table user

user\_id(int)(pk)

username(string)

picture(string)

email(string)

password(string)

user\_status(boolean)

last\_login(long)

reservation\_status(int 0 1 2) //0:on time 1:ivertime 2:waiting

account(int)

gender(string)

height(int)

weight(int)

total training hours(int)

Table record

Record\_id(int)(pk)

user\_id(fk->user pk)

month(pk)

activity\_name(string pk) Either user\_id or activity\_name deleted, the corresponding row should be deleted.

activity\_length(int)

table employee

employee\_id(int pk)

email(string)

employee\_name(string)

user\_status(boolean)

position(string)

table manager

manager\_id(int pk)

email(int)

manager\_name(string)

position(string)

table reservation

reservation\_id(int pk)

user\_id(int)(fk->user)

activity\_id(int)

venue\_id(int)

reservation\_start\_time(long)

reservation\_end\_time(long)

reservation\_make\_time(long)

table activity

activity\_id(int)(pk)

activity\_name(String)

max\_venuenumber(int)

current\_venuenumber((int)

current\_status(boolean)

Location(String)

table venue

venue\_id(int)(pk)

activity\_id(int)(fk->activity)

available(Boolean)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | URL | Parameter | Expected behavior | Return value |
| 1 | login | [ user\_email(String), password(String) ] | email to find the user example and compare the input password with the password in user table | Redirect to url:/user |
| 2 | create\_account | [ user\_name(String), position(String), password(String), password\_confirm(String), email(String) ] | Create a new ID for the new user, then add a new example in user table  If it choose the position as employee, the invitation code should match the invitation code as database, then add a new example to employee data set otherwise reject | Redirect to url:/user |
| 3 | del\_user | [ user\_id(Int) ] | Delete an entry in user table and all its relevant data | None |
| 4 | sort\_user | [ type(String: “by\_time”, “default”) ] | Select all instance in table and sort them based on the date of user’s user\_last\_login time in order from early to late | Return a sorted table list |
| 5 | search\_user | [ user\_input(string) ] | Search users’ names and emails and publish the instances with match feature | Return a table list |
| 6 | edit\_reservation | [ reservation\_id(int), ] |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | URL | Parameter | Expected behavior | Return value |
| 10. | edit\_user\_account | [userID(int),  reservation\_status(int)  amount(int)  ] | Change the factors (reservation status, amount) of User with Corresponding ID | Return an instance |
| 1. | add\_sports | [activity\_name (string), max\_venuenumber(int)  location(string)  ] | Create a new instance current\_status is free. current\_venuenumber would be 0. Set the location. Then add max\_venuenumber gyms with special ID and Free venues\_condition into the venues\_list. Then create a activity\_id for it. Add this instance to data set and table. | Return an instance |
| 4. | del\_sports | [None] | Del an instance in data set and table | None |
| 5. | bool\_sports | [ activity\_id(int)] | Change the sports\_ status into opposite boolean status | Return Instance |
| 10. | edit\_sports | [ activity\_id(int),  activity\_name(string)  max\_venuenumber(int)  location(string)  ] | Change the factors (activity\_name, max\_venuenumber, location) of sports with Corresponding ID | Return an instance |
| 11. | search\_sports | [user input(string)] | Search activity’ names and publish the instances with match feature | Return a table list |
| 12. | add\_venues | [ activity\_id(int),number(int)] | Add a number of examples to the venue table and the avaiable is Free | Return a instance |
|  | del\_venues | [ venue\_id (int)] | Del the venues with corresponding ID | None |
|  |  |  |  |  |

All functions in the table below need employee validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | URL | Parameter | Expected behavior | Return value |
| 4. | del\_employee | [None] | Del an instance in data set and table | None |
| 10. | edit\_employee | [ employee\_id(int)  email(string)  employee\_name(string)  ] | Change the factors (name, email | Return an instance |
| 11. | search\_employee | [user input(string)] | Search employees’ names and publish the instances with match feature | Return a table list |

Function for users:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | URL | Parameter | Expected behavior | Return value |
| 1. | get\_max\_venues | activity\_id(int) | Get the maximum number venues of the specific sport(can be directly find in database) | Max\_venues |
| 2. | get\_current\_venues | activity\_id(int), | Get the current number of occupied venues of the specific sport(can be directly find in database) | Current\_venues |
| 3 | check\_venue | venue\_id(int) reservatuin\_start\_time(long), reservation\_end\_time(long) | Search the reservation table by venue\_id, if the reservatuin\_start\_time and reservation\_end\_time match the parameter given, then return false,otherwise return true. | True or false |
| 4 | add\_reservation | activity\_id(int), user\_id(int), activity\_name(string), reservatuin\_start\_time(long), reservation\_end\_time(long), venue\_id(int) | add a instance to reservation table with a reservation\_id.  all attributes are given |  |
| 5 | Delete\_reservartion | reservation\_id(int) | delete from the reservation table by reservation\_id.  call this function when a reservation’s end\_time is on or other cases |  |
| 5.1 | search\_reservation | user\_id(int) | Search the reservation table by user\_id, return a list of reservation\_id | a list of reservation\_id of the specific user. |
| 5.2 | show\_reservation | reservation\_id | Search the reservation table by user\_id, and return other attributes | return: user\_id, activity\_name, reservation\_start\_time, reservation\_end\_time, venue\_id |
| 6 | create\_record | user\_id(int) | Add data to record table:  user\_id,the current month. For every activity,give a row.  Set the activity\_length as 0.  Call this function after create\_account function or the month is changed. |  |
| 7 | add\_record\_type | activivity\_id(int) | Add data to record table:  For every user,give a row.  The current month as month, activity\_id,set the activity\_length as 0. Call this function after add\_sports function. |  |
| 8 | edit\_record | user\_id(int),time(int),activity\_id(int) | add time to the attribute activity\_length in table record, this can be searched by user\_id and current month and activity\_id |  |
| 9 | get\_record | user\_id(int),month(int),activity\_id(int) | get the activity\_length by searching the record table using the three parameters. | activity\_length |
| 10 | get\_height\_and\_weight | user\_id(int) | get the height and weight from user table by searching using the user\_id | height,weight |
| 11 | edit\_profile | user\_id(int),gender(string),email(string)  username(string),height(int)weight(int)  picture(string) | search the table user by user\_id, and update attributes | update the username,picture gender,email,height and weight |
| 12 | get\_account | user\_id(int) | search the table user by user\_id, and return the account of user | account |
| 13 | add\_account | user\_id(int),money(int) | search the table user by user\_id,and add the account of user by money. | add money to the account attribute of name: |
| 14 | get\_reservation | reservation\_id(int) | search the table reservation by reservation\_id | Return these values:  Date:the day of reservation\_start\_time  Username:get the user\_id and use it to search user table  activity\_name:in reservation table.  venue\_id:in reservation table  during: the difference between start\_time and end\_time, both in reservation table. |
| 15 | log\_out | user\_id | Log out |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |