Math Department Resource Scheduling System

Rojen Shahi, Sean Keyse, Bikash Thapa

## Contents

1. Customer Statement of Requirements..............................................................................3
   1. Problem Statement................................................................................................3
   2. Glossary of Terms...................................................................................................6
2. System Requirements........................................................................................................
   1. Functional Requirements.......................................................................................
   2. Nonfunctional Requirements.................................................................................
   3. User Interface Requirements.................................................................................
3. Functional Requirements Specifications............................................................................
   1. Stakeholders...........................................................................................................
   2. Actors and Goals.....................................................................................................
   3. Use Cases................................................................................................................
   4. System Sequence Diagram......................................................................................
4. User Interface Specifications...............................................................................................
   1. Preliminary Design...................................................................................................
   2. User Effort Estimation..............................................................................................

# 1. Customer Statement of Requirements

## Problem Statement

Almost every type of organization in the world requires some type of scheduling system whether it's for scheduling resources, people, or in our case both. While a schedule itself is a useful tool, it provides little to no help if it's not well maintained and easy to use for everyone involved. Currently, the Math department has no decent way for scheduling rooms and the Math Assistance Center (MAC) doesn't have a good way of scheduling their tutors properly. The Math Department's rooms are fairly regularly reserved by both students- for tutoring or meeting with a group, and teachers - for meetings and conferences throughout the Math and Statistics Department. Where the MAC offers a wide variety of tutors that can cover all of their subjects offered at all times which requires a rather specific schedule making sure that all of the tutors are being used effectively and that each of their subjects offered has a tutor that is able to cover it. Our project will focus on taking these two scheduling needs for the two departments and creating an easy to use and intuitive scheduling system that will allow easy room reservation and tutor scheduling.

Currently, the math department's method for scheduling the labs and conference rooms that they have available is extremely outdated and quite a headache to deal with. After going to the department's office on the 5th floor, they have three binders: two of the binders handle the three labs they have available while the other handles all three of their conference rooms. The binders hold the schedule of each specific room in tabbed sections where they have a separate page with the schedule for each week of the semester. These weekly schedules have all of the classes in the labs printed out on the schedule at the times their held which leaves you room to reserve a room in the hour time blocks where classes aren't already scheduled. This means that once you find the right binder, you have to first flip to the correct tabbed section for your room, then flip through each page until you find the week that you need a reservation for before writing your name in the time slot, given that it's not holding a class or already been filled by someone else. This also means that if the time slot you need a reservation has already been taken, you have to go to either a different section or an entirely different binder, and then repeat the process again trying to find the correct week for your reservation. We do believe that it's necessary to keep information of past reservations made for looking at the statistics of the room reservations as well as seeing who reserved the room at any given time in the past. However, we think it's more important that you don't have to go through the entire schedule of the semester just to find the current week because it slows down the process and can just be confusing in general when looking for the right day. Room reservations are usually done by a member of the department, but we believe students should be able to reserve rooms as long they're able to get approval from someone in the department. Currently, the students have to speak with someone in the department to get approval for reserving a room before going through the process above of actually getting one reserved which means making a reservation is even more difficult and drawn out for them versus a department member.

An effective scheduling system should be quick and easy to use for everyone, so we believe that it's extremely important to show all of the rooms available for reservation in one spot where it's possible to easily see their current schedule. From this schedule you should be able to click on any of the rooms available and start up the process of reserving a room. We believe that members of the department should be able to simply reserve a room, automatically placing their reservation on the schedule for the time that they chose. Students on the other hand should have to start up a reservation request where they choose their time and enter the reason for their request which will block off that period of time until someone in the department either approves or denies the request. For this reason, we believe that department members should have accounts that distinguish them from students giving them the permission to make reservations instead of requests. For student requests, an email should be sent out to a select group of department members when the request is made so that they are able to either approve or deny the request, sending an email to the student with their decision regarding the request. While we want the main focus to be on the current schedule, we also believe that it's important to keep records of the past reservations. This allows us to obtain various statistics about room usage over a given time period, but it's more important to show everyone the current schedule so that should be hidden from someone who is just using the program to reserve a room. This means that either department members or the head of managing the schedule should have an option that opens up an administrative view allowing access to the past schedule as well as various statistics about the rooms.

