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

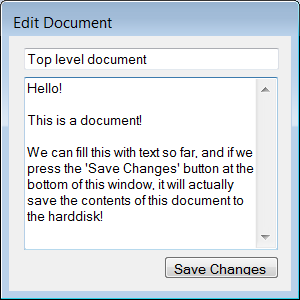
We started the day reviewing our design artifacts with a small meeting, and then sat down individually and coded separate parts of the program.   
At the end of the day a lot of coding had been successfully done, and we a lot of working functionality that had been tested.

## 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It also holds functionality to merge the document with a newer version of the same document.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.