

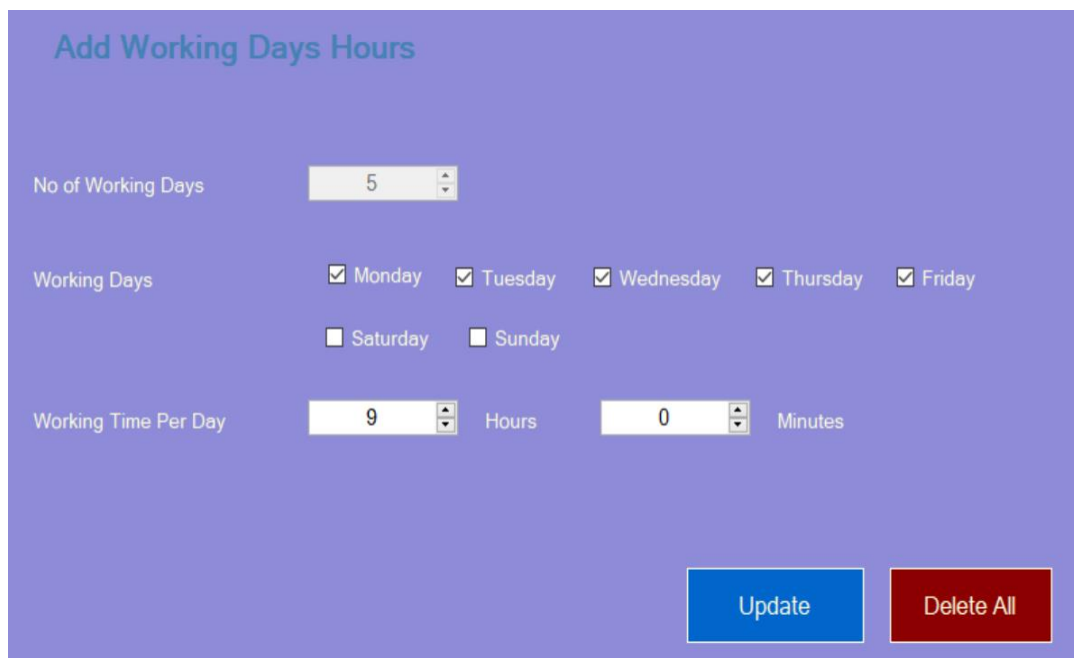
Project Specification

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 **desktop application** for managing the timetables 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)



The image shows a software interface titled "Add Working Days Hours" on a purple background. It contains three main input sections: "No of Working Days" with a spinner box set to 5; "Working Days" with checkboxes for Monday through Sunday, where Monday-Friday are checked; and "Working Time Per Day" with two spinner boxes for "Hours" (set to 9) and "Minutes" (set to 0). At the bottom right, there are two buttons: a blue "Update" button and a red "Delete All" button.

