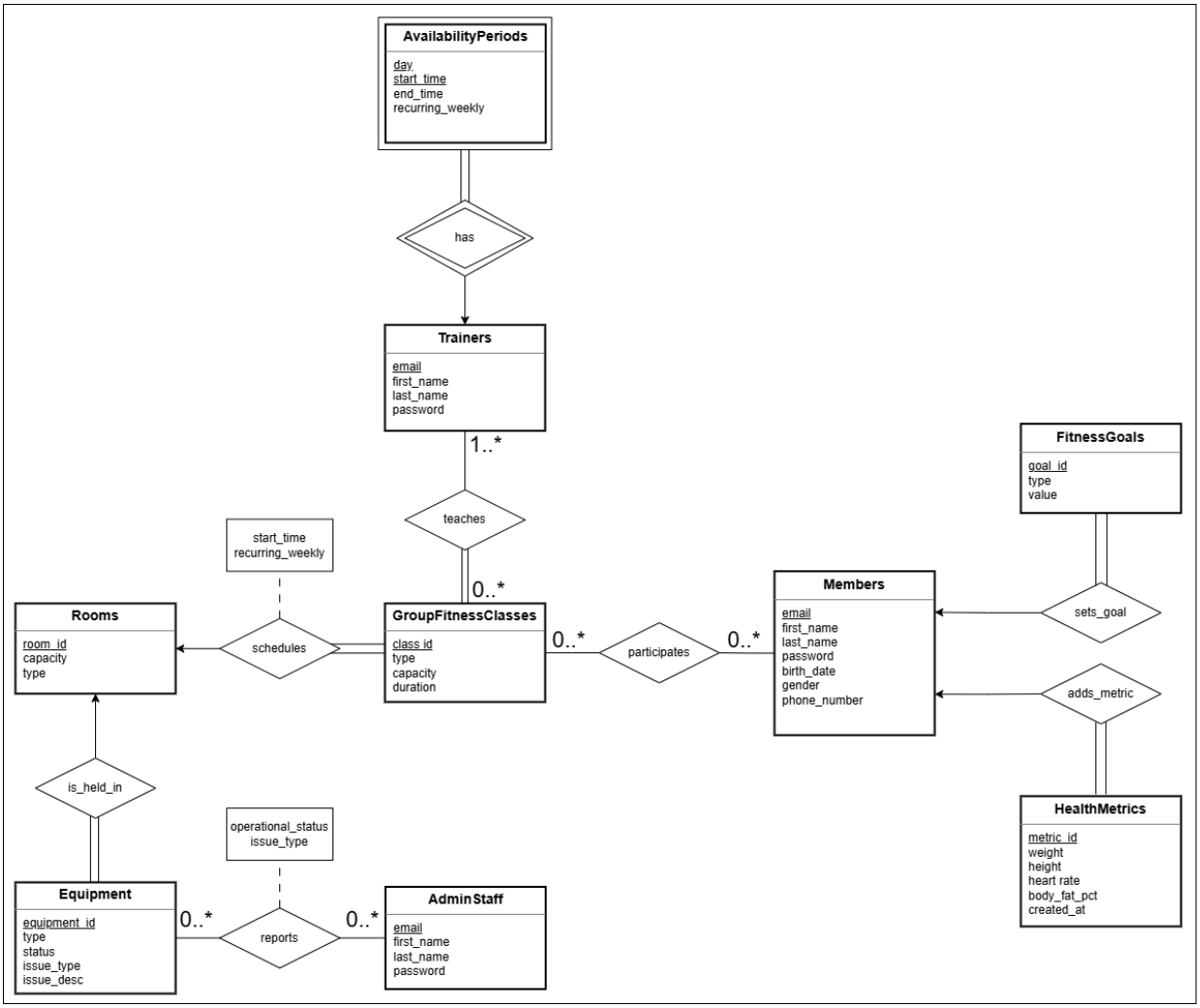
ER Diagram:



Assumptions:

- It's possible for group fitness classes to have multiple Trainers teaching it, but must always have at least one Trainer
- A member can have multiple fitness goals to create milestones for themselves
- Users log into their accounts using their email and password
- All Rooms exist within the same building (and thus have unique identifiers)
- Each Admin Staff member only makes 1 report per equipment, otherwise it gets overwritten
- Trainers can only look up members they are training in of their classes

Mapping (to Relational Database Schema):

We first created table schemas for each of our regular Entities (8 total) containing the attributes listed in the diagram.

We then handled the case of the Weak Entity “AvailibilityPeriods” by taking the Primary key “email” from its owner “Trainer” and combining it with the partial keys “day” and “start_time”.

We didn’t have any 1:1 relationships to represent, so we moved on to mapping the 1:N relationships. All our 1:N relationships had the N-side have total participation so we opted for the “Foreign Key” approach, giving every Entity on the side of N a foreign key that leads to the primary key of the Entity at the side with 1.

Lastly, we had the N:M relationships left, aka “reports”, “participates”, and “teaches”, so we created tables for them that cross reference the related Entities. As such those tables had composite primary keys that combine the primary keys of the Entities on both sides by “Referencing” them as foreign keys. In the case of reports, that table had the additional attributes “operational_status” and “issue_type”.

Normalization:

- The schema satisfies 1NF because all the attributes store single atomic values.

- It satisfies 2NF because all "primary key -> nonprime attribute" relations are fully functionally dependent. We made sure that was the case, by adding “ids” for certain entities to reinforce having unique records, while also working under the assumption that “All Rooms exist within the same building (and thus have unique identifiers)”.

Additionally, in the case of a table having a composite primary key, there are no nonprime attributes that depend on only part of that key; thus, we avoid having partial dependency.

Note that to also make this possible for our Reports relationship Table, we are working under the assumption that “Each Admin Staff member only makes 1 report per equipment, otherwise it gets overwritten”.

- It satisfies 3NF because no nonprime attribute depends on another nonprime attribute; all nonprime attributes depend directly on the primary key.
In other words, there is no relation such that X -> Z and Z -> Y hold, where Z is a nonprime attribute (no transitive relationships).