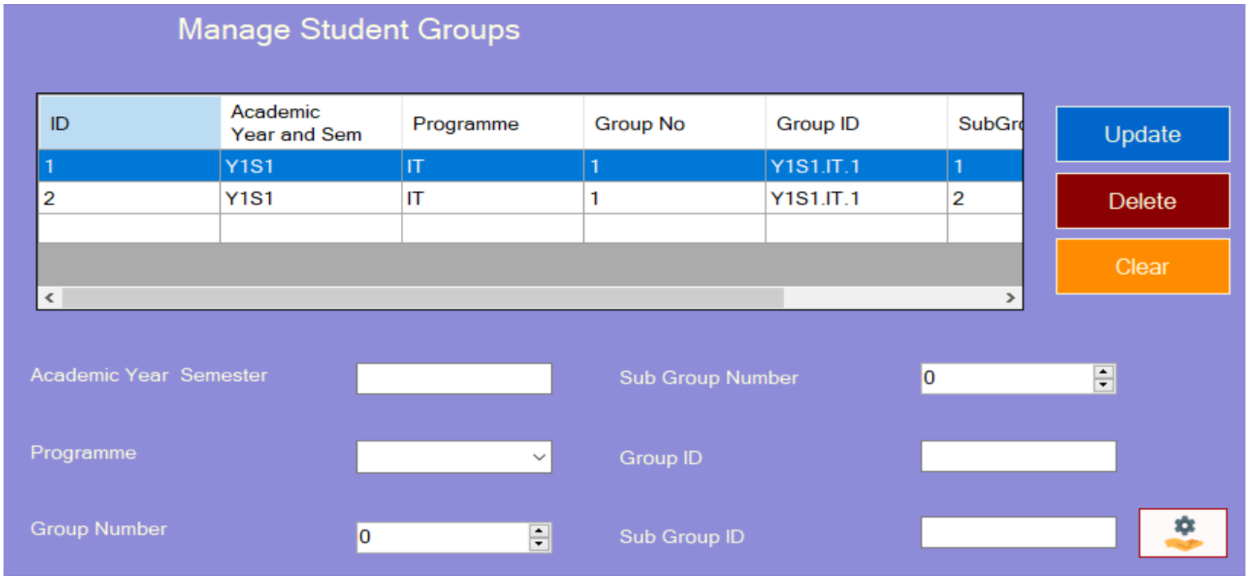
Group Number 0

Sub Group Number 0

Group ID

Sub Group ID

Fig. 5: Sample interface for adding student groups



ID	Academic Year and Sem	Programme	Group No	Group ID	SubGr
1	Y1S1	IT	1	Y1S1.IT.1	1
2	Y1S1	IT	1	Y1S1.IT.1	2

Academic Year Semester		Sub Group Number	0
Programme		Group ID	
Group Number	0	Sub Group ID	

Fig. 6: Sample interface for managing student groups

- The developed system should include an interface which facilitates the following entries related to the **tags**:
  - Adding tags (Eg: Lecture, tutorial, and practical)
  - Editing tags
  - Removing tags
  - Viewing added details of tags

