IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. VI (Sep – Oct. 2014), PP 04-14
www.iosrjournals.org

Teneur: A LINUX based offline content management system

Das Gupta Debasmita, Ganguly Sridisha, Paul Maumita, Saha Arunima
ESL Lab, Chandannagar, 2014

Abstract: *This technical section provides a brief synopsis about our offline content management system. The main purpose of this system is to manage the resources such as text files, images in a more synchronized manner. This system is also capable of administering over the messages send to and received from different users.*
The basic concept is that whenever a user signs up with his login id and password a new folder is automatically created with his id name. And then the user can access this area specified to him to store his belongings.
The different operations that can be done can be divided into modules. The first module is TEXT. One can open his own files or may want to view some other user's files. But he can do so only if he has the permission granted to him by the other user. This feature provides security to the contents. One may also add different text files within a specific file size limit. If enough memory is not present then he has an option to delete files either by his choice or the administrator will delete the most less frequently used files to accommodate the space for new files. The files will be stored in the folders specified by their extensions.
The second module is IMAGE. All the above mentioned operations performed with text files can also be implemented on Image files.
The third module is concerned about MESSAGES. Just like the e-mailing system a user can send and receive messages from other users. He can also search for messages by date or by sender's name.
An additional feature provided is that if a user wants to copy or transfer his contents or create a backup of his directory then he can do it easily by compressing his files in zipped format.
The methodology section shows its functionality in details. Hence this system provides a comprehensive and efficient administration of annals so that manipulation and storage becomes easy.

I. Introduction

Linux

Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open source software development and distribution. The defining component of Linux is the Linux kernel,an operating system kernel first released on 5 October 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux, which has led to some controversy.

Linux was originally developed as a free operating system for Intel x86-based personal computers. It has since been ported to more computer hardware platforms than any other operating system.It is a leading operating system on servers and other big iron systems such as mainframe computers and supercomputers.

The development of Linux is one of the most prominent examples of free and open source software collaboration. Typically, Linux is packaged in a format known as a Linux distribution for desktop and server use. Some popular mainstream Linux distributions include Debian, Ubuntu, Linux Mint, Fedora, openSUSE, Arch Linux, and the commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server.

Uses

Linux is ubiquitously found on various types of hardware. As well as those designed for general purpose use on desktops and servers, distributions may be specialized for different purposes including: computer architecture support, embedded systems, stability, security, localization to a specific region or language, targeting of specific user groups, support for real-time applications, or commitment to a given desktop environment. Furthermore, some distributions deliberately include only free software. Currently, over three hundred distributions are actively developed, with about a dozen distributions being most popular for general-purpose use.

- Desktop
  The popularity of Linux on standard desktop computers and laptops has been increasing over the years.Currently most distributions include a graphical user environment, with the two most popular environments being GNOME (which can utilize additional shells such as the default GNOME Shell and Ubuntu Unity), and the KDE Plasma Desktop.
- Netbook market

Linux distributions have also become popular in the netbook market, with many devices such as the ASUS Eee PC and Acer Aspire One shipping with customized Linux distributions installed.

- Smart devices
  Several OSes for smart devices, e.g. smartphones, tablet computers, smart TVs, and in-vehicle infotainment (IVI) systems, are Linux-based. The three major platforms are mer, Tizen, and Android.
- Embedded devices
  Due to its low cost and ease of customization, Linux is often used in embedded systems. In the non-mobile telecommunications equipment sector, the majority of customer-premises equipment (CPE) hardware runs some Linux-based operating system. OpenWrt is a community driven example upon which many of the OEM firmwares are based.
- Gaming
  There had been several games that run on traditional desktop Linux, and many of which originally written for desktop OS. However, due to most game developers not paying attention to such a small market as desktop Linux, there hasn't been many prominent games available for desktop Linux.
- Specialized uses
  Due to the flexibility, customizability and free and open source nature of Linux, it becomes possible to highly tune Linux for a specific purpose. There are two main methods for creating a specialized Linux distribution: building from scratch or from a general-purpose distribution as a base.
  Home theater PC
  A home theater PC (HTPC) is a PC that is mainly used as an entertainment system, especially a Home theater system. It is normally connected to a television, and often an additional audio system.
  Digital security
  Kali Linux is a Debian-based Linux distribution designed for digital forensics and penetration testing. It comes preinstalled with several software applications for penetration testing and identifying security exploits.
  System rescue
  Linux Live CD sessions have long been used as a tool for recovering data from a broken computer system and for repairing the system. Building upon that idea, several Linux distributions tailored for this purpose have emerged, most of which use GParted as a partition editor, with additional data recovery and system repair software

Shell Script

A shell script is a computer program designed to be run by the Unix shell, a command line interpreter.[1] The various dialects of shell scripts are considered to be scripting languages.
Typical operations performed by shell scripts include file manipulation, program execution, and printing text.

