

DESIGN DOCUMENT FOR Flashcard Software



For course: Software Engineering

***** Group 1:**

- Nguyễn Việt Anh
- Bùi Thị Phương Anh
- Trần Thị Thoa
- Phạm Nguyên 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.

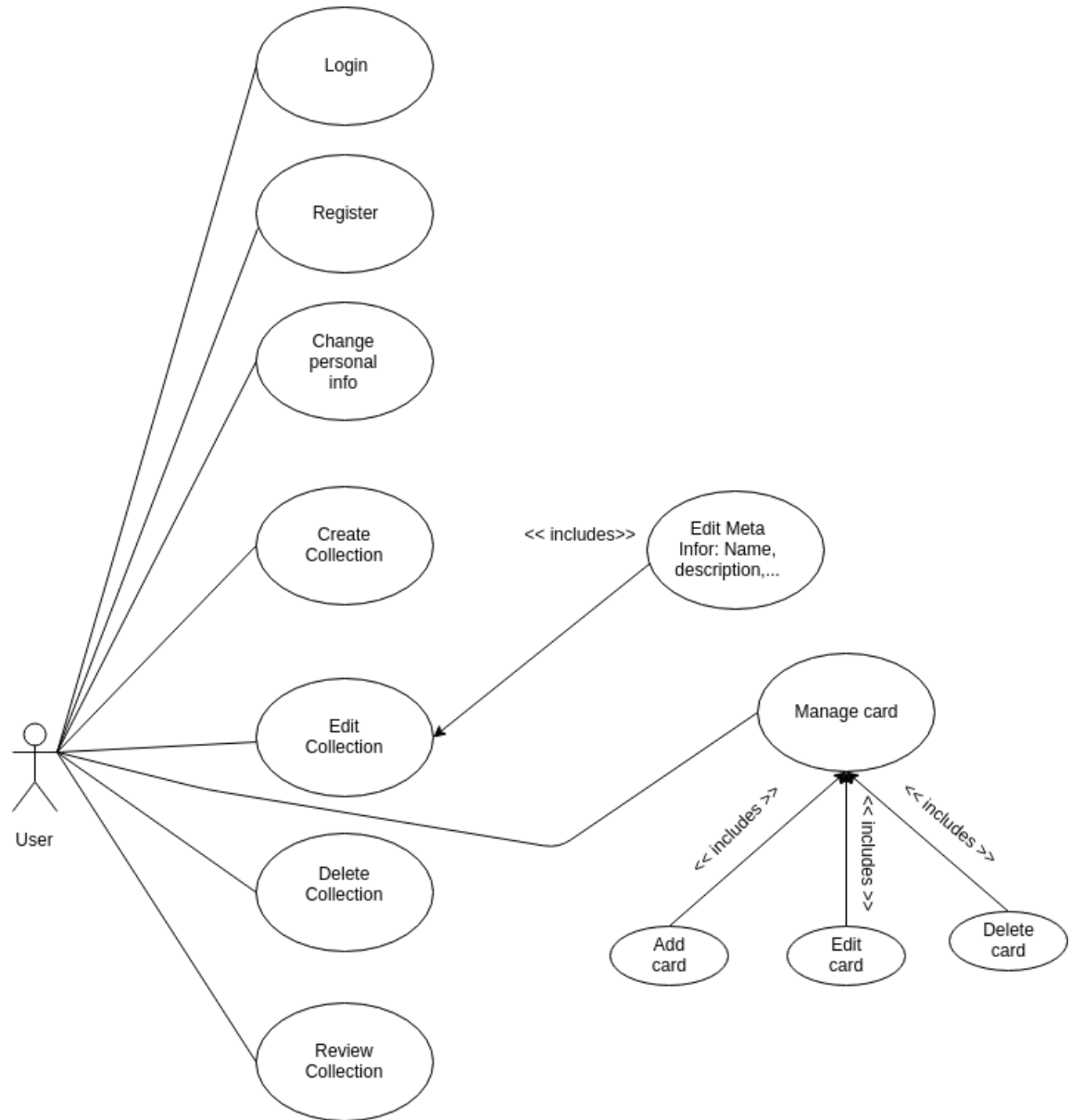
Main feature:

- + 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

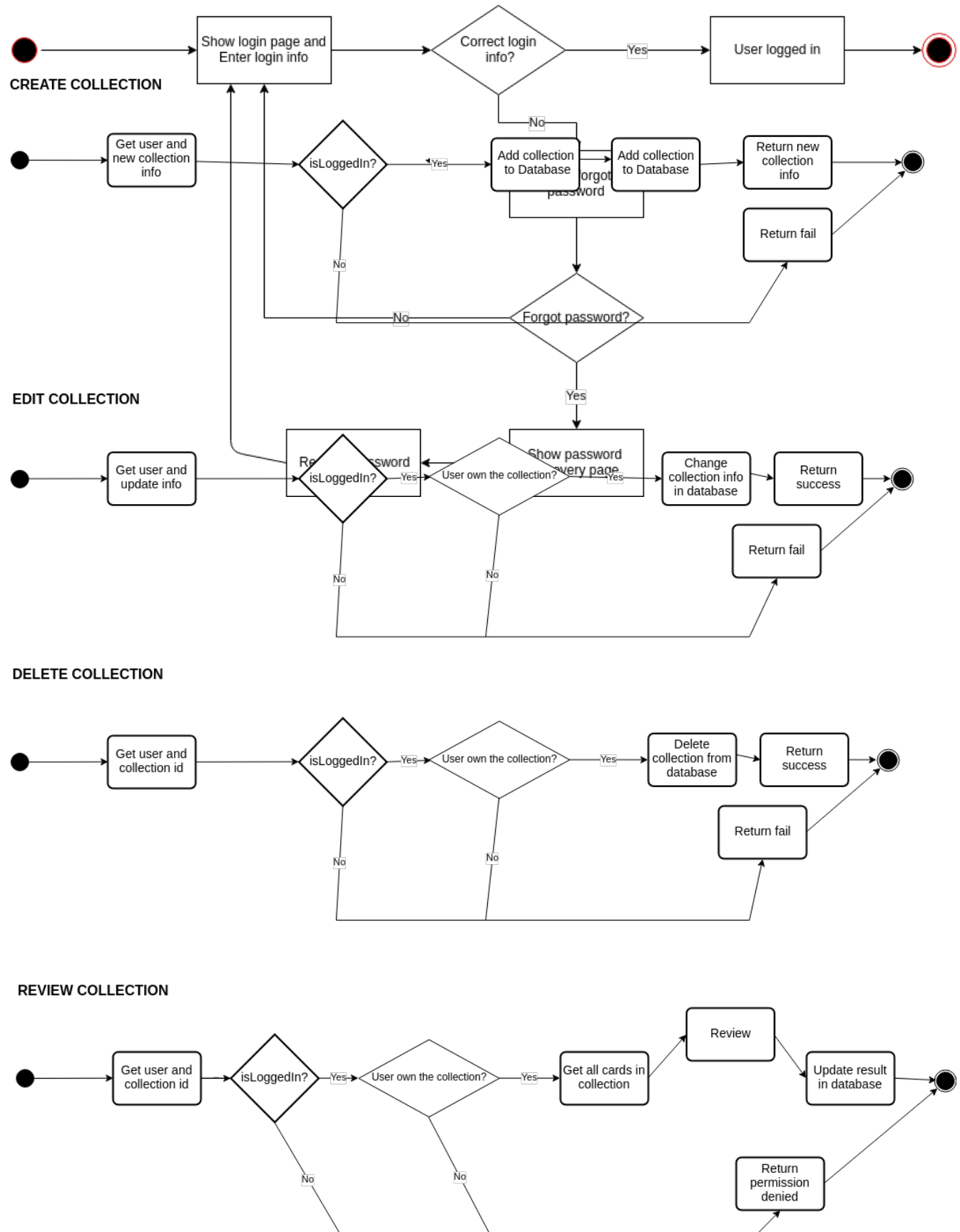
2.1. Use-case diagram

USECASE

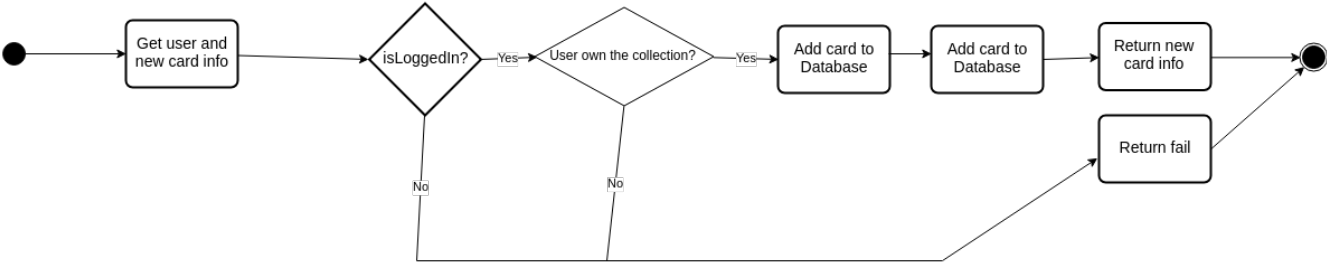


2.2. Activity diagrams

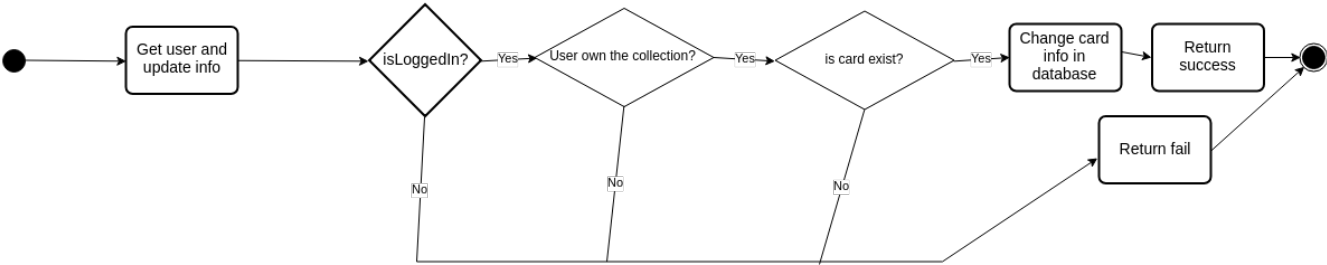
LOGIN



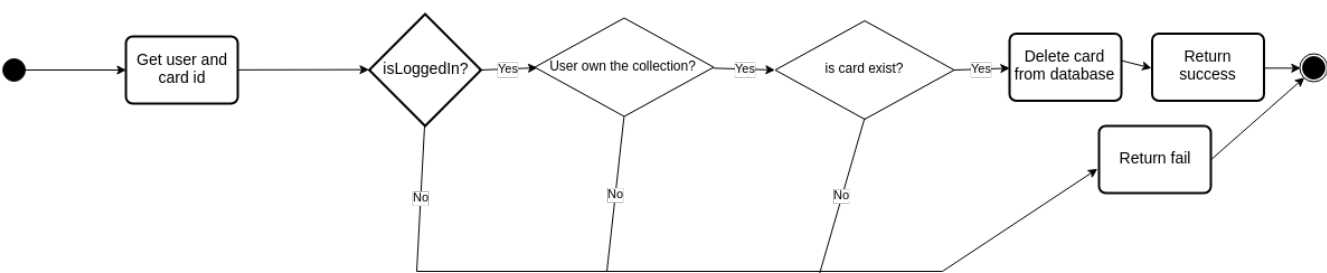
CREATE CARD



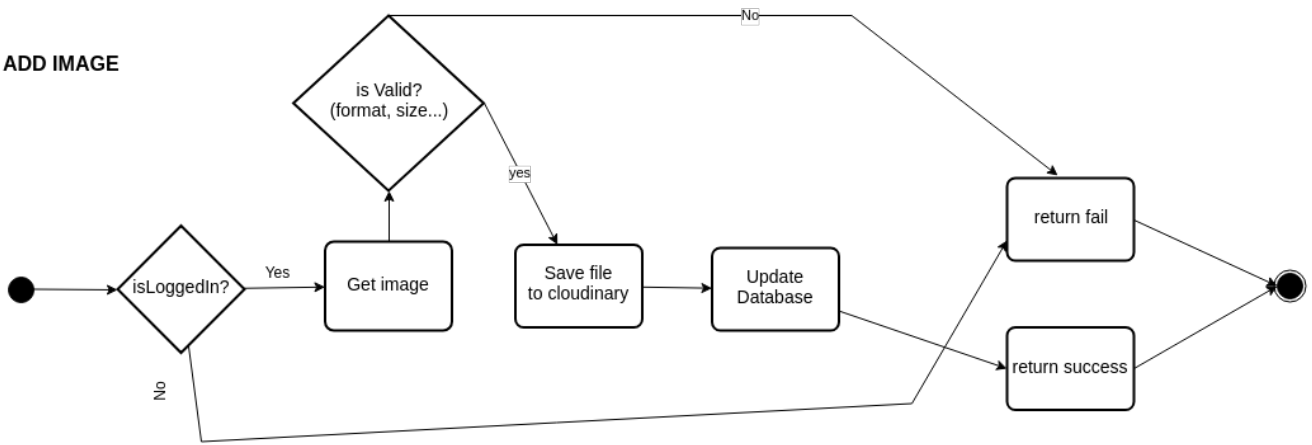
EDIT CARD



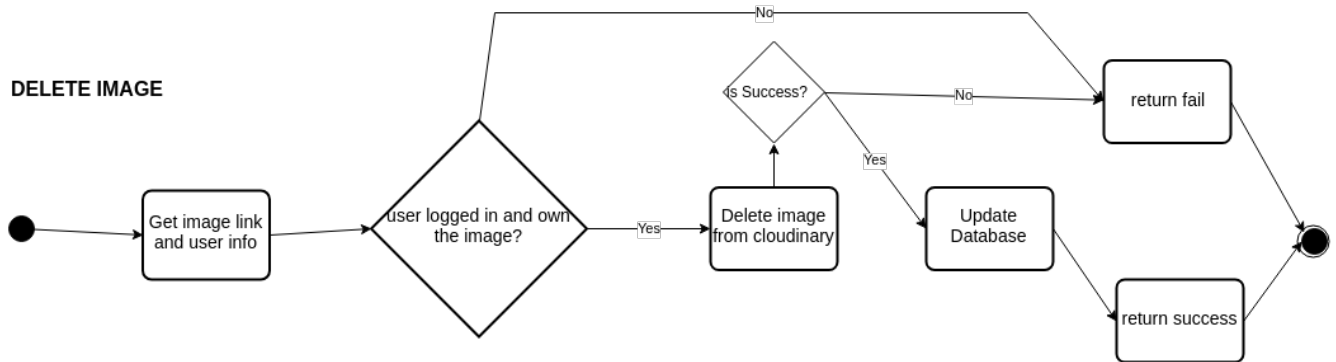