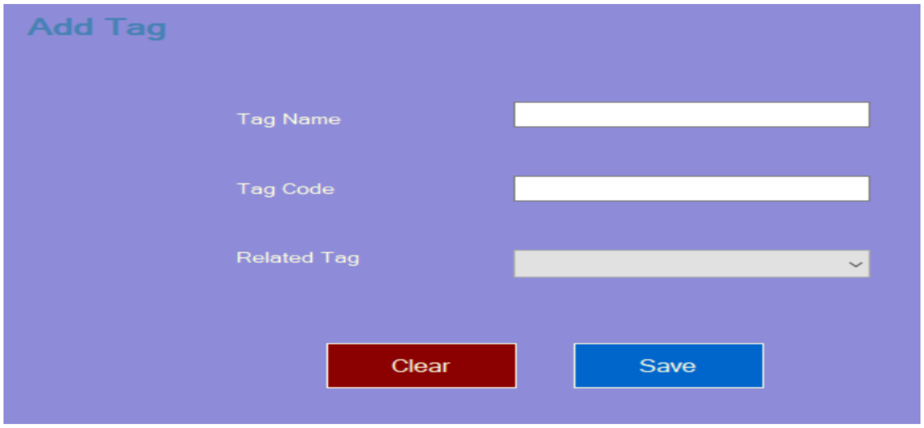
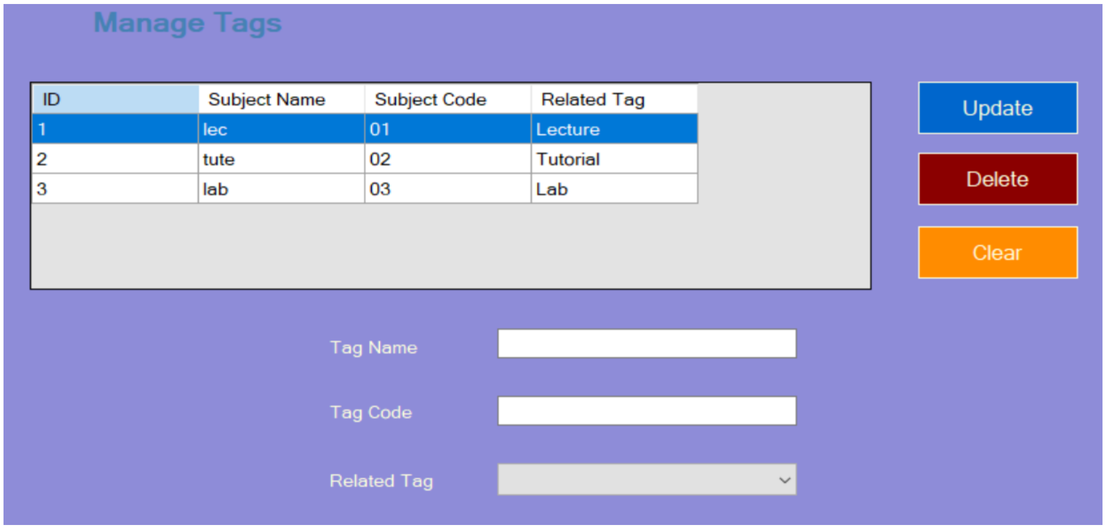


Fig. 7: Sample interface for adding tags



ID	Subject Name	Subject Code	Related Tag
1	lec	01	Lecture
2	tute	02	Tutorial
3	lab	03	Lab

Fig. 8: Sample interface for managing tags

- The developed system should include an interface which facilitates the following entries related to the **locations**:
  - Adding buildings (Eg: New building, D-block etc.)
  - Adding rooms (Eg: A501, B502, N3B-PcLab) and their capacities building-wise. A room can be a lecture hall or a laboratory.
  - Editing buildings
  - Editing rooms
  - Removing buildings
  - Removing rooms
  - Viewing added details of locations

