Room Scheduling Domain Model

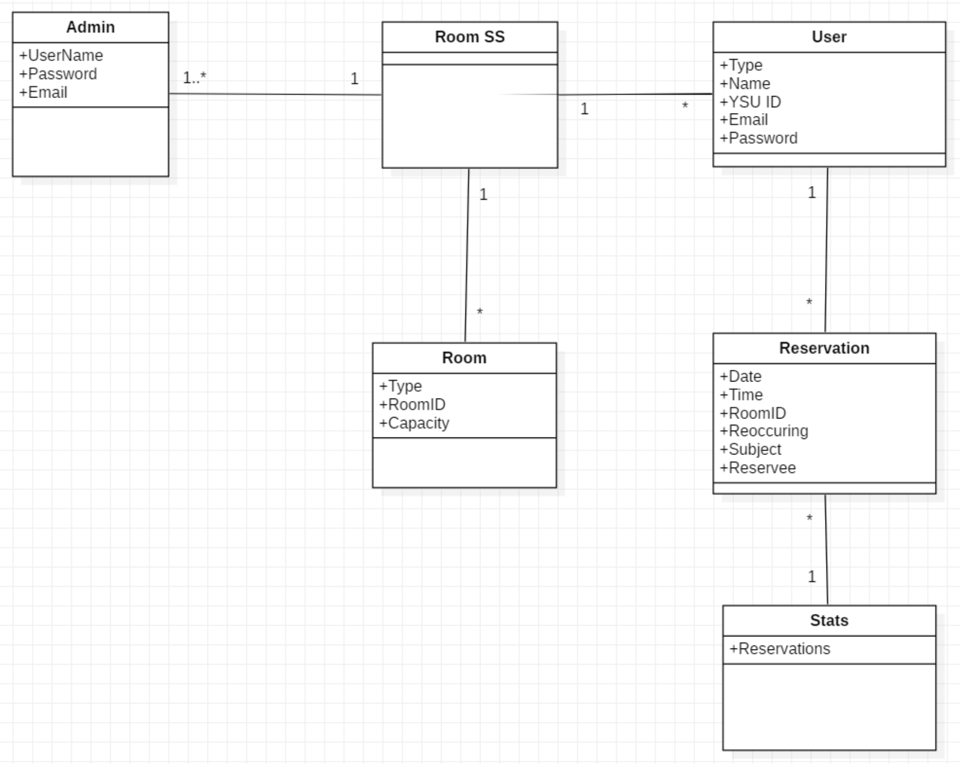

Figure 20: Domain Model (RSS)

Concept Definitions

- DC1: Room Scheduling System: This is the internal representation of the system which processes and schedules reservation requests as well as notifying users and sending request to the admin.
- DC2: Users: The students, faculty and other external actors which will interact with the system
- DC3: Admin: The Math Department Administrator who will manage room availability and request approvals within the system.
- DC4: Room: A room represents a physical room within the Math Department which can be scheduled with a reservation.
- DC5: Reservation: A scheduled block of time in which a user has reserved a specific room for an event.
- DC6: Stats: Statistical representation of the frequency at which reservations are placed.

Association Definitions

- Room Scheduling System to Admin: One room scheduling system has at least one admin
- Room Scheduling System to Rooms: One room scheduling system has many rooms
- Room Scheduling System to Users: One room scheduling system contains many users
- Users to Reservation: One user has many reservations
- Reservation to Stats: One stat analyzes many reservations

Attribute Definitions

- Admin
  - Username: To identify the person trying to access the system
  - Password: To allow access to the admin to enter the system
  - Email: Email address used for to contact the admin
- User
  - Type: To identify if user is a student or faculty outside of the math department
  - Name: Name of the user
  - YSU ID: unique ID for the user in the format of Y00765690
  - Email: Email address used to login in and contact the user
- Room
  - Type: if the room is a conference room or computer lab
  - Room ID: unique id to identity each room
  - Capacity: the number of people a room can hold
- Reservation
  - Date: the date a room is reserved for
  - Time: the specific hour that a room is reserved for
  - Room ID: unique id of the room being reserved
  - Reoccurring: true if the room is reserved for multiple days, false otherwise.
  - Subject: Title for the event
  - Reservee: The user who made the reservation
- Stats
  - Reservations: Reservation objects to get analytics from

Traceability Matrix

Use Case	Priority	Domain Concept
Scheduling a Room	1	1,2,3,4,5,6
Approval	2	3
Viewing Tutors	3	7,8,10
Room Unavailability	4	3
Obtaining a Suggested Schedule	5	7,8,9
Viewing Room Analytics	6	6

