

# Software Requirements Specification

Version 1.0

## File Management System

HNU\_CAMPUS LINK  
FSOFT ACADEMY



## Table of Contents

Table of Contents .....	1
1.0. Introduction .....	2
1.1. Purpose .....	2
1.2. Scope of Project.....	2
1.3. Glossary.....	2
2.0. Overall Description .....	3
2.1 System Environment.....	3
2.2 Functional Requirements Specification .....	3
2.2.1 File Use Case .....	4
Use case: View Files.....	4
Use case: Upload File .....	5
Use case: Download File .....	5
Use case: Delete File .....	6
2.2.2 Setting.....	6
Use case: System Setting .....	7
2.3 User Characteristics .....	7
2.4 Non-Functional Requirements .....	7
3.0. Requirements Specification .....	8
3.1 External Interface Requirements .....	8
3.2 Functional Requirements .....	8
3.2.1 View Files.....	8
3.2.2 Upload File .....	8
3.2.3 Download File .....	9
3.2.4 Delete File .....	9
3.2.5 Settings .....	9
3.3 Detailed Non-Functional Requirements .....	10
3.3.1 Logical Structure of the Data.....	10
3.3.2 Design Pattern.....	10
3.3.3 Java Best Practice .....	10

## **1.0. Introduction**

### ***1.1. Purpose***

The purpose of this document is to present a detailed description of the File Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the potential clients for its approval.

### ***1.2. Scope of Project***

This software system will be a File Management System for project managers, quality assurance team and any project members to manage their documents. This system will allow users control different versions of documents in different categories or projects. It also could set permissions on documents.

The file categories, account and project integration will be out of scope in this version.

### ***1.3. Glossary***

<b>Term</b>	<b>Definition</b>
File	All documents are managed inside the system.
Database	Collection of all the information monitored by this system.
Member	A member of a project.
User	End users of the system

## **2.0. Overall Description**

### **2.1 *System Environment***

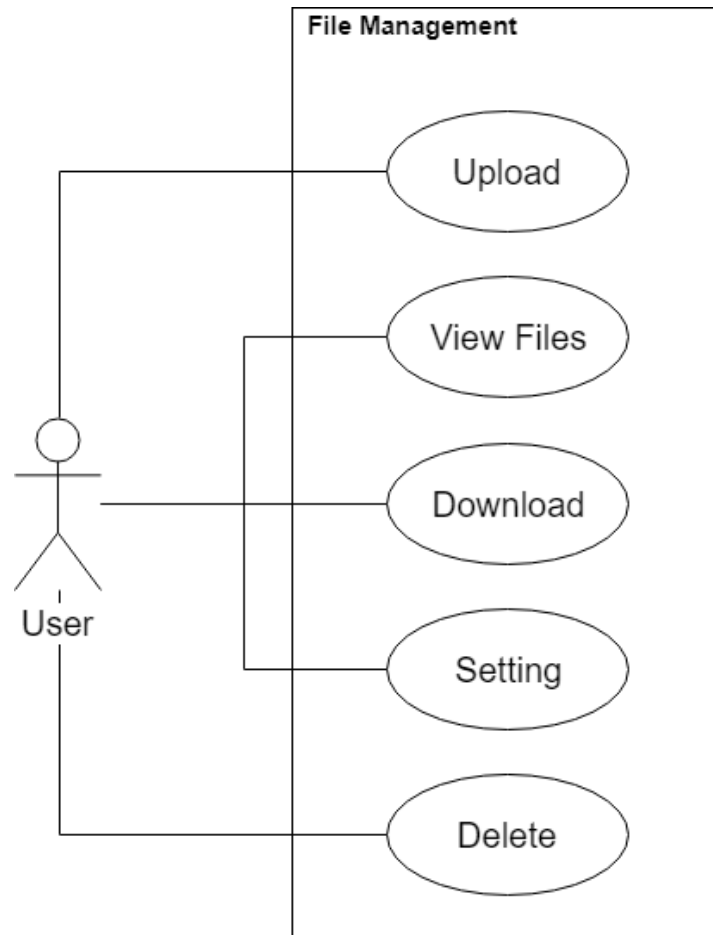
End users could be project managers, quality assurance team and other project members. There are more than 500 possible users.