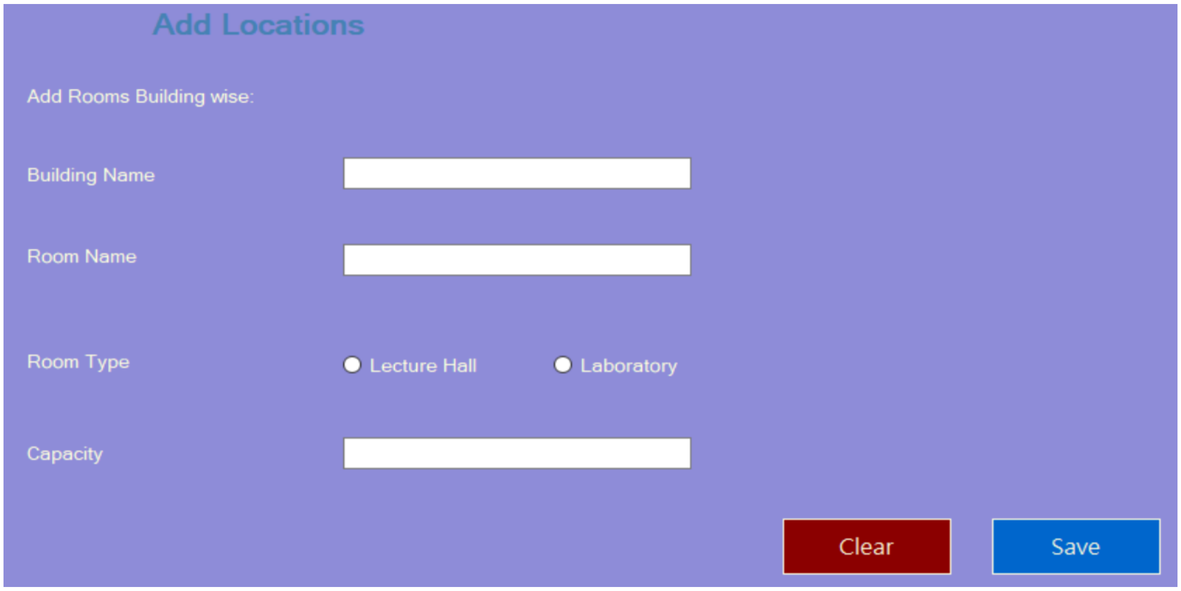
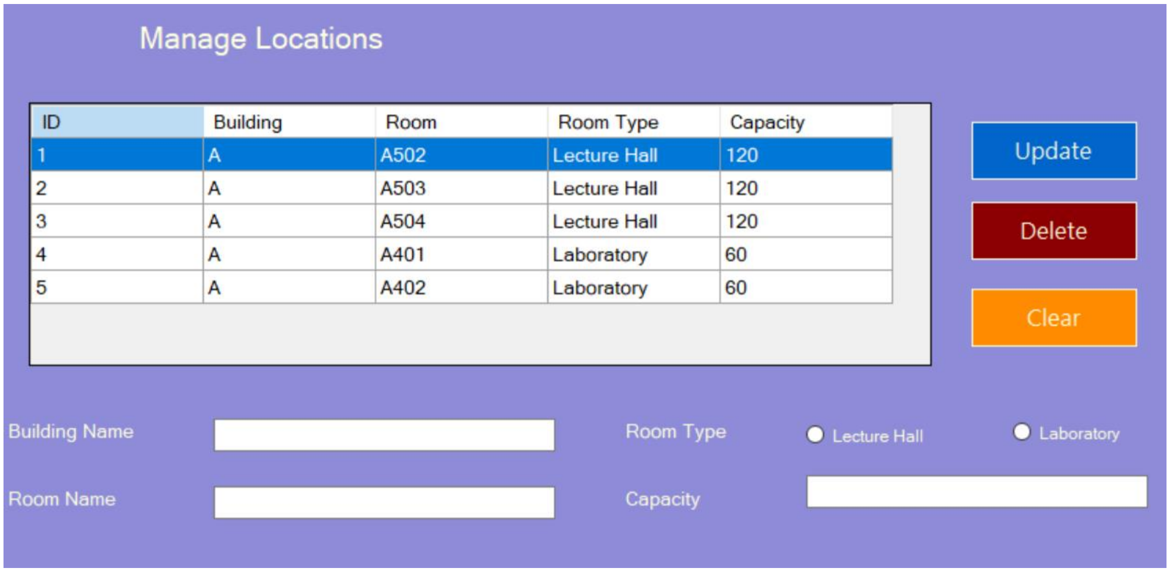


