

# **DESIGN DOCUMENT FOR Flashcard Software**

**\*\*\***

**For course: Software Engineering**

**\*\*\* Group 1:**

- Nguyễn Việt Anh
- Bùi Thị Phương Anh
- Trần Thị Thoa
- Phạm Nguyễn Khánh Phong
- Trần Vũ Hải

## **1. Requirement Specification :**

Flash card is a method that combining English word and a corresponding image to help people remembering words more easily and having fun with learning English.

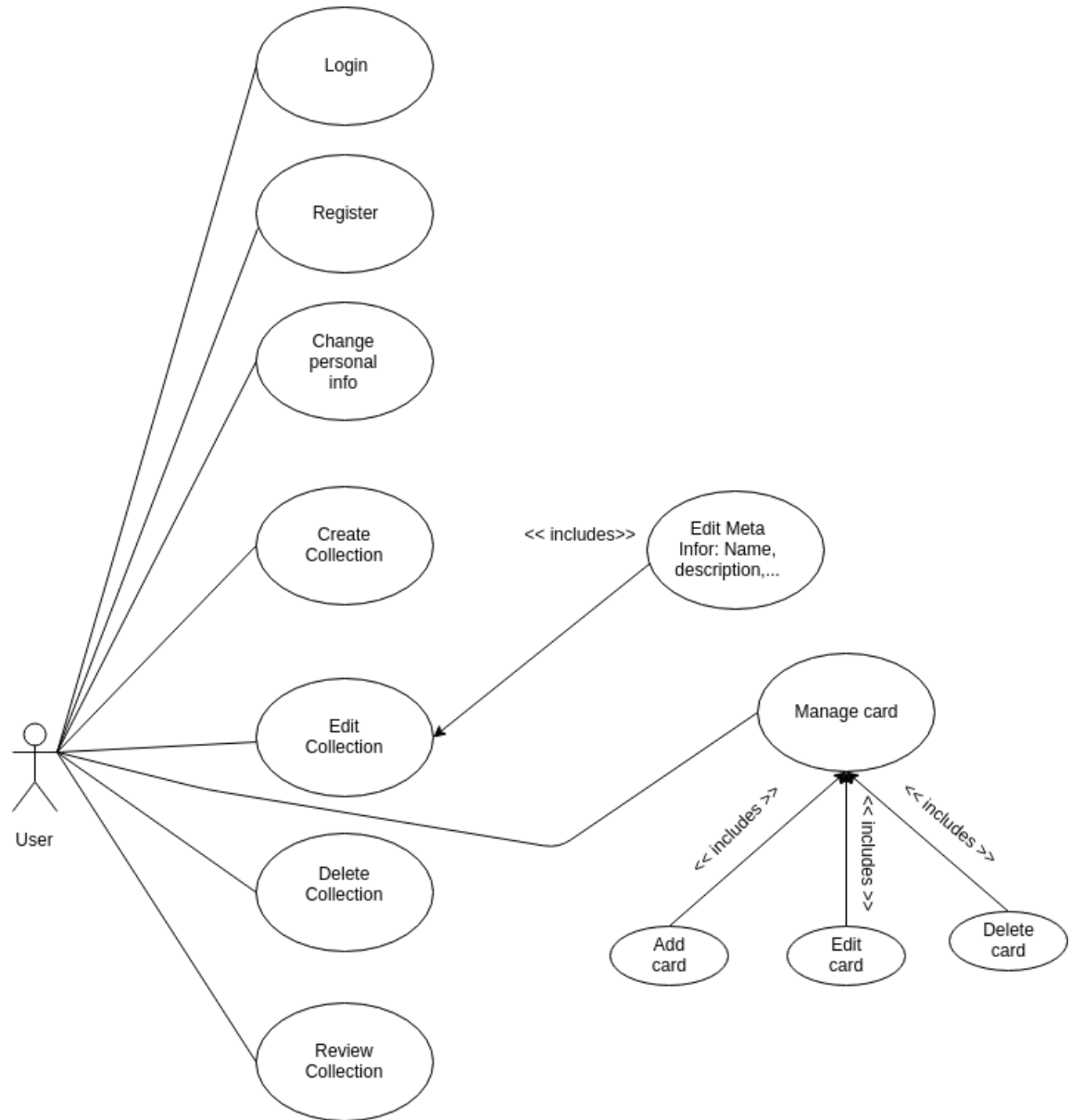
### **Main feature:**

- + Visualize flashcards, Flashcard collections on the computer screen so that people can use to learning without buying their paper collections.
- + Can be used in multiple platform (firstly on web and PC application) to learn every where.
- + Help people in using available Flashcard collections, creating/editing their own collections.

## 2. Design diagram

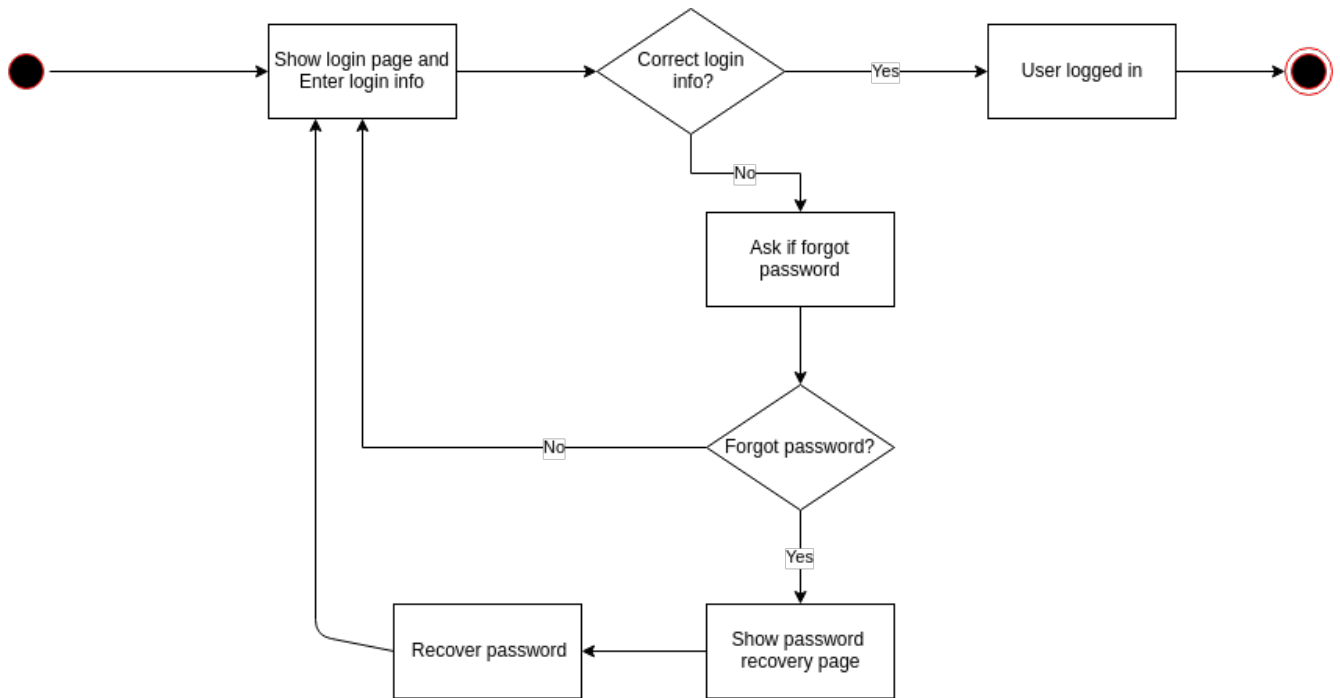
### 2.1. Use-case diagram

USECASE

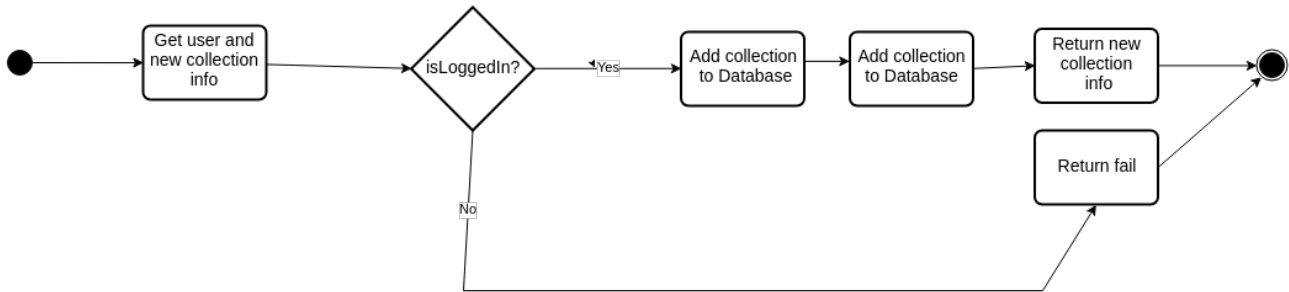


## 2.2. Activity diagrams

