Documentation of sprint 1

# Description of each work day

## Day 1

We spent all working hours on creating various design artifacts, including domain model, SSD’s and more. At the end of the day we had a good idea of how to start the coding the next day.

## Day 2

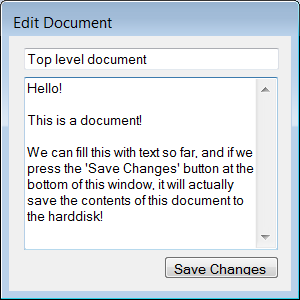
We started the day reviewing our design artifacts with a small meeting, and then sat down individually and coded separate parts of the program.   
At the end of the day a lot of coding had been successfully done, and we a lot of working functionality that had been tested.

## Day 3

On day 3 there was not a lot of work done due to other real life activities coming in the way of the work. Some slight coding, documenting and testing was done, but not at a satisfactory level.

# The programs functionality at the end of this sprint

## C:\Users\DE\git\SliceOfPie\Artifacts\Sprint 1\mainScreen2.pngGUI

The user interface is an initial version with basic functionality. The main window shows a tree hierarchy of the folders and documents created, and the user can open one of the documents and start editing. For now it only reads in a set of test data already defined in code, but it will save to the hard disk when the ‘Save Changes’ button is pressed in the ‘Edit Document’ window. (The ‘Create New Document’ button is currently just a placeholder)

## Controller

The controller is used as a gateway to the functionality that lies in the storage class, so it is not directly accessed from the GUI.

## Document

The document class is at this stage done unless new requirements for it arise. It holds all information related to a document, including a Log which holds entries on changes made to the document.   
It also holds functionality to merge the document with a newer version of the same document.

## Folder

The folder class holds information relevant to a folder in the file system, and functions to add and remove children, as well as getting all children.

## IFileSystemComponent

Our interface that covers over both folders and documents, not much to say other than it at this point serves its functionality.

## IFileSystemComponentEnum

An enum describing whether a component is a folder or a document.

## Storage

The storage at its current stage can perform most functionality needed from it, but still needs further development.  
At this stage it is able to create documents on the file system from a document object without a change log. It can also create folders from folder objects, even if they have a document inside or another folder it will still create a representation of it on the harddrive, but it still needs some optimization before it’s done.  
We can also read from the harddrive and make new documents, but not folders yet.

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

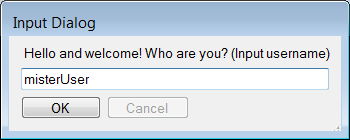
We did a lot of researching on server functionality, and started implementing it as well as being busy with including new readjustments from day 2, the day was very productive and a lot of work was put into the program, resulting in good progress of the program.

# The programs functionality at the end of this sprint

## C:\Users\Crelde\git\SliceOfPie\SliceOfPie\Artifacts\Sprint 2\mainScreen.pngGUI

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.

Documentation of sprint 3

# Description of each work day

## Day 1

We sat down looking at diagrams and models, checking which should be updated, and which are as they should, we also started looking at our webgui.

## Day 2

Almost the entire day was devoted to coding, spending more time than allotted, but spending the time well on implementing many features across the program.

## Day 3

Day 3 was very much like day 2, we spent way more time than was allotted to the day, but the time was spent doing hardcore coding implementing new features every single hour.

# The programs functionality at the end of this sprint

## OfflineGui

The “offlineGui” for clients working offline is completely done, all it needs is to be hooked up with the server.

## Controller

The controller has been expanded with a lot of functionality to cope with all the new features that were added during the sprint, including pictures, projects and more.

## Document

Our document Class has been updated to be able to contain pictures, as well as slight fixes to the way the documents log is written.

## 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

The folder class not been changed since the last sprint.

## Project

The Project class not been changed since the last sprint.

## Storage

Our storage has received its final additions, including server functions, improved functionality to functions like reading and writing to files, getting the hierarchy. New functions include adding and deleting pictures.

## Picture

The picture class was added to allow documents to hold images as well as their Id.   
The picture Class simply has two properties, an Id and a System.Drawing.Bitmap, being the picture object.

## Server

The server has been deployed and is fully operational, all that’s needed is for it to be implemented by our various clients.

## WebGui

Our program now also has the online client, being the WebGui. At this stage it holds functionality to open and edit both folders and documents. For it to be completed it still needs picture support and possibly some styling, as well as allowing editing and creation of projects.