Fig. 9: Sample interface for adding locations

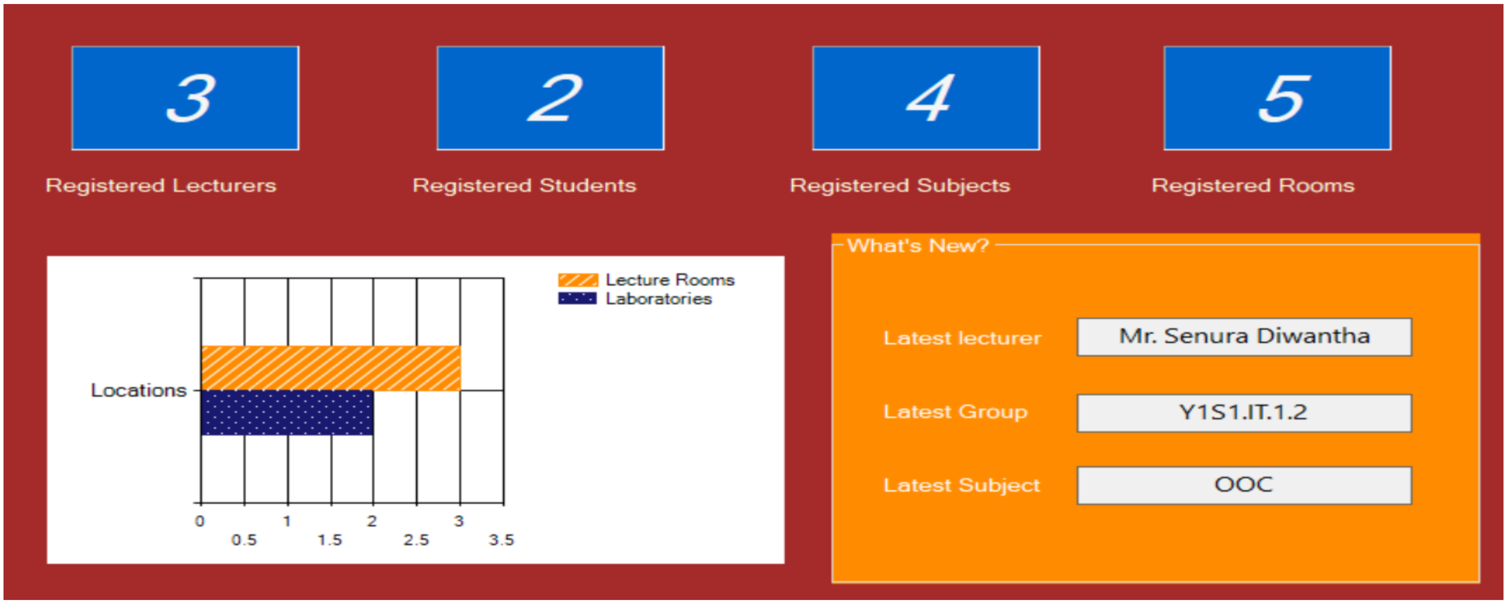


ID	Building	Room	Room Type	Capacity
1	A	A502	Lecture Hall	120
2	A	A503	Lecture Hall	120
3	A	A504	Lecture Hall	120
4	A	A401	Laboratory	60
5	A	A402	Laboratory	60

Fig. 10: Sample interface for managing locations

Section 2

- The developed system should include interfaces to visualize the following **statistics**
  - Statistics related to lecturers
  - Statistics related to students
  - Statistics related to subjects



3 Registered Lecturers

2 Registered Students

4 Registered Subjects

5 Registered Rooms

What's New?

Latest lecturer: Mr. Senura Diwantha

Latest Group: Y1S1.IT.1.2

Latest Subject: OOC

Fig. 11: Sample interface for visualizing statistic

- Add sessions. Steps related to adding of sessions are as follows:
  - Load lecturers and select the relevant lecturer for the session
  - Load tags and select the relevant tags for the session
  - Load students and select the relevant group or sub-group for the session
  - Load subjects and select the relevant subject for the session
  - Add the number of students for the session
  - Add the duration for the session
  - Finally add the session with the loaded specifications above
- List or visualize the sessions in detail.
- Add filters to search the sessions based on a particular lecturer, year, etc.