The technologies are Optional we can see some technologies for Java development:

- Oracle Java 8
- MySQL
- JDBC/Hibernate.....
- Maven 3
- Jsp/Servlet, Spring boot Thymeleaf, Spring MVC....
- Bootstrap 4
- HTML/CSS/ jQuery

TBU

### **2.2 *Functional Requirements Specification***



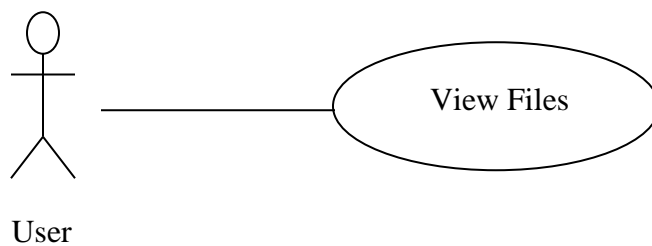
**Figure 1 - File Management System Use Case**

This section outlines the use cases for each of the active user separately.

#### 2.2.1 File Use Case

Use case: **View Files**

**Diagram:**



#### **Brief Description**

The user accesses the file management website to see all files.

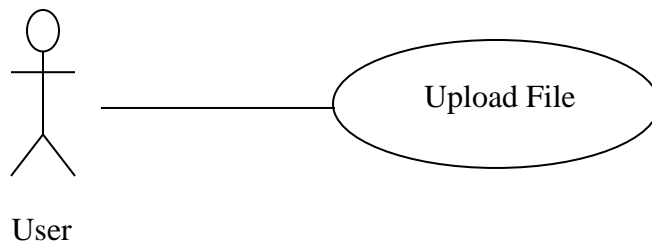
### Initial Step-By-Step Description

Before this use case can be initiated, the user has already accessed the file management website.

1. The user open the homepage to view files.
2. The system displays all files in separated pages.
3. The system allows the user to perform other activities.

Use case: **Upload File**

**Diagram:**



### Brief Description

The user accesses the file management website, choose files to upload into system. Files could be new or existed.

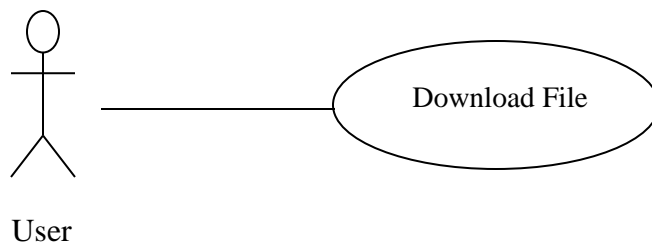
### Initial Step-By-Step Description

Before this use case can be initiated, the user has already accessed the file management website.

1. The user clicks button to choose files to upload.
2. The system displays a dialog to browse files.
3. The user selects one or many files to upload.
4. The system will validate files to upload or reject.
5. The system will update file lists to display.
6. The system will store files on a configured hard disk location.
7. The system allows the user to perform other activities.

Use case: **Download File**

**Diagram:**



**Brief Description**

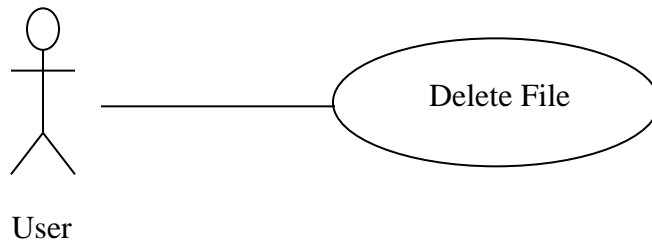
The user accesses the file management website to download files on the list view.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already accessed the file management website.

1. The user clicks button to download a file.
2. The system downloads files into the client machine.
3. The system increases the number of download.
4. The system allows the user to perform other activities.

Use case: **Delete File**

**Diagram:****Brief Description**

The user accesses the file management website, choose files to delete. Files will be permanent deleted.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already accessed the file management website.

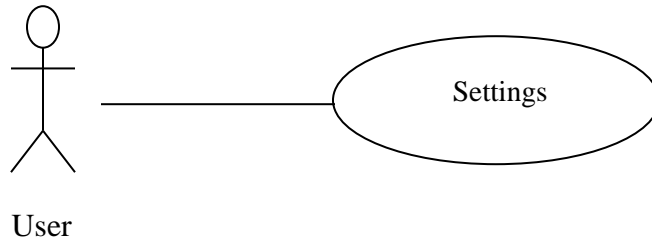
1. The user clicks button to delete a file.
2. The system delete selected files on the database and the file location.
3. The system will update the list view.
4. The system allows the user to perform other activities.

### 2.2.2 Setting



Use case: **System Setting**

**Diagram:**



### **Brief Description**

The user accesses the file management website to change and update new settings for the system.

### **Initial Step-By-Step Description**

Before this use case can be initiated, the user has already accessed the file management website.

5. The user clicks button to change settings.
6. The system displays a setting board.
7. The user changes values of parameters.
8. The user clicks a save button to update changes
9. The system will update new settings.
10. The system allows the user to perform other activities.

