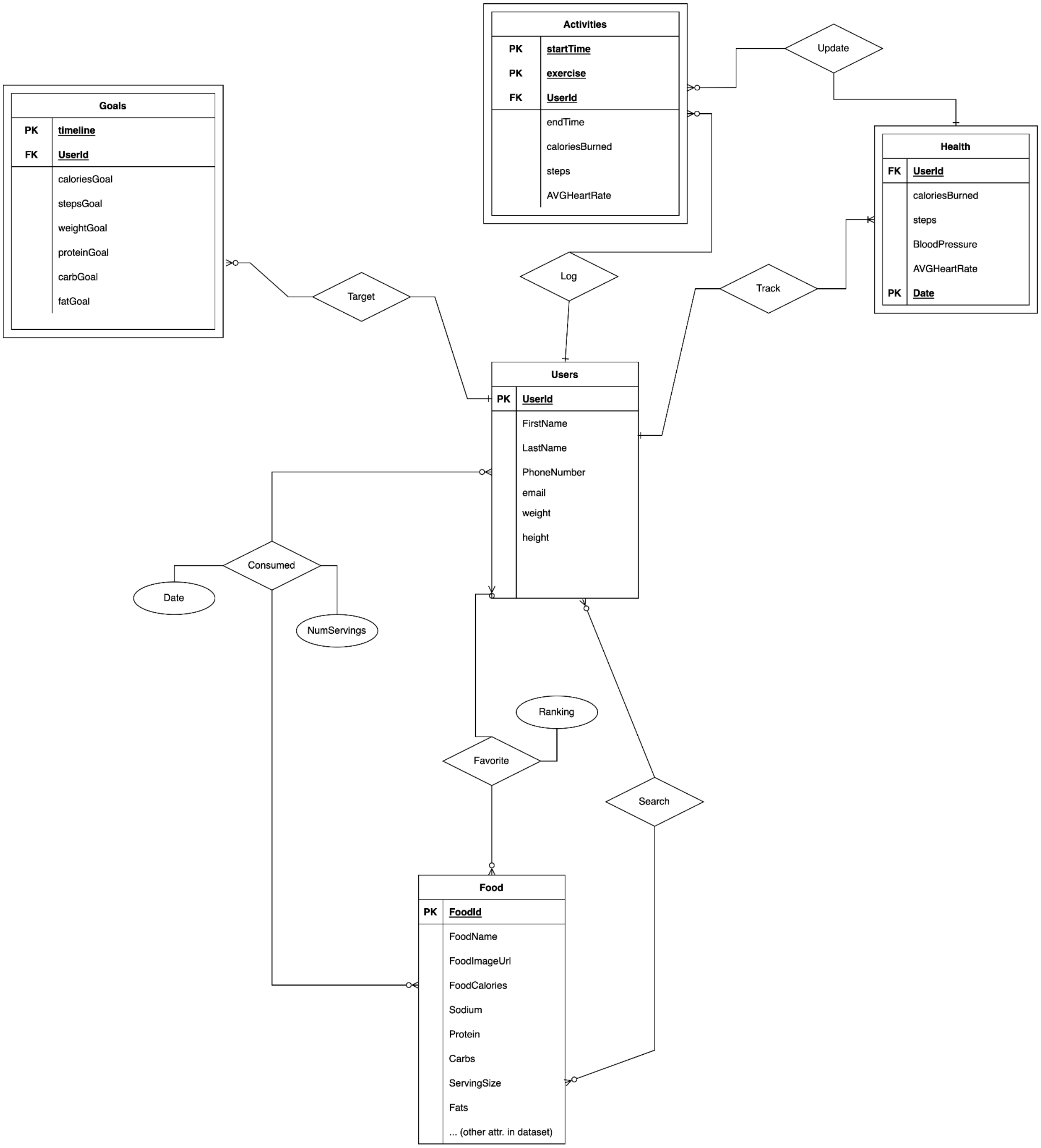
CMP003 - Project Stage 2: Conceptual and Logical Database Design