The MAC is a tutoring center on campus that allows students to schedule a block of time with a tutor that is assigned to the requested subject. They have three different tables for Pre-Calculus, Calculus, and Statistics classes where they try to keep one or two tutors at each table, however when there's a lot of students requesting a specific subject, they may need to schedule more tutors. This means that the person in charge of making the schedule has to first make sure that they make the original schedule with right set of people that are able to cover all of the tables available. They also have to make sure they have tutors reserved for each of the subjects in case one is highly requested, and they need to have another tutor come in. Currently when a tutor is hired they provide a list of the subjects that they're willing to tutor in, the time slots that they have available to work, and any stipulations that they have for working – such as what times of the day they would prefer working as well as time restrictions like only being able to work 12 hours every week. After that, the scheduler has to make their own list of all of the tutors detailing the information provided that they will have to refer to when making their schedules. This list can be helpful, however it's difficult to go through all of that information, even in a list, and figure out what would be the best schedule to properly accommodate all of the needs of the subjects and tutors while keeping in mind the reserves that may be needed throughout the week.

For a scheduling system like this, we want to be able to make it easier for both the person in charge of creating the schedule as well as the tutors. In regard to new hires, we want to be able to both let the new tutor enter in their own availability, preferred subjects, and work stipulations as well as letting the scheduler enter in the information on their own for the new tutor. This means the scheduler will need different permissions that allow them to create a new account for a tutor as well as editing or deleting tutors that already exist in the system. The tutors themselves should also be able to change their own information in case their availability changes or one of their provided stipulations changes. When either a tutor or the scheduler logs in, we'd like to display the current week's schedule showing which tutors are working throughout the week. This means everyone has access to view the schedule to see when they are working, and tutors can see what subject they will be working on in case they have multiple available subjects. The scheduler should have the option to create a schedule for future weeks that haven't already been scheduled which will provide the scheduler with a suggested list of which tutors should be assigned to the shifts. The system is going to have to take a lot of different restraints into account when providing the suggested schedule. The most important would be the tutors' availability because we don't want the system to suggest a tutor that isn't available during the time frame suggested. Equally important, because the MAC is broken down into three different tables, we need to make sure that tutors are only being scheduled to tables with their preferred subjects because we don't want a tutor that specializes in Pre-Calculus scheduled to the Statistics table. The hours worked and time of day restraints must also be taken into consideration when creating the schedule. The restraint of maximum hours per week worked will be a little more difficult to take into account as it will have to update the schedule's suggestions throughout the week as they are confirmed for the schedule. For example, if a tutor gets scheduled for four hours on both Monday and Tuesday, the system will need to take into account that they can only be suggested for a two hour session throughout the rest of the week and update its suggestions as it gathers that information. A tutor's time of day preference, however is much easier to handle. If a tutor prefers a certain time of day, we simply put them lower on the suggestion list meaning that they are less likely to be selected during that time of day. This means that if a tutor has availability outside of their preferred time range, they will only be scheduled during a busy time if they're needed. One last piece of information that we'll have to take into account when suggesting tutors to schedule is the last time that they were scheduled. We'd like to keep the scheduling evenly distributed among the tutors so that one of them isn't getting either over or under scheduled. Using all of these different stipulations we want to be able to create scheduling system that will update automatically as tutors are being put onto the schedule so that the MAC is able to schedule all of their tutors as quickly and evenly distributed as possible.

Obviously, our project requires two different scheduling systems for different areas of the math department, but our intention is to incorporate both of these systems into one program. We hope that combining both of the scheduling systems into a single program will help people that work with both the MAC and the math department to view everything that they need. This includes their tutoring schedule for the week as well as a room reservation schedule in case they may need to reserve a lab. Because of the current systems in place, our main goal is to optimize each scheduling process to make sure that everything is as easy and as quick as possible. We want to make sure that each new user is able to easily figure out what they need to do and how to do it.

