Scheduling System for Math Department Resources

Matthew Mooney

Brandon Rozzi

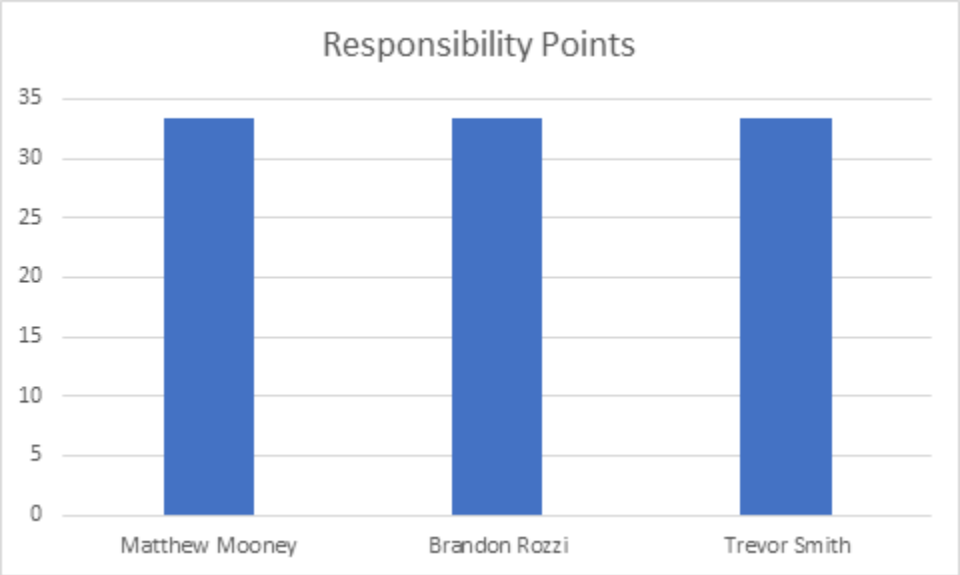
Trevor Smith

Individual Contribution Breakdown

Table 1: Responsibility Matrix

Allocation by Team Member		**Team Member Name**		
		Matthew Mooney	Brandon Rozzi	Trevor Smith
Responsibility levels	Sec 1: Interaction Diagrams	33%	33%	33%
	Sec 2: Class Diagram and Interface Specification	33%	33%	33%
	Sec 3: System Architecture and System Design	33%	33%	33%
	Sec.4: Algorithms and Data Structures (if applicable)	33%	33%	33%
	Sec.5: User Interface Design and Implementation	33%	33%	33%
	Sec.6: Design of Tests	33%	33%	33%
	Sec.7: Project Management and Plan of Work	33%	33%	33%

