**Table 3.1 – Server_Application**

| | |
|---|---|
| **Name** | *Server* |
| **Base Class(es)** | |
| **Purpose** | *The server is used to connect to the database and store,delete or update information for the user, workouts, and bank in the application.* |
| **States** | *Change/Update, Validate, Create, Delete*<br>*Active or Inactive* |
| **Constructors** | *Default: makes server ready for data information and processing* |
| **Operators Mutators** | *validateWorkout(user:user_Information, workout:Workout, activity)*<br>*updateBalance(balance:Integer)*<br>*createOrUpdateUser(user:User_Information)*<br>*deleteUser(user:User_Information)*<br>*checkForMissedActivities()*<br>*changePassword(User_Session, :string)*<br>*requestUsersInformation()*<br>*addWorkout(user:User_Information)* |
| **Accessors** | *connectToDatabase(location: String,db:Credentials)*<br>*connectClient(Credentials)*<br>*getWorkoutsForUser(user:User_Information)*<br>*getBalanceForUser(user:User_Information)*<br>*getUsersCanValidate(user:User_Information)* |
| **Fields** | *User_Session* |

**Table 4.1 – Server_Application:connectClient**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:connectClient(client:Credentials )* |
| **Purpose** | *To connect the client to a user session in the server* |
| **Receives** | *The clients credentials to authenticate the server* |
| **Returns** | *TRUE if the credentials were valid*<br>*FALSE if the credentials were invalid* |
| **Remarks** | *The operation may fail if the user forgot to log out last time.* |

**Table 4.2 – Server_Application:connectToDatabase**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:connectToDatabase(location:String,db:Credentials)* |
| **Purpose** | *To connect the user to the database to access the user's saved information* |
| **Receives** | *The database location and database name as a string object*<br>*The database credentials from the server* |
| **Returns** | *TRUE if the user and server connect to the correct database with the correct credentials*<br>*FALSE if the user and server do not connect correctly or if the the credentials are invalid* |
| **Remarks** | *The operation may fail if the database cannot be found or the connection is interrupted at anytime with the current connection* |

**Table 4.3 – Server_Application:validateWorkout**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:validateWorkout(user:User_Information, workout:Workout, workout:Activity)* |
| **Purpose** | *To connect the user to the database to access the user's workout history and information* |
| **Receives** | *The server receives the user's information along with the workout and the activity associated with that workout.* |
| **Returns** | *TRUE if the facilitator was able to validate the workout correctly*<br>*FALSE if the facilitator was unable to find the saved workout or if there are no workouts to validate* |
| **Remarks** | *The operation might fail if the user forgot to save the work out or if the workout does not have any activities associated with it.* |

**Table 4.4 – Server_Application:getWorkoutsForUser**

| | |
|---|---|
| **Prototype** | *WORKOUTS:Server_Application:getWorkoutsForUser(user:userInformation)* |
| **Purpose** | *This will connect the server to the databse to receive the user's list of workouts from the database.* |
| **Receives** | *The server receives the user's information to find the list of workouts.* |
| **Returns** | *WORKOUTS if the the list of workouts for the user is found in the database*<br>*NULL if the user does not have any current workouts* |
| **Remarks** | *This operation might fail if the user is not working out for a few weeks or if it the first time and there are no workouts created.* |

**Table 4.5 – Server_Application:getBalanceForUser**

| | |
|---|---|
| **Prototype** | *Int:Server_Application:getBalanceForUser(user:User_Information)* |
| **Purpose** | *This will connect the server to the bank database to access the user's current balance.* |
| **Receives** | *The server receives the user's information to find the user in the bank database* |
| **Returns** | *INTEGER if the user's balance was found in the bank database*<br>*NULL if the user currently does not have any funds available.* |
| **Remarks** | *This operation might fail if the user has not set up bank information with a starting balance.* |

