IT-632-Software Engineering

Multiple Cloud Account Manager

Feasibility Analysis Version 1.0

Team-2

Instructor – Professor Asim Banerjee

Date: 24th August, 2013

Table of Contents

1. Project Specification/Overview

In this project, we intend to build a web based software, in which a user can add his multiple cloud accounts. Basically we intend to add support of multiple cloud account providers. A shortlist of those providers was prepared which is as follows.

1. Dropbox
2. Google Drive
3. Box
4. SkyDrive

A Brief research was done, which revealed that such a system is possible. Also, there exists many apps in the Apple AppStore which already provided such functionality. Thus, we thought to take it a step further. A system which allowed the user to upload a file of any size. The uploaded file should be uploaded by distributing the file into different pieces and stored into the multiple accounts of the user automatically. The system should be capable of managing the same. This will allow an unlimited space on the cloud. The splitting and joining of the file can be done using HJSplit for Java. It is a small .class file which has the facility to split files into unequal pieces.

Hence, the main idea behind this project is to use the free space available on the internet, and save your file distributed within multiple cloud

space providers. The file remains secure on the internet. Even the cloud space provider won't be able to use your file because they will have only a part of your file.

2. Feasibility

2.1 Technical feasibility

I. The project is supposed to be a web portal so we would majorly be dealing with HTML, CSS, JavaScript, AJAX and JQuery.
II. The back-end or the server-side scripting language we will be using is Java which is a popular and highly used open source programming language. So, we would be getting help and learning new concepts from various forums regarding the java language.
III. We are planning to explore and work on Struts 2 application framework for Java web development, which is also very good documented and highly renowned.
IV. The database that we are going to use would be either MySql or PgSQL which are both open source.
V. The team needs to learn as many as 6 to 7 different API's and a person who is capable to integrate all of them into a single working system.
VI. The project is too large for a team of 10 who have no experience in the API's needed in this project.
VII. A strong and robust system is needed. Because we are using 3rd party cloud storage providers. So there is always a chance that the provider API may malfunction unexpectedly.

VIII. Server speed may be a problem, because even the 3rd party providers also use someone else's servers to store their data. Thus, it needs strong knowledge of the API's and a lot of experience to work with the problems which may occur.

IX. The entire project may malfunction if the provider changes or upgrades the API's.

X. High internet speed is required to run this type of software smoothly.

2.2 Financial feasibility

I. The tools required to develop the application are largely open source so there would be no cost involved on the software side of the project.

II. The server costs if hosted on the Internet are to stand not too high for a strong and robust system.

2.3 Scheduling feasibility

I. As multiple API's are included and expertise in all of them is necessary is quite a challenge to create a timeline in which all the modules can be implemented.

II. The only thing which is possible in this scenario is that firstly multiple API's basic functionality classes be implemented, and then similar functions for all the API's be made to co-ordinate the process for all the API's.

III. At the end all of them have to be implemented altogether into single system. Which is quite a tedious task to do. Mainly, different API's have been implemented in different ways, also by different people. Thus, debugging all of them together will be tiresome.

2.4 Operational feasibility

I. A server with unlimited storage and bandwidth will be required to host such a web software.
II. As this is a commercial application, no special hardware requirements are required. Any system with a web browser can use it.
III. A time to time check has to be maintained if the API provider has made any changes which may lead to an unexpected malfunction.

3. Cost Benefit Analysis

3.1 Costs

- Nominal server hosting is required, as we are not using any kind of space of our own. Instead we are managing the space provided on the internet for free, which helps us saving heavy server costs.
- The development cost can be a disappointment. Not monetarily, but other aspects like the time provided, the skill required is too high to build such a system.

3.2 Benefits

- No high-end system required to run this software. A nominal personal computer can be more than enough to runt this system.
- The user can access his data of any size from anywhere in the world.
- Given enough bandwidth speed, the basic limitation of physical memory will be no more. In any of the devices supporting the software.

4. Conclusion

Overall, if the system being developed, will be a great help for the society especially in the western countries where internet speed is not much slow than file transmission speed of a personal computer/flash drive. But, the basic drawback is that this kind of system cannot be implemented within the resources we have at our disposal. The skillset which we possess may help us to build such a software, but the limitation of the required time still persists. Hence, the system is not feasible.

• • •