Capabilities

- Shortcuts
  In its most basic form, a shell script can provide a convenient variation of a system command where special environment settings, command options, or post-processing apply automatically, but in a way that allows the new script to still act as a fully normal Unix command.
  One example of a shell script that could be used as a shortcut would be to print a list of all the files and directories within a given directory.
  #!/bin/sh
  clear
  ls -al
  In this case, the shell script would start with its normal starting line of #!/bin/sh. Following this, the script executes the command clear which clears the terminal of all text before going to the next line. The following line provides the main function of the script. The ls -al command list the files and directories that are in the directory from which the script is being run. The ls command attributes could be changed to reflect the needs of the user.
- Batch jobs
  Shell scripts allow several commands that would be entered manually at a command-line interface to be executed automatically, and without having to wait for a user to trigger each stage of the sequence. For example, in a directory with three C source code files, rather than manually running the four commands required to build the final program from them, one could instead create a C shell script, here named build and kept in the directory with them, which would compile them automatically:
  #!/bin/csh
  echo compiling...

cc -c foo.c
cc -c bar.c
cc -c qux.c
cc -o myprogfoo.obar.oqux.o
echo done.
The script would allow a user to save the file being edited, pause the editor, and then just run ./build to create the updated program, test it, and then return to the editor.

- Generalization
  Simple batch jobs are not unusual for isolated tasks, but using shell loops, tests, and variables provides much more flexibility to users. A Bash (Unix shell) script to convert JPEG images to PNG images, where the image names are provided on the command line—possibly via wildcards—instead of each being listed within the script, can be created with this file, typically saved in a file like /home/username/bin/jpg2png
- Verisimilitude
  A key feature of shell scripts is that the invocation of their interpreters is handled as a core operating system feature. So rather than a user's shell only being able to execute scripts in that shell's language, or a script only having its interpreter directive handled correctly if it was run from a shell (both of which were limitations in the early Bourne shell's handling of scripts), shell scripts are set up and executed by the OS itself.
- Programming
  Many modern shells also supply various features usually found only in more sophisticated general-purpose programming languages, such as control-flow constructs, variables, comments, arrays, subroutines, and so on. With these sorts of features available, it is possible to write reasonably sophisticated applications as shell scripts. However, they are still limited by the fact that most shell languages have little or no support for data typing systems, classes, threading, complex math, and other common full language features, and are also generally much slower than compiled code or interpreted languages written with speed as a performance goal.

Python

Python is an interpreted, high-level programming language, pure object-oriented and powerful server-side scripting language for the Web. Like all scripting languages, Python code resembles pseudo code. Its syntax's rules and elegant design make it readable even among multiprogrammer development teams. The language doesn't provide a rich syntax, which is really helpful. The idea behind that is to keep you thinking about the business rules of your application and not to spend time trying to figure out what command you should use.

It is also true that Python is interactive, portable, easy to learn, easy to use, and a serious language. Furthermore, it provides dynamic semantics and rapid prototyping capabilities.

Why Python?

Python is simple: Python has clear and simple rules, and is closer to English than any of the languages we use. Creating Python programming is so straightforward that it's been called "programming at the speed of thought."

Statements are terminated by end of line, and block structure is indicated by indentation. Python programs look like executable pseudo-code. This eliminates a host of troublesome errors for beginning programmers, especially placement of semi-colons, bracketing and indentation. The simplicity of Python makes it easy to learn. In addition to the list (dynamic array) data structure, Python provides tuples (immutable lists) and dictionaries (hash tables). Together with the class mechanism, these can be used to quickly build sophisticated data structures for interesting projects. The absence of type declarations makes for less code and more flexible programming.

