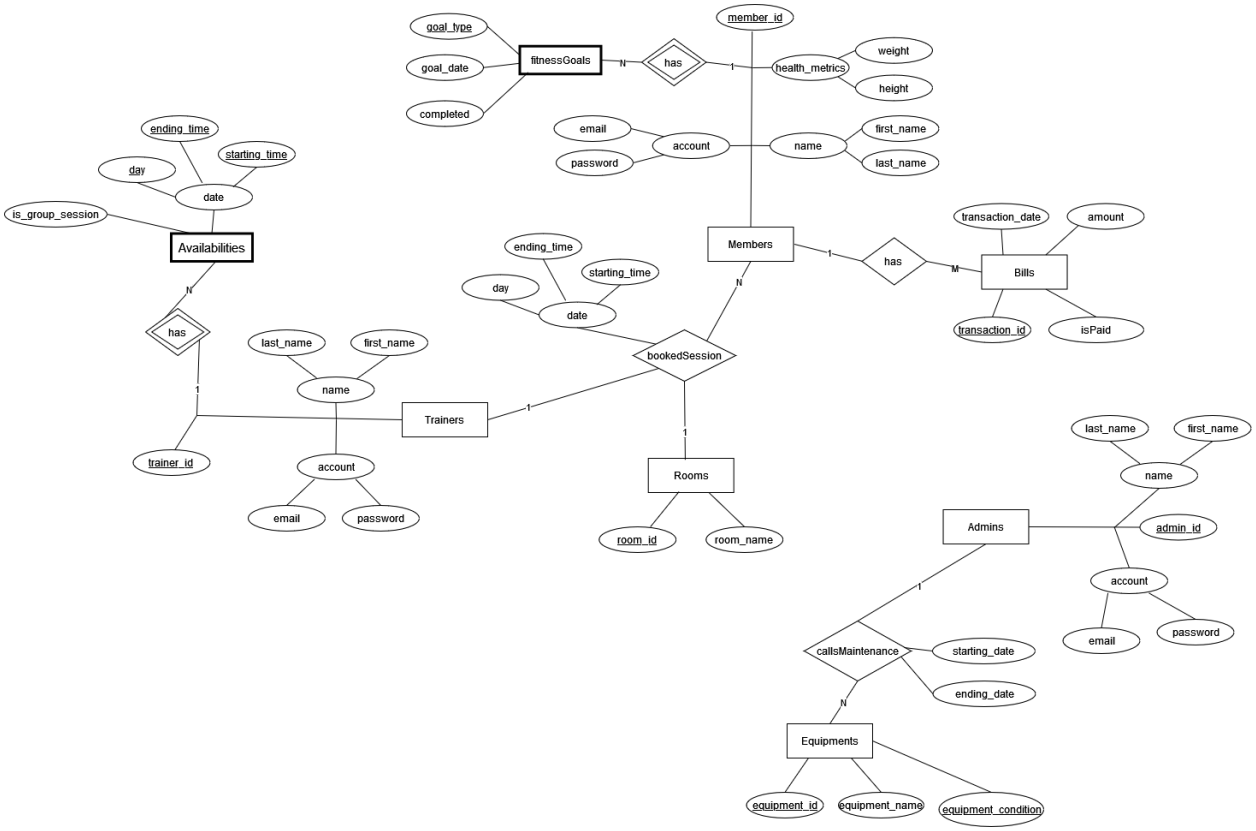
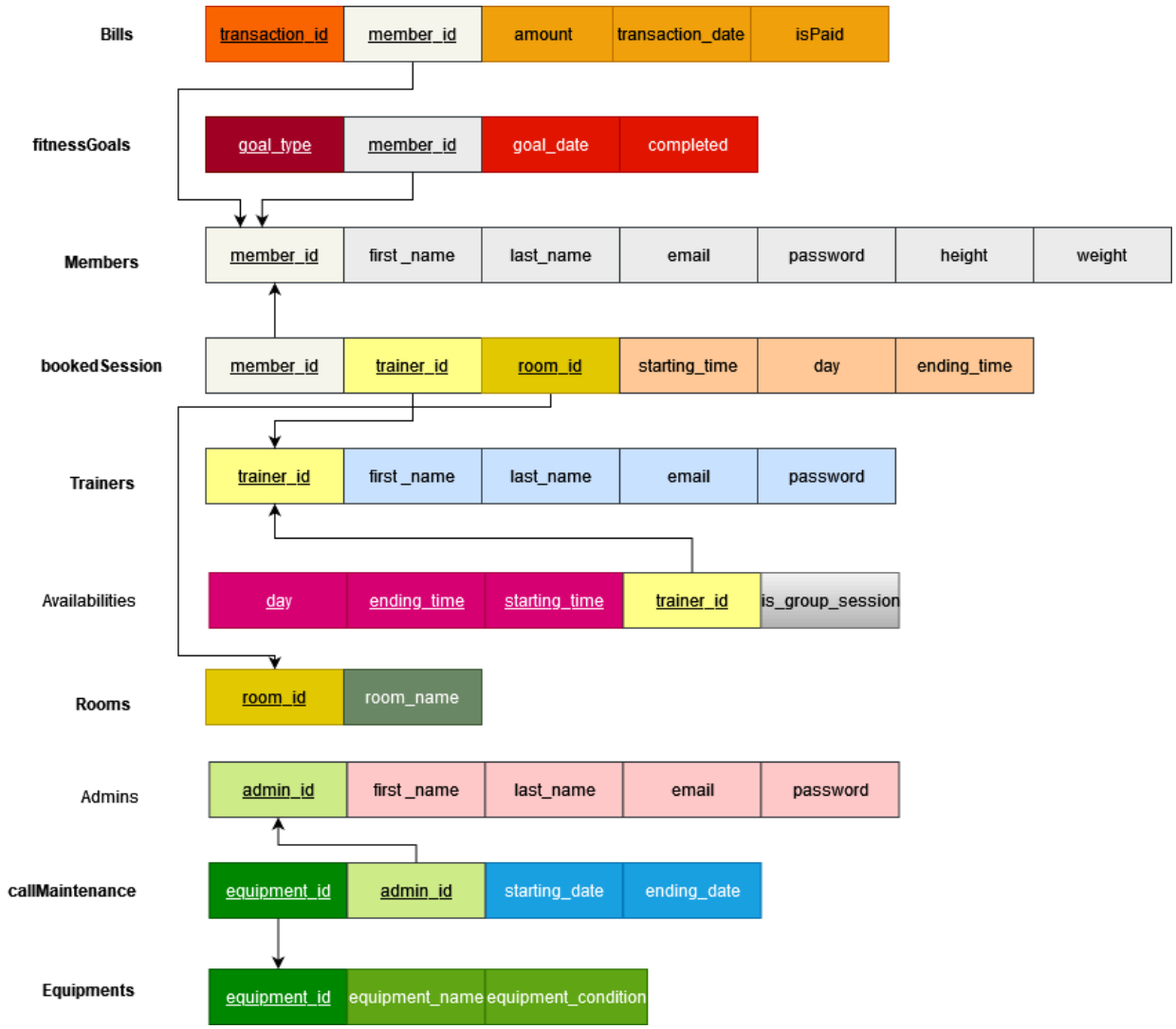
ER-Diagram (check pdf file if its too small or blurry):



