Documentation of sprint 2

# Description of each work day

## Day 1

Day one we started out looking at our diagrams to see what updated had to be made, when that was done, we started coding on our separate parts of the program, extending functionality in all ends.

## Day 2

We did a lot of testing on day 2, to make sure everything was working as intended, found several minor issues which were corrected.

We also realized that the structure of our program had to be slightly readjusted to cope with requirements as well as just make more sense in the way data was stored and handled within the program.

## Day 3

We did a lot of researching on server functionality, and started implementing it as well as being busy with including new readjustments from day 2, the day was very productive and a lot of work was put into the program, resulting in good progress of the program.

# The programs functionality at the end of this sprint

## GUI

The User interface is nearing a finished state. All the controls the user needs to fully operate the program has been added. Some of these controls are not functional yet, and only serve as placeholders until the back end is ready to be tied up to these buttons.

This version of the User Interface does not include the recent design change our program underwent, that being the introduction of projects.

## Controller

The Controllers functionality is unchanged since the last sprint, but will be updated to allow interaction with server, as well as the remaining functionality that is required from our readjustment of the structure.

## Document

Our Document Class was changed in several ways since last sprint. The biggest change being the MergeWith function, along with several others.  
The MergeWith function now takes all changes in the document into account, and generates a fitting changeLog according to the changes.

A document is no longer “shared”, as that is handled by our new class “Project”.

## Folder

The folder class has not been changed since the last sprint.

## IFileSystemComponent

The IFileSystemComponent has not been changed since the last sprint.

## Doctype

The enum “IFileSystemComponentEnum” was renamed to Doctype, and a new third value was added, which is “Project”.

## DocumentStruct

This sprint also included the introduction of the DocumentStruct, which is used to store enough information to create a suitable Gui from, with information taken from the storage, without pulling out and handling large amounts of data, of which very little is needed. The documentStruct also inherits from IFileSystemComponent.

## Project

The Project class was added to allow users to shared entire “folders”, which are named projects. Projects function in almost the same manner as folders, being that they can contain children that are IFileSystemComponents, which is why it extends the folder class.   
Projects also include an owner and a list of users it is shared with.

## Storage

Our storage class can now save an object of our new Project class on the file system, it creates a folder and some metainformation(a .txt file) which says something about the owner of the project and who it shared with. We can also at this moment, given a project id, create a new project from the files the file system, and return the project object as it should look like for gui representation.

The storage was also updated to read and write documents with their entire log to and from the file system.

## Server

A lot of time during this sprint was spent doing reserach and considering the specifics of the implementation of the server. In its current state, the server is runnable and the client application is able to contact the server correctly. The server has no behavior implemented yet. The server is a WCF service hosted in a separate application.