Glossary of Terms

# 2. System Requirements

## Functional Requirements

* + - 1. **Resource Reservation System**

|  |  |  |
| --- | --- | --- |
| ID | Priority weight | Requirement |
| REQ 1 | 5 | System should show the status of each room |
| REQ 2 | 5 | System should show the reservation and reservation time of each room |
| REQ 3 | 5 | System should show the available period of each room |
| REQ 4 | 5 | System should allow user to reserve the desired room in desired time |
| REQ 5 | 3 | System should allow user to make changes to their reservation |
| REQ 6 | 2 | System should update daily |

Table 1: Functional requirement for Resource Reservation System

* + - 1. **Scheduling System**

|  |  |  |
| --- | --- | --- |
| ID | Priority weight | Requirement |
| REQ 7 | 5 | System should show the status of subject and instructor in given time |
| REQ 8 | 4 | System should show the schedule of the course and instructors |
| REQ 9 | 4 | System must show the available courses |
| REQ 10 | 4 | System must comply with the student schedule |
| REQ 11 | 4 | System must allow user to reserve seat to a instructor and desired course at a given time. |
| REQ 12 | 3 | System must show the statistics of tutoring |

Table 2: Functional requirement for Scheduling System

## Nonfunctional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Priority weight | Requirement |
| REQ 1 | 5 | System should update in real time |
| REQ 2 | 4 | System should have extreme security. only the administrator must be able to have access to the database. |
| REQ 3 | 4 | System must be instant to users operation |
| REQ 4 | 4 | System must be easy when they are broken. |

Table 3: Non-Functional requirement for Resource Management and Scheduling System

## User Interface Requirements

The section contains a hand drawn visual representation of the user’s interface. The images are the visual representation of the interface we w ill make in the future.

