CPE111

Programming with Data Structures

Topic : Photo Search Engine

Team name : GET A

Team members :

1. Kodchaphon Treesak (Janjao) 62070503401

2. Kulchaya Songkwan (Mew) 62070503405

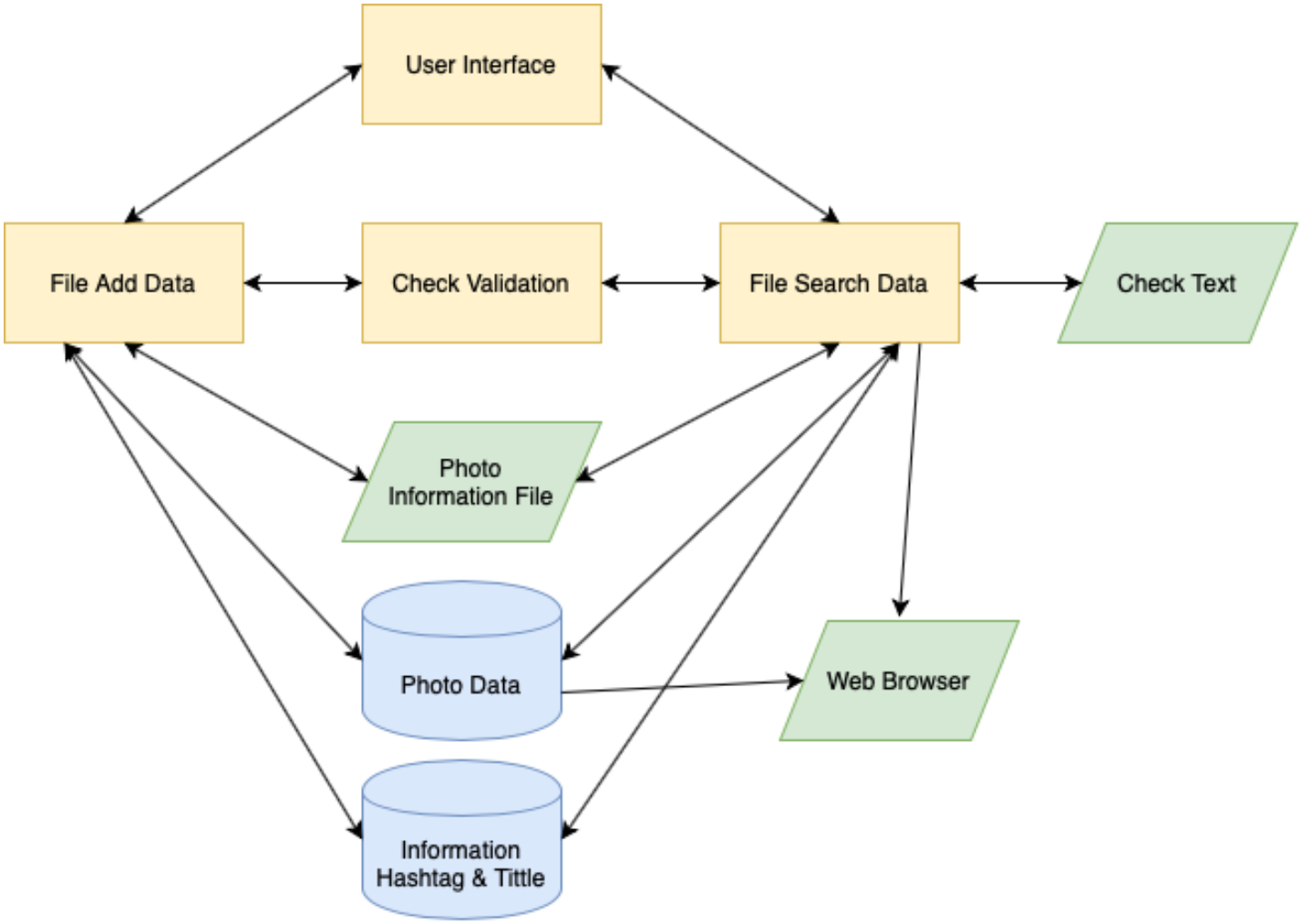
3. Chotiya Pertpring (Fon) 62070503413

4. Prakasit Issanapong (Tar) 62070503431

ABSTRACT

The photo search engine program uses to find images in the folder. The image types consist of .jpg, .JPG, .png or .PNG. Before the user uses the program. They must copy their photo to the system folder (folder : picture). This program has 2 use case. The first is addData. The program will ask the user to input their filename photo. If they don't already add this photo or the photo already keep in the system. The program will show the error message and ask you again. After that, you must add information about the photo to the system. The second use case is searchData. The program will ask the user browser and input the tag. If the browser not exists the program will show the error message. The tags exceed is 5. After you input the system will find a relative photo from the tag and show the filename the user can choose the filename and input it again to show the photo on their web browser.