Graph 1: Responsibility Chart



Contents

Interaction Diagrams

Approval

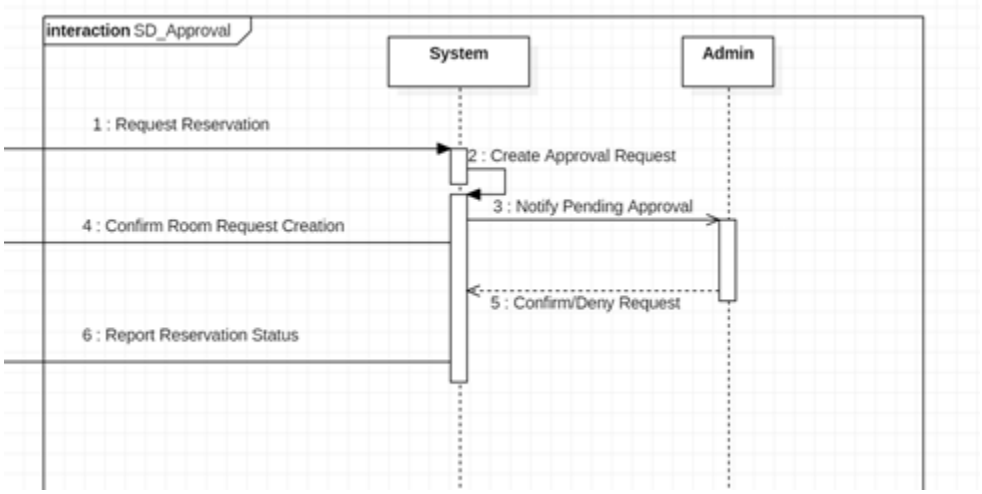


Figure 1: Approval Sequence Diagram

The Approval Use Case has two main contributors, and one outside actor who starts the flow of events. The design principle of the System actor is the Expert Doer Principle. The System actor must be notified of an outside user wanting to reserve something, create the request, and then notify the Admin of the pending request. Since the System is tasked with both communication and computation responsibilities, using Expert Doer Principle for the System makes sense. The Admin's design principle is High Cohesion since the Admin shouldn't need to perform any computations. The Admin should only need to communicate with the system that the request has been approved or denied.