Operation: SubmitReservationRequest() Cross References: Use Case - Room Scheduling Preconditions: The user is logged in and his filled out the form correctly. Postconditions: The request has been submitted
Operation: SendApprovalRequest(Reservation: Reservation) Cross References: Use Case - Room Scheduling Preconditions: A reservation request has been submitted and requires approval Postconditions: The reservation request has been sent to the Admin for approval
Operation: ScheduleReservation(Reservation: Reservation) Cross References: Use Case - Room Scheduling Precondition: A reservation has been approved or a reservation request submitted that did not require approval Postcondition: The reservation is scheduled, and the time block is removed from the availability pool
Operation: NotifyUser(User: User, Reservation: Reservation) Cross References: Use Case – Room Scheduling Preconditions: A request has been accepted and reservation scheduled, or a request has been denied Postconditions: User is notified of the change
Operation: ViewApprovals()Cross References: Use Case: Approval Preconditions: Admin successfully authenticates into the scheduling system. Postconditions: -Admin views list of pending requests for room reservation and user faculty status.
Operation: ApproveRequest(Reservation: Reservation) Cross References: Use Case: Approval Preconditions: Admin successfully views pending requests and selects a request. Postconditions: Request is approved, notifying the user.
Operation: DenyRequest(Reservation: Reservation) Cross References: Use Case: Approval Preconditions: Admin successfully views pending requests and selects a request. Postconditions: Request is denied, notifying the user.
Operation: ApproveRequest(User: User) Cross References: Use Case: Approval Preconditions: Admin successfully views pending requests and selects a request. Postconditions: Request is approved, notifying the user.
Operation: DenyRequest(User: User) Cross References: Use Case: Approval Preconditions: Admin successfully views pending requests and selects a request. Postconditions: Request is denied, notifying the user.

Tutor Scheduling System Domain Model

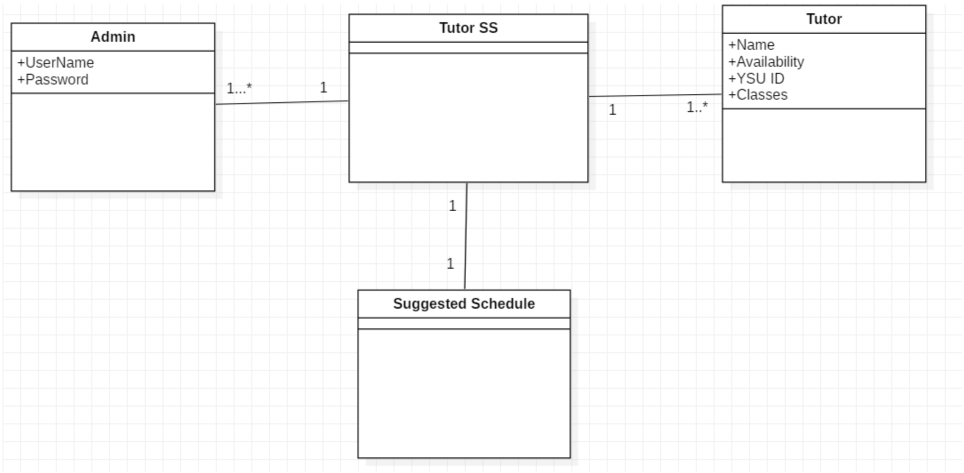

Figure 21: Domain Model (TSS)

Concept Definitions

- DC7: Tutor Scheduling System: This is the internal representation of the system which is responsible for storing tutor information as well as generating the suggested schedule.
- DC8: Admin: The MAC Coordinator who will manage and interact with the Tutor Scheduling System
- DC9: Suggested Schedule: A suggested tutor work schedule built from analyzing tutor availability
- DC10: Tutors: The MAC personnel who provide tutoring to the students

Association Definitions

- Tutor Scheduling System to Admin: One tutor scheduling system has at least one admin
- Tutor Scheduling System to Tutor: One tutor scheduling system has many tutors
- Tutor scheduling system to Suggested Schedule: One tutor scheduling system has many suggested schedule

Attribute Definitions

- Admin
  - Username: To identify the person trying to access the system
  - Password: To allow access to the admin to enter the system
- Tutors
  - Name: Name of a specific tutor
  - Availability: Hours they can work each week
  - YSU ID: unique ID for the user in the format of Y00765690
  - Classes: class subjects that a tutor can teach

Traceability Matrix

Use Case	Priority	Domain Concept

Scheduling a Room	1	N/A
Approval	2	N/A
Viewing Tutors	3	7,8,10
Room Unavailability	4	N/A
Obtaining a Suggested Schedule	5	7,8,9
Viewing Room Analytics	6	N/A

Operation Contracts

Operation: SelectTutor(tutor) Cross References: Updating Tutors Preconditions: Admin has logged in and is viewing list of tutors Postconditions: - A new/updated information about tutor is entered
Operation: checkTutor(tutor) Cross References: Updating Tutors Preconditions: A database with the tutors exists Postconditions: Tutor information is sent back to admin
Operation: submitTutorInfo() Cross References: Updating Tutors Preconditions: Admin has the tutor information form Postconditions: The new tutor information has been submitted
Operation: updateTutor(tutorInfo) Cross Reference: Udating Tutors Preconditions: - A database with tutor information exists - Edited tutor information is submitted Postconditions: Tutor information is updated in the database