
Architecture Diagram & Object Models

FH Mobile Application

Version 2.0

Prepared by

Omar Rivera
Andrew Poirier
Daven Amin
Rick Rejeleene

Overview

Within the FH mobile application there are three parts; the client, database and server. This high-level view is described in Figure 1. The client and the server are described in detail in Figure 2 and Table 1. The database design is described in detail using an ERD below in Figure 3.

High Level Diagram



Figure 1 – High Level Diagram for FH Mobile Application System

Class Diagram / Cardinalities

All of the relationships between Classes in Figure 2 are not shown for legibility. Table 1 specifies the relationship between each Class and the classes it has a relationship with. Similar relationships are grouped together.

<i>Object Name</i>	<i>Relationship /Cardinality</i>	<i>Relationship Object</i>	<i>Description</i>
Client Application	N to 1 Many to one In a fully functional system, there will be multiple instance of a client applications interacting with only one server application.	Server_Application	The role of the client application is to basically run the whole show of the application. It has the ability to log the user into the system and reset their password if they forgot it or lost it. Along with the log and user information the Client_Application is also used for creating and validating a workout. Along with those functions, the user can also get basic help from this main page if they have trouble logging in or changing their password.
	1 to N One to many. One Client Application will interact and store multiple users information retrieved from the server.	User_Information	
Server Application	1 to N One to many. The server Application will interact with multiple clients and the database simultaneously.	User_Session	Server_Application - this class acts as a facilitator between the client and the database. It contains methods to retrieve information from the database and return it in a format usable by the client. It also contains methods to add/update/delete users from the database, and to add an amount to a user's balance. Finally, it contains logic (called periodically by the system) to query the database for user activities which

			were not validated and occurred in the past - and to deduct accordingly from the user's balance.
User Information	1 to 1 One User Information contains a single primitives for userID, UserDOB, UserHeight, UserWeight and user Passwords.	(Integer) userDOB, userHeight, userWeight, (String) userID, password	The user information class in a few words would be the sign up page for a new user that wants to use the application. It has a userID field along with their new password, date of birth, height, weight, workouts, and if the user has the ability to validate someone else workout.
	1 to N One user Information contains many workouts created by the user and any user validated by the user for that account.	Workout, User_Information	
Workout	1 to 1 One Workout Information contains one single start date and number of weeks	(Integer) startdate, numberOfWeeks	The workout class basically is the class that is used to create a new workout schedule for the current user that is logged in. The workout class will get a start date, the number of weeks the workout should run, and the activities involved in a workout.
	1 to N One workout contains one or more activities	Activity	
Activity	1 to 1 One activity only contains one , quantity, name,description, numberPerWeek	(Integer) quantity, numberPerWeek (String) name, description	In the activity class, there are a bunch of activities pre-configured in the system like swimming, weight lifting, running, etc. This class will link the activities to the users workout profile with the name of the activity, the number per week, the quantity of the activity(reps, laps, distance, etc.) and the description and performance notes from the workout.

Credentials	1 to 1 The Credentials class contains the userID and the password of a single user.	(String) userID, password	This class is the basic log in class for the system. It can get the user id and the user's password and will validate to see if they are a match and will log them into the system if they are a match. This class will get information from the Client_Application Class.
User Session	1 to 1 The User session only contains the user information and the session ID of the current user.	(Integer) sessionID, User_Information	This class is used for the system to log each session the user logs in. It will create a session ID and associate this session ID along with the user name and information for the login.