Room Scheduling

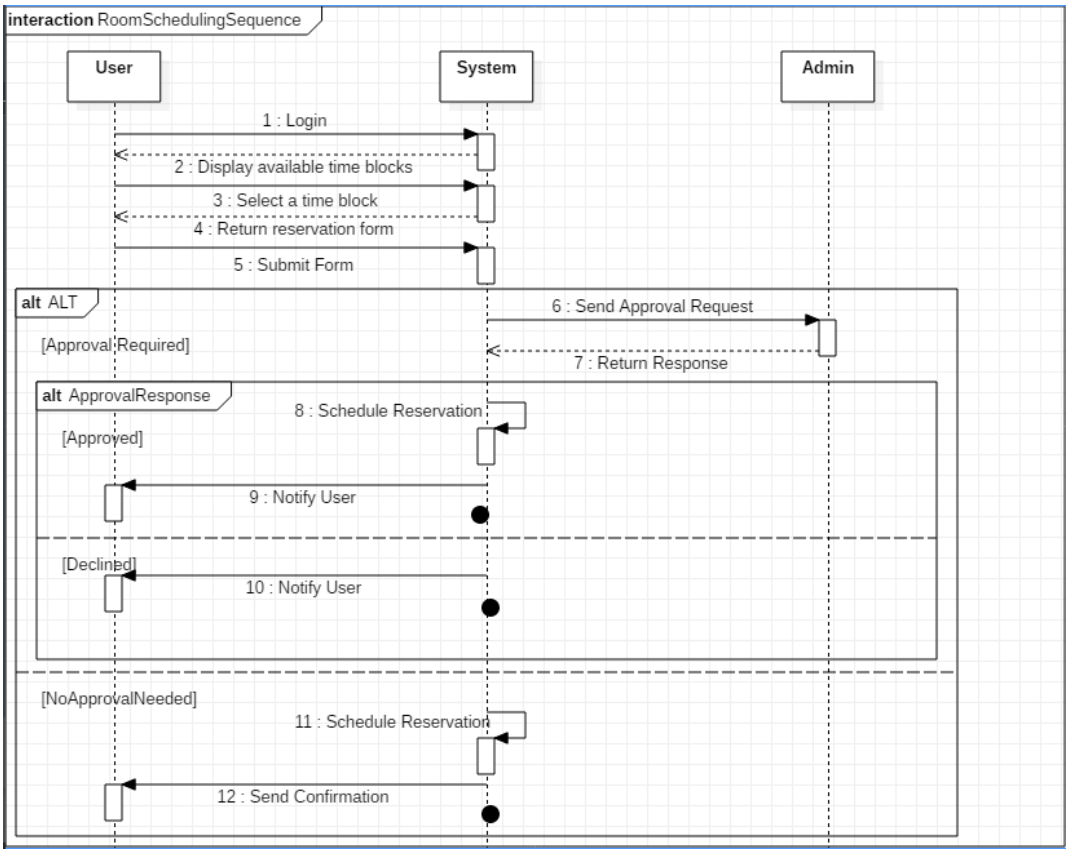


Figure 2: Room Scheduling Sequence Diagram

The room scheduling use case has two main contributors and an outside actor who starts the flow of events. The System actor must be notified of an outside user wanting to reserve a room, create the request, send the request, schedule the event and notify the end user. Since the System is tasked with both communication and computation responsibilities, using Expert Doer Principle for the System makes sense. The Admin's design principle is High Cohesion since the Admin shouldn't need to perform any computations. The Admin should only need to communicate with the system that the request has been approved or denied.

Adding/Editing Tutors

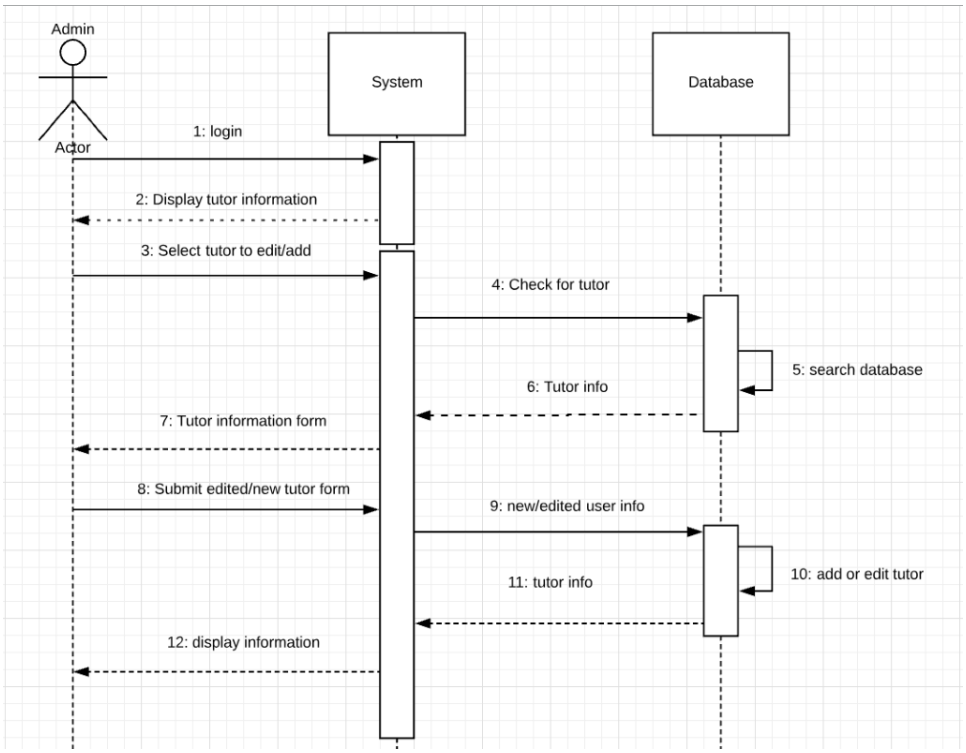


Figure 3: Adding/Editing Tutors Sequence Diagram

The Adding/Editing Tutors Use Case has one main contributor who starts the flow of events. The admin actor must login and select a tutor to edit/add, give information about the tutor, and submit the form. The design principle of the System actor is the Expert Doer Principle. Since the system actor does not need to do any computations then the admins design principle is of high cohesion. The admin only needs to communicate with the system to add or edit a tutor.

Project Management

Merging Contribution Conflicts

- The team worked in an online document simultaneously while discussing design elements.
- Each team member completed each section for their assigned use case in order to keep the language and terminology consistent for each use case.

