Initial Plan

Project DIY Cloud

Ka-Him, HO

Supervisor: Ralph Martin

Moderator: Irena Spasic

Module: CM3203 One Semester Project (40 Credits)

February 2, 2015

Project Description

The aim of this project is to build a DIY cloud solution, based on an FTPS server controlled by the user. Nowadays, more and more general users using cloud services and cloud storage. However with the increase in concern about data privacy, people may need to trade-off between the convenience of storage services that synchronise automatically in background and the concern about service providers giving out data to government[1]. In addition, data ownership is a problem with third-party cloud solutions, it is debatable that data uploaded to providers' servers own by which parties.

Many individuals or companies have their own FTP(S) servers. By using own FTP(S) server, we can ensure data is not giving out without notice, but these are not applications, they are file transfer protocols, they do not synchronise data in background automatically like Dropbox and Google Drive. Therefore, we need a tool that automatically synchronise data in the cloud to a local computing device like various cloud solutions do, in a form that based on FTPS server controlled by the user.

The client application will be provided in various platforms, for examples, Android (Mobile device), and Mac or Windows (Desktop). Users will be able to select folders to be synchronised to FTPS server. Any changes on the master copy on the server will download automatically in background in a time frame. Thus, any updates to local files will upload to the server automatically. There are several major concerns of this project, including, issues like clients going offline for a long time, conflicts if two clients change the file at the same time, battery usage on mobile devices, data loss during updating the file, ease of add and remove files, and ease of selection of folders to be synchronised.

This application will benefit users that want to use their own controllable FTPS server as the backend of cloud solutions, who need to synchronise data between different clients.

Project Aims and Objectives

The aim for this project is to create an application that automatically synchronise files between client and FTPS server where directories to be synchronised are controllable by the user.

1. The application should store a master copy of selected files on FTPS server.
2. The application should do file exchange and connection using FTPS protocol.
3. The application should download updates when master copy on the server changes.
4. The application should upload files when local copy changes
5. The application should run in the background without command synchronous action manually or any user interaction after the initial setup.
6. The application should handle conflicts like updating the same file at the same time.
7. The application should handle users that offline for a long time of period.
    7.1. Either save a copy of list of master file states and compare which changes
    7.2. Or able to notice user that they have a long period of time have not update, therefore replace the whole directory selected.
8. The application should consider battery usage on mobile devices.
    8.1. The application should give users control on synchronous in certain condition. I.e. in mobile network or/and in Wi-Fi network.
    8.2. The application should give users control the interval of check time. I.e. able to custom check every X seconds or minutes.
9. The application should avoid data loss while updating local and server's data.
    9.1. The application should not delete any files before it ensure files updated successfully.
10. The application should be easy to select or remove directories to be synchronised.
11. The application should be easy to add or remove files in selected synchronize directory.
12. The application should be implemented on various platforms, e.g. Mac or Windows (Desktop), and Android (Mobile).

Work Plan

Below highlights how the aims and objectives are broken down across the project timeframe and the expected deliverables for each. A scheduled weekly supervisor meeting will be taken to ensure the project progress as planned. In order to better documenting the progress of implementation, choices made for each steps when implementing various solutions will be recorded and written in the final report.

Week 1-2, 26 January 2015.

- Meeting to discuss initial plan
- Background research
    - Research on similar open-source projects and FTPS client projects on different platforms and languages
    - Research on existing solutions to solve conflicts when synchronised client offline for a long time, and 2 or more client update at the same time.
    - Conduct requirement analysis for each component of the system and document these requirements. Design the functionality of the system.
    - Research on IDE features and how to use them.
- Work on android samples code
- Implementation (Android)
    - basic FTPS connection for uploading and downloading files.
- Milestone: Submit Initial Plan, 2 February

Week 3-4, 9 February 2015.

- Understanding the way relevant Android API works and try different functions which will be used in the Android application, including:
    - access and write to data storage
    - background job scheduler, asynchronous task
- Implementation (Android)
    - Selection of directories to be synchronise
    - Comparing two file directories.
- Implementation (Desktop)
    - basic FTPS connection for uploading and downloading files.

- Design the user interfaces for the application, taking into account the usability features researched, by creating a non-functional prototype of each view.
- Create a use case diagram to model interactions between the user and different system components.
- Create a class diagram to model structure of the system.
- Milestone: Design phase complete.

Week 5-6, 23 February 2015.

- Implementation (Android)
    - Handling conflicts when two file update at the same time.
    - Handling user offline for a long time.
    - Application interfaces.
- Implementation (Desktop)
    - Selection of directories to be synchronise
    - Comparing two file directories.
    - Application interfaces.
- Milestone: complete manually synchronous files with FTPS server

Week 7, 9 March 2015.

- Buffer week, allowing time for the unexpected.
- Project progress review meeting

Week 8, 16 March 2015.

- Implementation (Android)
    - synchronous in background.
- Implementation (Desktop)
    - Handling conflicts when two file update at the same time.
    - Handling user offline for a long time.
    - synchronous in background.
- Complete all the layouts and designs for both mobile and desktop applications.
- Milestone: Implementation phase complete.

Week 9, 23 March 2015.

- Refine and test the application. (Ongoing)
- Produce test cases based on the requirements specified and the intended functionality.
- Conduct the testing using the test cases created.

Easter Recess, 30 March 2015.

- To be used if project runs over plan deadlines.

Week 10-12, 20 April 2015.

- Project progress review meeting
- Evaluate the system using results from testing.
- Report writing (Ongoing)
    - Overall project findings from research
    - Draft design of the application
    - Complete code of the application
    - User guide for the application
    - Testing application results
    - Identify and document future improvements.
    - Write up reflection of project and finalise report.
- Milestone: Submit Final Report, 5 May, and Project VIVA.

Reference

[1] Lucas Mearian, *No, your data isn't secure in the cloud [Online]*, Computerworld, 2013. Available at: http://www.computerworld.com/article/2483552/cloud-security/no--your-data-isn-t-secure-in-the-cloud.html [Accessed : 30 January 2015].