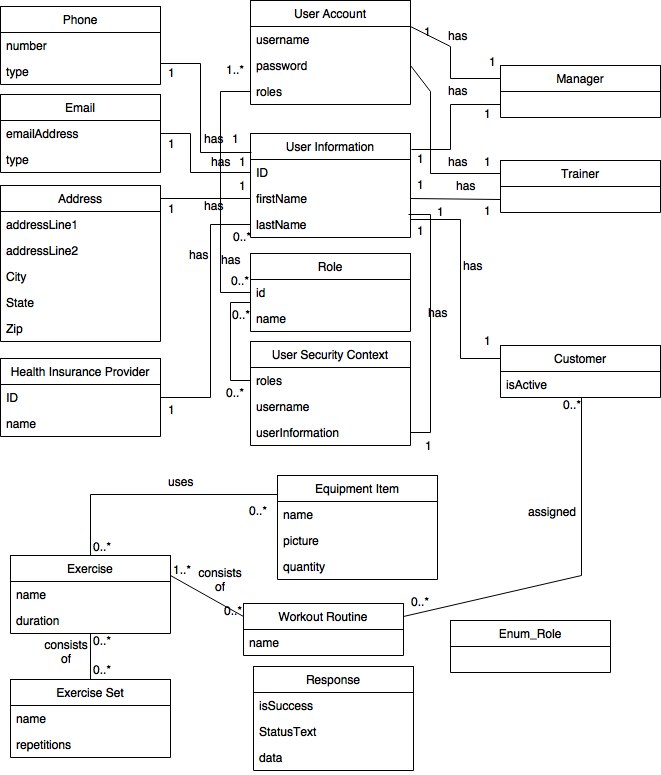
Domain Model

Note: Please see P2\_UML.png in GitHub if image resolution is not readable

Glossary

1. User Information

Object to hold all common user information data for Managers, Trainers, and Customers

* 1. Attributes:
     1. **ID**: Unique identifier
     2. **FirstName:** Attribute to store first name
     3. **LastName:** Attribute to store last name
  2. Associations:
     1. **Phone**: Association of a phone object to this object. This User Information object can only have one associated phone
     2. **Email**: Association of an email address object to this object. This User Information Object can only have one associated email
     3. **Address**: Association of an Address object to this object. This User Information object can only have one associated address
     4. **Health Insurance Provider**: Association of a Health Insurance Provider object to this object. This User Information object can only have one Health Insurance Provider

1. Phone

Object to represent a phone number within the application

* 1. Attributes:
     1. **Number**: Attribute to store the actual phone number
     2. **Type**: Attribute to store type of phone. E.g. Cell, Home, Work
  2. Associations:
     1. **User Information**: Each phone object is associated with exactly one User Information object

1. Email

Object to represent an email address within the application

* 1. Attributes:
     1. **EmailAddress**: Attribute to store the actual email address
     2. **Type**: Attribute to store the type of email. E.g. School, Work, Personal
  2. Associations:
     1. **User Information**: Each email object is associated with exactly one User Information object

1. Address

Object to represent an address within the application

* 1. Attributes:
     1. **addressLine1**: Attribute to store the first line of the address
     2. **addressLine2**: Attribute to store the second line of the address
     3. **City**: Attribute to store the City of the address
     4. **State**: Attribute to store the State of the address
     5. **Zip**: Attribute to store the zip code of the address
  2. Associations:
     1. **User Information**: Each addressobject is associated with exactly one User Information object

1. Health Insurance Provider

Object to represent a health insurance provider entity within the application

* 1. Attributes:
     1. **ID**: Attribute to store the unique identifier for the health insurance provider
     2. **Name**: Attribute to store the name of the health insurance provider
  2. Associations:
     1. **User Information**: Health insurance providers can be associated with many user information entities

1. Manager

Object to represent a manager entity within the application

* 1. Attributes: None
  2. Associations:
     1. **User Information**: Each manager entity is associated with exactly one User Information entity
     2. **User Account**: Each manager entity is associated with exactly one user account in order to log into the system

1. Trainer

Object to represent a trainer entity within the application

* 1. Attributes: None
  2. Associations:
     1. **User Information**: Each trainer entity is associated with exactly one User Information entity
     2. **User Account**: Each trainer entity is associated with exactly one user account in order to log into the system

1. User Account

Object to represent a user account entity within the application in order to allow a user to log into the application

* 1. Attributes:
     1. **Username**: Attribute to store the username to be used to into the application
     2. **Password**: Attribute to store the password to allow a user to log into the application
  2. Associations:
     1. **Manager**: User account must be associated with at most one manager entity and no trainer entity

OR

* + 1. **Trainer**: User account must be associated with at most one trainer entity and no manager entity

1. Customer

Object to represent a customer entity within the application

* 1. Attributes:
     1. **IsActive**: Attribute to denote if this customer is active / not active within the application
  2. Associations:
     1. **User Information**: Each customer entity is associated with exactly one User Information entity
     2. **Assigned Workout Routine**: Each customer entity is assigned 0 to many workout routines

1. Workout Routine

Object to represent a workout routine entity within the application

* 1. Attributes:
     1. **Name**: Attribute to store the name of the workout routine
  2. Associations:
     1. **Customer**: Each workout routine can be assigned to 0 to many customers
     2. **Exercise**: Each workout routine entity consists of 1 to many exercises

1. Exercise

Object to represent an exercise within the application

* 1. Attributes:
     1. **Name**: Attribute to store the name of the exercise
     2. **Duration**: Attribute to store the duration of the exercise
  2. Associations:
     1. **Exercise Set**: An exercise may consist of 0 to many exercise sets
     2. **Equipment Item**: An exercise may make us of 0 to many equipment item entities

1. Exercise Set

Object to represent an exercise set within the application.

* 1. Attributes:
     1. **Name**: Attribute to store the name of the exercise set
     2. **Repetitions**: Attribute to store the number of repetitions the exercise set requires
  2. Associations:
     1. **Exercise**: An exercise set may be used in 0 to many exercises

1. Equipment Item

Object to represent an item of equipment within the gym. E.g. treadmill, weight bench, barbells

* 1. Attributes:
     1. Name: Attribute to store the name of the equipment item
     2. Picture: Attribute to store a picture of the equipment item
     3. Quantity: Attribute to store the quantity of the equipment item within the gym
  2. Associations:
     1. Exercise: Each equipment item may be used in 0 to many exercises within the application

1. Roles

Object to represent a user’s role in the system. Many to many relationship with User Account

* 1. Attributes:
     1. Id: Role identifier
     2. Name: Name of the role
  2. Associations:
     1. User Account: Many roles to many user accounts

1. User Security Context

Object to represent a user’s security context in the system.

Attributes:

* + 1. roles: user roles
    2. username: User’s username
    3. user’s userInformation object
  1. Associations:
     1. Roles: 0 to many association with Roles object

1. User Security Context

Object to represent a user’s security context in the system.

Attributes:

* + 1. roles: user roles
    2. username: User’s username
    3. user’s userInformation object
  1. Associations:
     1. Roles: 0 to many association with Roles object

1. Response

Object to represent a response from a controller.

Attributes:

* + 1. isSuccess: denotes if action was successful
    2. StatusText: Text message to display to user
    3. data: Additional data to pass to UI for presentation
  1. Associations:
     1. None