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

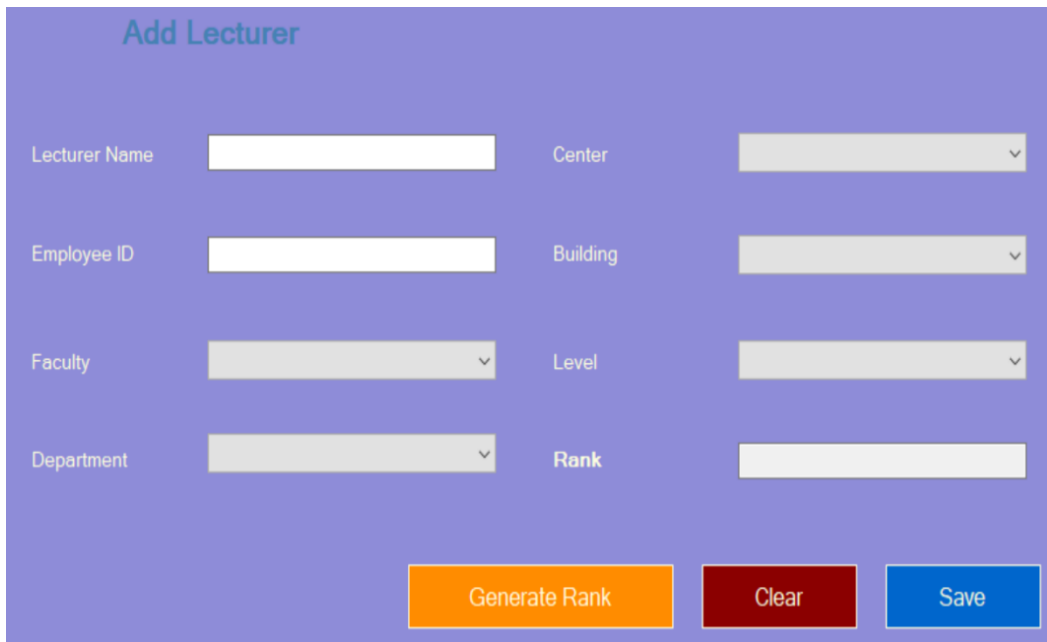
- The developed system should include an interface which facilitates the following entries related to the **lecturers**:
 - Adding the following lecturer details:
 - Name
 - Employee ID. This should be 6 digit number (Eg: 000150).
 - Faculty (Eg: Computing, Engineering, Business, Humanities & Sciences, 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



The image shows a web interface titled "Add Lecturer" with a purple background. It contains several input fields and dropdown menus for adding a new lecturer. The fields are arranged in two columns. The first column includes "Lecturer Name" (text input), "Employee ID" (text input), "Faculty" (dropdown), and "Department" (dropdown). The second column includes "Center" (dropdown), "Building" (dropdown), "Level" (dropdown), and "Rank" (text input). At the bottom right, there are three buttons: "Generate Rank" (orange), "Clear" (red), and "Save" (blue).