Python is self-adjustable: Python provides full dynamic runtime type checking and bounds checking on array subscripts. Python employs garbage collection so there is no problem with dangling pointers or memory leaks. It is impossible for user code in Python to produce a segmentation violation. In this respect Python is similar to Java, and both are much safer than C++.

Python supports object-oriented programming: Languages like C#, Java, and Python are all object-oriented. But Python does them one better. In C# and Java, OOP is not optional. This makes short programs unnecessarily complex, and it requires a bunch of explanation before a new programmer can do anything significant. Python takes a different approach. In Python, using OOP techniques is optional. You have all of OOP's power at your disposal, but you can use it when you need it. Got a short program that doesn't really require OOP? No problem.

Got a large project with a team of programmers that demands OOP? That'll work too. Python gives you power and flexibility.

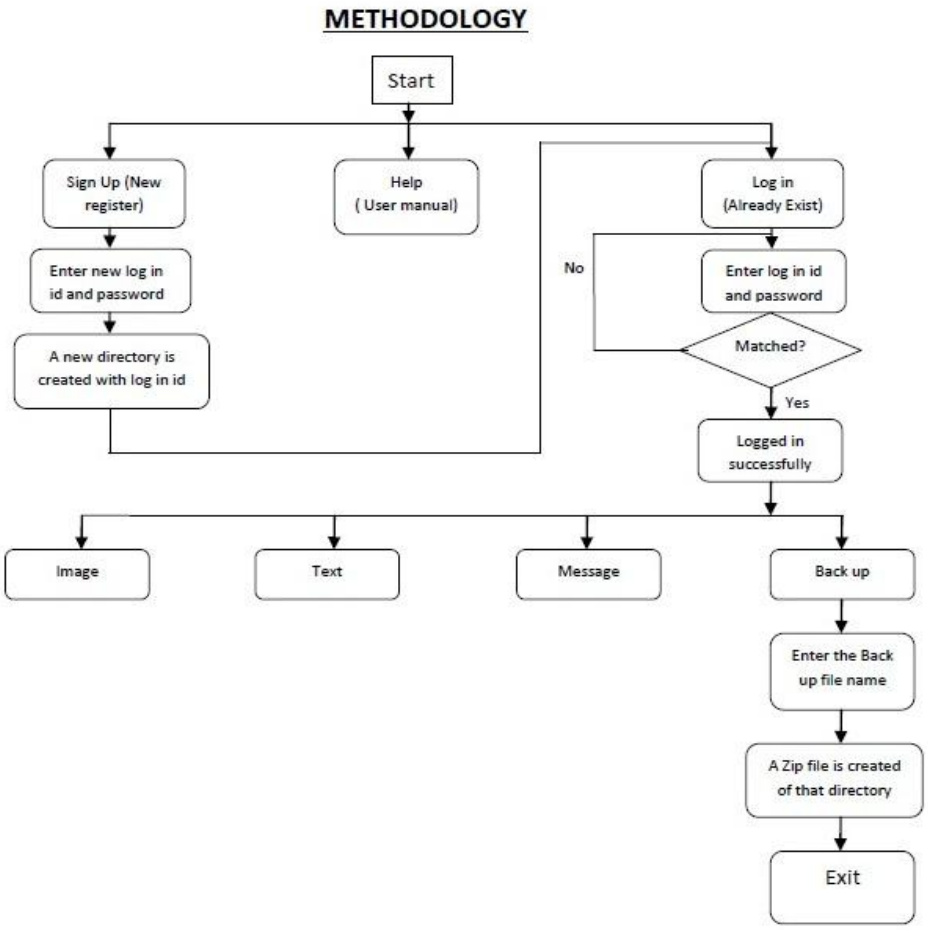
Python Runs Everywhere: Python programs are platform independent, which means that regardless of the operating system you use to create your program, it'll run on any other computer with Python. So if you write a game on your PC, you can e-mail a copy to your friend who runs Linux or to your aunt who has a Mac and the program will work.