Note: A session should include the following:

- Lecturer
- Subject code
- Subject
- Tag (Eg: Lecture, Tutorial, Practical)
- Group ID (if the tag is a lecture or tutorial) or sub-group ID (if the tag is a practical)
- Student count
- Duration (Number of hours for the session)

- Accordingly, the format of a generated session should be as follows:
  - Dr. Nuwan Kodagoda – IT2030 – OOC – Lecture – Y1.S1.IT.01 – 120 - 2

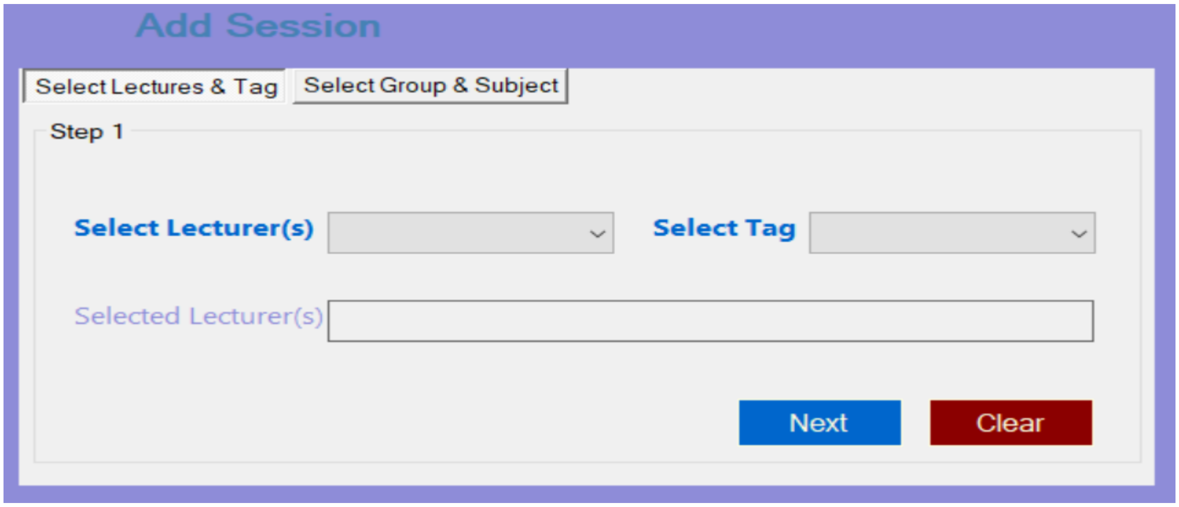


Fig. 12: Sample interface for adding sessions

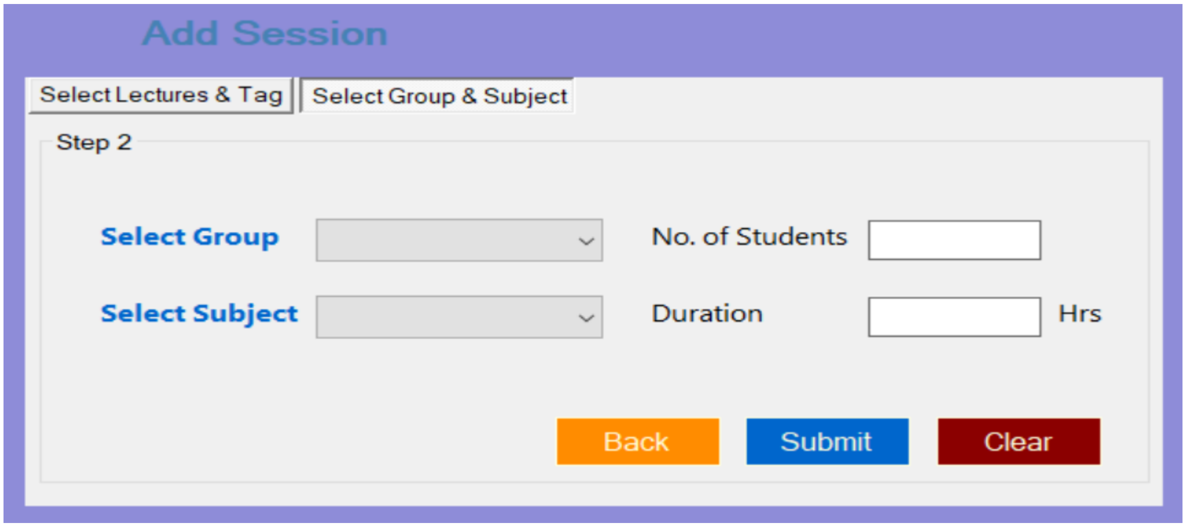
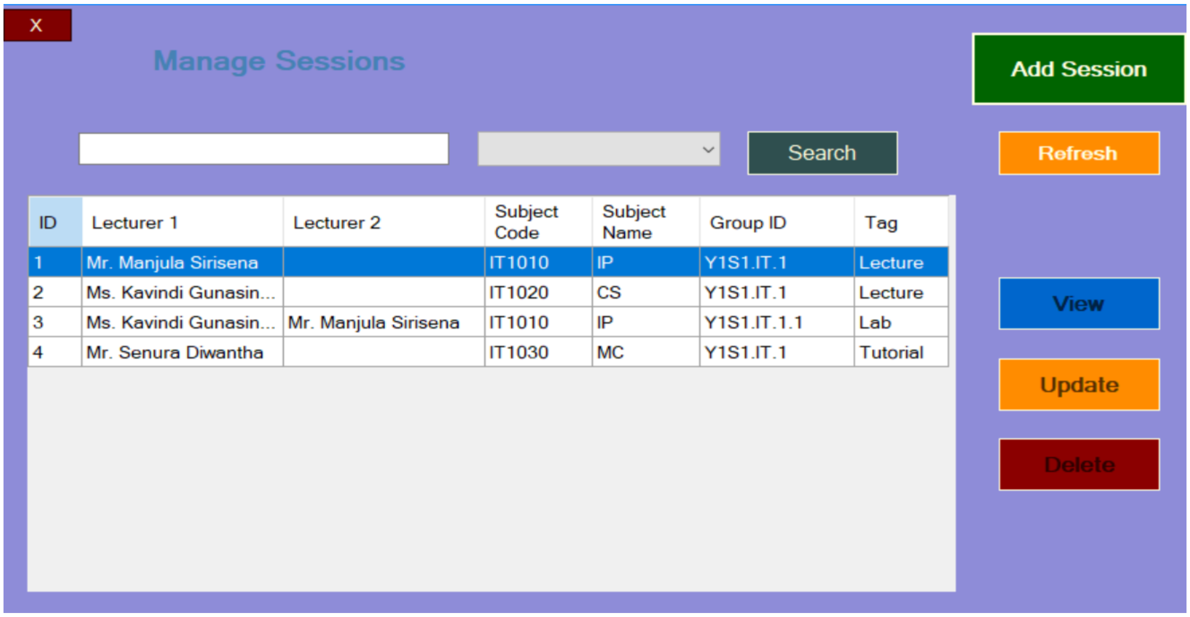


Fig. 13: Sample interface for adding sessions



ID	Lecturer 1	Lecturer 2	Subject Code	Subject Name	Group ID	Tag
1	Mr. Manjula Sirisena		IT1010	IP	Y1S1.IT.1	Lecture
2	Ms. Kavindi Gunasin...		IT1020	CS	Y1S1.IT.1	Lecture
3	Ms. Kavindi Gunasin...	Mr. Manjula Sirisena	IT1010	IP	Y1S1.IT.1.1	Lab
4	Mr. Senura Diwantha		IT1030	MC	Y1S1.IT.1	Tutorial

Fig. 14: Sample interface for managing sessions

The distribution of the functions for Sprints should be as follows(**This is an individual project**):

- **Sprint 1 –**
  - Should implement all the features of **Section 1** above
- **Sprint 2 –**
  - Should implement all the features of **Section 2** above

- System should be converted into an **exe** file(installer) which could be installed and run on any PC or laptop.
- **There would be a reduction of marks allocated for the Sprint 2 assessment if the developed application cannot be installed on the evaluator's PC or laptop. When submitting the .exe file for Sprint 2 submit a demo video which is not more than 15 minute duration.**