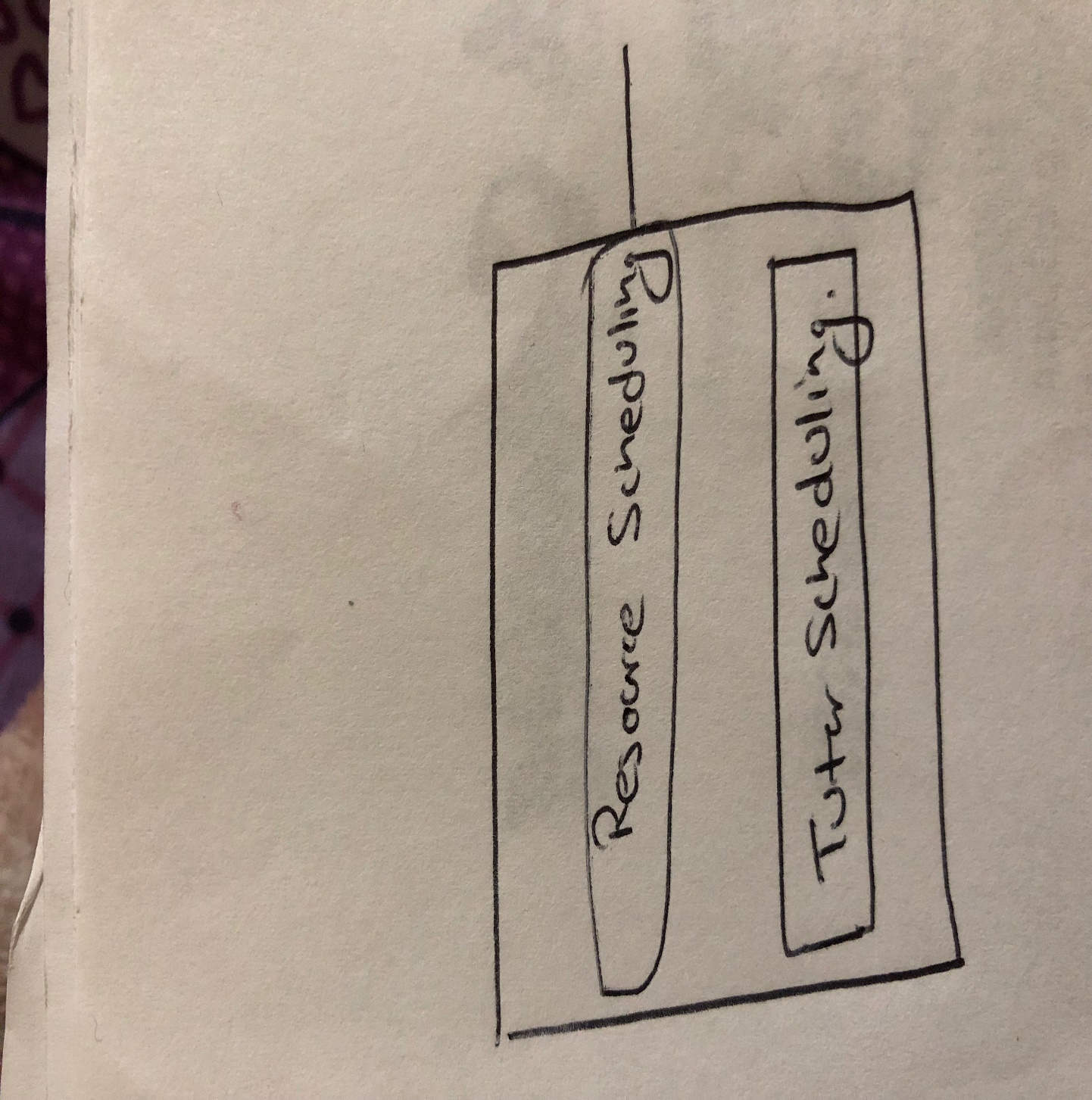


Fig 1 . This is the main interface of the system. This is from where we either go to reasource scheduling system or tutor scheduling