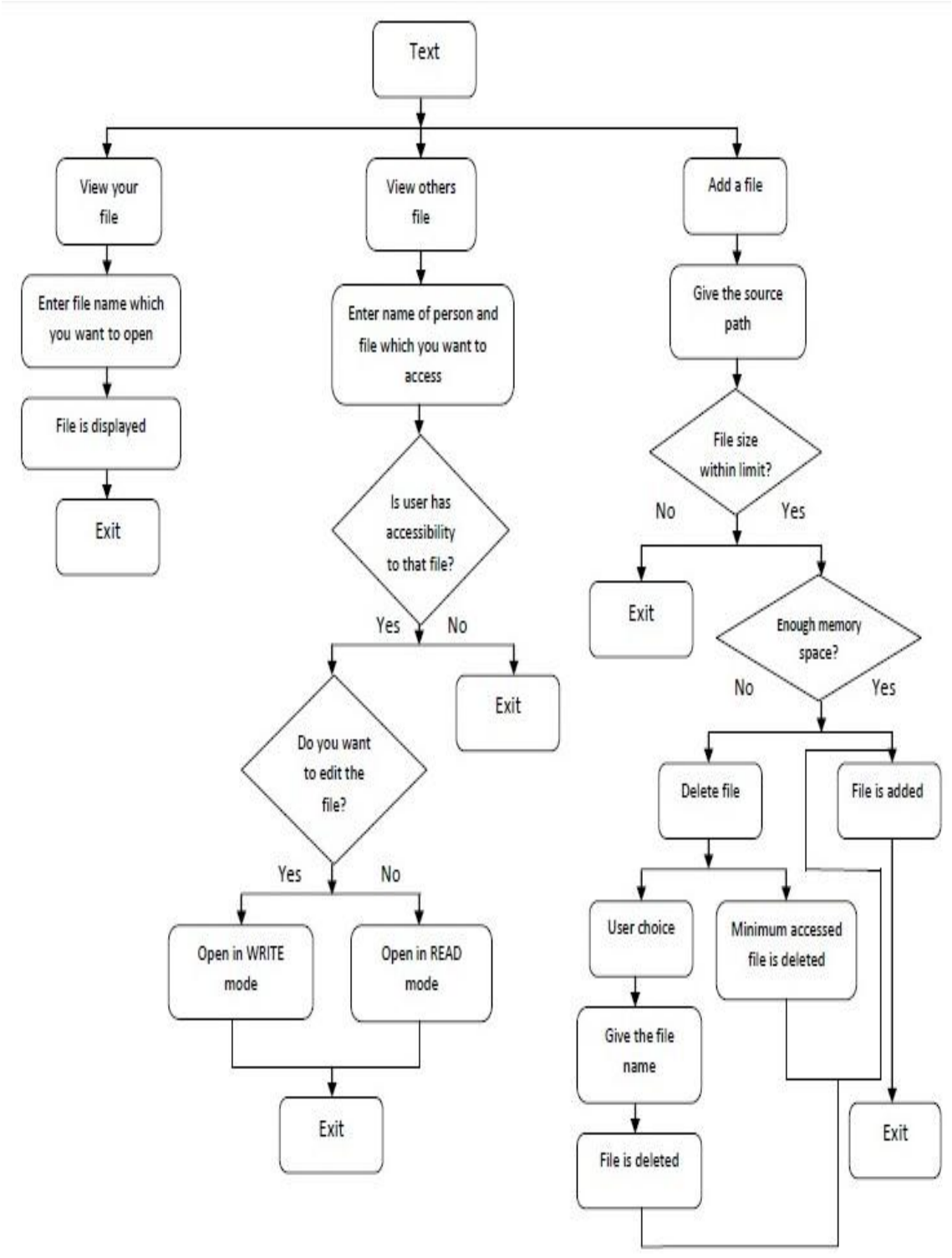
Python is Open Source: Python is free. You can install it on your computer and never pay a penny. But Python's license lets you do much more than that. You can copy or modify Python. You can even resell Python if you want (but don't quit your day job just yet). Embracing open-source ideals like this is part of what makes Python so popular and successful. Python is a programming language that lets you work more quickly and integrate your systems more effectively. You can learn to use Python and see almost immediate gains in productivity and lower maintenance costs.

II. Motivation

The main motive behind this project of ours is the thought to develop an offline content management system which will aid different organizations. We are all aware of the online Content Management systems like Wordpress and Drupal to name a few. As it is online we may face certain difficulties at times. Say for example, we may not always get a proper internet connection and so then it will be practically impossible to use them. Moreover there is a lot of memory wastage as no backup is created. So what we thought of is to design something which will efficiently work in an offline environment and hence the internet issues can be ignored. The problem of memory wastage is also minimized as we have kept a check on the maximum size of files that a user can upload, the total size of each user's directory and also the backup facility is granted.