Table 1 – Object Name Description Table

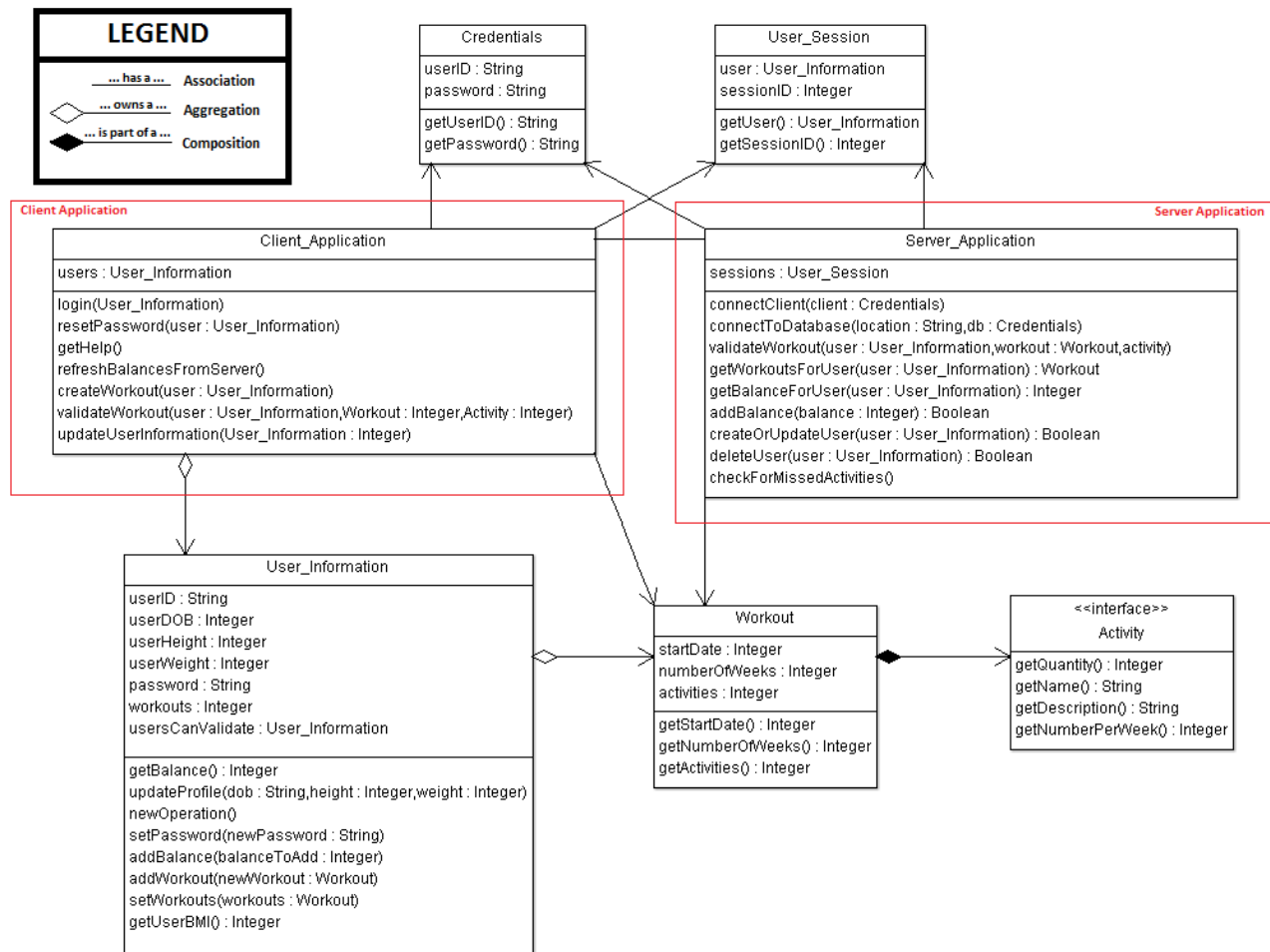


Figure 2 – Class Diagram for Client and Server Applications

Database Entity Relationship Diagram

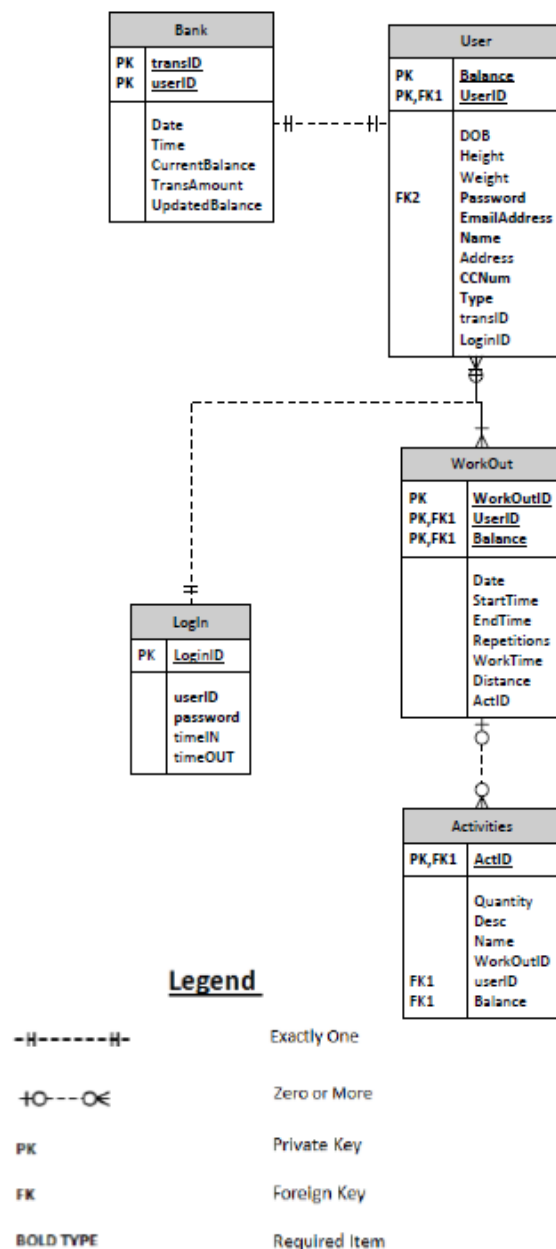


Figure 3 – Database Design

In Figure 3 there is a simulation of how the database for the Fitness Health application will be distributed. The database has five major tables in its database. The high level table where all of the information is coming from is the User table. This table will be used to store all of the information that will be needed for the application to retrieve workout, bank, login and to calculate information. The way that the system is able to do it is through the userID field. This field is a public key, foreign key and a requirement for any data alteration. This userID will be used throughout the system as a way to store and retrieve information for a particular person.