

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

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* ht[]);  
void writeData(PHOTO_T* pHead, HASHITEM_T* ht[]);  
void createFile();
```

model.c

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(PHOTO* pHead, char* name);  
  
void searchByTag(HASHITEM_T* ht[] ,char* tags[]);  
  
void searchCondition(HASHITEM_T* ht[],  
char* tags[],char* except[]);  
  
void findSimilar(HASHITEM_T* ht[],PHOTO_T* photo);  
void addTag(PHOTO_T* photo,char* tags[],int count);  
void deleteTag((PHOTO_T* photo,char* tags[],int count);
```

view.c

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

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

```
void handlemenu(PHOTO* pHead, HASHITEM_T* ht[]);
void handleSearchByTag(HASHITEM_T* ht[]);
void handleSearchCondition(HASHITEM_T* ht[]);
void handleAddTag(PHOTO* pHead);
void handleDeleteTag(PHOTO* pHead);
void handle

int main();
```

dtype.c

This file contain implement data structure such as, hash

```
void isNull(*void pointer);
```

LINKLIST ADT

```
void insertNode(void** pHead, void* newNode);
void freeList(void* pHead);
```

HASH ADT

```
int hashFunction(char* key);
void insertitem(char* key, PHOTO_T* item);
void freeHash(HASHITEM_T* ht[]);
```

Structure

Photo Structure

```
typedef struct _listtag
{
    char* nametag;
    struct _listtag *next;
}LIST_TAG_T;

typedef struct _photo
{
    char photoname[256];
    char path[512];
    int count; /*use for calulate the similiar*/

    LIST_TAG_T *alltag;

    struct _photo *next;
}PHOTO_T;
```

Hash Table Item

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


