**Team:** AlphaX

**Function Description**

- Add a new photo with new tags

- Search for photo(s) with the tags the user given

- Search for photo(s) with the tags condition that user given

- User input photo and display list of three most similar photos

- Choose the photo and display to the browser

**Module Description**

**readwrite.c (index builder)**

This file handle reading and writing the information of the photos to the file and also setup the data structure for using

void readData(PHOTO\_T\* pHead,HASHITEM\_T\* hashphoto[],HASHTAG\_T\* hashtag[]);

void add\_photo\_2\_hashphoto(PHOTO\_T\* photo,HASHITEM\_T\* hashphoto[]);

void add\_photo\_2\_hashtag(PHOTO\_T\* photo,HASHITEM\_T\* hashtag[]);

void add\_photo\_2\_masterlist(PHOTO\_T\* photo,PHOTO\_T\* pHead);

void freeAll(PHOTO\_T\* pHead,HASHITEM\_T\* hashphoto[],HASHITEM\_T\* hashtag[]);

void writeData(PHOTO\_T\* pHead);

void createFile();

**model.c (data access)**

This file contain function handling search for photo(s) that user want , list three most similar photos and Add a new photo with new tags

PHOTO\_T\* findPhoto(char\* namephoto,HASHITEM\_T\* hashphoto[]);

PHOTO\_T\* searchByTag(char\* tag[],HASHITEM\_T\* hashtag[]);

PHOTO\_T\* searchCondition(char\* tag[],char\* except[],HASHITEM\_T\* hashtag[]);

void findSimilar(char\* namephoto,HASHITEM\_T\* hashtag[]);

int checktag(PHOTO\_T\* photo,char\* tag[]);

void addTag(char\* namephoto);

void deleteTag(char\* namephoto);

**view.c (user interface)**

This file contain function for display all user interface

void menuPage();

void searchByTagPage();

void searchConPage();

void similarPage();

void addTagPage();

void deleteTagPage();

void clearScreen();

void displayData(PHOTO\_T\* photo);

**controller.c (controller)**

This file contain function that receive command from the user and response what user want to see

void handlemenu();

void handleSearchByTag();

void handleSearchCondition();

void handleAddTag();

void handleDeleteTag();

int main();

**dtype.c (ADT)**

This file contain implement data structure such as, hash

**LINKLIST ADT**

void insertNode(void\*\* pHead,void\* newNode);

void freeList(void\* pHead);

**HASH ADT**

HASHITEM\_T\*\* initalHash();

int hashFunction(char\* key);

void insertitem(char\* key, PHOTO\_T\* item,HASHITEM\_T\* hash[]);

HASHITEM\_T\* getlist(char\* key,HASHITEM\_T\* hash[]);

void freeHash(HASHITEM\_T\* hash[]);

**Data structures used**

Photo Structure

typedef struct \_listtag

{

/\*linklist of tags\*/

char\* nametag; /\*name of the tag\*/

struct \_listtag \*next;

}LIST\_TAG\_T;

typedef struct \_photo

{

char namephoto[256];/\*name of the photo\*/

char path[512];/\*path of photo\*/

int numtag; /\*number of the tag\*/

int count; /\*use for calculate the similar\*/

int state; /\*use as a flag of calculation,use in search tag and find similar

Block duplicate in list result

true => already be the result of the search

False => not already the result of the search\*/

LIST\_TAG\_T\* alltag; /\*array of all tag\*/

struct \_photo \*nextResult;/\*use for search return as a list\*/

struct \_photo \*next;/\*next photo (use in masterlist)\*/

}PHOTO\_T;

\*mark in PHOTO\_T state is use for block duplicate result when add to the list result

Hash of the tagTable and photoTable Item

typedef struct \_hashitem

{

PHOTO\_T\* photo;

struct \_hashitem \*next;

}HASHITEM\_T;