Fig. 2: Sample interface to display adding lectures

Manage Lecturers

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

Update

Delete

Clear

Lecturer Name

Employee ID

Faculty

Department

Center

Building

Level

Rank

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 ☐ 2nd Semester

Subject Name

Subject Code

Number of Lecture Hours

Number of Tutorial Hours

Number of Lab Hours

Number of Evaluation Hours

Clear

Save

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

Update

Delete

Clear

Offerd Year ▼

Number of Lecture Hours 2

Offerd Semester
☐ 1st Semester
☐ 2st Semester

Number of Tutorial Hours 1

Subject Name

Number of Lab Hours 2

Subject Code

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

Group ID

Programme

Sub Group ID

Group Number

Sub Group Number

Generate IDs

Clear

Save

Fig. 6: Sample interface for adding student groups

Manage 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

Update

Delete

Clear

Academic Year Semester

Sub Group Number

Programme

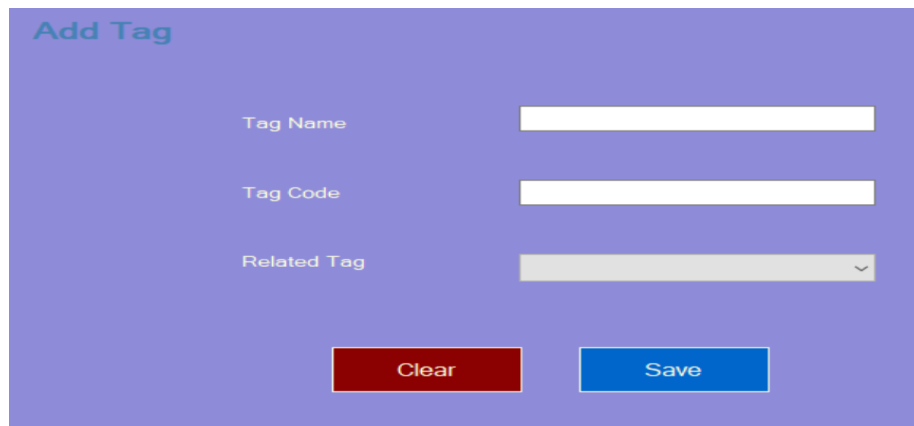
Group ID

Group Number

Sub Group ID

Fig. 7: 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



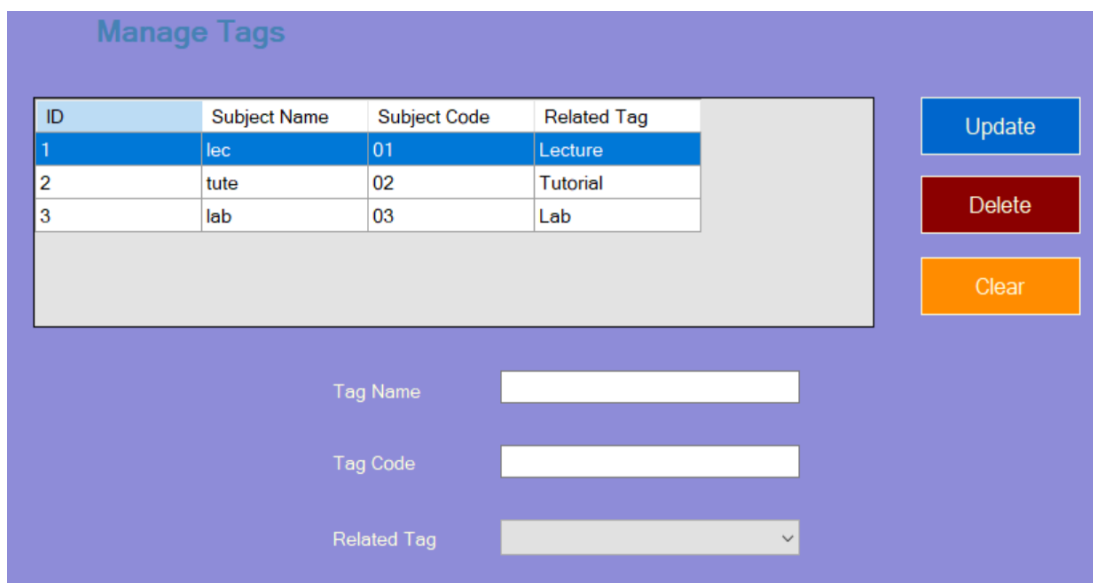
Add Tag

Tag Name

Tag Code

Related Tag

Fig. 8: Sample interface for adding tags



Manage Tags

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

Tag Name

Tag Code

Related Tag

Fig. 9: 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

Add Locations

Add Rooms Building wise:

Building Name

Room Name

Room Type ☐ Lecture Hall ☐ Laboratory

Capacity

Fig. 10: Sample interface for adding locations

Manage 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

Building Name Room Type ☐ Lecture Hall ☐ Laboratory

Room Name Capacity

Fig. 11: Sample interface for managing locations

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

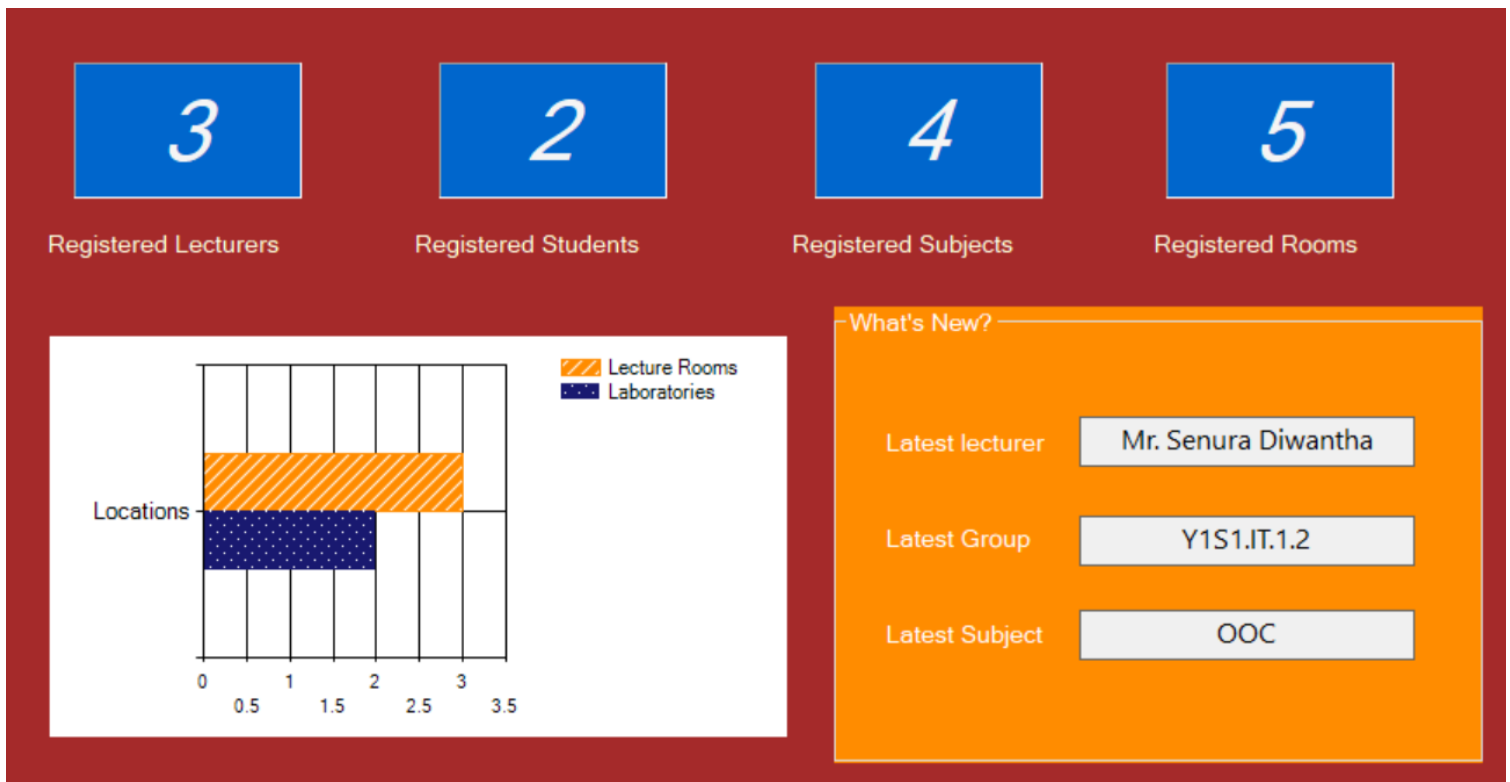


Fig. 12: Sample interface for visualizing statistic

Section 2

- 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

Add Session

Select Lectures & Tag

Select Group & Subject

Step 1

Select Lecturer(s) Select Tag

Selected Lecturer(s)

Next Clear

Fig. 13: Sample interface for adding sessions

Add Session

Select Lectures & Tag

Select Group & Subject

Step 2

Select Group No. of Students

Select Subject Duration Hrs

Back Submit Clear

Fig. 14: Sample interface for adding sessions

X

Manage 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

Add Session

Refresh

View

Update

Delete

Fig. 15: Sample interface for managing sessions

Section 3

- Add consecutive sessions (Eg: Lecture and Tutorial can be considered as consecutive sessions).
- Add set of sessions with the **same starting time, day, and duration** for scheduling parallel sessions. (For example, in SLIIT, the fourth year optional modules are grouped into several categories. When creating the 4th year timetable, two or more of the modules that belongs to the one category should be assigned as parallel events. This function is there to handle such cases).
- Add set of sessions that **should not overlap**(Eg: When considering the example in the previous point, two or more modules belonging to different categories should not be overlapped).