**Table 4.6 – Server_Application:updateBalance**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:updateBalance(balance:Integer)* |
| **Purpose** | *This will connect to the server and will update the users balance off of the current balance that was received* |
| **Receives** | *The server receives the current balance for the user connected* |
| **Returns** | *TRUE if the balance was updated successfully*<br>*FALSE if the balance was not updated successfully* |
| **Remarks** | *This operation might fail if the user does not have a balance or the user does not have a penalty/reward amount set in the system* |

**Table 4.7 – Server_Application:createOrUpdateUser**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:createOrUpdateUser(user:User_Information)* |
| **Purpose** | *This will allow the user to update or create an account on the server for the application.* |
| **Receives** | *The server recieves the current user's information if they are updating. And it will receive new user information if a user is creating an account* |
| **Returns** | *TRUE if the user was created correctly or the information was updated correctly*<br>*FALSE if the user did not supply a required field when creating an account or the user does not update information with the correct information designated for that field* |
| **Remarks** | *This operation might not work if a user tries to create a user account with the same name that is already on the server.* |

**Table 4.8 – Server_Application:deleteUser**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:deleteUser(user:User_Information)* |
| **Purpose** | *This will allow the user to remove their account and information from the server* |
| **Receives** | *The user's information that is store on the server* |
| **Returns** | *TRUE if the user was deleted successfully*<br>*FALSE If the user was not deleted successfully.* |
| **Remarks** | *The operation will fail if the user tries to delete a user that does not exist on the server.* |

**Table 4.9 – Server_Application:checkForMissedActivities**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:checkForMissedActivities()* |
| **Purpose** | *This will allow the user to check to see if they missed activities while performing their workout.* |
| **Receives** | *The server will receive the users information to check for missed activities.* |
| **Returns** | *TRUE if the user has missed any activities*<br>*FALSE if the user has completed all of their activities for the workout* |
| **Remarks** | *The operation will fail if the user does not have any activities listed under a workout* |

**Table 4.10 – Server_Application:changePassword**

| | |
|---|---|
| **Prototype** | *Bool:Server_Application:changePassword(user:User_Information ;string)* |
| **Purpose** | *This will allow the user to change the current password saved and used on the server* |
| **Receives** | *The server will receive the user's session information and a sting containing the new password* |
| **Returns** | *TRUE if the password was changed successfully*<br>*FALSE if the user could not update the password sucessfully(length was not correct or did not pass password specifications)* |
| **Remarks** | *The operation might fail if the user log outs before the success or fail statement is sent back to the user.* |

**Table 4.11 – Server_Application:requestUsersInformation**

| | |
|---|---|
| **Prototype** | *UserInformation:Server_Application:requestUsersInformation()* |
| **Purpose** | *This will allow the server to return all users information* |
| **Receives** | *All of the users that are currently on the server* |
| **Returns** | *TRUE if the information was passed correctly*<br>*FALSE if the user does not exist or the information was not passed correctly* |
| **Remarks** | *The operation might fail if there are no users on the server.* |

**Table 4.12 – Server_Application:addWorkout**

| | |
|---|---|
| **Prototype** | *Bool:UserInformation:Server_Application:addWorkout(user:User_Information)* |
| **Purpose** | *This allows the user to add a workout to their account* |
| **Receives** | *The user's information* |
| **Returns** | *TRUE if the workout was added*<br>*FALSE if the workout could not be added* |
| **Remarks** | *The operation might fail if the user tries to add a workout with an existing workout name* |

**Table 4.13 – Server_Application:getUsersCanValidate**

| | |
|---|---|
| **Prototype** | *Bool:UserInformation:Server_Application:getUsersCanValidateWorkout(user:User_Information)* |
| **Purpose** | *Allows the user to set the users that can validate their workout.* |
| **Receives** | *The user's information* |
| **Returns** | *List of users that can validate workout* |
| **Remarks** | *This operation might fail if there are now users listed to validate a workout* |