# Vision document

We want our front-end for our clients to include an Explorer-like system, with some directories in which the clients’ documents are supposed to be, clicking on the document would then open it in a different window and there would be options to save and delete the document, along with other trivial document editing services. We want to be able to let several users use the same document at the same time without problems.

We wish for our proof of concept to be simple rather than overly complicated packed with features, but instead an intuitive working prototype, demonstrating the basic features, though as we progress we might want to add additional features.

# Use cases

Create new document

Bla bla bla <short text describing the use case goes here>

Change the name of a document

Equally bla

Delete a document

Save a document

“Open a document”?

Share a document with another user(s)

View all documents user has permission to edit

Insert picture to document

# Initial project Glossary

Document

The text document the user can insert images and text into.

Explorer system

Like the explorer from windows where you can browse through directories and files.

# Iteration plan

We plan on doing a total of 3 sprints, spanning 3 days each, starting on the 4th of December, ending the 13th of December. After this we plan on doing a release sprint until the project is ready for release.

Each member of the Scrum team will roughly spend 6-7 hours per day, 7 days per week.

# Furps+

Functional

The following points describe functions we want from our program.

* The ability to include both text and images.
* Allow the program to support being able to arrange the text files into folder and subfolders.
* Show a list of previous versions of the document.
* Synchronization of local content to a “server”.
* Ability to work while not connected to server.
* Changes to documents should be merged for everyone who has access to it.

Usability

Our program to be run exclusively in a GUI we set up, without any need for command line interaction.

The training time for a new user to be familiar with our program and its functionality should be minimal, since we want a simple and intuitive gui.

Reliability

If the system happens to crash the saved data should be kept intact via physical storage, which can then be loaded next time the program is opened.

Performance

The program should be able to support at least 3 users being connected to the server the time.

Supportability

The product code will follow the coding standards as well as naming convention that is commonly used in C#.

Implementation

Interfaces

Operations

Packaging

Legal