Sessions and Not Available Times Allocations

Consecutive Parallel Non Overlapping Not Available Times

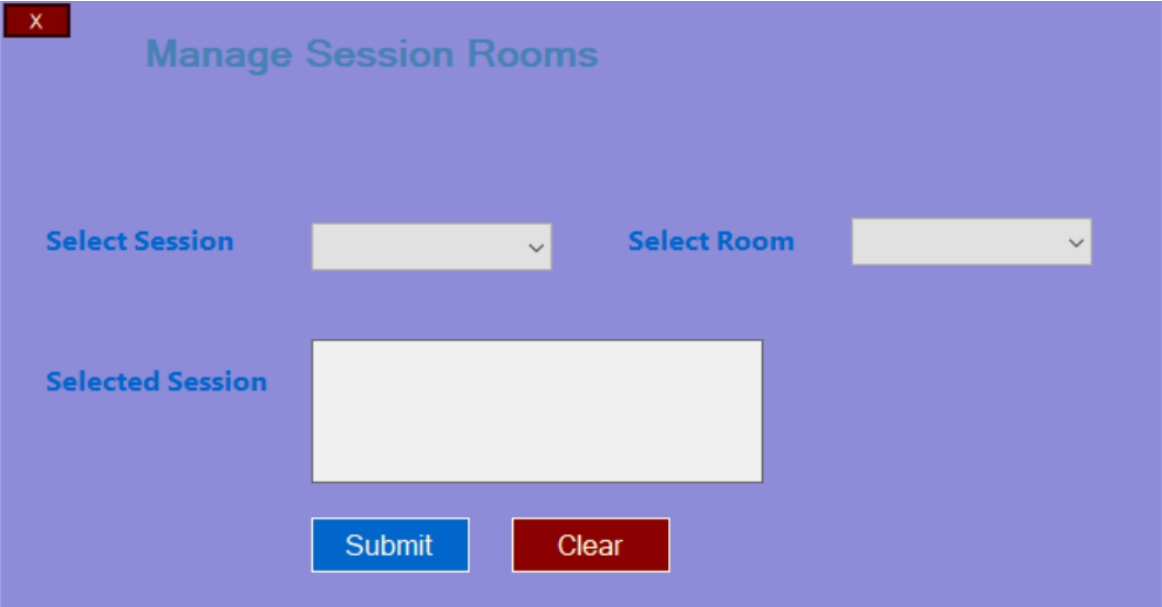
	ID	Lecturer 1	Lecturer 2	Subject Code	Subject	Group ID	Tag
<input checked="" type="checkbox"/>	1	Mr. Manjula Siris...		IT1010	IP	Y1S1.IT.1	Lectu
<input checked="" type="checkbox"/>	2	Ms. Kavindi Gun...		IT1020	CS	Y1S1.IT.1	Lectu
<input type="checkbox"/>	3	Ms. Kavindi Gun...	Mr. Manjula Siris...	IT1010	IP	Y1S1.IT.1.1	Lab
<input type="checkbox"/>	4	Mr. Senura Diwa...		IT1030	MC	Y1S1.IT.1	Tutor

Consecutive Sessions Added Successfully

Fig. 16: Sample interface for managing activities given in section 3

Section 4

- Add a suitable room or rooms for a subject
- Add a suitable room or rooms for a tag (If tag is a lecture or tutorial, then lecture halls would be allocated for all lectures and tutorials and if the tag is a practical, then laboratories would be allocated for all practical sessions)
- Add a suitable room or rooms for a lecturer
- Add a suitable room or rooms for a group or sub-group.
- Add a suitable room or rooms for a session
- Add consecutive session in the same room (Eg: Lecture and tutorial can be considered as consecutive sessions)
- Add the preferred room or rooms for a subject and relevant tag (Eg: The lecture (i.e., the tag) of the Game development module (i.e., the subject) should be allocated in the IM lab)