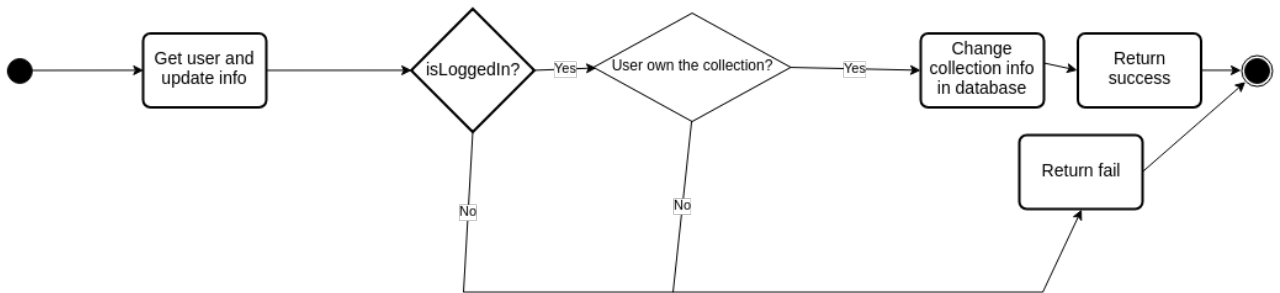
### LOGIN



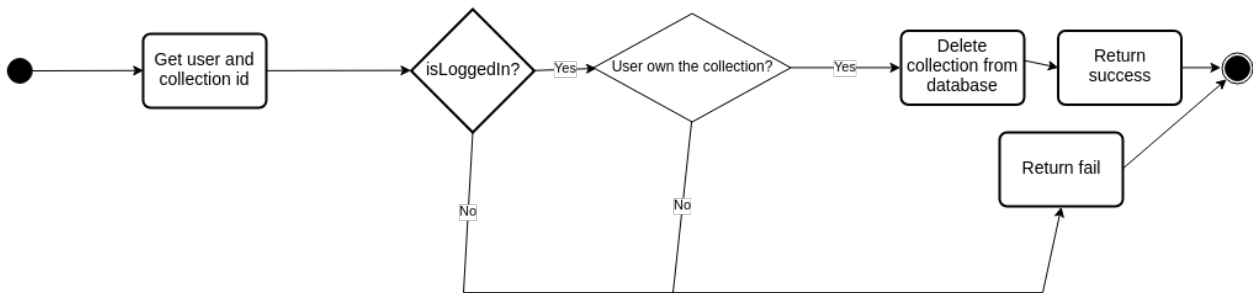
## CREATE COLLECTION



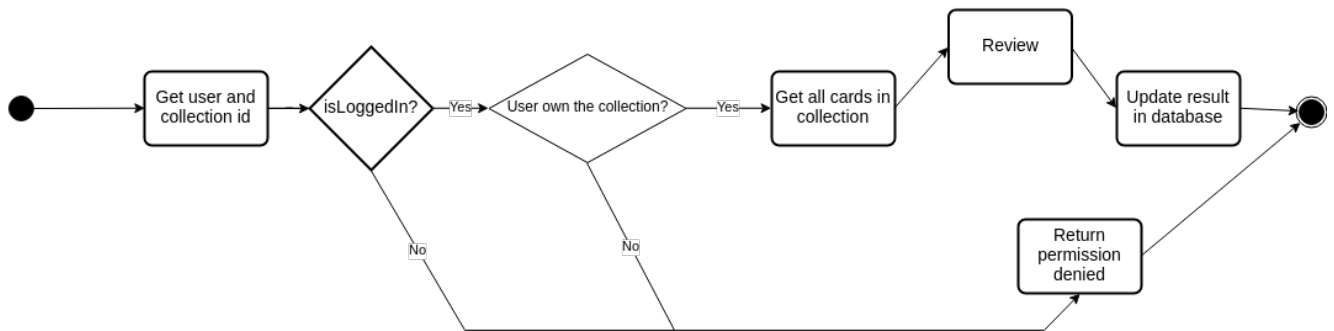
