Kevin Wu 101258379

COMP 3005 Project V2

2.1

Assumptions

Requirement	Assumption	Representation
Member function: User Registration	Users should have a corresponding username and password, which should be set by the user. I'll assume the username to be their email, since it would be unique and can be the primary key.	members entity Attributes: member_email (Primary key), member_password
Member function: Profile Management (Updating personal information, fitness goals, health metrics)	Users have a first name and last name as personal information. Their health metrics will be their height and weight. Their fitness goals will be stored in the form of a target weight, and a general text description.	members entity Attributes: height, curr_weight, goal_weight, goal_desc
Member function: Dashboard Display (Displaying exercise routines, fitness achievements, health statistics)	Exercise routines will display the personal training and group sessions that the user has booked. For this to work, the sessions have to track which users have booked them. Fitness achievements will be set by admin and viewable by the corresponding user.	fitness_achievements entity Attributes: member_email (references members), content training_sessions entity Attributes: member_email (references members) group_sessions entity group_sessions_registered relation
Member function: Schedule Management (Scheduling personal training sessions or group fitness classes. The system must ensure that the trainer is available)	Each training session should only be bookable if no other member has booked it to ensure trainer availability. Group fitness classes should allow more than one member to register, so this information should be stored in a separate table from the group sessions. This is done with a foreign key approach.	training_sessions entity Attributes: member_email (references members) group_sessions_registered entity Attributes: member_email (references members)
Trainer function: Schedule	Trainers should be able to create	training_sessions entity

Management (Trainer can set the time for which they are available.)	private training sessions with a desired timeslot. Availability of a session is determined by whether the foreign key member_email is null. The relation between trainers and training sessions is done by a foreign key approach, and so is the relation between members and training sessions. This way, each training session is associated with one trainer and one or null members.	Attributes: session_id (primary key), trainer_email (references trainers), member_email (references members), timeslot
Trainer function: Member Profile Viewing (Search by Member's name)	Trainers should be able to search for member information by name. I'll allow trainers to see a member's name, height, and weight. The searching will be done by finding matching last names.	Program functionality
Admin function: Room Booking Management	Admin should be able to create new room bookings for members. Members can specify which room to book and a time slot to book it. I'll assume for this that there are 10 rooms, numbered 1-10. I'll assume this process happens in person, and since the rooms are meant to be under control of admin, the booking process will only refer to admin_email, with no mention of the member requesting the room. With the process being the relation, it will connect admin with the rooms entity.	room_bookings relation Attributes: booking_id (primary key), admin_email (references admin), timeslot, room_number (references rooms) rooms entity Attributes: Room_number (primary key), name
Admin function: Equipment Maintenance Monitoring	Admin should be able to monitor all equipment that is under maintenance. This includes information about the piece of equipment under maintenance and the start date of the maintenance. Admin should also be able to update, add, and remove these entries. There should be an equipment table	equipment entity: Attributes: equipment_id (primary key), equipment_desc equipment_maintenance relation Attributes: equipment_id (references equipment), admin_email (references admin_email),

	containing all equipment, and the maintenance relation should be between equipment and admin.	date_start
Admin function: Class Schedule Updating	Admin should be able to create new group class sessions, which are bookable by members. Admin should also be able to cancel group sessions. These sessions should have a name and a time. I assume these sessions should not be run by trainers, since I assumed the trainers would stick to doing personal training sessions. Upon creation, the session should be given a name and a timeslot, omitting the need for a schedule relation between admin and group_sessions. There should be a registers relation between members and group sessions.	Program functionality group_sessions entity Attributes: session_id (primary key), session_name, timeslot group_session_registered relation Attributes: session_id, member_email
Admin function: Billing and Payment Processing	When a member books a personal training session, it should automatically create a bill payment when the user agrees to the fee. The same should occur when a new member registers. Admin can view a list of all incoming bill payments, and assuming the existence of a payment processing service, could then process the payments as desired. The relation between payments and admin is a foreign key approach.	training_sessions entity: Attributes: fee payments entity: Attributes: payment_id (primary key), member_email (references members), amount