Appendix

**Analysis, Design and Software Architecture-E2012  
Exam Project:  
Slice of Pie**



Kewin Bent Pedersen (kbep@itu.dk)  
Christian Martin Henriksen (cmah@itu.dk)  
Kasra Tahmasebi Shahrebabak (ktah@itu.dk)

Indhold

[Appendix 1: Use cases 4](#_Toc343485814)

[1.0 Revision Table 4](#_Toc343485815)

[1.1: Create new document 4](#_Toc343485816)

[1.2: Change the name of a document 4](#_Toc343485817)

[1.3: Delete a document 5](#_Toc343485818)

[1.4: Open a document 5](#_Toc343485819)

[1.5: Save a document 5](#_Toc343485820)

[1.6: Create a project 5](#_Toc343485821)

[1.7: Choose a project to work in 6](#_Toc343485822)

[1.8: Share a project with another user 6](#_Toc343485823)

[1.9: Insert picture to a document 6](#_Toc343485824)

[1.10: View a picture attached to a document 7](#_Toc343485825)

[1.11: Remove picture attached to a document 7](#_Toc343485826)

[1.12: Rename folder 7](#_Toc343485827)

[1.13: Move object in explorer 7](#_Toc343485828)

[1.14: Synchronize local project with server 8](#_Toc343485829)

[1.15: Add project from server to offline client. 8](#_Toc343485830)

[Appendix 2: Domain Model 9](#_Toc343485831)

[2.0 Revision Table 9](#_Toc343485832)

[Appendix 3: System Sequence Diagrams 9](#_Toc343485833)

[3.1 OfflineClient-Server SSD 9](#_Toc343485834)

[3.2 OfflineClient-Server SSD diagram Revision Table 10](#_Toc343485835)

[3.3 OfflineClient-Storage SSD 11](#_Toc343485836)

[3.4 OfflineClient-Storage SSD Revision Table 12](#_Toc343485837)

[3.5 WebClient-Server SSD 12](#_Toc343485838)

[3.6 Web-Server diagram SSD Revision Table 13](#_Toc343485839)

[Appendix 4: Scrum Burndown charts 14](#_Toc343485840)

[Appendix 5: Use case diagram 14](#_Toc343485841)

[5.1 Use case Model 14](#_Toc343485842)

[5.2 Use case diagram Revision Table 14](#_Toc343485843)

[Appendix 6: Class diagram 15](#_Toc343485844)

[Appendix 6.0: Class diagram 16](#_Toc343485845)

# Appendix 1: Use cases

## 1.0 Revision Table

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 04-12-2012 | Initial version | All |
| 04-12-2012 | Details and diagram added for Use case 1 and 2.[[1]](#footnote-1) | Christian |
| 13-12-2012 | Diagrams edited for use case 1 and 2.  Details and diagram added for use cases 2-8. | Kewin |
| 16-12-2012 | Major overhaul of how use cases are written. | Kewin |
| 16-12-2012 | Added use cases 9 to 15, and rewrote others. | Kewin |

## 1.1: Create new document

The user wants to create a new document.

Basic flow

1. The user chooses a place where the document should be and specifies a name for it.
2. The program saves the document to the storage, and updates the list of documents to include the new document.

Preconditions:

* The user must be logged into a client.

## 1.2: Change the name of a document

The user wants to change the name of a document.

Basic flow

1. The user chooses which document he wants to rename, and specifies the new name for it.
2. The program edits the document and saves the change to the storage, while also adding the change to the document’s logs. And then finally updates the list of documents to show the changed name.

Preconditions:

* The user must be logged into a client and be working in a project.
* The user must have access and rights to a document.

## 1.3: Delete a document

The user wants to delete a document.

Basic flow

1. The user specifies which document he wants to delete.
2. The program deletes the document from the storage, and updates the list of documents is show that the document is no longer there.