## **2.3 User Characteristics**

TBD

## **2.4 Non-Functional Requirements**

TBD

### 3.0. Requirements Specification

#### 3.1 External Interface Requirements

TBD

#### 3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

##### 3.2.1 View Files

<b>Use Case Name</b>	View Files
<b>XRef</b>	Section 2.2.1, View Files
<b>Trigger</b>	The user assesses the File Management Website
<b>Precondition</b>	None
<b>Basic Path</b>	<div>1. The user open the homepage to view files. 2. The system displays all files in separated pages. 3. The system allows the user to perform other activities.</div>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	The user could download, upload or delete files as well as change settings on the home page.
<b>Exception Paths</b>	The files may not be found and the error page will be displayed.
<b>Other</b>	Files are separated and displayed on different pages.

##### 3.2.2 Upload File

<b>Use Case Name</b>	Upload File
<b>XRef</b>	Section 2.2.1, Upload File
<b>Trigger</b>	The user assesses the File Management Website
<b>Precondition</b>	The homepage is displayed including the upload button.
<b>Basic Path</b>	<div>1. The user clicks button to choose files to upload. 2. The system displays a dialog to browse files. 3. The user selects one or many files to upload. 4. The system will validate files to upload or reject. 5. The system will update file lists to display. 6. The system will store files on a configured hard disk location. 7. The system allows the user to perform other activities.</div>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	The uploaded files must be stored into the file location. The database is also updated (file details, file versions)
<b>Exception Paths</b>	The upload process could be interrupted because of connections.
<b>Other</b>	The files are validated by setting conditions.

### 3.2.3 Download File

<b>Use Case Name</b>	Download File
<b>XRef</b>	Section 2.2.1, Download File
<b>Trigger</b>	The user assesses the File Management Website
<b>Precondition</b>	The homepage is displayed including the download button.
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. The user clicks button to download a file.</li><li>2. The system downloads files into the client machine.</li><li>3. The system increases the number of download.</li><li>4. The system allows the user to perform other activities.</li></ol>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	The number of download is increased.
<b>Exception Paths</b>	The download process could be interrupted because of connections. The file is not found and the error page will be displayed.
<b>Other</b>	None

### 3.2.4 Delete File

<b>Use Case Name</b>	Delete File
<b>XRef</b>	Section 2.2.1, Delete File
<b>Trigger</b>	The user assesses the File Management Website
<b>Precondition</b>	The homepage is displayed including the delete button.
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. The user clicks button to delete a file.</li><li>2. The system delete selected files on the database and the file location.</li><li>3. The system will update the list view</li><li>4. The system allows the user to perform other activities.</li></ol>
<b>Alternative Paths</b>	None
<b>Postcondition</b>	The files are deleted on database and on hard disk permanent. The list view is updated.
<b>Exception Paths</b>	The process could be interrupted because of connections.
<b>Other</b>	None

Error in modal  
notif

### 3.2.5 Settings

<b>Use Case Name</b>	Settings
<b>XRef</b>	Section 2.2.2, Settings
<b>Trigger</b>	The user assesses the File Management Website
<b>Precondition</b>	The homepage is displayed including the setting button.
<b>Basic Path</b>	<ol style="list-style-type: none"><li>1. The user clicks button to change settings.</li><li>2. The system displays a setting board.</li><li>3. The user changes values of parameters.</li><li>4. The user clicks a save button to update changes.</li><li>5. The system will update new settings.</li><li>6. The system allows the user to perform other activities.</li></ol>
<b>Alternative Paths</b>	None

<b>Postcondition</b>	The new settings are updated into database and the rules are applied to upload file processes.
<b>Exception Paths</b>	None
<b>Other</b>	The input values are validated

### 3.3 Detailed Non-Functional Requirements

#### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored on the system is given below:

File	Setting
<ul style="list-style-type: none"> <li>- ID: long</li> <li>- name: string</li> <li>- path: string</li> <li>- createdDateTime: DateTime</li> <li>- fileSize: long</li> <li>- mime: string</li> <li>- numberOfDownload: int</li> <li>- version: int</li> <li>- status: boolean</li> <li>- versionIds: string</li> </ul>	<ul style="list-style-type: none"> <li>- ID: int</li> <li>- maxFileSize: long</li> <li>- itemPerPage: int</li> <li>- mimeTypeAllowed: string</li> <li>- lastUpdatedTime: DateTime</li> </ul>

Figure 2 - Logical Structure of the File Management System

#### 3.3.2 Design Pattern

Apply at least 1 design pattern in system.

#### 3.3.3 Java Best Practice

Apply at least 5 java best practice in coding.