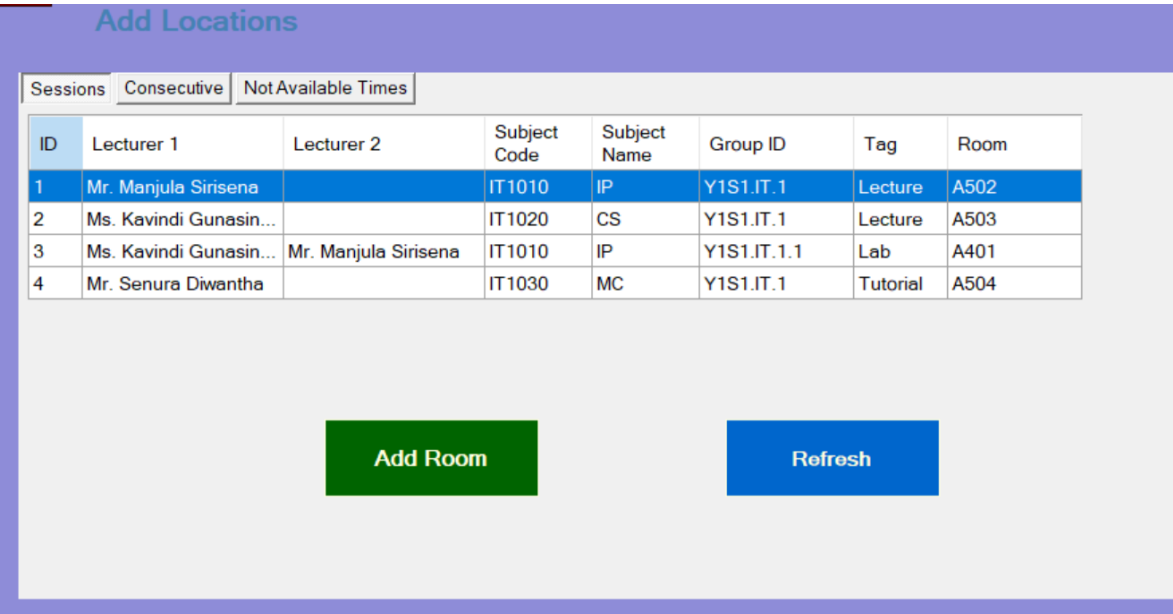
Manage Session Rooms

Select Session

Select Room

Selected Session

Fig. 17: Sample interface for adding location related activities given in section 4



Add Locations

Sessions Consecutive Not Available Times

ID	Lecturer 1	Lecturer 2	Subject Code	Subject Name	Group ID	Tag	Room
1	Mr. Manjula Sirisena		IT1010	IP	Y1S1.IT.1	Lecture	A502
2	Ms. Kavindi Gunasin...		IT1020	CS	Y1S1.IT.1	Lecture	A503
3	Ms. Kavindi Gunasin...	Mr. Manjula Sirisena	IT1010	IP	Y1S1.IT.1.1	Lab	A401
4	Mr. Senura Diwantha		IT1030	MC	Y1S1.IT.1	Tutorial	A504

Fig. 18: Sample interface for managing location related activities given in section 4

Section 5

- Allocate **not available** time of lecturers, sessions, groups and sub-groups.
- Allocate **not available** time of a student group/sub group
- Add the time that a room **cannot** be reserved
- Add a session with a preferred starting time and day

Sessions and Not Available Times Allocations

Consecutive Parallel Non Overlapping Not Available Times

Lecturers, Groups & Sub Groups

Select Lecturer

Select Group

Select Sub Group

Select Session ID

Time

Submit

View

Clear

Fig. 19: Sample interface for activities related to allocating not available time given in section 5

Manage Not Available Times

ID	Duration	Lecturer	Group ID	Sub Group
1	Saturday ,08.30...	Ms. Kavindi Gun...	Y1S1.IT.1	Y1S1.IT.1
<div style="display: flex; justify-content: space-between; align-items: center;"> < > </div>				

Refresh

Back

Delete

Fig. 20: Sample interface for managing activities related to not available times given in section 5

Add Locations

Sessions Consecutive Not Available Times

Select Room

Select Day

Start Time

End Time

Add Session

Clear

Fig. 21: Sample interface for managing activities related to location given in section 5

Section 6

- Generate timetables
- Show if there are any conflicts (Can be a pop out message)
- View the timetable of a particular lecturer (refer Fig 1)
- View timetable for a particular student group (refer Fig 2)
- View timetable for a particular room (refer Fig 3)
- Print timetable