Preconditions:

* The user must be logged into a client and be working in a project.
* The user must have access and rights to a document.

## 1.4: Open a document

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

Basic flow

1. The user chooses which document he wants to open.
2. The program reads all info about the document from the storage, and shows it to the user.

Preconditions:

* The user must be logged into a client and be working in a project.
* The user must have access and rights to a document.

## 1.5: Save a document

The user wants to save a document.

Basic flow

1. The user has changed attached pictures and or changed the documents text.
2. The program saves the document to the storage.

Preconditions:

* The user must be logged into a client and be working in a project.
* The user has opened the document, and made changes to it.

## 1.6: Create a project

The user wants to create a project.

Basic flow

1. The user specifies he title for his new project.
2. The program saves the new project to the storage, and makes sure the user has access to his new project through the explorer.

Preconditions:

* The user must be logged into the web client.

## 1.7: Choose a project to work in

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

Basic flow

1. The user logs in with his username.
2. The program presents the user with a list of projects available to him.
3. The user selects which program he wants to work in.
4. The program gets the information about the project from the storage, and presents the project in the explorer.

Preconditions:

* The user must be logged into a client.
* The user must either own or have projects be shared with him.

## 1.8: Share a project with another user

The user wants to share a project.

Basic flow

1. The user writes the name of the person he wants to share his project with.
2. The program updates the storage to reflect that the project is now shared with the person the user specified.

Preconditions:

* The user must either be the owner of the project, or the project has to be shared with him.
* The user must have opened the project in the web client.

## 1.9: Insert picture to a document

The user wants to attach a picture to a document.

1. The user chooses a picture from his local file system to add to the document.
2. The program saves the picture to the storage, and updates the document to reflect that the picture is attached to it.

Preconditions:

* The user must be logged into a client.
* The user must have opened the document he wants to insert the picture to.

## 1.10: View a picture attached to a document

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

Basic flow

1. The user selects the picture he wants to view.
2. The program shows the picture to the user in a separate window.

Preconditions:

* The user must be logged into a client and be working in a project.
* The user must have opened a document that has a picture attached to it.

## 1.11: Remove picture attached to a document

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

Basic flow

1. The user chooses which picture in the document he wants to delete.
2. The program will remove the picture from the storage, and update the document view to reflect that the picture is no longer attached to it.

Preconditions:

* The user must be logged into a client and be working in a project.
* The user must have opened a document that has a picture attached to it.

## 1.12: Rename folder

The user wants to rename a folder.

1. The user chooses which document he wants to rename, and specifies the new name for it.
2. The program edits all documents in the folder and underlying folders’ paths and saves the changes to the storage, while also adding the change to all impacted document’s logs. And then finally updates the list of documents to show the changed name.

Preconditions:

* The user must be logged into a client and be working in a project containing a folder.

## 1.13: Move object in explorer

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

Basic flow

1. The user chooses which folder or document should be moved.
2. The program asks the user where the object should be moved to.
3. The user chooses which folder the object should be moved to.
4. The program changes the single documents path if the object was a document, or all documents the folder contained if the object was a folder. The program then saves the changes to the storage, and updates the explorer to reflect the changes.

Preconditions:

* The user must be logged into a client and be working in a project..

## 1.14: Synchronize local project with server

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

Basic flow

1. The user tells the program he wants to synchronize the project he is working in.
2. The program contacts the server and handles all synchronization with the server, and saves the changes made to the local storage, and updates the explorer to reflect changes made to local content.

Preconditions:

* The user must be logged into the offline client and be working in a project.

## 1.15: 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.

1. The user tells the program he would like to add a remote project to his local client.
2. The program contacts the server which supplies it with a list of project titles on the server that the user does not have in his local client, that he either owns or is shared with him.
3. The user chooses which project he wants to add to his local client from the list.
4. The program gets the project and all its content from the server, and saves it to the storage, and then finally updates the list of projects available on the local client to reflect the change.