ER-Mapping:


Bills
transaction_id
member_id
amount
transaction_date
isPaid
fitnessGoals
goal_type
member_id
goal_date
completed
Members
member_id
first _name
last_name
email
password
height
weight
bookedSession
member_id
trainer_id
room_id
starting_time
day
ending_time
Trainers
trainer_id
first _name
last_name
email
password
Availabilities
day
ending_time
starting_time
trainer_id
is_group_session
Rooms
room_id
room_name
Admins
admin_id
first _name
last_name
email
password
callMaintenance
equipment_id
admin_id
starting_date
ending_date
Equipments
equipment_id
equipment_name
equipment_condition

Requirement	**Assumption**	**Representation in ER Model**
“Members should be able to … establish personal fitness goals (you can determine suitable fitness goals such as weight and time, and members will set the values)”	I will make a weak entity called fitness_goal which will contain: - goal_type(partial key) - Goal_date - Completed (is goal complete if so then make it an achievement) Additionally, I will also assume that a member can have multiple fitness goals, so it will be a one(member) to many(fitness_goals) relationship.	Members will have a “has” relation with the weak entity(double lined box) fitness_goals. Additionally, a N has been placed on fitness_goals and a 1 on members.
Members “scheduling a session. Following that, the system generates a bill, the member makes a payment, and then the system confirms the transaction.”	I make entity called Bills which will contain: - transaction_id(pri. key) - amount - transaction_date - isPaid (checks if bill has been paid) Additionally, I will assume that bills can only bill a single person but a member can have multiple bills. So this is a one(member) to many(bills) relationship.	Members will have a “has” relation with Bills. Bills will have a m near it and members will have one near it.
“Schedule Management (Trainer can set the time for which they are available.)”	I will make a weak entity called availabilities which will contain: - day - ending_time - starting_time Additionally, I will assume that trainers have many “availabilities”. So this is a one(trainer) to many(availability) relationship.	Trainers will have a “has” relationship with availabilities . Availability will have a N near it and Trainers will have a one near it.