Architecture Diagram



Module Descriptions

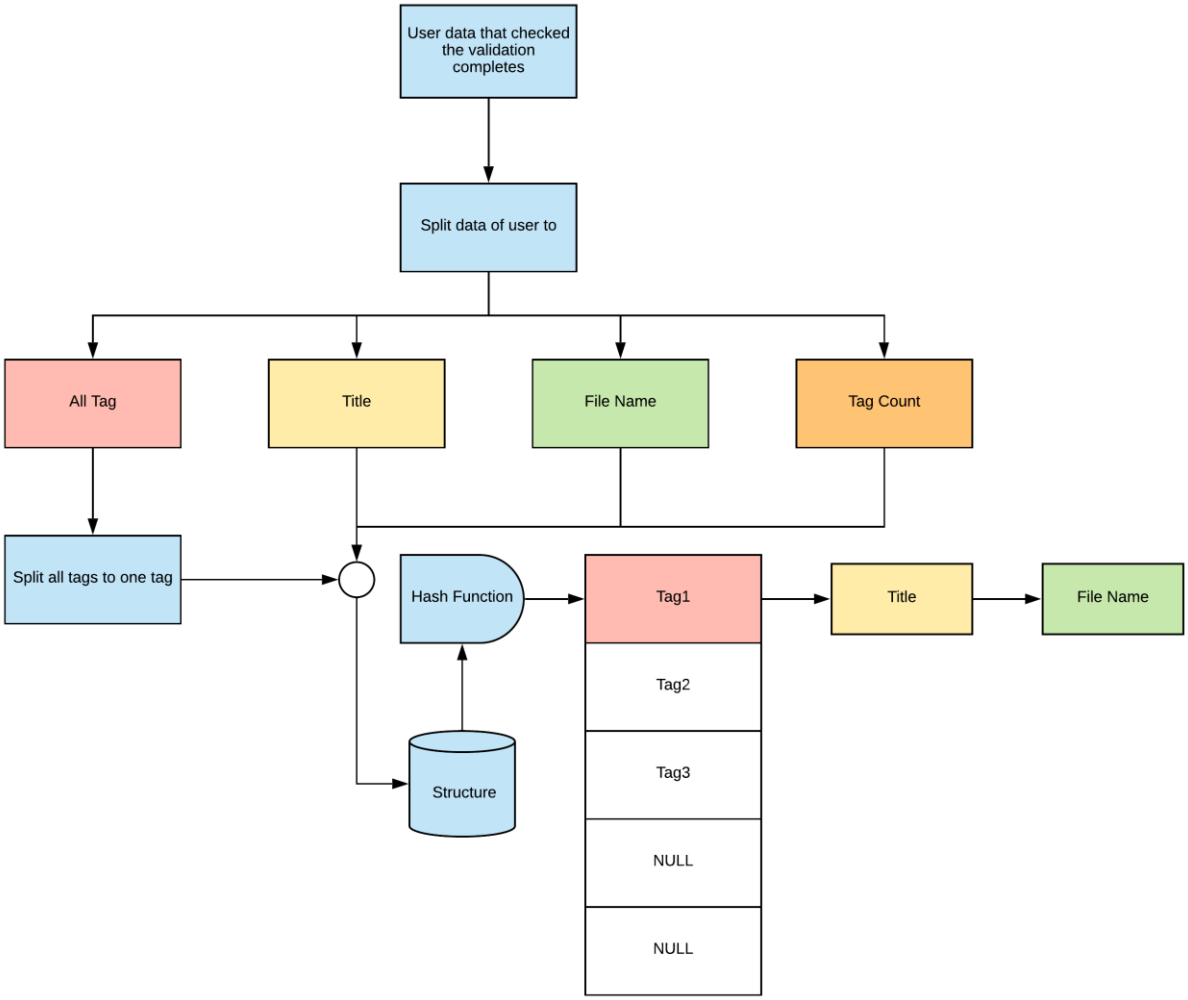
Component	Functionality
interface.c	• main - use to display what option that user want to do (add, search or exit). - in case user choose add then calls AddOption function in addData.c - in case user choose search then call searchOption function in searchData.c
validation.c	• rules - use to display the rule that user must follow when user want to add the data of photo. • checkNameSystem - use to find all file name of photo in folder that user keep all photos. • checkCompareTitleInFile - use to check the title of the photo that user is adding does not same with title in the file. • checkSpecialStringIntag - use to check tags that user is adding only alphabet and number. • checkSpecialString - use to check data(file name and title) that user is adding only alphabet and number. • checkFilename - use to check file name of the photo that user is adding is .jpg, .JPG, .png or .PNG. • checkAllFileName - use to check the file name of photo that user is adding does not same with file name in the file.

Component	Functionality
addData.c	• renameFunction - use to rename the file name when user want to rename it. • writeInFile - use to write the data that user input in to the file. • freeList - use to free the data in variable. • AddOption - use to get file name that user want to add, title of photo, how many tags in this photo, tags of photo. And set the data to check about validation.
searchData.c	• bitwiseOpHash - use to fine index of data. • buildHashTable – use to build hash table from text file • printOnWeb - use to ask user what browser does user want to use and ask what name of photo that user want to display on the browser. • printSingleData - use to print all data information of searching process(in case user input only one tag). • printData - use to print all data information of searching process(in case user input more than one tag but less than six tags). • lastCheck - use to printf file name that match with user input tags. • checkDash - use to find how many tags that user input. • checkSpace – use to find space in tags that user input

Component	Functionality
	• searchOption - get the tags that user want to search (Maximum 5 tags).
chainedHashTable.c	• hashTableSize - use to find the number of slots in hash able. • hashTableItiemCount - use to find the number of items currently stored in the hash table. • hashTableInit - Initialize the hash table. • hashTableFree - free the hash table. • hashTableInsert - insert the data in to hash table. • hashTableRemove - use to remove the data in hash table. • hashTableLookup - look up only one data in hash table. • multiHashTableLookup - look up multi data in hash table.

Data Structure Documentation

Data Structure Duagram



Explain :

The user has added the data and the program has verified the correct format, the program will split the user data into 4 parts is 1.all tags, 2.title, 3.file name, 4.tag count and import all tags split into one tag. Collect Title and File name and tags into structure and each data in the structure is stored in the form of a hash table using the hash function and using the tag as the address of the data and in the one slote of the hash table, use a list link to collect data of title and file name.

This main structure use hold data of the user :

```
/* structure of all photo data */
typedef struct _file
    {
    char title[128];            /* array of titlename */
    char filename[128];         /* array of filename */
    char alltag[256];           /* array of all hastag */
    char tags[MAXTAGS][256];    /* array of tag in each photo */
    char* pTags[MAXTAGS];
    int tagCount;               /* how many tags are associated with this photos */
    int matchCount;             /* temporary, user in searching algorithm */
    struct _file *pNext;
    } PHOTODATA_T;
```

This structure use hold data about file name of photo from subdirectory

```
/* struct to keep all name file.*/
typedef struct
    {
    char namefile[128];         /* keep the name file of photo*/
    } PHOTODATA2_T;
```