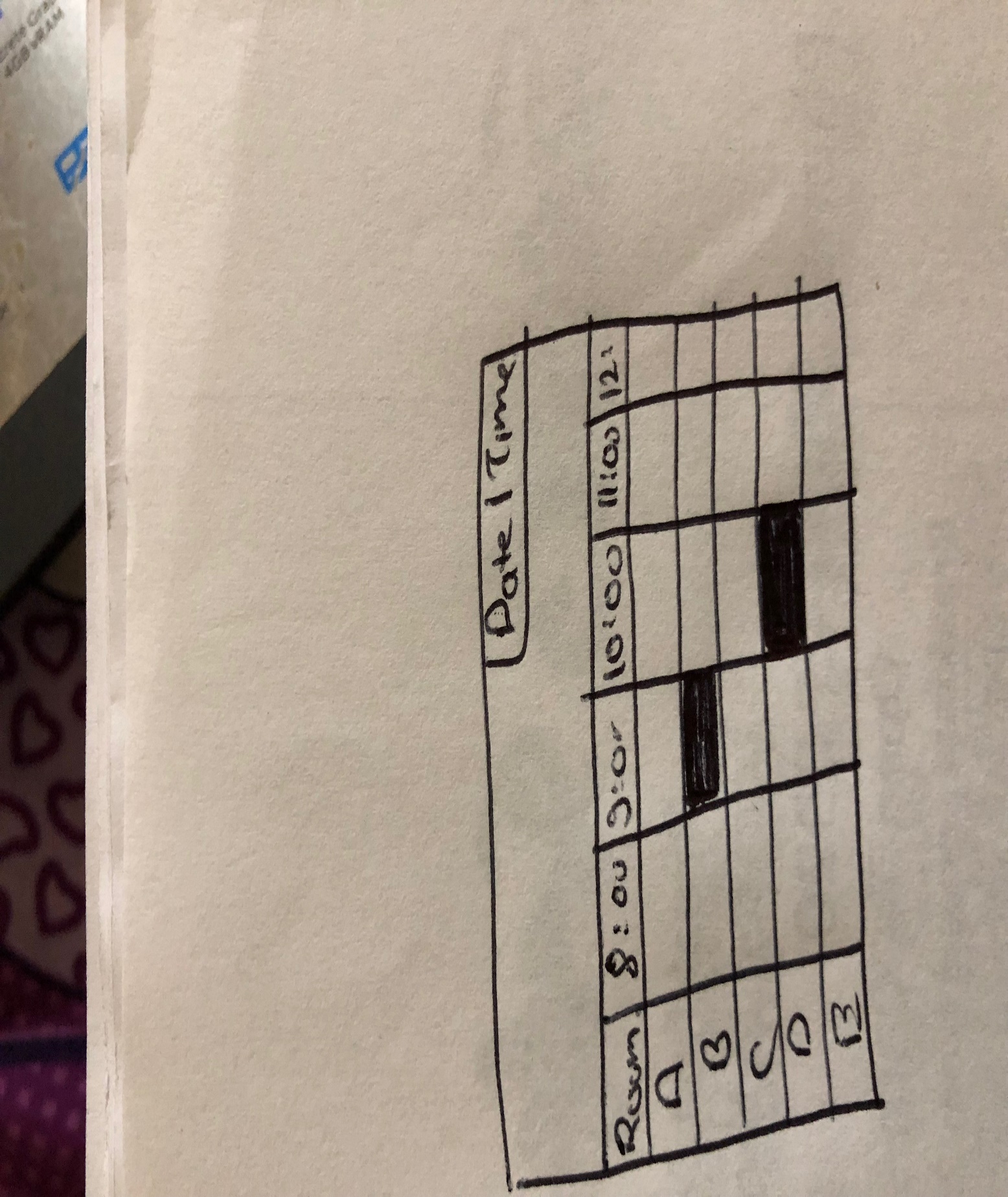


Fig 2: This is the resource scheduling UI from where we can choose rooms on specific time.