	Furthermore, the "day" attribute is an integer because trainers' availability have to be on a weekly basis. So if it's sunday, then they are expected to work on sundays for the entire year. We also have starting hour and duration of how long there shift is on that day	
Administrative Staff Functions: 2. Equipment Maintenance Monitoring	I will make a entity that will be called equipment which will contain: - equipment_id(pri. key) - equipment_name - equipment_status	There exists a box with entity, and its attributes
Administrative Staff Functions: 2. Equipment Maintenance Monitoring	I will make it so admins will have a relationship with equipment called “callsMaintenance.” This relation will have the following attributes: - starting_date - ending_date Admins can call for the maintenance of any equipment, providing the starting and ending date of the maintenance for the equipment.(one(admin) to many(equipment) relationship)	Admin has a callMainteance relation with equipment. Admin has a 1 near it and equipment has n near it, to show the one to many relationship.
Design and implement an application for a Health and Fitness Club Management System. This system will serve as a comprehensive platform catering to the diverse needs of club members, trainers, and administrative staff.	It makes sense that members, trainers, and admins all will require a name(first and last name), email, and password for the accounts used for this system.	Members, Trainers, and Admins all have the following attributes: - first name - last name - email - password
Member Functions: 2. Profile Management (Updating personal	Members has 2 attributes: - Height - Weight	In the UML diagram member has an attribute called health_metrics which further

information, fitness goals, health metrics		has weight and height
Administrative Staff Functions: 1. Room Booking Management	Because admins can manage room bookings, this is why a room entity has been created which has the following attributes: - room_id - room_name	In the diagram I made a box that contained rooms and the attributes mentioned prior
Administrative Staff Functions: 1. Room Booking Management 3. Class Schedule Updating	I believe adding these functions to the ER diagram wouldn't make sense with what I've built as these functions don't necessarily create a relationship between admins and rooms/class-schedules (in sql).	Didn't include in ER diagram
Trainer Functions: 2. Member Profile Viewing (Search by Member's name)	I also didn't include this in the ER-diagram because this cannot be displayed as a relationship in the diagram	Didn't include in ER diagram
"Schedule Management (Trainer can set the time for which they are available.)" "Member Functions: 4. Schedule Management (Scheduling personal training sessions or group fitness classes. The system must ensure that the trainer is available)"	There is a relation called bookedSession. This requires both the member(s) who booked the session, a trainer, a room, and the time of the session. Process of booking: Trainer post availability -> Member books session -> Trainer availability removed from Availabilities table -> Add information of availability along with room_id, member_id, and trainer_id into the relation bookedSession. But if a trainer puts an availability time with the implicit intent of it being a group session then that availability is not removed from availabilities.	Added relation bookSession. It connects 3 entities: room, member, and trainer. Also we made it so n was near the member and the 1 was near the trainer and room boxes to represent the many to one relationship.

	Additionally, this is a many(members) to one(room and trainer) relationship.	
Member Functions: 4. Schedule Management (Scheduling personal training sessions or group fitness classes. The system must ensure that the trainer is available)	I will make it so trainers will show their availabilities to members. Then members can book sessions based on those availabilities. Then if the availability was for a single session then its gone, otherwise it stays.	N/A
Administrative Staff Functions: 1. Room Booking Management 2. Equipment Maintenance Monitoring 3. Class Schedule Updating 4. Billing and Payment Processing (Your system should assume integration with a payment service [Note: Do not actually integrate with a payment service])	I made it so admins can set a sessions attributes which makes it so admins can manage room booking and can update class scheduling. Additionally, I made it so admins can check up on all equipments and can send them for maintenance if they feel the condition of the equipment is low Finally admins can check all bills and either bill a member for x amount or cancel a pre-existing bill that hasn't been paid off. I'm also keeping bills that have been paid because I feel it makes sense to keep track of all financial records.	N/A
Member booking session	Members use the availability of the trainers and the room they would like to book to complete the booking of a session. Any contradictions will cancel the session.	N/A
Members paying bills	I made it so whenever a member books a session a $20 fee will be sent to the member. The member can pay it using paypal. Only bills that haven't been paid will be	N/A

	displayed to the member. So this is a session payment model and there are no subscriptions or membership fees.	
N/A	The fitness club will have opening hours of 7am to 11:59 pm to schedule appointments.	N/A
N/A	Passwords are hashed using bcrypt	N/A