Abstract

In this report you will be able to read about our scrum-based development of our program “Slice of Pie”.

// NEEDS MOAR

# Indholdsfortegnelse

[Indholdsfortegnelse 2](#_Toc343467289)

[Software Analysis 4](#_Toc343467290)

[Use cases 4](#_Toc343467291)

[1: Create new document 4](#_Toc343467292)

[2: Change the name of a document 4](#_Toc343467293)

[3: Delete a document 4](#_Toc343467294)

[4: Open a document 5](#_Toc343467295)

[5: Save a document 5](#_Toc343467296)

[6: Create a project 5](#_Toc343467297)

[7: Choose a project to work in 5](#_Toc343467298)

[8: Share a project with another user 6](#_Toc343467299)

[9: Insert picture to a document 6](#_Toc343467300)

[10: View a picture attached to a document 6](#_Toc343467301)

[11: Remove picture attached to a document 7](#_Toc343467302)

[12: Rename folder 7](#_Toc343467303)

[13: Move object in explorer 7](#_Toc343467304)

[14: Synchronize local project with server 8](#_Toc343467305)

[15: Add project from server to offline client. 8](#_Toc343467306)

[Artifacts 8](#_Toc343467307)

[FURPS+ 9](#_Toc343467308)

[Software Design 9](#_Toc343467309)

[Class diagram 9](#_Toc343467310)

[Interaction diagrams 9](#_Toc343467311)

[Design patterns 9](#_Toc343467312)

[Software Architecture 9](#_Toc343467313)

[Architechture analysis 9](#_Toc343467314)

[Scenarios 9](#_Toc343467315)

[Factor tables 9](#_Toc343467316)

[Logical and deployment views (4+1)? 9](#_Toc343467317)

[Development documentation (Scrum) 9](#_Toc343467318)

[Testing, strategy and results 9](#_Toc343467319)

[Appendix 10](#_Toc343467320)

[HUSKELISTE: 11](#_Toc343467321)

# Software Analysis

## Use cases

In this segment we have only taken the use case titles and short descriptions, for a more detailed description, see appendix 1.

### 1: Create new document

The user wants to create a new document.

### 2: Change the name of a document

The user wants to change the name of a document.

### 3: Delete a document

The user wants to delete a document.

### 4: Open a document

The user wants to open a document he has selected in the explorer.

### 5: Save a document

The user wants to save a document.

### 6: Create a project

The user wants to create a project.

### 7: Choose a project to work in

The user wants to choose which project he would like to work in.

### 8: Share a project with another user

The user wants to share a project.

### 9: Insert picture to a document

The user wants to attach a picture to a document.

### 10: View a picture attached to a document

The user wants to see a picture that is attached to a document.

### 11: Remove picture attached to a document

The user wants to remove a picture attached to a document.

### 12: Rename folder

The user wants to rename a folder.

### 13: Move object in explorer

The user wants to move either a document or a folder to another folder in the explorer.

### 14: Synchronize local project with server

The user wants to synchronize his local project with the servers version of the project.

### 15: Add project from server to offline client.

The user wants to add a project that is shared with him on the server to his local client.

## Domain model

In the initial state of our program we had problems identifying the different domains that were relevant to the end-user. As a result of this our initial software architecture ended up being flawed, and that could have been avoided had we been better at identifying the different domains at an earlier stage.

In its final version, it is kept very simple, but it helped us identify the structure of how the user sees our program, and to build a program based on exactly that.

You can find the final version of our domain model in Appendix 2.

## Artifacts

* Vores forskellige artifacts

## FURPS+

# Software Design

## Class diagram

## Interaction diagrams

## Design patterns

* Everything regarding software-architecture
* Design patterns.

## Software Architecture

## Architechture analysis

## Scenarios

## Factor tables

## Logical and deployment views (4+1)?

## Development documentation (Scrum)

* Everything that regards to Scrum

# Testing, strategy and results

# Appendix

* Usecase model
* Class diagram
* Etc etc etc

# HUSKELISTE:

Server og klient “burde” ikke dele storage.

Vi har ikke kastet os ud i omfattende exceptionhandling :<

Et sprint i Scrum skal aldrig kunne have sin varighed eller sit mål ændret under selve sprintet. Selvom vi har tilføjet nye stories til sprintlog’en under vores sprints, har vi ikke ændret selve målet med sprintet, så noget har vi i hvert fald gjort rigtigt.

User kunne sagtens kun være en string frem for at være en hel klasse, men det er både mere scalable, mere ’typesikkert’ (somehow) og vi bryder os mere om at have User som sin egen klasse.

” Top Reasons To Not Go Scrum #5: you have a fixed deadline, with a fixed set of requirements”

SKRIV: HVORFOR VI IKKE LAVEDE COMMUNICATION DIAGRAM, VI MENTE SYSTEM SEQUENCE VAR BESKRIVENDE/LET FORSTÅELIGE.