We have used Linux to design he basic framework of our project and Python at the back end. This is apt for organizations having multi users as this allows the users to have their own separate directory where they can store their own files, access other's files as per the permission given and the system admin can keep a check on all the users' directories and their files. The several employees working in the organization can communicate with each other by the Messaging facility of this project. In all " " will give the feel of an online CMS when it's actually working offline.




Text
View your file
Enter file name which you want to open
File is displayed
Exit
View others file
Enter name of person and file which you want to access
Is user has accessibility to that file?
Yes
No
Exit
Do you want to edit the file?
Yes
No
Open in WRITE mode
Open in READ mode
Exit
Add a file
Give the source path
File size within limit?
No
Yes
Exit
Enough memory space?
No
Yes
Delete file
File is added
User choice
Minimum accessed file is deleted
Give the file name
File is deleted
Exit

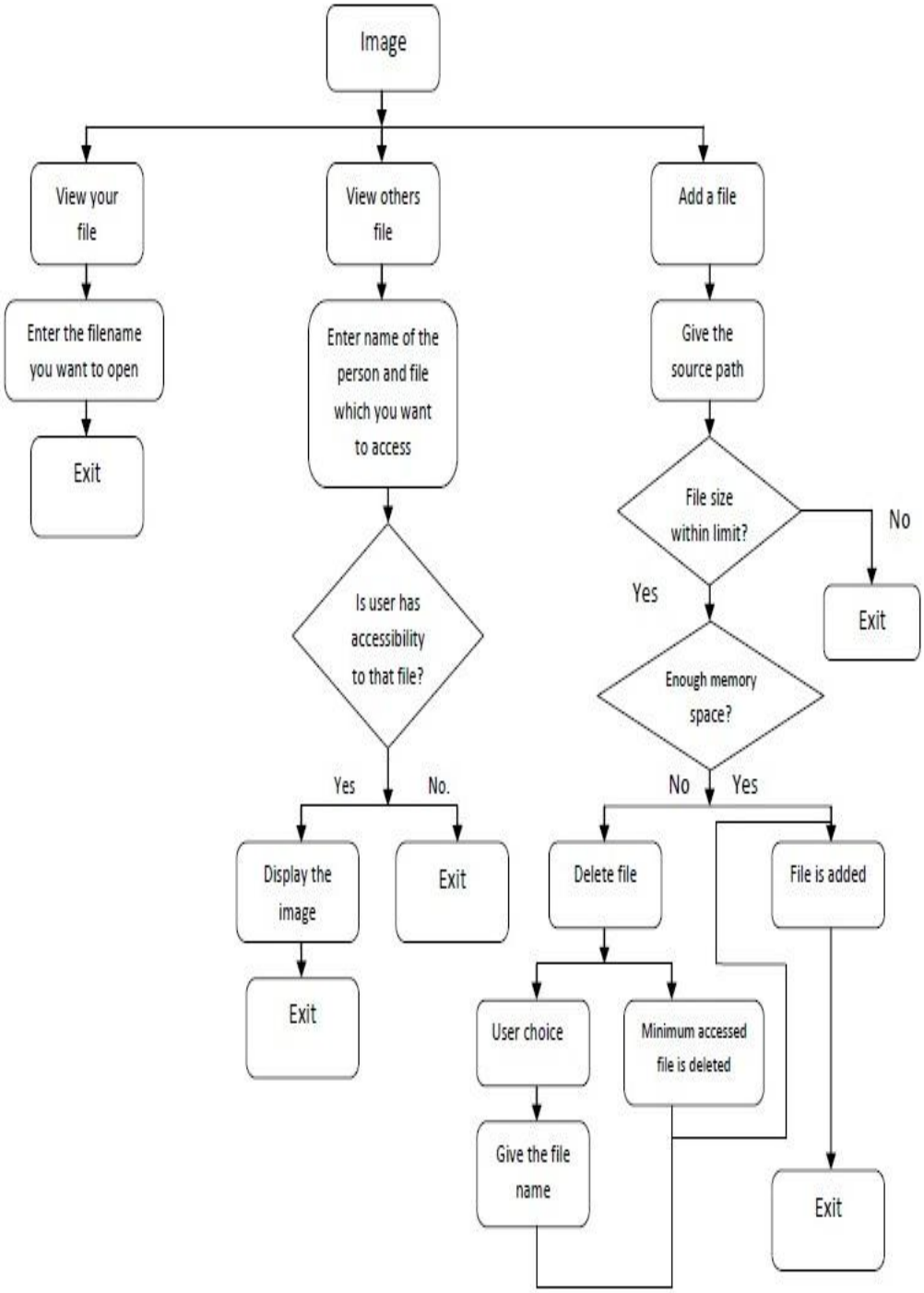
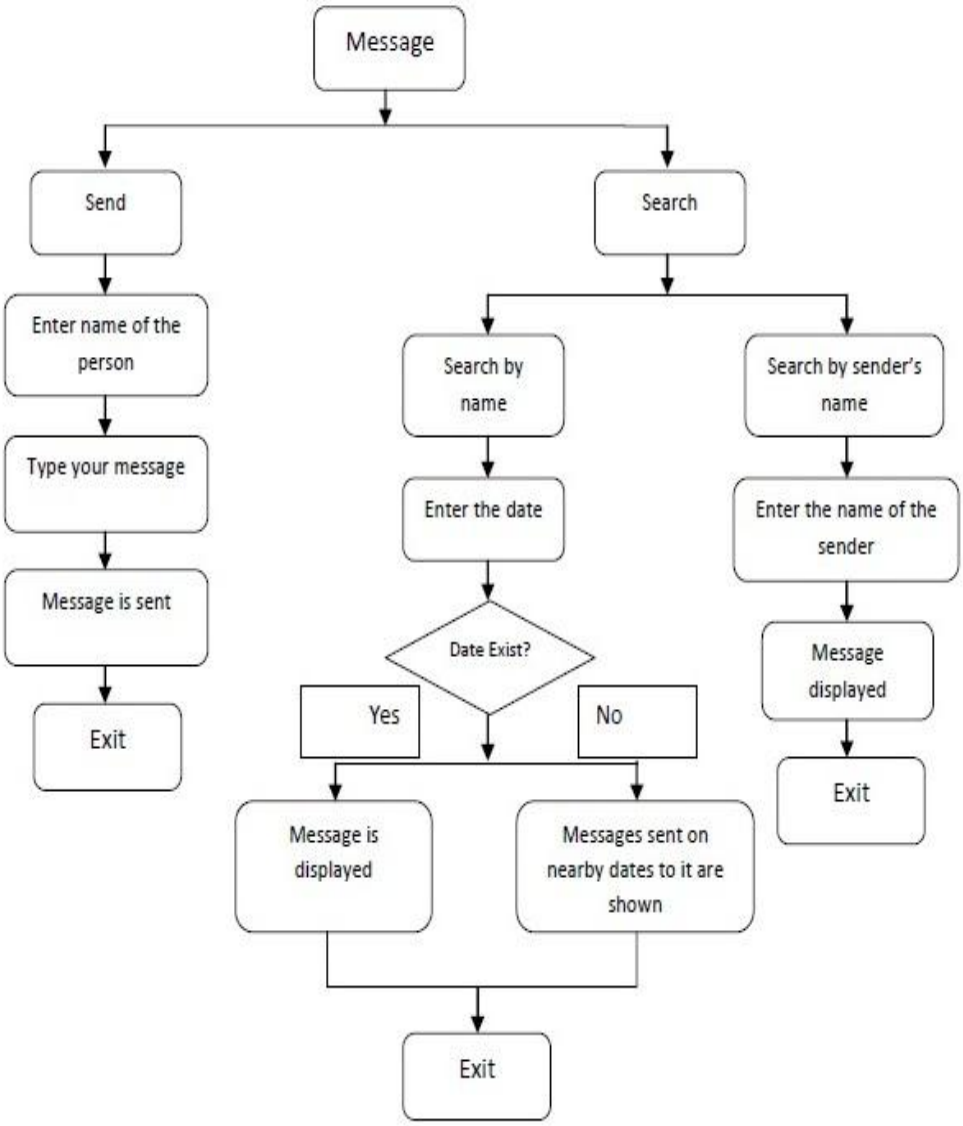