## EDIT COLLECTION



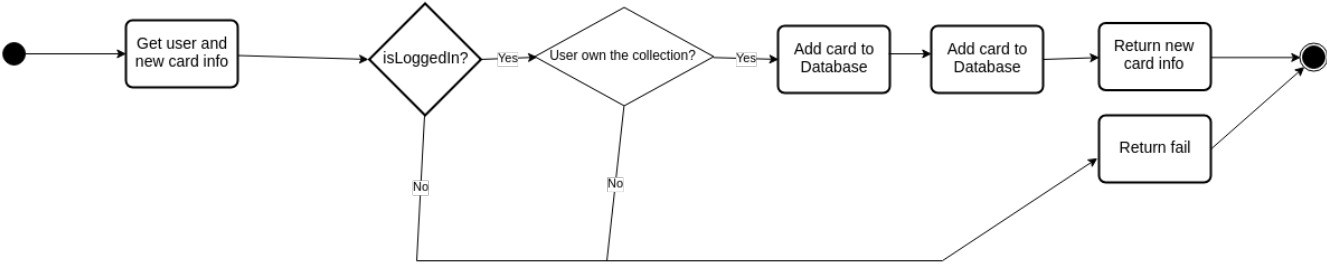
## DELETE COLLECTION



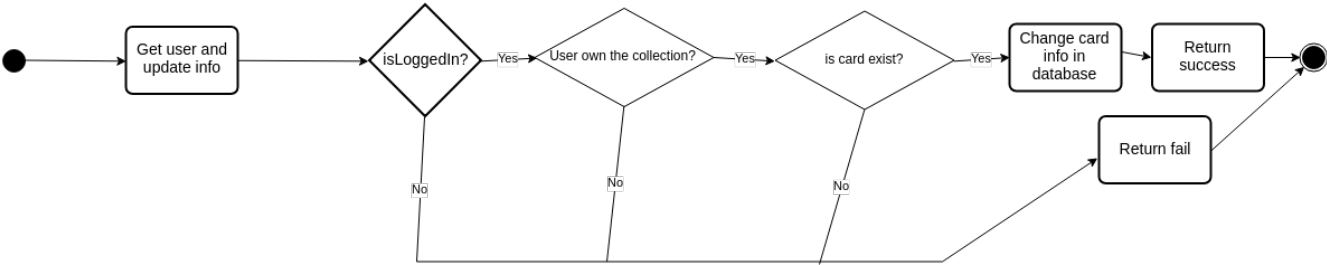
## REVIEW COLLECTION



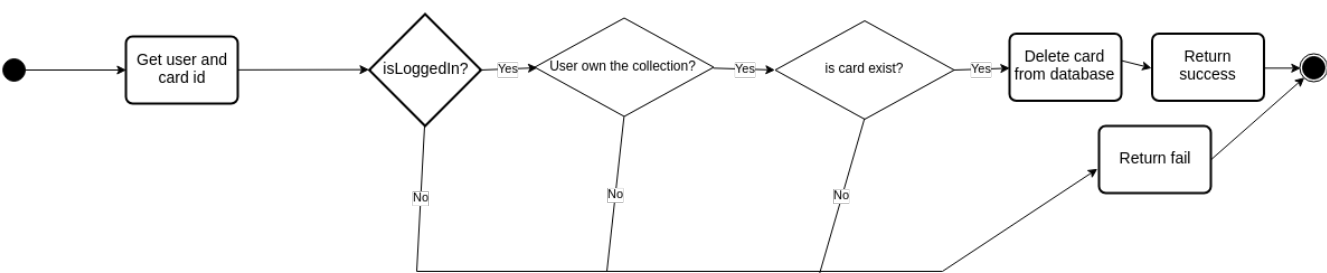
CREATE CARD