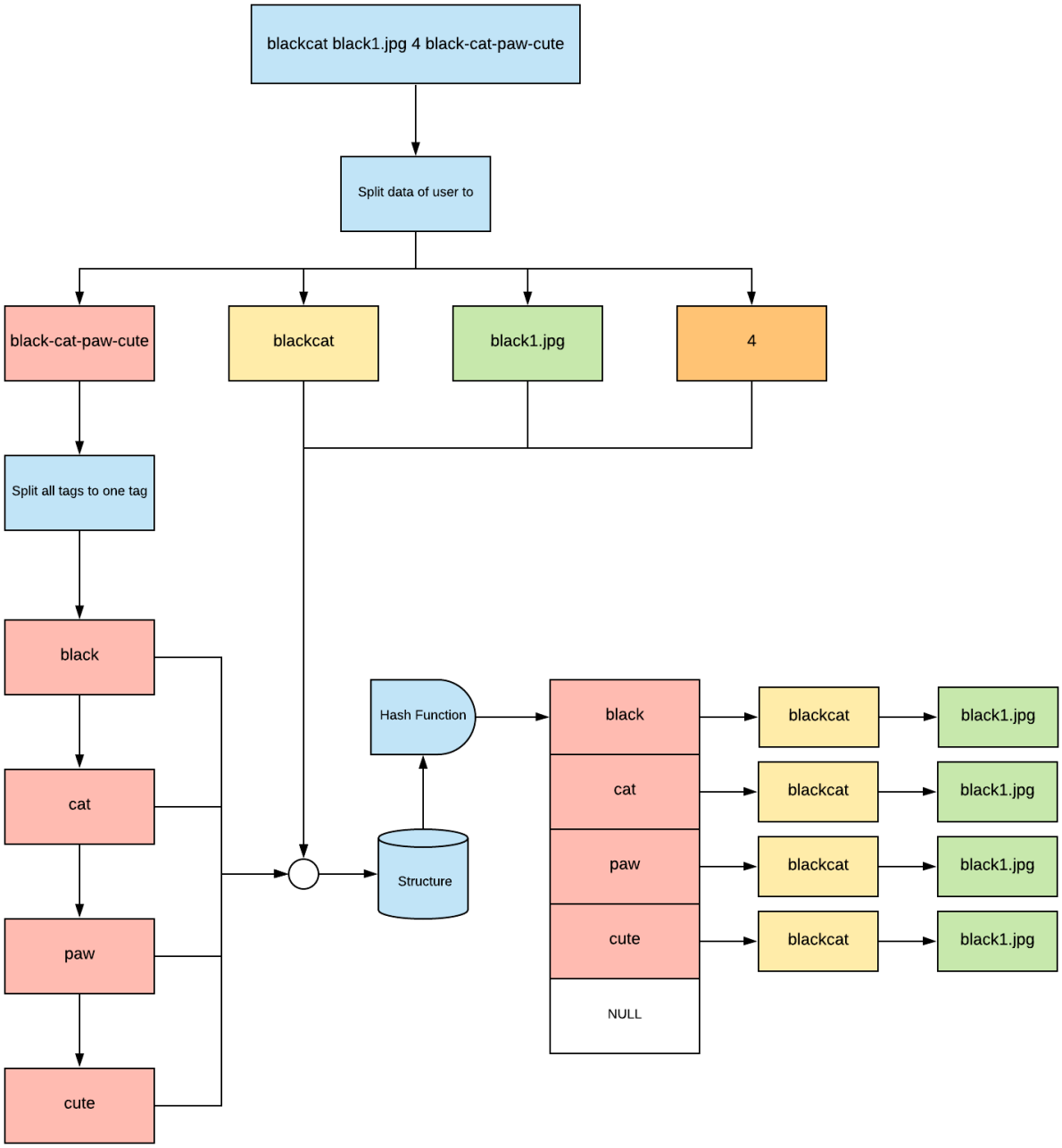
External Files structure :

input.txt - This file will hold each data of the user that completed the validation. The title will store in **title[128]**, The file name will store in **filename[128]**, the number of tag wil store in **tagCount** and The hastag will store in **alltag[256].** For example input.txt

```
blackcoffee black2.jpg 4 black-coffee-cup-hand
blackguitar black3.jpg 4 black-guitar-mp3-note
blackwolf black4.jpg 4 black-wolf-dark-killer
blacklipstick black5.jpg 4 black-lipstick-channel-luxury
bluedonut blue1.jpg 4 blue-donut-sugar-sweet
bluehair blue2.jpg 4 blue-hair-girl-hilight
bluewhale blue3.jpg 4 blue-whale-ocean-sea
bluedress blue4.jpg 4 blue-dress-girl-shoes
```