Image
View your file
View others file
Add a file
Enter the filename you want to open
Exit
Enter name of the person and file which you want to access
Give the source path
File size within limit?
No
Yes
Exit
Is user has accessibility to that file?
Enough memory space?
Yes
No.
No
Yes
Display the image
Exit
Delete file
File is added
Exit
User choice
Minimum accessed file is deleted
Give the file name
Exit



III. Data Structure

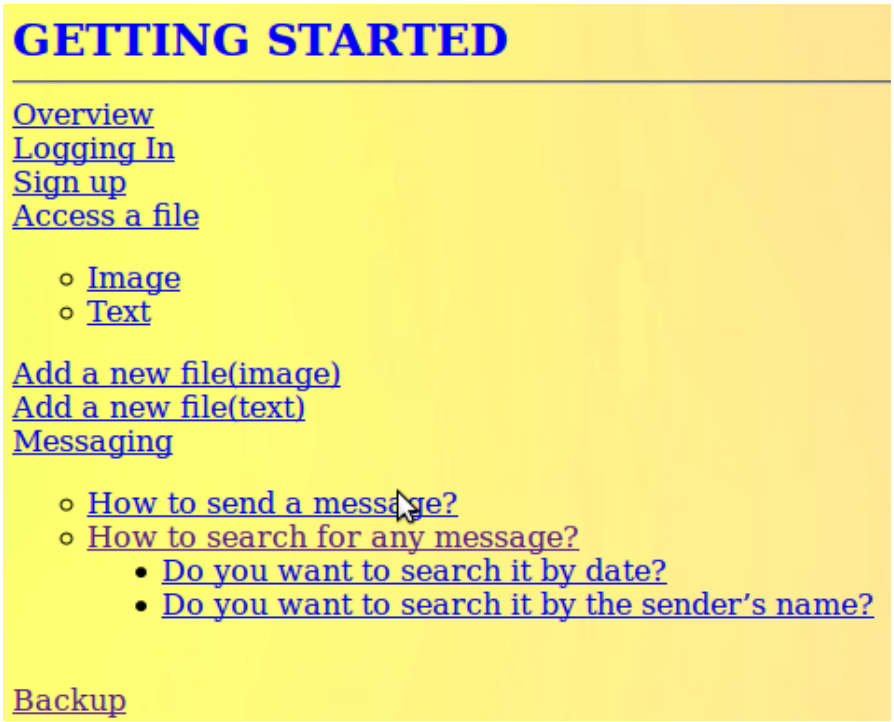
Dictionary: A mutable associative array of key and value pairs. It can contain mixed types (keys and values). Keys must be an immutable type. Here we use the following dictionaries:-
Permission={'Maumita': [(['DSC_0476.jpg', ''], 'r')], 'Sridisha': [('DSC_0476.jpg\n', 'r')]}
List: Mutable list, can contain mixed types.
E.g. total=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']

Tuple: Immutable, can contain mixed types.
E.g. (" Hello all! ",2,"EE")

File: Afile is a collection of data stored on mass storage (e.g., disk or tape). The data is subdivided into records. Each record contains a number of fields. One (or more) field is the key field.
We have used many text files in our project to store various information of the users like their Login-id, password, the name of the text files and image files which they can access etc.

