Table<user>

|  |  |
| --- | --- |
| Column | Data Type |
| user\_id | INT |
| username | VARCHAR(64) |
| picture | VARCHAR(64) |
| email | VARCHAR(64) |
| password | VARCHAR(64) |
| user\_status | BOOLEAN |
| last\_login | BIGINT |
| reservation\_status | TINYINT(0: ON\_TIME; 1: OVERDUE; 2: WAITING) |
| amount | INT |
| gender | VARCHAR(16) |
| height | INT |
| weight | INT |
| total\_training\_hours | INT |

TODO: Table record

|  |  |
| --- | --- |
| Column | Data Type |
| record\_id | INT |
| user\_id | INT |
| month | TINYINT |
| venue\_id | INT |
| activity\_length | INT |

Table employee

|  |  |
| --- | --- |
| Column | Data Type |
| employee\_id | INT |
| email | VARCHAR(64) |
| employee\_name | VARCHAR(64) |
| position | VARCHAR(64) |

Table manager

|  |  |
| --- | --- |
| Column | Data Type |
| manager\_id | INT |
| email | VARCHAR(64) |
| manager\_name | VARCHAR(64) |
| position | VARCHAR(64) |

Table reservation

|  |  |
| --- | --- |
| Column | Data Type |
| reservation\_id | INT |
| user\_id | INT |
| venue\_id | INT |
| reservation\_start\_time | BIGINT |
| reservation\_end\_time | BIGINT |
| reservation\_make\_time | BIGINT |

Table activity

|  |  |
| --- | --- |
| Column | Data Type |
| activity\_id | INT |
| activity\_name | VARCHAR(64) |
| max\_venue\_number | INT |
| current\_venue\_number | INT |
| current\_status | BOOLEAN |
| location | VARCHAR(256) |

Table venue

|  |  |
| --- | --- |
| Column | Data Type |
| venue\_id | INT |
| activity\_id | INT |
|  |  |

Functions for management

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Parameter List | Expected Behavior | Response |
| login | user\_email : String  password : String | Try to login to the system with given email(as account) and password.  TODO: behavior if email is not found or password is incorrect | Redirect to url:/user |
| create\_account | user\_name : String  email : String  password : String  password\_confirm : String  position : String | Create a new account with given email, password, user name and position.  TODO: behavior if email is already exist or the password is different from the confirmed password | Redirect to url:/user |
| del\_user | user\_id : int | Delete an entry in user table and all its relevant data.  TODO: behavior if user of given id is not found | None |
| sort\_user | type : String(“by\_time”, “default”) | Sort the user list by last login time if parameter type is “by\_time” and return sorted list. Otherwise just return the user list in native order. | A list of users |
| search\_user | user\_input : String | Find all users with their name or email contains the given string. | A list of users |
| edit\_reservation | reservation\_id : int | TODO: defination |  |
| edit\_user\_account | user\_id : int  reservation\_status : int  amount : int | Update user’s relevant states.  TODO: behavior if user of given id is not found | WARN: Return an instance |
| add\_sport | activity\_name : String max\_venue\_number : int  location : String | Create a new activity with given name and location. Create venues for this activity of required number. | WARN: Return an instance |
| del\_sport | activity\_id : int | Delete required activity and all its relevant data  TODO: behavior if activity of given id is not found | None |
| bool\_sport | activity\_id : int | Flip the status of required activity  TODO: behavior if activity of given id is not found | WARN: Return Instance |
| edit\_sport | activity\_id : int  activity\_name : String  max\_venue\_number : int  location : String | Update attributes of given activity  TODO: behavior if activity of given id is not found | WARN: Return an instance |
| search\_sport | user\_input : String | Find all activities with its name contains the given string. | A list of sports |
| add\_venue | activity\_id : int | Create a new venue that belongs to the given activity.  TODO: behavior if activity of given id is not found | WARN: Return an instance |
| del\_venue | venue\_id : int | Delete the given venue and move it to garbage venue list.  TODO: behavior if venue of given id is not found | None |

Functions for employee validation

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Parameter | Expected behavior | Return value |
| del\_employee | employee\_id : int | Delete the given employee and all its relevant data. | None |
| edit\_employee | employee\_id : int  email : String  employee\_name : String | Update attributes of the given employee.  TODO: behavior if employee of given id is not found | WARN: Return an instance |
| search\_employee | user\_input : String | Find all employees with their name contains the given string. | A list of employees |

Functions for users

|  |  |  |  |
| --- | --- | --- | --- |
| URL | Parameter | Expected behavior | Return value |
| get\_max\_venues | activity\_id(int) | Get the maximum number venues of the specific sport(can be directly find in database) | Max\_venues |
| 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 |
| 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 |
| 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 |  |
| 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 |  |
| 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. |
| 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 |
| 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. |  |
| 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. |  |
| 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 |  |
| 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 |
| get\_height\_and\_weight | user\_id(int) | get the height and weight from user table by searching using the user\_id | height,weight |
| 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 |
| get\_account | user\_id(int) | search the table user by user\_id, and return the account of user | account |
| 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: |
| 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. |
| log\_out | user\_id | Log out |  |