1. ER:



2. Assumptions and Implementation Design:

1. We will use a hash function for the user's username and password to generate a unique userId.
2. Every user must have their health tracked
3. The user will only have 1 goal per timeline (i.e., they will not have multiple calorie goals for the same day).
4. The user can only perform 1 activity at a time.
5. It does not matter what time a user eats, only the day they eat.
6. Goals, Activities, and Health are all User-specific and thus weak entities which require foreign keys to the user.
7. Multiple activities can update one instance of health.
8. Health is recorded on a daily basis based on User's timezone.
9. Users can rank their favorite food, and the ranking can be used to make recipe suggestions for the same user and others.
10. User's food consumption is tracked daily with the number of servings specified. This relational table will keep track of the User's diet (not a separate entity to reduce redundancies).

3. A description of each relationship and its cardinality:

1. Users ***track*** Health: The relationship is **one-to-many** as we want multiple users to have their health but a particular health only belongs to one person
2. Users ***log*** Activities: The relationship is **many-to-many** as we want multiple users to be able to log multiple activities
3. Users ***target*** Goals: The relationship is **many-to-many** as we want multiple users to be able to have multiple goals (one user should be able to have daily, monthly, and yearly goals)
4. Users ***search*** Food: The relationship is **many-to-many** as we want multiple users to be able to search up multiple foods (all foods practically)
5. Users ***favorite*** Food (Ranking): The relationship is **many-to-many** as we want multiple users to be able to have multiple favorite foods and one food can be the favorite food of multiple users
6. Users ***consumed*** (Date and NumServings) Food: The relationship is **many-to-many** as we want multiple users to be able to consume multiple foods and one food can be consumed by multiple users
7. Health ***updates*** Activities: The relationship is **one-to-many** as we want multiple activities that the user does to update the user's health status for a particular date

4. Convert ER to the logical design (relational schema)

//Entities:

Users(UserId: VARCHAR(10), FirstName: VARCHAR(255), LastName: VARCHAR(255), PhoneNumber: REAL, email:VARCHAR(255), weight: Numeric(5,2), height: Numeric(5,2));

Activities(startTime: DATETIME[PK], exercise: VARCHAR(255)[PK], endTime: DATETIME, caloriesBurned: Numeric(8,2), steps: INT , AVGHeartRate: Numeric(5,2), UserId: VARCHAR(10) [FK to Users.UserId], Date: DATE[FK to Health.Date])

Health(Date: DATE[PK], caloriesBurned: Numeric(8,2), steps: INT , AVGHeartRate: Numeric(5,2), BloodPressure: INT, UserId: VARCHAR(10) [FK to Users.UserId])

Goals(timeline: VARCHAR(50)[PK], caloriesGoal: Numeric(8,2), stepsGoal: INT , weightGoal: Numeric(5,2), proteinGoal: Numeric(8,2), carbGoal: Numeric(8,2) , fatGoal: Numeric(8,2), UserId: VARCHAR(10) [FK to Users.UserId])

Food(FoodId: INT[PK], FoodName: VARCHAR(255), FoodImageUrl: VARCHAR(255) , FoodCalories: Numeric(8,2), Sodium: Numeric(8,2), Protein: Numeric(8,2) , ServingSize: VARCHAR(255), Carbs: Numeric(8,2), Fats: Numeric(8,2))

//Relationship tables:

Favorite(UserId: VARCHAR(10) [PK][FK to Users.UserId], FoodId: INT [PK][FK to Food.FoodId], Ranking: INT)

Consumed(UserId: VARCHAR(10) [PK][FK to Users.UserId], FoodId: INT [PK][FK to Food.FoodId], Date: Date, NumServings: INT)

Search(UserId: VARCHAR(10) [PK][FK to Users.UserId], FoodId: INT [PK][FK to Food.FoodId])