IV. Results

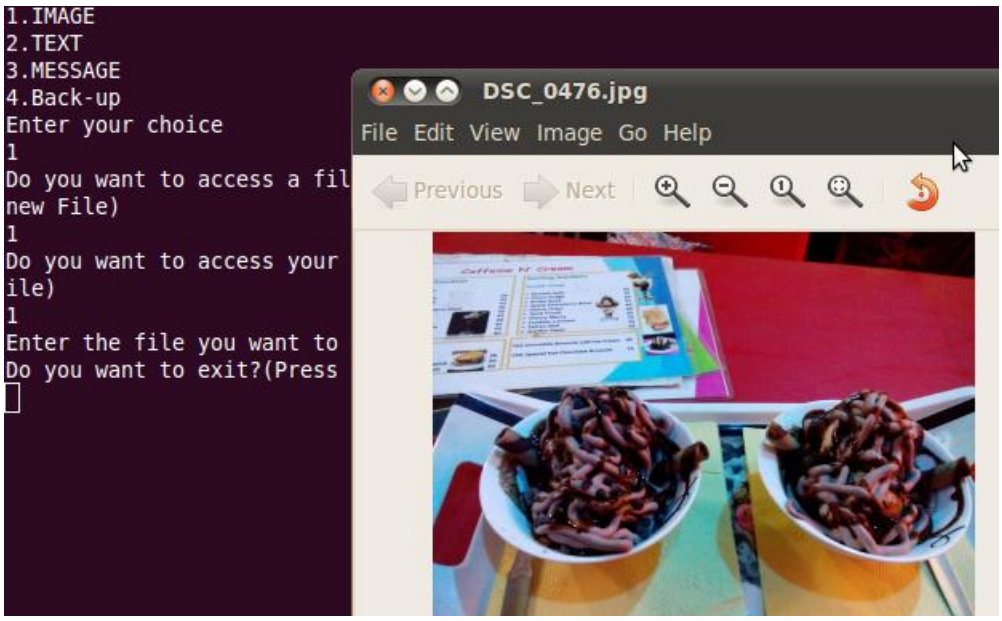
1. Main Page where Signing-up is done if new user wants to access and then he/she is logged in.



2. When help option is selected User Guide is displayed.

```
1.IMAGE
2.TEXT
3.MESSAGE
4.Back-up
Enter your choice
1
Do you want to access a file or add a new file?(Press 1=Access Press 2=Create a
new File)
1
Do you want to access your file or someone else's file?(Press 1 to access your F
ile)
1
Enter the file you want to open:- DSC_0476.jpg
```

3. Image is selected for access (similarly for the text part).



4. Image is opened for viewing.

```
1.IMAGE
2.TEXT
3.MESSAGE
4.Back-up
Enter your choice
1
Do you want to access a file or add a new file?(Press 1=Access Press 2=Create a
new File)
2
Enter the Source path:
/home/maumita/DSC_0476.jpg
Some item needs to be deleted
Do you want to delete it yourself otherwise system will delete it?(Press 1 for y
es)
1
Enter the file name you want to delete
1.jpg
```

5. If new file is to be added then allocated memory is checked and if enough space not available some file needs to be deleted.

```
1.IMAGE
2.TEXT
3.MESSAGE
4.Back-up
Enter your choice
1
Do you want to access a file or add a new file?(Press 1=Access Press 2=Create a
new File)
2
Enter the Source path:
/home/maumita/DSC_0476.jpg
File size larger than permissible size..Try again
Do you want to exit?(Press 1=>YES)
0
Do you want to access a file or add a new file?(Press 1=Access Press 2=Create a
new File)
```

6. If new file that is to be added is more than the permissible file size then it is not accepted.

```
1.IMAGE
2.TEXT
3.MESSAGE
4.Back-up
Enter your choice
3
You have no unread messages...
Do you want to view all your read messages(Y/N)==>N
1.Send Message
2.Search any Message
1
Whom do you want to send?==> Maumita
Write your message==> Hello!..When will you come?
```

7. Message is sent.

```
maumita@maumita-laptop:~/new_project$ sh menu.sh
1.IMAGE
2.TEXT
3.MESSAGE
4.Back-up
Enter your choice
3
You have 2 unread messages
Do you want to view the unread messages(Y/N)==> Y
Arunima ==> Hello dear!!how are you?    Aug 24 21:07:13 2014
Debasmita ==> Hello    Jul 21 01:55:02 2014
Do you want to view all your read messages(Y/N)==>N
1.Send Message
2.Search any Message
2
Search
1.Search by date
2.Search by sender
2
Whose message do you want to search==>Arunima
Hello dear!!how are you?    Aug 24 21:07:13 2014
```

8. Unread messages are displayed and message is searched (here according to sender's name, similarly for search by date).

V. Conclusion

So as to conclude we can say that this offline content management system is an efficient tool to manage document without taking the help of web. Apart from the different features described in previous sections this offline data management system has its own advantages. The ability to work on the content even when disconnected from the internet, no online database requirements to dispense content makes this system feasible and user-friendly.

In addition to easy data accessibility, data security and compression of files gives this system a superfluous configuration. Hence in a nutshell this system provides a well-coordinated platform for the management of data items with utmost security.