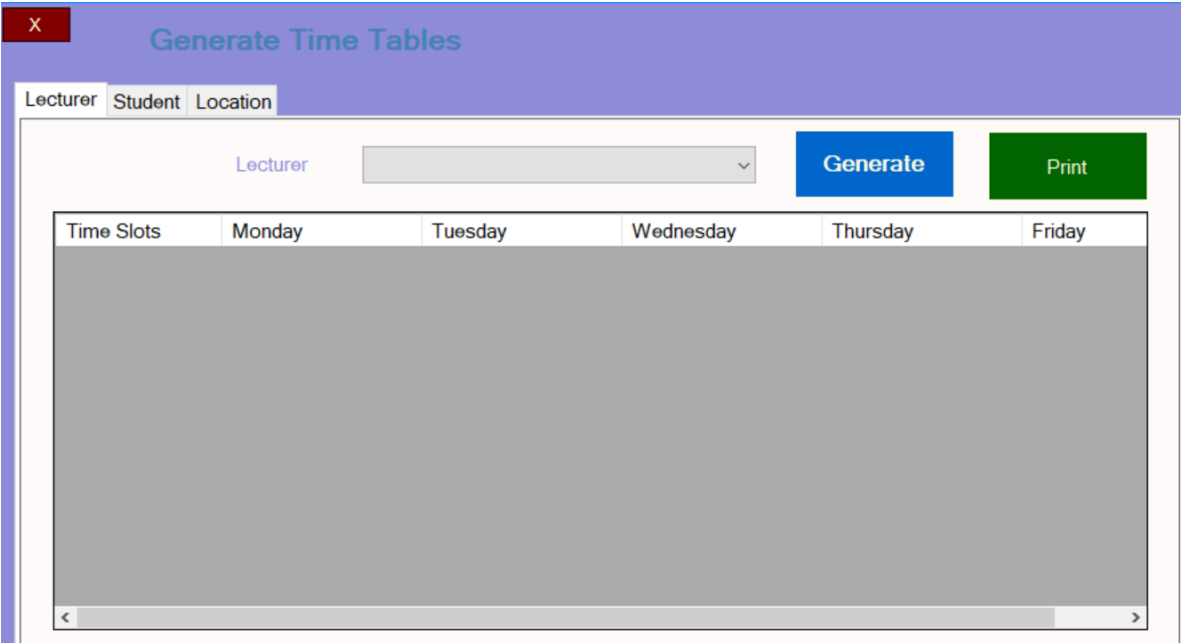


Fig. 22: Sample interface for activities given in section 6

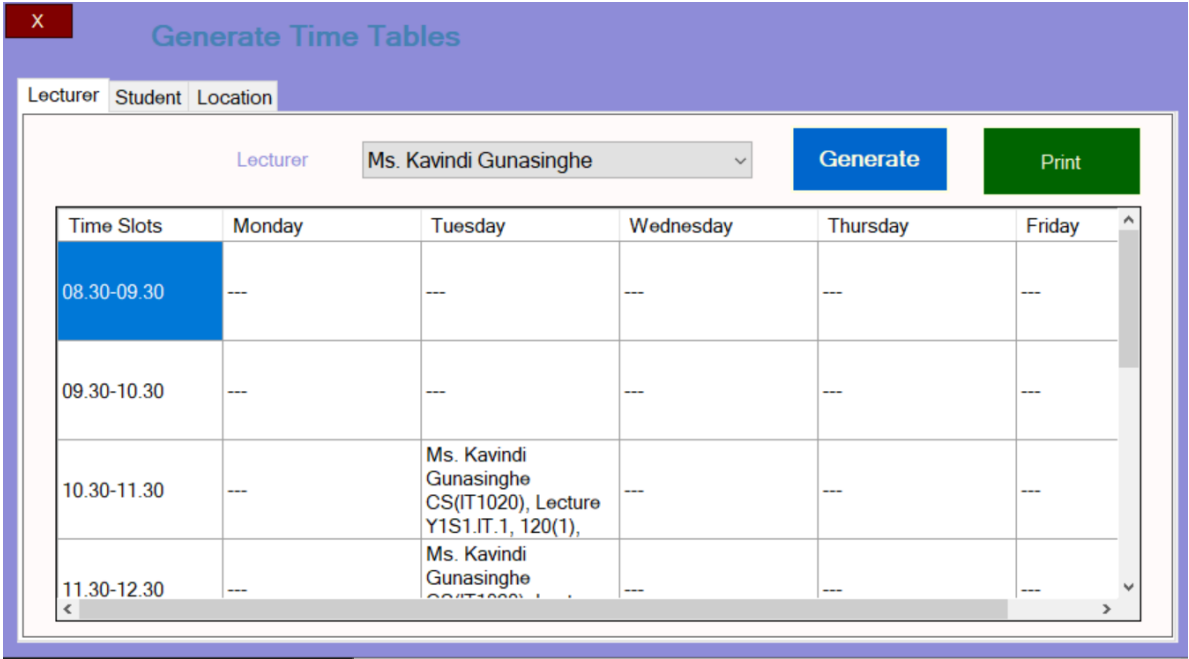


Fig. 23: Sample interface for generating timetable for a particular lecturer

X

Generate Time Tables

Lecturer
Student
Location

Group
Y1S1.IT.1
Generate
Print

Time Slots	Monday	Tuesday	Wednesday	Thursday	Friday
08.30-09.30	Ms. Kavindi Gunasinghe CS(IT1020), Lecture Y1S1.IT.1, 120(1),	---	---	---	---
09.30-10.30	Ms. Kavindi Gunasinghe CS(IT1020), Lecture Y1S1.IT.1, 120(1),	Mr. Senura Diwantha MC(IT1030), Tutorial Y1S1.IT.1, 120(1), A504	---	---	---
10.30-11.30	Mr. Manjula Sirisena IP(IT1010), Lecture Y1S1.IT.1, 120(1), A502	Ms. Kavindi Gunasinghe, Mr. Manjula Sirisena IP(IT1010), Lab	---	---	---
11.30-12.30	---	Ms. Kavindi Gunasinghe, Mr. Manjula Sirisena	---	---	---

Fig. 24: Sample interface for generating timetable for a particular student group

X

Generate Time Tables

Lecturer
Student
Location

Location
A502
Generate
Print

Time Slots	Monday	Tuesday	Wednesday	Thursday	Friday
08.30-09.30	Mr. Manjula Sirisena IP(IT1010), Lecture Y1S1.IT.1, 120(1), A502	---	---	---	---
09.30-10.30	Mr. Manjula Sirisena IP(IT1010), Lecture Y1S1.IT.1, 120(1), A502	---	---	---	---
10.30-11.30	---	---	---	---	---
11.30-12.30	---	---	---	---	---

Fig. 25: Sample interface for generating timetable for a particular lecture hall

The distribution of the functions among the group members should be as follows:

Note:

S1 = Functions required to complete for the Sprint 1 assessment

S2 = Functions required to complete in Sprint 2 assessment

➤ **Member 1 –**

- Should implement all the features of **Students** and **Tags** mentioned in **Section 1** above – **S1**.
- Should implement all the features in **Section 3** mentioned above - **S2**.

➤ **Member 2-**

- Should implement all the features of **Lecturers** and **Subjects** mentioned in **Section 1** above – **S1**.
- Should implement all the features in **Section 2** mentioned above – **S2**.

➤ **Member 3-**

- Should implement all the features of **Working Days and Hours** mentioned in **Section 1** above – **S1**.
- Should implement all the features in **Section 5** mentioned above – **S2**.

➤ **Member 4 –**

- Should implement all the features of **Location** and **Statistics** mentioned in **Section 1** above- **S1**.
- Should implement all the features in **Section 4** mentioned above - **S2**.

➤ **Group Implementation**

- Should implement all the features in **Section 6** mentioned above – **S2**.

➤ The final output should be **an integrated system** with all the mentioned functions.

➤ The 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.**