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,char* tag[]);
   void deleteTag(char* namephoto,char* tag[]);
```

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

<u>Photo Structure</u>

```
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

<u>Hash of the tagTable and photoTable Item</u>

```
typedef struct _hashitem
{
   PHOTO_T* photo;
   struct _hashitem *next;
}HASHITEM_T;
```