Project Coordination and Progress Report

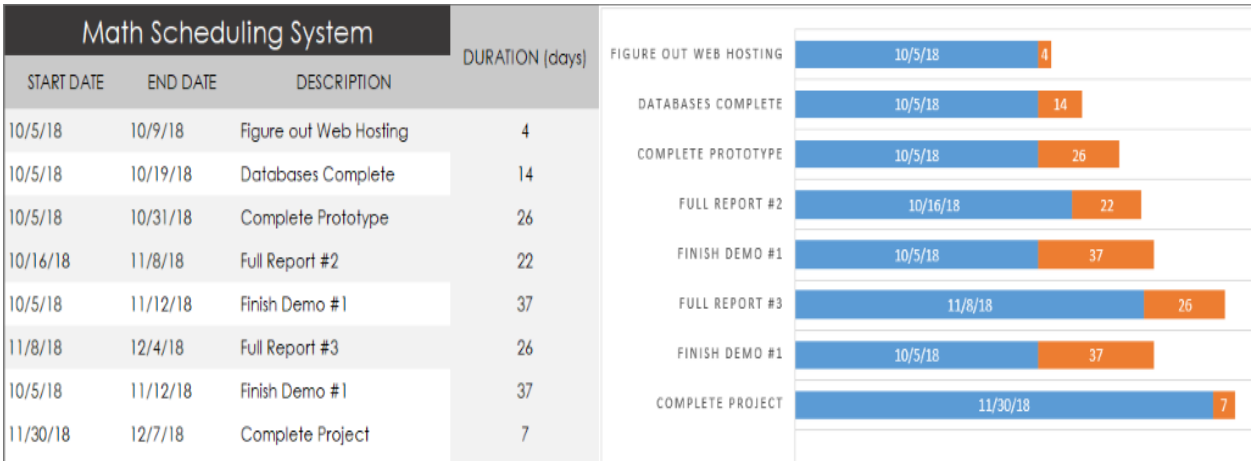
- Implementation has not begun.
- The team plans to begin implementation in the next few days to keep on schedule.

Plan of Work

As of today, we have all split the work of the schedule system evenly as seen in *table 1:*

Responsibility Matrix. The scheduling system has a total of nine functional requirements that can be split up evenly amongst the group. Trevor will be handling REQ-6, REQ-7, and REQ-9 all of which will deal

with the analytics of the scheduling system. Brandon will be handling the requirements that deal with the user's viewing of the scheduling system. These include the requirements REQ-1, REQ-5, and REQ-8. Matt will be handling the requirements of room availability and requests. These Include REQ-2, REQ-3, and REQ-4. Our immediate plans are to figure out the web hosting. After that is squared away, we will work on our databases for the system that are needed. Our short-term goal is to get all of this done by the 19th of October as seen in *figure 4*. Staying on track with this time line we will have a functioning prototype to present to our client by the end of October. This will give us more than enough time to be able to address any unseen problems before the completed project deadline. After the prototype, we will finish up the second report and that will lead us into the first demo we can present. After the first demo we will begin to implement any nonfunctional requirements that need to be added. Then be able start combining report one and two for the final report for the system. Once all this is completed, we will focus on and finish the final version of the scheduling system.



Math Scheduling System

START DATE	END DATE	DESCRIPTION	DURATION (days)
10/5/18	10/9/18	Figure out Web Hosting	4
10/5/18	10/19/18	Databases Complete	14
10/5/18	10/31/18	Complete Prototype	26
10/16/18	11/8/18	Full Report #2	22
10/5/18	11/12/18	Finish Demo #1	37
11/8/18	12/4/18	Full Report #3	26
10/5/18	11/12/18	Finish Demo #1	37
11/30/18	12/7/18	Complete Project	7

Figure 4: Gannt Chart for Scheduling System

Breakdown of Responsibilities

Team Member	Assigned Task
Brandon Rozzi	Design the tutor management modules
Matthew Mooney	Design the approval module
Trevor Smith	Design the room scheduling module

References