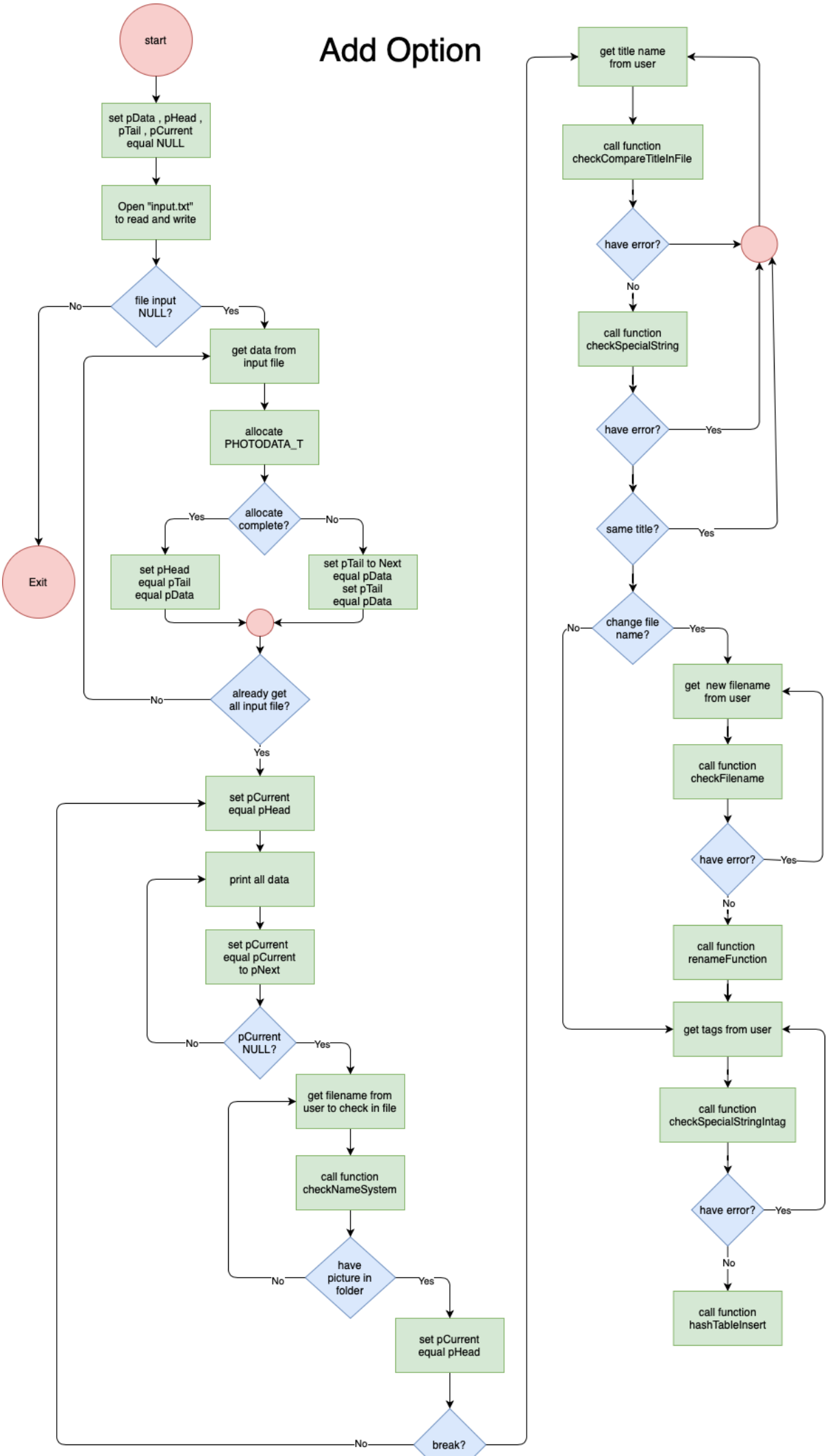
Example :

