CloudSync
Software Requirements Specification

Version 1.1

Project Supervisor: Prof. Gihan Dias

Co- Supervisor: Dr. Dilum Bandara

Group Members:	090022B - P.D.D.A. Anicitus
	090236J - Y. Jenananthan
	090324D - M. Mayuresan
	090471A - S. Sathyavarathan

1. Introduction

1.1 Purpose

The purpose of this SRS is providing scope, functional and nonfunctional requirements of the project CloudSync, a file synchronization application. This will be used by the project team during the design and development phase.

1.2 Scope

The project will be a file synchronization application, which keeps most up to date copies of the same set of files in many geographically distributed devices. This project will result an efficient file synchronization software, to low bandwidth connection, comparing to currently available solutions by applying new mechanisms.

- avoid sending a file that is already exists in the server
- send the difference only when a similar file exists in the server
- avoid uploading files that are downloaded from internet by sending its source URL to the server to be downloaded.
- keep templates of file types, e.g : doc, so that its enough to send the contents of the file only

The file synchronization application will have two components, one for server side and one for client side. For the moment there is one device and cloud but our design should be able to extend to more devices in future.

1.3 Definitions, Acronyms, and Abbreviations

Term	***Description***
URL	Uniform Resource Locator
Sync	Synchronization
User directory	Directory in the file system of any device, selected by the user during installation and copies of which will be kept in all locations used for synchronization

1.4 References

http://www.ts.mah.se/RUP/RationalUnifiedProcess/webtmpl/templates/req/rup_srs.htm

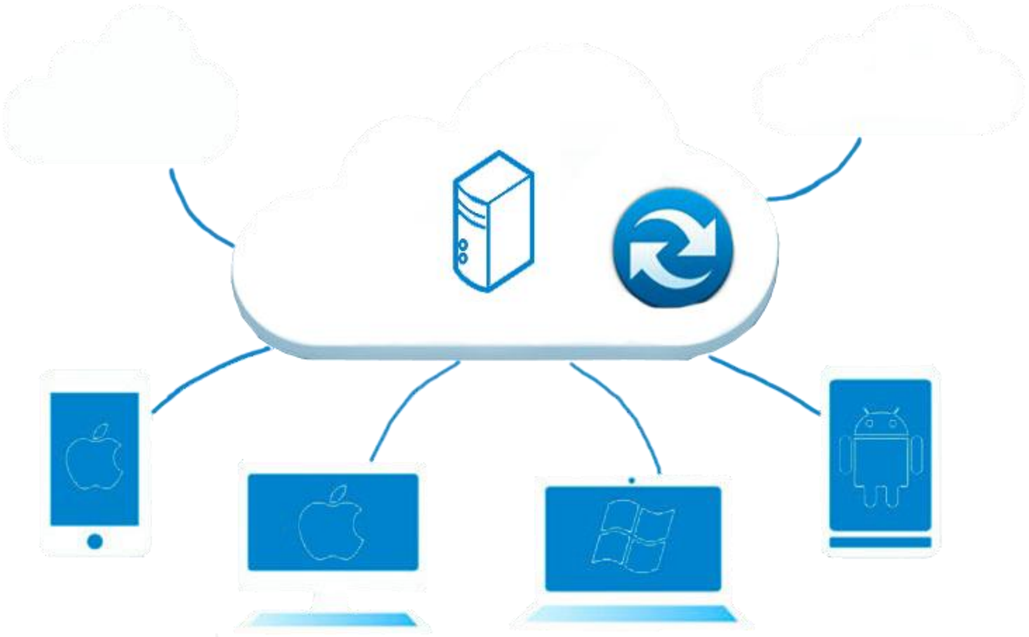
1.5 Overview

This document contains a well structured approach to describe the overall initial requirement specification of the system. First, a brief overview of the system is done through an overall description. Then, the document continues on describing the specific requirements, divided into two main categories as functional requirements and non-functional requirements. Description of these requirements is done in detail in that section.

2. Overall Description

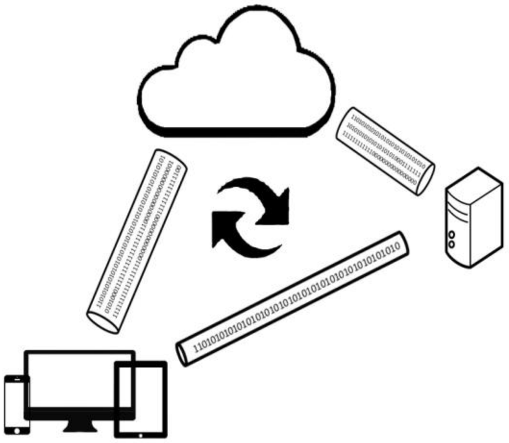
2.1 Product Perspective

Today people use different devices on different occasions like PCs at work, Laptops at home, tablets and mobiles on the go. It is vital for users having access to their files regardless of where they are and what they do. Nowadays it has become common that users work from home where it is inevitable to collaborate with other co-workers working on the same set of files. Above all it is very important not to have a situation where losing a single device would cost us all very important files. Existing file synchronization systems use some protocols which are proved to be inefficient and not optimized for users with low bandwidth connections, although low bandwidth networks are common and users have to pay to ISPs based on their data usage.



2.1.1 System overview

The following design shows the system overview.



There will be a centralized server, in which the server side application will run in the background, being plugged in the file system, notifying any changes in that to the clients, changing its file system based on the changes in the corresponding clients, and displaying a control panel on request for the user to change any configurations or preferences.