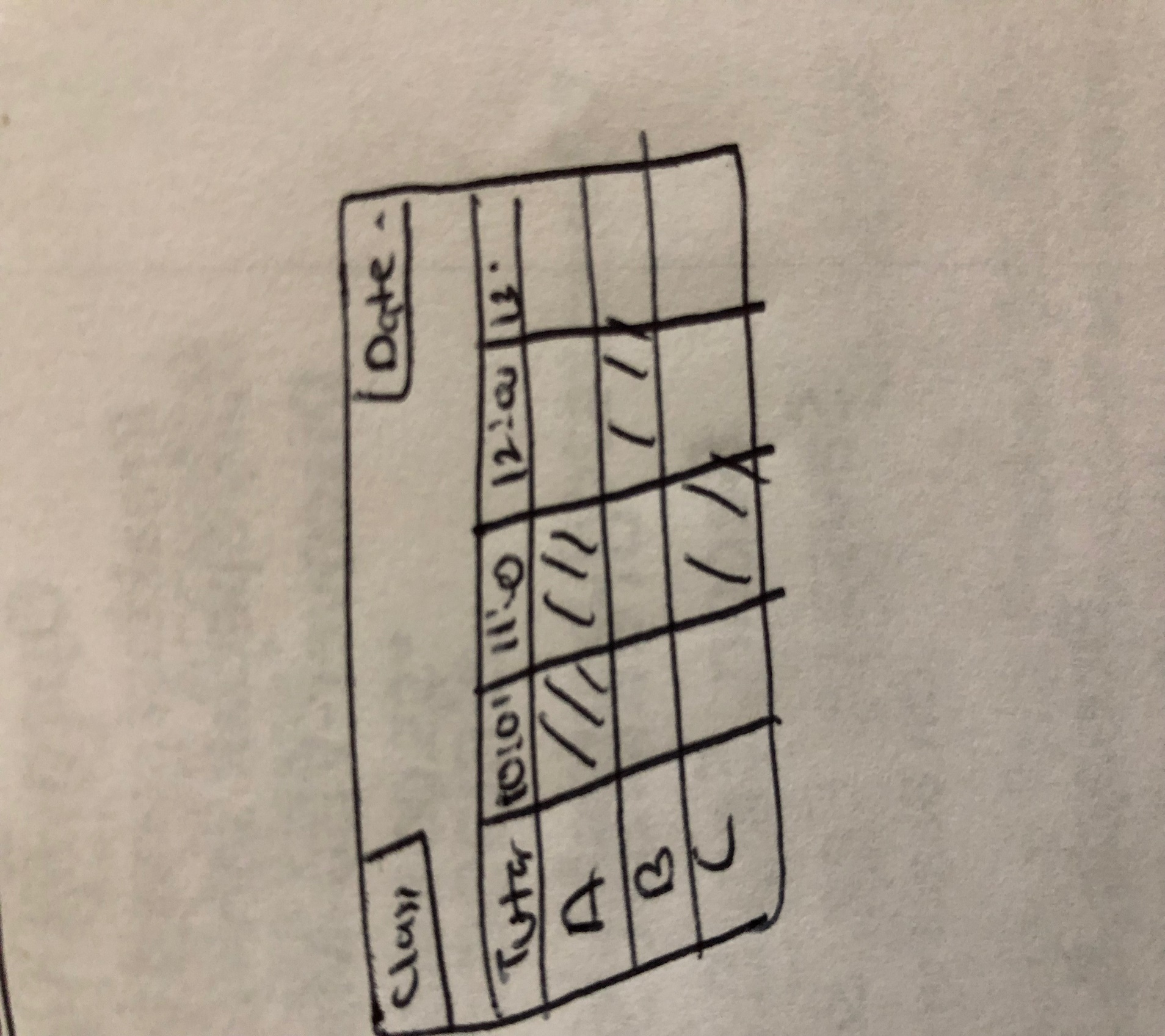


Fig 3 : This is the window to schedule tutor to a particular time.

# 3. Functional System Requirements

## Stakeholders

Office Manager: They need to be able to verify room reservations made by different students as well as accessing statistics about past room usage

Teachers: They need to be able to reserve different rooms that are available.

Students: They need to be able to request reservations for the rooms that are available

Tutors: They need to be able to access the system to view their schedule as well as enter their availability and different stipulations for working.

Schedule Maker: They need to be able to view all of the available tutors and use this information to create a weekly schedule that will adhere to all of the requirements.

## Actors and Goals

|  |  |  |
| --- | --- | --- |
| Actor | Goals | Use Case |
| Tutor, Schedule Maker  Tutor DB | Be able to enter availability, and different stipulations for the schedule | EnterTutorAvailability(UC-4) |
| Student, Teacher, Manager  Room DB | View the current schedule of all the rooms available for reservation | ViewCurrentSchedule(UC-1) |
| Student, Teacher, Manager  Room DB | Reserve a room given that it’s available | ReserveRoom(UC-2) |
| Manager  Room DB | View the statistics of each of the rooms showing how often rooms are reserved | ViewRoomStatistics(UC-3) |
| Schedule Maker  Tutor DB | Create a schedule using the availability provided by the tutors | CreateSchedule(UC-6) |
| Schedule Maker  Tutor DB | Manage all available tutors allowing their information to be changed or deleted | ManageTutors(UC-5) |
| Schedule Maker  Tutor DB | View statistics about previous schedules, like when more tutors are generally needed | TutoringStatistics(UC-7) |

## Use Cases

**Casual Description:**

UC#1 Viewing Current Room Schedule

Any user should be able to open the program and look at the current schedule of rooms. This will show which rooms are currently available as well as which are currently in use. If a room is available, it will give the user an option to reserve the room. The schedule should be updated in real time so that multiple requests are not made to the same room.

UC#2 Reserving a Room

Users should be able to go from the current schedule to a screen that will allow them to reserve a room. Teachers and department staff should be able to reserve a room with no questions asked. Students trying to make a reservation will have a request sent to the Office Manager that will then have to either approve or deny the request. Reserved rooms are not able to be requested.

UC#3 Viewing Statistics of Room Reservations

The office manager should be able to open the program and have an option to view statistics about past room reservations. This will allow the user to see which rooms are reserved the most often as well as what times are the most popular for reservations.

UC#4 Entering Tutoring Availability

Tutors should be able to log in and provide the schedule maker with their availability. This will allow them to enter their hours available for each week. They will also be able to enter their work stipulations that will affect when and how they are scheduled. They should be able to update the provided information when necessary.

