# Irithim Task

## First database tables:

### owner:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | User\_name | Email | Password | Role |

### Unconfirmed\_Users:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | User\_name | Email | Password | Role |

### Users:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | User\_name | Email | Password | Role |

### Lists:

|  |  |  |
| --- | --- | --- |
| ID | Title | Created\_by |

### User lists:

|  |  |  |
| --- | --- | --- |
| ID | User\_Id | List\_Id |

### Cards:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Title | Description | Comments\_count | List\_id | Created\_by |

### Comments:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Content | Card\_id | Replies\_count | Created\_by |

### Replies:

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Content | Comment\_id | Created\_by |

## Todos:

### Users:

1. Authentication: Sign-up / Login.
2. Get all users paginated.
3. Get user lists/cards/comments/replies.

### Lists:

1. Get all lists paginated.
2. Get a list with cards in it.
3. Create/update.
4. Delete (will delete all cards/comments/replies in it).
5. Search list by title.

### User lists:

1. Assign to list (create).
2. Unassign to list (delete).

### Cards:

1. Get all cards ordered by comments count paginated.
2. Get all cards for a certain list ordered by comments count.
3. Get a card with the first three comments.
4. Create/update.
5. Delete (will delete all comments/replies in it).
6. Search cards by title/description.

### Comments:

1. Get all comments for a card paginated.
2. Create/update.
3. Delete (will delete all replies in it).

### Replies:

1. Get all replies on a certain comment paginated.
2. Create/update/delete.

## Authorization

### Admin:

* List:
  + Can create lists.
  + Can remove/update his own lists,
  + Can read all lists.
* Card:
  + Can Create Cards on any list.
  + Can remove/update his own cards and any other card in his own lists.
  + can read all cards.
* Comments:
  + Can update/remove his own comments and any other comment on his own lists.
  + Can create a comment anywhere, can read all comments.
* Member: can assign/unassign members on his own lists.

### Member:

* List:
  + can access only lists assigned to him.
* Cards:
  + Can create a card on his lists.
  + can read all cards on his lists.
  + can update/remove his own cards
* Comments:
  + can create comment on any card.
  + can reply to comments in his lists.
  + can read all comments on his lists.
  + can remove his own comments including other people’s replies on it.
  + can update his comments/replies.

### Functions:

1. set password:
   1. will take user password as an argument.
   2. Will return hashed password
2. check password:
   1. will take hashed password and entered password as an argument.
   2. Will return true or false
3. get permissions:
   * will take user\_id\entity\_id as an argument.
   * Will return access token (jwt)
4. Check permissions:
   * will take entity\_id as an argument.
   * Will return true or false

### Assumptions:

1. Role: Boolean if any value received then it is admin(true) if not then member(false)
2. Owner:
   * get all unconfirmed registered users from unconfirmed users table.
   * confirm an unconfirmed user to be added to the users database and deleted from the unconfirmed users table.
   * delete an unconfirmed user which will be deleted from the unconfirmed users table.
   * change role of a confirmed user from member to admin or vice versa.