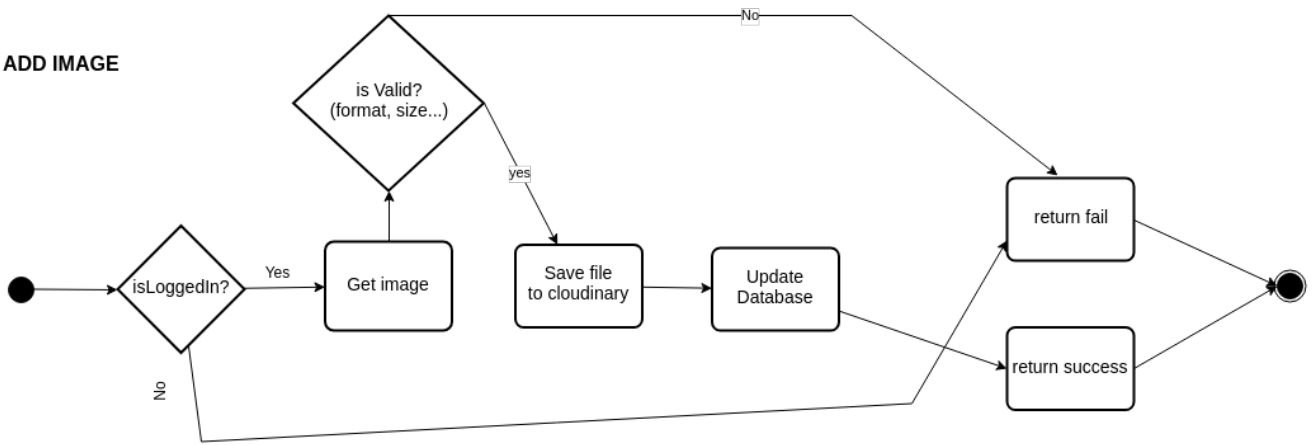
EDIT CARD



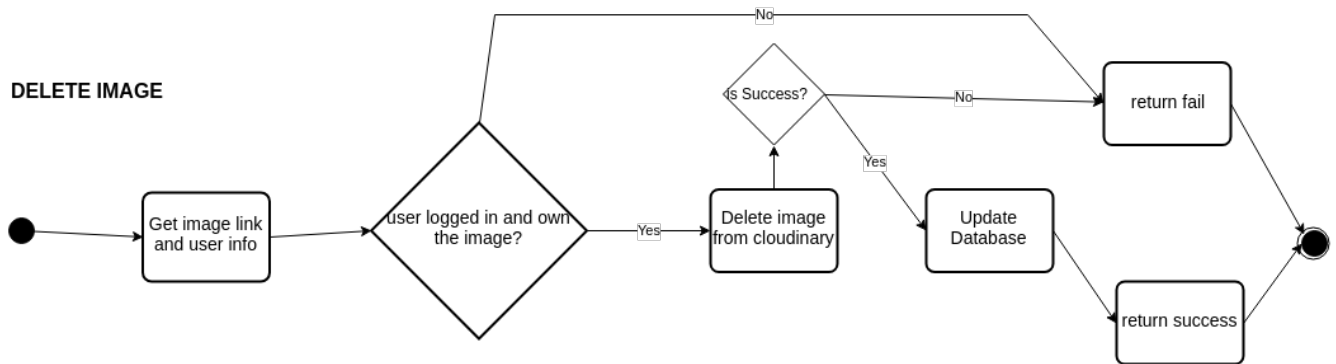
DELETE CARD



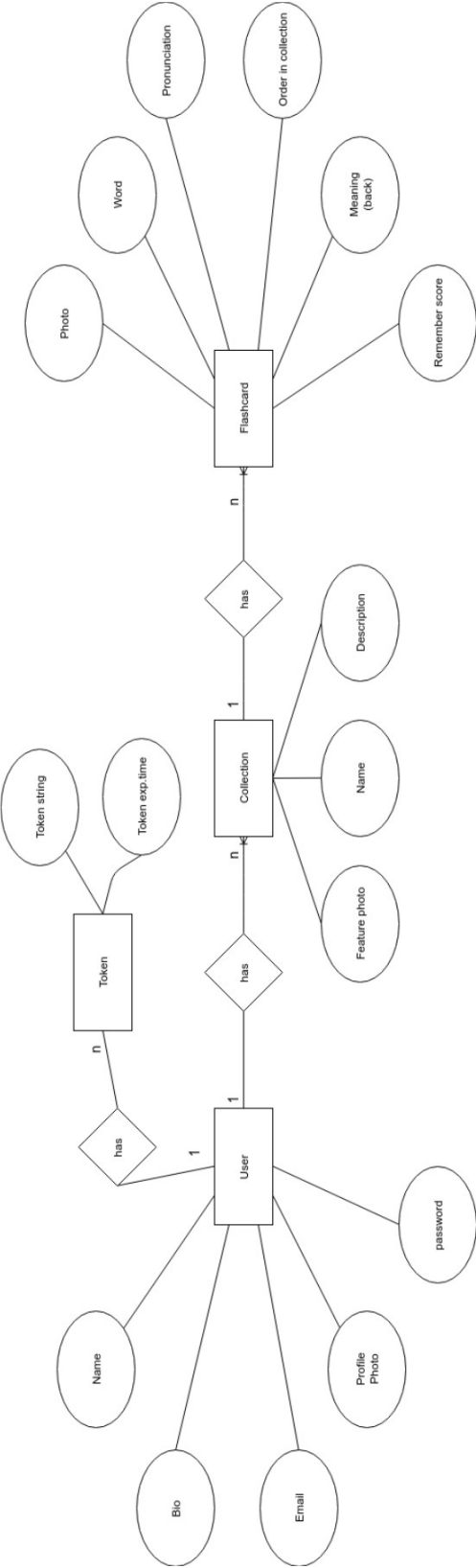
## ADD IMAGE



## DELETE IMAGE



2.3. Entity diagram





### 3. Web APIs

The APIs in this part are used in communication between server (Node.js) and clients (Web frontend and PC application).