UC#5 Managing Current Tutors

The schedule maker should be able to view all of the current tutors that are in the system. They should be able to add new tutors, edit currently existing tutors and delete any of the existing tutors. Deleting should require verification before fully deleting the tutor.

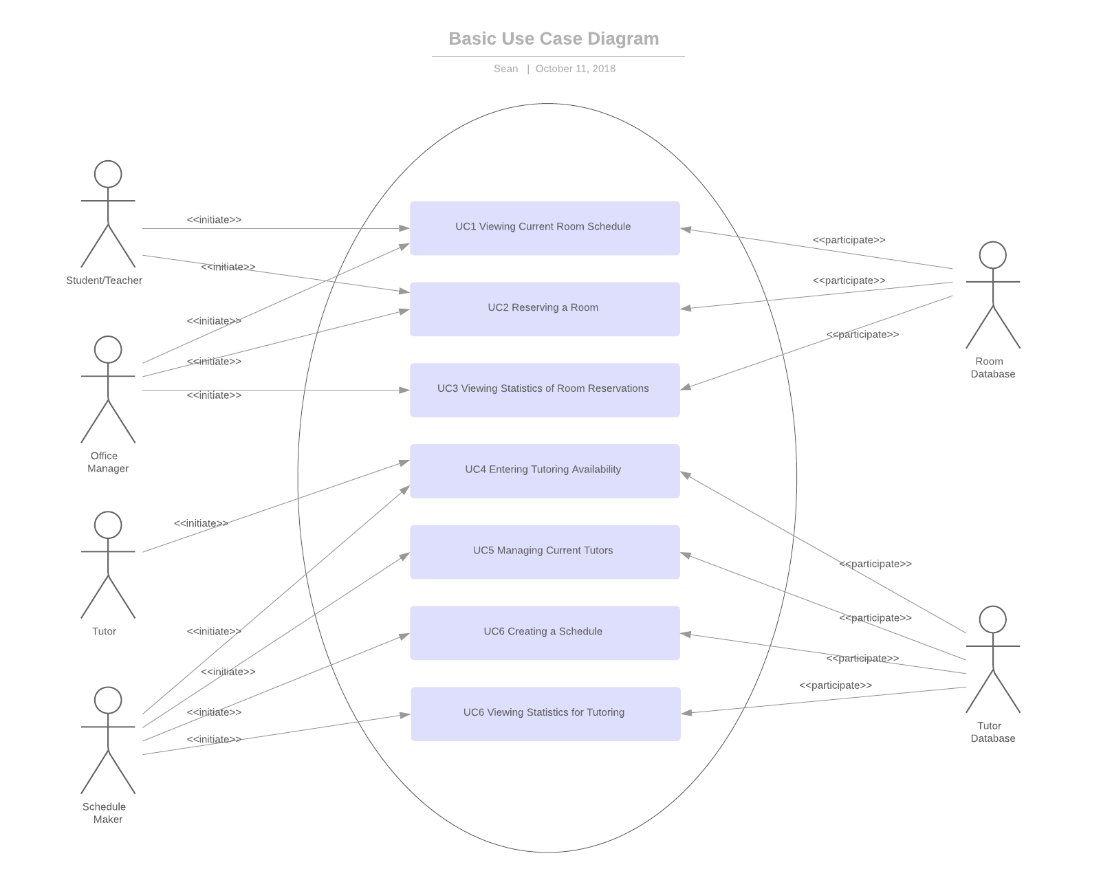
UC#6 Creating a Schedule

The schedule maker needs to be able to create a weekly schedule based on the tutors that they have available. The program should suggest tutors so that each table is covered by a tutor with that subject listed at all times of the day. The schedule maker should just have to confirm the schedule will work and then let the tutors know.

UC#7 Viewing Statistics for Tutoring

The schedule maker should be able to view statistics for when tutors are most in demand. They should have a way to view when extra tutors needed to be scheduled as well as how often. This will help them determine whether they should initially schedule more tutors on certain days or at certain times.