DELETE CARD



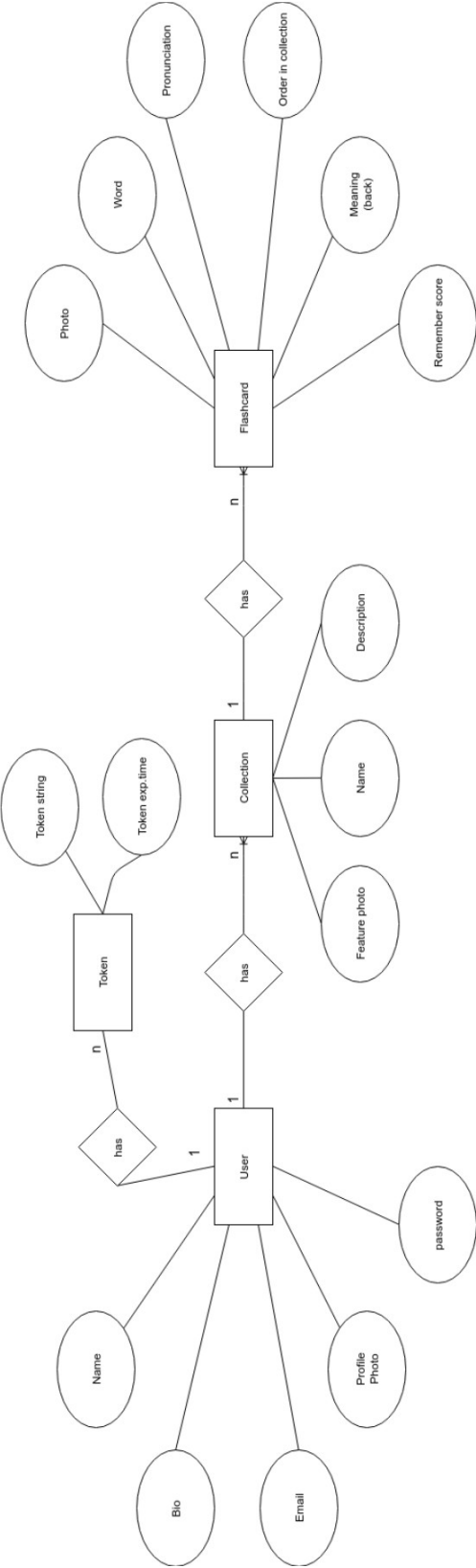
ADD IMAGE



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, password, email, fullname

success: return {'success': true}

fail: return {'sucess': 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: username, token

success: return {'success': true, 'data': {<email, bio, fullname, username, phofile_photo>}} }

3.3. Image mamangement

- Upload image

Params: {

'user': '<username>',

'token', '<token>',

'image': <image-base64>

}

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

fail: return {'success': false}

- Delete image

Params: {

'user': '<username>',

'token', '<token>',

'imageUrl': '<image-address>'

}

success: return {'success': true}

fail: return {'success': false}