#### 3.1. Dictionary

##### - E-V Dictionary Lookup:

**GET:** /api/dictionary/ev?lookup=<word>

#### 3.2. User management

##### - User Register

**POST:** /api/user/register

**params:**

```
{  
    'username': '<username>',  
    'password': '<password>',  
    'email': '<password>',  
    'fullname': '<password>'
```

**success:** return {'success': true}

**fail:** return {'success': false, 'message': '<why fail?>'}

##### - User Login

**POST:** /api/user/login

**params:**

```
{  
    'username': '<username>',  
    'password': '<password>'
```

}

**success:** return {'success': true, 'token': '<token>'}

**fail:** return {'success': false}

## - User Logout

**POST:** /api/user/logout

**params:** {'username': '<username>', 'token': '<token>'}

**success:** return {'success': true}

**fail:** return {'success': false}

## - Check login state

**POST:** /api/user/islogin

**params:** {'username': '<username>', 'token': '<token>'}

**success:** return {'success': true, 'isLoggedIn': true}

**fail:** return {'success': false}

or: return {'success': true, 'isLoggedIn': false}

## - Update user info

**POST:** /api/user/updateinfo

**params:** {'username': '<username>', 'token': '<token>',

'oldPassword': '<oldPassword>'

'newUserInfo': {

    'email' : '<email>',

    'bio' : '<bio>',

    'fullname' : '<fullname>',

    'newPassword' : '<newPassword>',

}

}

=> 'oldPassword', 'newPassword' are used when user wants to change their password

=> all properties of 'newUserInfo' are optional

**success:** return {'success': true}

**fail:** return {'success': false}

## - Get user info

**POST:** /api/user/getinfo

**params:** {

    'user': '<username>',

    'token', '<token>'

}

**success:** return

{

    'success': true,

    'data': {

        'email': '<email>',

        'fullname': '<fullname>',

        'bio': '<bio>',

        'username': '<username>',

        'phofile\_photo', '<phofile\_photo>'

    }

}

### 3.3. Image mamangement

#### - Upload image

**POST:** /api/image/upload

**Params:** {

'user': '<username>',  
'token', '<token>',  
'imageURI': <image-base64>

}

**success:** return {'success': true, 'imageUrl': '<image-address>'}

**fail:** return {'success': false}

#### - Delete image <not implemented>

**Params:** {

'user': '<username>',  
'token', '<token>',  
'imageUrl': '<image-address>'

}

**success:** return {'success': true}

**fail:** return {'success': false}

### 3.4. Collection mamangement

#### - Create collection

**POST:** /api/collection/create

**Params:** {

'username': '<username>',  
'token', '<token>',  
'photo': '<photoUrl>',  
'name': '<name>',

```
        'description': '<description>',
    }
success: return
{
    'success': true,
    'collection_id': '<collection_id>',
}
fail: return {'success': false}
```

## - Update collection

**POST:** /api/collection/update

**Params:** {

```
    'username': '<username>',
    'token', '<token>',
    'collection_id': '<collection_id>',
    'photo': '<photoUrl>',
    'name': '<name>',
    'description': '<description>',
    'remember_score': '<remember_score>'
}
```

**success:** return

```
{ 'success': true }
```

**fail:** return

```
{ 'success': false }
```

## - Delete collection

**POST:** /api/collection/delete

**Params:** {

```
    'username': '<username>',  
    'token', '<token>',  
    'collection_id': '<collection_id>' }  
}
```

**success:** return

```
{ 'success': true }
```

**fail:** return

```
{ 'success': false }
```

#### - Get collection info

**POST:** /api/collection/get

**Params:** {

```
    'username': '<username>',  
    'token', '<token>',  
    'collection_id': '<collection_id>',  
}
```

**success:** return

```
{ 'success': true,
```

```
  'data' : {
```

```
    'id': '<collection_id>',
```

```
    'photo': '<photoUrl>',
```

```
    'name': '<name>',
```

```
    'description': '<description>',
```

```
    'remember_score': '<remember_score>',
```

```
    'flashcards': [<list of ids of flashcard in the collection>]
```

```
  }
```

```
}
```