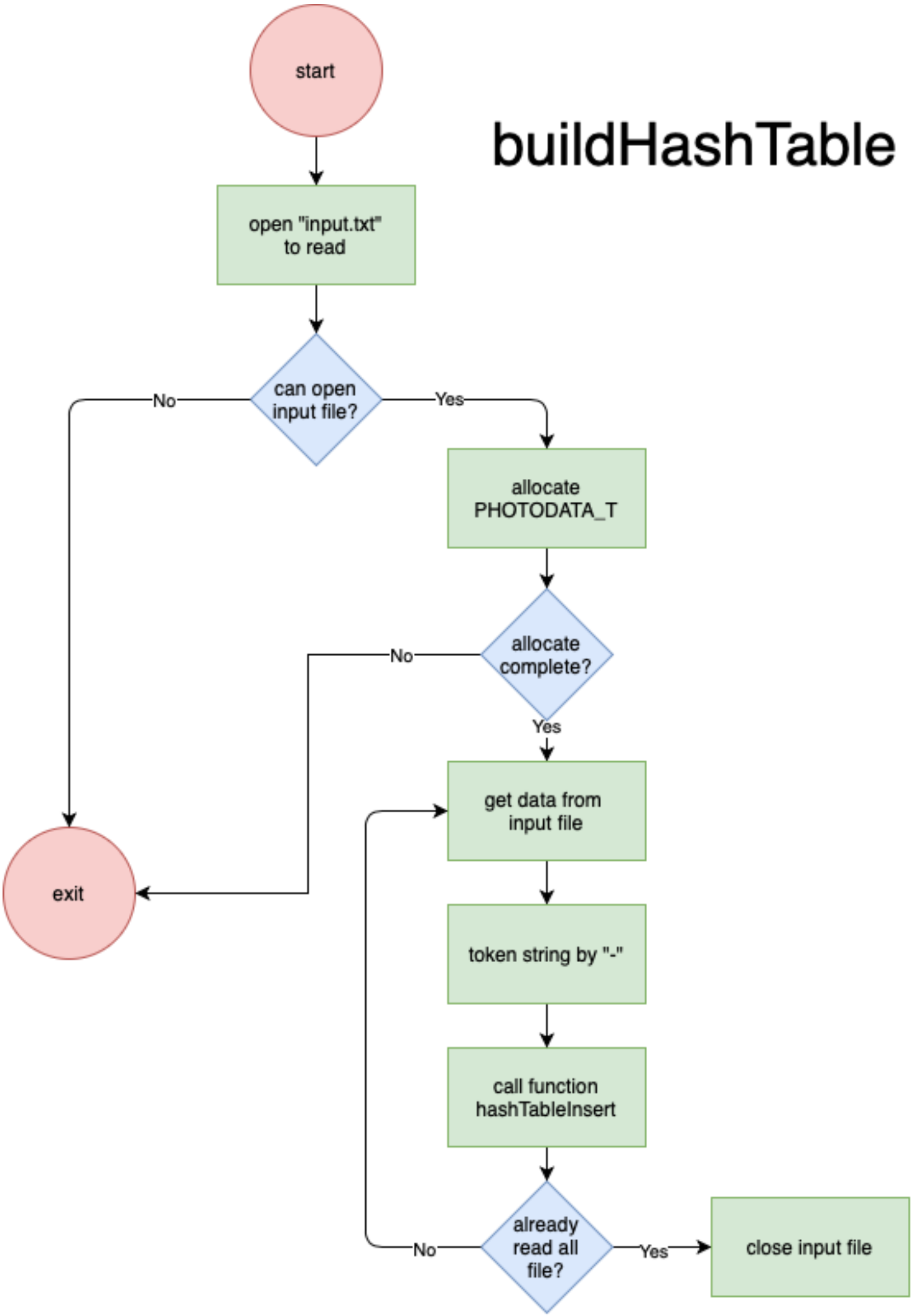
If user input photo have the title is **blackcat, the** file name is **black1.jpg** and the hastag is **black-cat-paw-cute** ”