Preconditions:

* The user must be logged into the offline client.

# Appendix 2: Domain Model

**Document**

Includes text and pictures.

**User**

Owns a

Is shared with

1

0..\*

**Folder**

**Project**

1

0..\*

1

0..\*

Contains

1

Contains

1

Contains

0..\*

0..\*

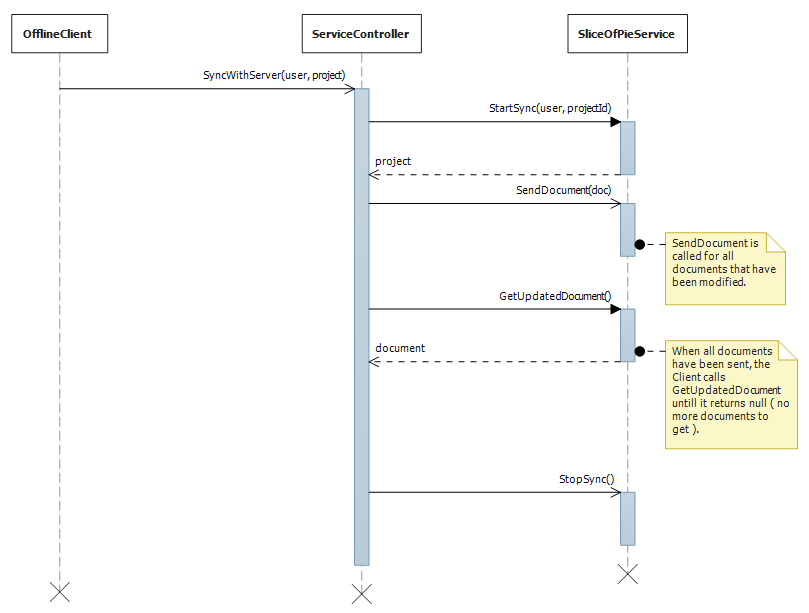
## 2.0 Revision Table

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 04-12-2012 | Initial version | Kewin & Kasra |
| 16-12-2012 | Updated model, added folder and project | Kewin |
|  |  |  |

# Appendix 3: System Sequence Diagrams

These diagrams were all created in our visual studio solution, and can also be found there in the project “UML Slice of Pie”.