**fail:** return

```
{ 'success': false }
```

#### - Get all collection

**POST:** /api/collection/getall

**Params:** {

'username': '<username>',

'token', '<token>',

'collection\_id': '<collection\_id>',

}

**success:** return

```
{ 'success': true,
```

```
  'data' : [{
```

```
    'id': '<collection_id>',
```

```
    'photo': '<photoUrl>',
```

```
    'name': '<name>',
```

```
    'description': '<description>',
```

```
    'user_id': '<user_id>'
```

```
  },...]
```

```
}
```

**fail:** return

```
{ 'success': false }
```

### 3.5. Flashcard mamangement

#### - Create flashcard

**POST:** /api/flashcard/create

**Params:** {

'username': '<username>',

'token', '<token>',

```
'collection_id': '<collection_id>',
'word': '<word>',
'pronunciation': '<pronunciation>',
'meaning': '<meaning>',
'image': '<image>',
'order': '<order in collection>',
}
success: return
{
    'success': true,
    'flashcard_id': '<flashcard_id>',
}
fail: return {'success': false}
```

## - Update flashcard

**POST:** /api/flashcard/update

**Params:** {

```
'username': '<username>',
'token', '<token>',
'flashcard_id': '<flashcard_id>',
'word': '<word>',
'pronunciation': '<pronunciation>',
'meaning': '<meaning>',
'image': '<image>',
'order': '<order in collection>',
}
success: return
```



```
{  
    'success': true  
}  
fail: return {'success': false}
```

#### - Delete flashcard

**POST:** /api/flashcard/delete

**Params:** {  
 'username': '<username>',  
 'token', '<token>',  
 'flashcard\_id': '<flashcard\_id>',  
}

**success:** return  
{ 'success': true }

**fail:** return  
{ 'success': false }

#### - Get flashcard info

**POST:** /api/flashcard/get

**Params:** {  
 'username': '<username>',  
 'token', '<token>',  
 'flashcard\_id': '<flashcard\_id>',  
}

**success:** return  
{  
 'success': true,  
 'data': {

```
        'collection_id': '<collection_id>',
        'id': '<flashcard_id>',
        'word': '<word>',
        'pronunciation': '<pronunciation>',
        'meaning': '<meaning>',
        'image': '<image>',
        'order': '<order in collection>'
    }
}
```

**fail:** return {'success': false}

#### - Get all flashcards

**POST:** /api/flashcard/getall

**Params:** {

```
    'username': '<username>',
    'token', '<token>',
    'collection_id': '<flashcard_id>',
}
```

**success:** return  
{

```
    'success': true,
    'data': [{
        'collection_id': '<collection_id>',
        'id': '<flashcard_id>',
        'word': '<word>',
        'pronunciation': '<pronunciation>',
        'meaning': '<meaning>',
        'image': '<image>',
```

```
        'order': '<order in collection>'
    }, ...]
}
```

**fail:** return {'success': false}

#### - Update all flashcards

**POST:** /api/flashcard/updateall

**Params:** {

```
    'username': '<username>',
    'token', '<token>',
    'collection_id': '<flashcard_id>',
    'data': [{
        'collection_id': '<collection_id>',
        'id': '<flashcard_id>',
        'word': '<word>',
        'pronunciation': '<pronunciation>',
        'meaning': '<meaning>',
        'image': '<image>',
        'order': '<order in collection>'
    }, ...]
}
```

**success:** return  
{

```
    'success': true,
    'data': [
        {
            'collection_id': '<collection_id>',
            'id': '<flashcard_id>',
```

```
    'word': '<word>',  
    'pronunciation': '<pronunciation>',  
    'meaning': '<meaning>',  
    'image': '<image>',  
    'order': '<order in collection>'`  
    }, ...]
```

```
}
```

```
fail: return {'success': false}
```