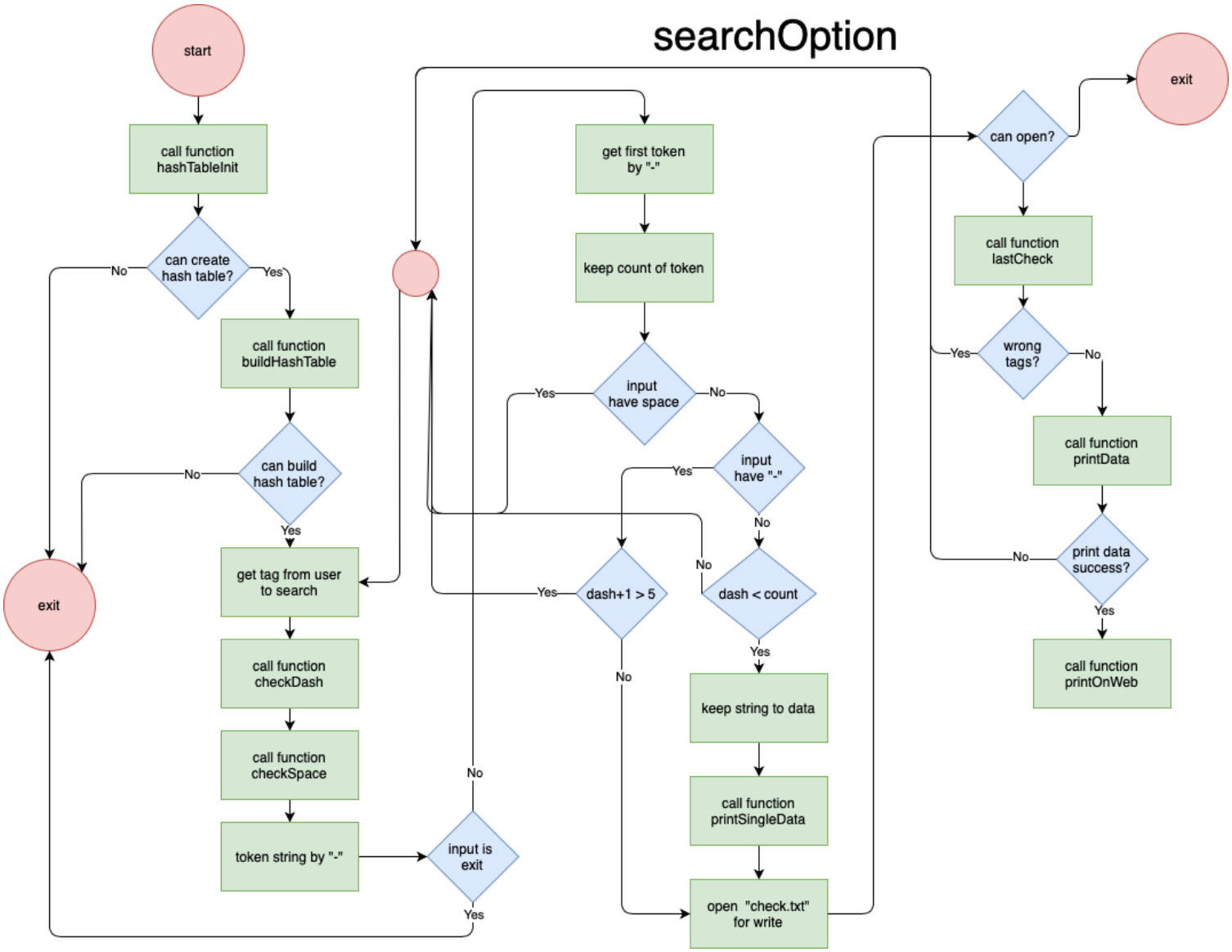


Flowcharts for Key Algorithms

Add Option


start
set pData , pHead , pTail , pCurrent equal NULL
Open "input.txt" to read and write
file input NULL?
No
Yes
Exit
get data from input file
allocate PHOTODATA_T
allocate complete?
Yes
No
set pHead equal pTail equal pData
set pTail to Next equal pData set pTail equal pData
already get all input file?
No
Yes
set pCurrent equal pHead
print all data
set pCurrent equal pCurrent to pNext
pCurrent NULL?
No
Yes
get filename from user to check in file
call function checkNameSystem
have picture in folder
No
Yes
set pCurrent equal pHead
break?
No
get title name from user
call function checkCompareTitleInFile
have error?
No
call function checkSpecialString
have error?
Yes
same title?
Yes
change file name?
No
Yes
get new filename from user
call function checkFilename
have error?
Yes
No
call function renameFunction
get tags from user
call function checkSpecialStringIntag
have error?
Yes
No
call function hashTableInsert


buildHashTable
start
open "input.txt" to read
can open input file?
No
Yes
exit
allocate PHOTODATA_T
allocate complete?
No
Yes
get data from input file
token string by "-"
call function hashTableInsert
already read all file?
No
Yes
close input file


searchOption
start
call function hashTableInit
can create hash table?
No
Yes
exit
call function buildHashTable
can build hash table?
No
Yes
exit
get tag from user to search
call function checkDash
call function checkSpace
token string by "-"
input is exit
No
Yes
get first token by "-"
keep count of token
input have space
Yes
No
input have "-"
Yes
No
dash+1 > 5
Yes
No
dash < count
No
Yes
keep string to data
call function printSingleData
open "check.txt" for write
can open?
exit
call function lastCheck
wrong tags?
Yes
No
call function printData
print data success?
No
Yes
call function printOnWeb

Top Level Flowchart