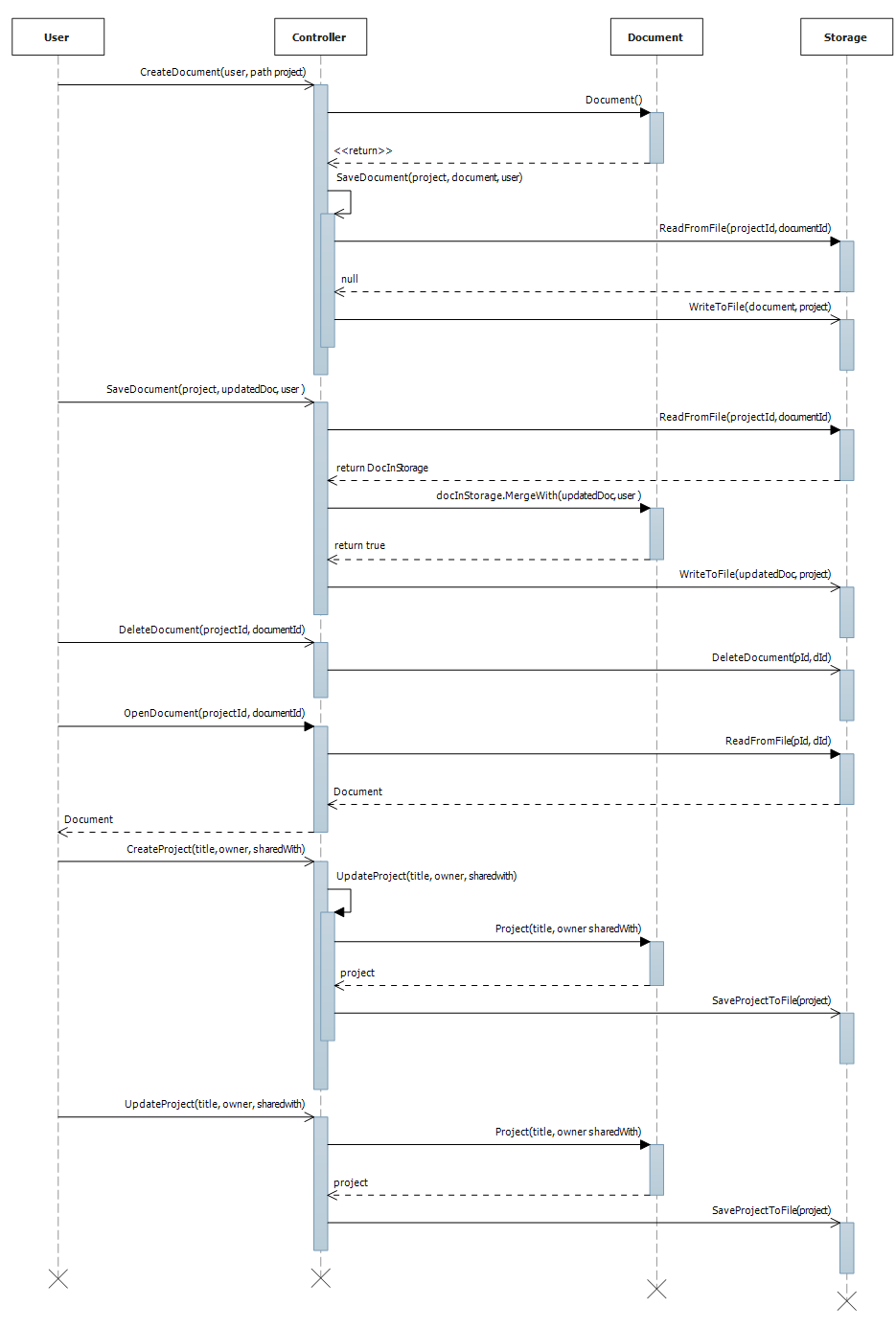
## 3.1 OfflineClient-Server SSD



## 3.2 OfflineClient-Server SSD diagram Revision Table

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 16-12-2012 | Initial version | Kewin |
|  |  |  |
|  |  |  |
|  |  |  |

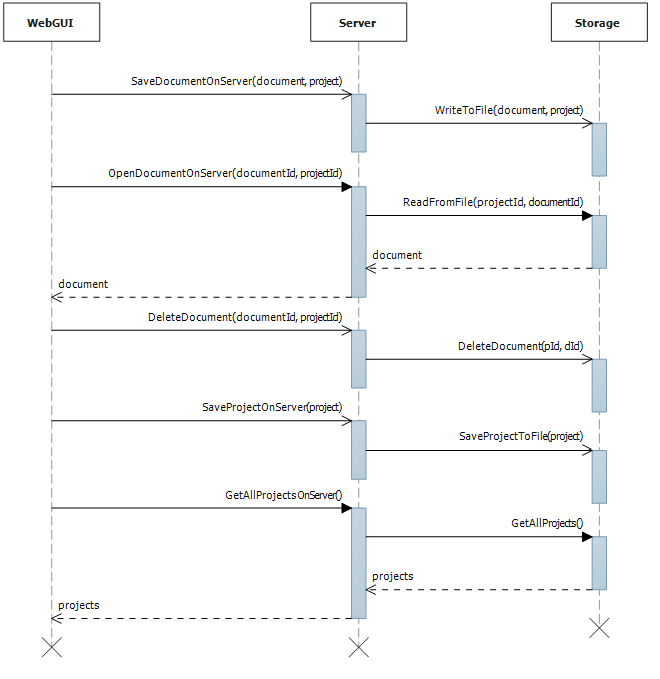
## 3.3 OfflineClient-Storage SSD



## 3.4 OfflineClient-Storage SSD Revision Table

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 04-12-2012 | Initial version | Christian |
| 14-12-2012 | Changed according to new architechture. | Kewin |
| 16-12-2012 | Small changes | Kewin |
|  |  |  |

## 3.5 WebClient-Server SSD



## 3.6 Web-Server diagram SSD Revision Table

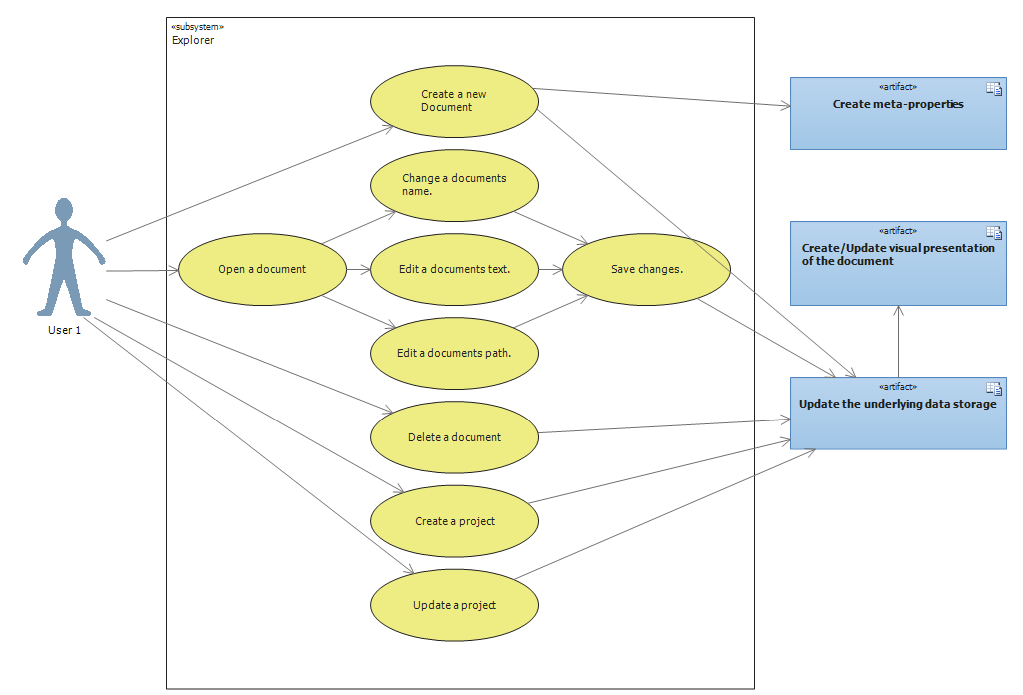
|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 16-12-2012 | Initial version | Kewin |
|  |  |  |
|  |  |  |

# Appendix 4: Scrum Burndown charts

# Appendix 5: Use case diagram

This diagrams was created in our visual studio solution, and can also be found there in the project “UML Slice of Pie”.