**Use case Diagram:**



**Traceability Matrix:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Priority weight | UC#1 | UC#2 | UC#3 | UC#4 | UC#5 | UC#6 | UC#7 | UC#8 |
| REQ-1 | 5 | X |  | X |  |  |  |  |  |
| REQ-2 | 5 | X |  | X |  |  |  |  |  |
| REQ-3 | 5 | X |  | X |  |  |  |  |  |
| REQ-4 | 5 |  | X |  |  |  |  |  |  |
| REQ-5 | 3 |  | X |  |  |  |  |  |  |
| REQ-6 | 2 |  |  |  |  |  |  |  |  |
| REQ-7 | 5 |  |  |  |  |  |  |  |  |
| REQ-8 | 4 |  |  |  | X | X | X |  |  |
| REQ-9 | 4 |  |  |  |  |  |  |  |  |
| REQ-10 | 4 |  |  |  | X | X | X |  |  |
| REQ-11 | 4 |  |  |  |  |  |  |  | X |
| RE-12 | 3 |  |  |  | X |  |  | X | x |
| Total Weight |  | 15 | 8 | 15 | 11 | 8 | 8 | 3 | 7 |

**Fully -Dressed description of use case**

|  |
| --- |
| **Use Case UC-1: Viewing Current Room Schedule** |
| **Related Requirements:** REQ-1,REQ-2,REQ3 |
| **Initiating Actor :** Student, Teacher and Office manager |
| **Participating Actor:** Room database |
| **Actor’s Goal:** View the status of the room and the schedule of the tutor. |
| **Pre condition:**  The system has a reservations for a programs on a given time and tutors are assigned a particular calsss according to their schedules. |
| **Flow of events For Main Success Scenarios:**  User selects “View Room Schedule” in main UI.  Systems view the schedule of the rooms. |

|  |
| --- |
| **Use Case UC-3: Viewing Statistics for Room Reservation** |
| **Related Requirements:** REQ-1,REQ-2,REQ-3 |
| **Initiating Actor :** Student, Teacher and Office manager |
| **Participating Actor:** Room database |
| **Actor’s Goal:** Reserve the room for the events |
| **Pre condition:**  The system has a reservations space for room to be reserved for different events.  **Post conditions :**  The system views thestats for different rooms for the given date and time. |
| **Flow of events For Main Success Scenarios:**  Admins selects the available time period and views different stats about the rooms.  Systems provides the stats for different rooms. |
| **Use Case UC-4 : Entering Tutor availability** |
| **Related Requirements:** REQ-8,REQ-10,REQ-12 |
| **Initiating Actor : Tutor, Schedule Maker** |
| **Participating Actor: Tutor** database |
| **Actor’s Goal:** Entering the tutor available hours into the system |
| **Pre condition:**  The system has a space for tutor to enter their available hours into the system.  **Post conditions :**  The system adds the tutors hours into the system. |
| **Flow of events For Main Success Scenarios:**  User select “Enter tutor hours “ in main UI.  System gives the form for tutor to enter their available hours. |

|  |
| --- |
| **Use Case UC-6 : Creating a Schedule** |
| **Related Requirements:** REQ-8,REQ-11 |
| **Initiating Actor : Schedule Maker** |
| **Participating Actor: Tutor** database |
| **Actor’s Goal:** Creating a schedule for the tutors |
| **Pre condition:**  The system has got schedule of tutors for making the schedule.  **Post conditions :**  The system makes the schedule for the tutors |
| **Flow of events For Main Success Scenarios:**  User select “Create Schedule “ in main UI.  System reviews the schedule of the tutors and creates schedules.. |

## System Sequence Diagram

UC-1 Viewing current room Schedule

Database

UI

User

Asks for Schedules System refers database

Database sends

information to UI

UC-2 **: Viewing Statistics for Room Reservation**

User

Database

UI

User ask for Statistics System refers database

For room Information sent back.

**Use Case UC-4 : Entering Tutor availability**

Enters the tutor Systems updates the

User

UI

Database

Available hours Database

**Use Case UC-6 : Creating a Schedule**

Request the tutor’s Request the schedules

User

UI

Database

Schedules

Views the schedule Provides the schedule

Creates a schedule Delivers schedule to database

# 4. User Interface Specifications

## Preliminary Design

## User Effort Estimation