There will be clients which contains the similar application but with less scale, less CPU consumption and less memory usage, running in the background and will be notified any changes, is plugged into clients file system and with a user control panel.

The server will have a high bandwidth internet connection, while the client devices will have low bandwidth ones. Copies of the synchronized files will be stored in a directory in the file system of the server, which is allocated to the specific user, and in the file systems of the users' devices.

2.2 Product functions

- Detecting new files and copying
- Identify file changes and transfer
- Detecting file movements, deletions and do same actions on the other side
- Renaming files and folders should be done to other locations' files
- Select execution time from control panel

2.3 Constraints

- Expensive internet connections, for which users have to pay based on the data usage, so for a cost effective synchronization, so number of bytes transferred should be as minimum as possible.
- Low speed internet connections, so synchronization should not occupy the bandwidth for a long time to make other network applications starve.
- Connection can be closed at any time, so synchronization should be able to resume whenever the connection is established again
- Should run on mobile devices, which have low computational power
- Should be done while user is online of which the time is limited

2.4 User characteristics

Once system is installed in the device, user will not often interact with this system directly like normal software applications like Word, Powerpoint etc. He will rarely interact with the system in case of any configuration changes.

- Uses mobile devices like laptops, smart phones, iPads
- Familiar with basic graphical user interface components of the operating system
- Have minimum technical knowledge
- Ability to read and understand English

2.5 Assumptions and dependencies

The client has a low bandwidth connection with server and the internet while the server has high bandwidth connection to internet.

Since client has low bandwidth connection with server real time synchronization is not required, files can be synchronized whenever bandwidth is free.
Same file in the client and server will not be changed at same time, hence there will no conflicts handling needed.
If a file is downloaded from the internet, the source URL will be a direct download link (which should not need login information, entering captcha etc)
Root access will be given to server component

3. Specific Requirements

3.1 Functionality

3.1.1 Select user directory

The user should be able to select a user directory in his/her device's file system, which will be synchronized with others.

3.1.2 Adding a new file in the user directory should be reflected in other locations

File added to the user directory should be eventually added in other synchronizing locations' user directories. This can be done at the next time the other client opens the directory, because manipulating other places while they are working can hold their work

3.1.3 Changes done in file in the user directory should be transferred to other locations

When a user opens a file in the user directory edits and saves, the changes done in the file should be done in the corresponding files in other locations.

3.1.4 File deleted or removed from the user directory should be deleted in other locations

When a file is deleted from the user directory or moved away from it, the corresponding file in the other locations should be removed

3.1.5 File moved within the directories inside the user directory should be moved simultaneously

3.1.6 Renaming a file or directory

When a file or directory is renamed in one location the system should rename it all corresponding locations

3.1.7 Server should keep logs about the file changes, uploads and deletes

3.2 Usability

3.2.1 Getting used to the system

Once the setup is done, application will be almost invisible to the user; there will be small icon in the taskbar which will pop up a small control panel when user clicks it. Through this control panel user can pause and resume sync, also user can remove and change the user directory.
The user, who has used similar systems before should not find much differences in the interface or processes. New users, who have basic knowledge of using file systems and graphical user interface of the operating systems can easily understand. Additionally every user will be provided with a manual with all the descriptions of the functions

3.2.2 Easy to understand

The control panel will include Basic English, which can easily be understood by a normal user of information systems, without any technical knowledge

3.3 Availability

There will be a very slow connection between the server and client.
Server availability is high for any user.
Client availability might be fairly low.

Availability of server should be more than 99%. In order to have 99% availability server can be down only for approximately 53 minutes during a year and recovery time should be minimal.

3.4 Performance

System should reduce the network bandwidth usage by 15% to 80%
CPU consumption < 5 %,
Memory usage < 20 %
By addressing following techniques

- When the file is already in the server, no need to send the entire file and check, rather send the checksum of the file to server and check whether any file has same checksum in server.
- When there is a similar file in the server send only the difference between those files
- When a file needs to be synced which is already downloaded from the internet, just send the source URL of that file to server, and server will download it from the internet.
- When there is a unique file to be synced, templates of the files can be used in order to minimize the data transfer over network by sending only the content of that file.

As the importance is reducing the network bandwidth usage, the system can take an acceptable amount of storage

3.5 Security

- Data should not be corrupted while transmitting over the network, checksum of the files should be checked in order to ensure there is no corruption
- For the user authentication and authorization required for the application, an efficient, and available solution component can be used
- Encryption will be used while transferring the files for secure data transfer

3.6 Maintainability

Software must have a simple structure to avoid complication after project completion. Full documentation must be provided for maintainability and transferring the software from our development environment to other production environment.
Application developers will maintain the same code base for the two or more platforms.

3.6 Design Constraints

3.6.1 Language to be used for development

As the application should run on both windows and Linux environment, language should not be platform dependent.

3.7 On-line User Documentation and Help System Requirements

A website will be maintained for the project, which will contain manuals for the project and demonstrations as images or videos. Forums also might be maintained if possible.

3.8 Interfaces

3.9.1 User Interfaces

A user friendly small control panel which will indicate the sync status and recent changes

3.9.2 Software Interfaces

- Interfaces that interact with Operating System Kernel, to observe file system changes
- Server file system will be used for storing the files
- Interface to create and maintain logs periodically
- Interact with database for storing chunk details
- Plug-in for browser to detect file downloads and store the metadata with downloaded file checksum

3.9.3 Communications Interfaces

A port will be assigned to communicate with server whenever synchronization happens.
TCP will be used for synchronizing, as reliable connection is essential
Server side port will be always open and listening for the clients to connect.