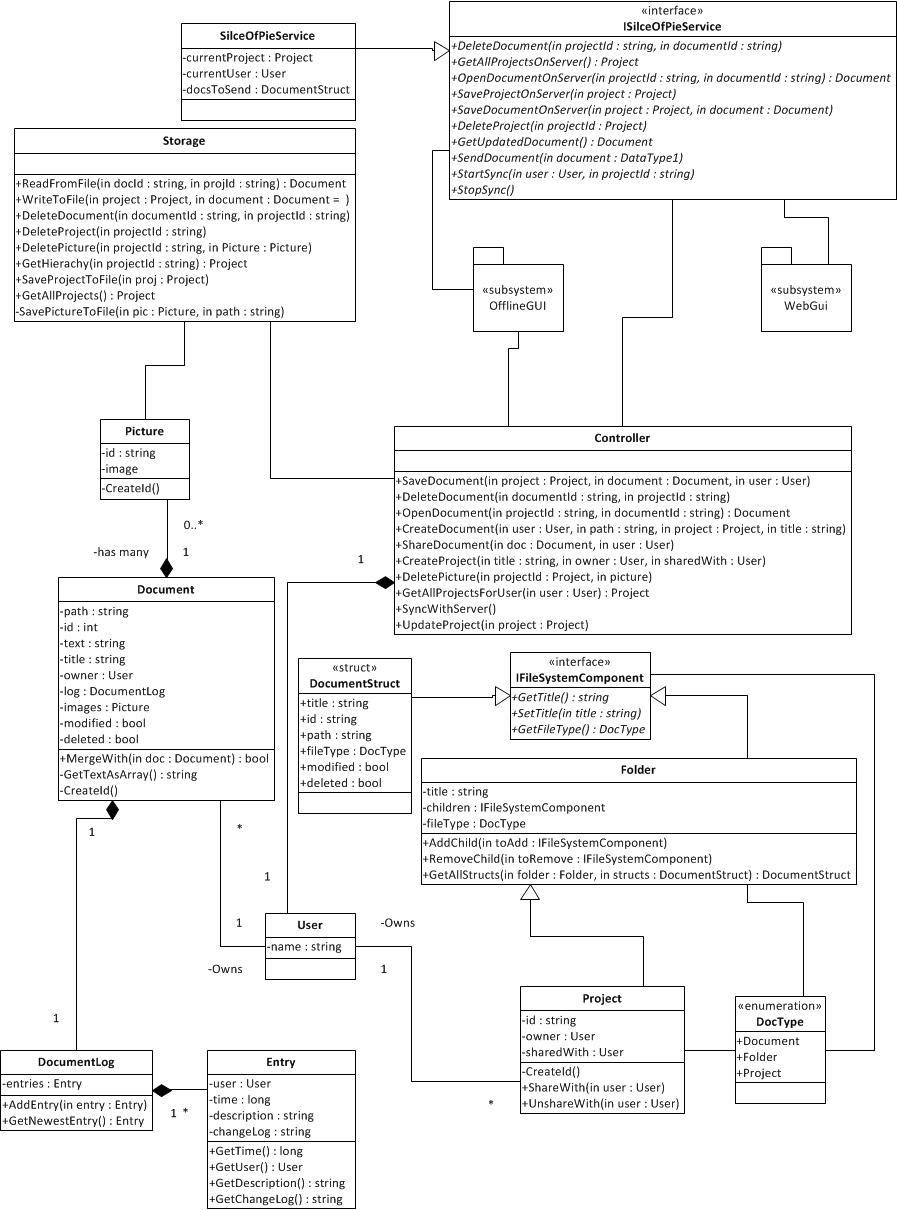
## 5.1 Use case Model



## 5.2 Use case diagram Revision Table

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 04-12-2012 | Initial version | Christian |
| 14-12-2012 | Added many use cases | Kewin |
|  |  |  |
|  |  |  |

# Appendix 6: Class diagram



## Appendix 6.0: Class diagram

|  |  |  |
| --- | --- | --- |
| Date | Description | Author |
| 04-12-2012 | Initial version | All |
| 04-12-2012 | Small additions to controller and storage | Kasra |
| 08-12-2012 | Completely reworked class diagram to conform to new decisions. | Kasra & Kewin |
| 11-12-2012 | Reworked class diagram again. | Kewin |
| 16-12-2012 | Reworked again | Kewin |
| 17-12-2012 | Adjusted to fit final implementation | Kasra |

# Appendix 7: Sprint reviews and retrospectives

1. Diagram added in visual studio project [↑](#footnote-ref-1)