1. Cover page (Completed)
2. Project Description
    a. The domain of the application is fitness/nutrition management. It allows users to keep track of their nutrition goals and log food which helps them by tracking their nutrition and giving recommendations based on dietary restrictions, favorite foods, and activity levels.

    b. The database models a user who wants to track their nutrition. A user has dietary restrictions, activity levels, and goals. A user eats meals that contain individual foods with nutrition information. This information is added to a daily record, keeping track of the nutrition according to the user's goals. The user is given meal recommendations based on goals, dietary restrictions, and activity levels.

    Ultimately, the project addresses the problem of a user tracking their nutrition following their goals, which can take time to do manually. It allows the user to input the food into the application, which then calculates everything for them. It also gives recommendations that fit the user's goals by taking their favourite foods into account.

3. Database Specification:
    - Provide nutrition facts (calories, protein, vitamins, minerals, etc.) about meals that users enter
    - Users can view their nutritional summaries over time, allowing them to assess their dietary habits and make informed decisions
    - Users can set their goals (e.g., daily calorie intake) and activity level so that the database can track and provide feedback to help them stay on track
    - Users can mark their favourite food, and identify as a dairy restriction/vegan which allows the system to provide customized recommended foods to help them achieve their daily intake
4. Description of the application platform
    a. This project will use "MySQL" as the database system
    b. We will use JavaScript/Node.js/ Express JS and Oracle Server provided by the department

5. ER Diagram
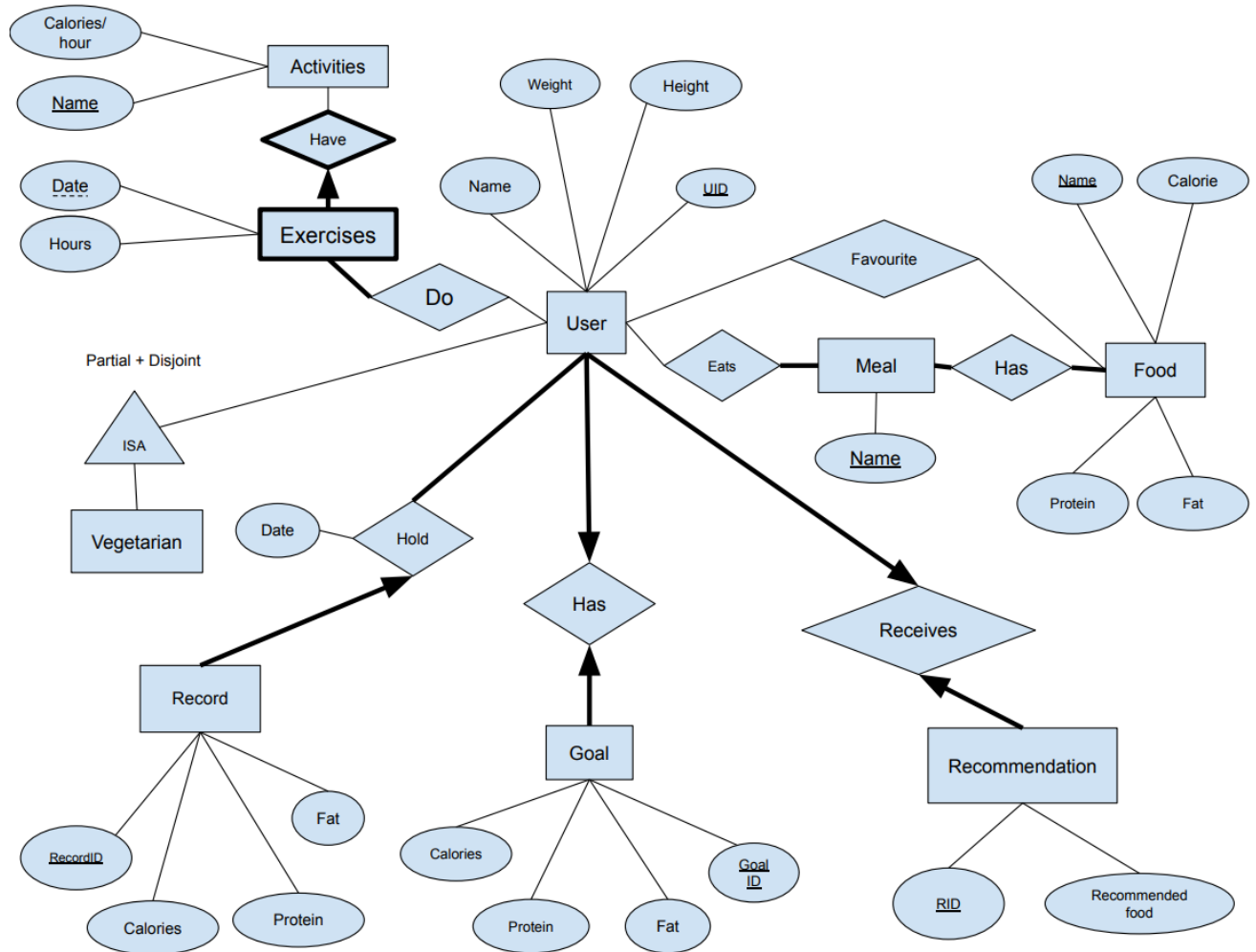
6. ER diagram description
- The number of entities: (7) User, Activities, Meal, Food, Record, Goal, Recommendation.
  +) User (<u>UID</u>, Name, Weight, Height)
  +) Meal (<u>Name</u>)
  +) Food (<u>Name</u>, Calorie, Protein, Fat)
  +) Record(<u>RecordID</u>, Calorie, Protein, Fat)
  +) Goal (<u>GoalID</u>, Calorie, Protein, Fat)
  +) Recommendation (<u>RID</u>, Recommended food)
  +) Activities (<u>Name</u>, Calories)
- Relationship, cardinality and participation constraint:
  +) Do (User and Exercise): many to many, all exercises must be in this relationship.
  +) Eats (User and Meal): many to many, every meal must be in the relationship.
  +) Has (Meal and Food): many to many, all meals and food must be in the relationship.
  +) Favorite (User and Food): many to many
  +) Has (User and Goal): one to one, all users and goals must be in the relationship.
  +) Receive (User and Recommendation): one to one, all users and recommendations must be in the relationship.
  +) Hold (User and Record): many to one, all users and records must be in the relationship.
- ISA hierarchy: "User" with 1 child: "Vegetarian" (we decided to categorize users based on dietary preference: vegetarian). This ISA hierarchy is partial and disjoint.
- Weak Entity/ Additional ISA relationship:
  "Exercises" is a weak entity of Activities.