

# **Remote Document Vault**

## **Operational Concept Document**

Document version 1.0

**Prateek Sharma**  
**CSE681- Software Modelling and Analysis- Project 3**  
**October 21, 2013**  
**SUID: 263406294**

**Document Revision History**

| Version   | Reason for Change                                                     | Date             | Author         |
|-----------|-----------------------------------------------------------------------|------------------|----------------|
| 0.1 Draft | Created document.                                                     | October 14 ,2013 | Prateek Sharma |
| 0.2       | Added Executive Summary, Use cases and User interface sections        | October 17, 2013 | Prateek Sharma |
| 0.3       | Added Partitions, Activity Diagram section and Updated Views section. | October 19, 2013 | Prateek Sharma |
| 0.4       | Added Critical issues and conclusion sections                         | October 20, 2013 | Prateek Sharma |
| 1.0       | Finalized Document                                                    | October 21, 2013 | Prateek Sharma |

## Contents

---

|                                                    |           |
|----------------------------------------------------|-----------|
| <b>1. EXECUTIVE SUMMARY.....</b>                   | <b>4</b>  |
| <b>2. INTRODUCTION.....</b>                        | <b>5</b>  |
| 2.1 Overall System Architecture.....               | 5         |
| 2.2 Client View. ....                              | 6         |
| 2.3 Organizing Principles .....                    | 8         |
| <b>3. USERS AND USE-CASES .....</b>                | <b>9</b>  |
| 3.1 Potential Users.....                           | 9         |
| 3.2 Use Cases: .....                               | 10        |
| <b>4. VIEWS.....</b>                               | <b>13</b> |
| <b>5. CONTEXT DIAGRAM .....</b>                    | <b>20</b> |
| <b>6. PARTITIONS .....</b>                         | <b>22</b> |
| 6.1 Document Map Builder .....                     | 22        |
| 6.2 Remote Document Vault System Partitions: ..... | 23        |
| <b>7. ACTIVITY DIAGRAM .....</b>                   | <b>31</b> |
| 7.1 Document Vault Client Activities.....          | 31        |
| 7.2 Document Vault Server Activities.....          | 33        |
| <b>8. CRITICAL ISSUES .....</b>                    | <b>35</b> |
| 8.1 Document Vault Client: .....                   | 35        |
| 8.2 Document Vault Server: .....                   | 36        |
| <b>9. CONCLUSION .....</b>                         | <b>41</b> |
| <b>REFERENCES.....</b>                             | <b>42</b> |

## 1. Executive Summary

---

The purpose of Remote Document Vault software is to enable insertion and extraction of file into remote vault and display the file relationships among files. Remote document vault provides client side user interface for the purpose of insertion and extraction of various file formats and to display the relationship between files. Additionally, all the processing is done at the server side which is handled by the Remote Document Vault server component.

The main objective behind the Remote Document Vault is to adapt itself according to the user needs. For example, in a software firm, remote document vault can be used to hold the software code and it can be useful to find file dependencies. Similarly, remote document vault can also hold various design documents for example functional specification documents, High level design documents and other related documents. So, the system will be very helpful to find the relationships among documents. Thus, the Remote Document Vault's flexibility can be very helpful in various situations.

Primary users of the remote document vault are content providers, architects, developers, administrators, project managers, quality assurance analysts and customers. Developers, architect and managers can use this software to manage the code repositories and software documentation. Similarly content providers can use this software by giving the child dependencies accordingly.

In this document architecture details of remote document vault is discussed and next sections of this document discuss about the packages and activities of document vault client and server and build an architecture which can successfully be implemented. Additionally, graphical user interface of document vault client is discussed in the Views section and various client level graphical user views are explained in detail.

In the last section, some critical issues related to the design of remote document vault system are analyzed so that while implementing this system, these issues are considered so that implemented system can successfully cater the customer needs. Some of the critical issues are following:

- Cache coherency and cache management issues at client side.
- Network and security issues related to the client and server communication.
- Concurrent file access issues.
- Concurrent user access issues.
- Issues related to file insertion and editing.

## 2. Introduction

---

Remote document vault system provides the insertion and extraction of files in the remote vault and shows the various file properties and parent-children relationship of files in the vault. Since, document vault is a network based application, it can be divided into two parts: Document Vault client and Document Vault Server. Document Vault client runs at the client end and provides the graphical user interface to the user to perform the various functions like, to view the meta-data and text content, parent-child relationships of a file, various categories and associated files. It also gives the text-search and meta-data search option to the user.

Similarly, document vault server handles requests from the clients and performs all the processing related to user requests. Server's primary task is to listen the incoming request and on the basis of request type, processing of the request. Server also uses the map builder tool at the start-up which scans all the meta-data files in the server and creates a map of parents.

In the next sub sections client and server architecture is discussed in detail.

### 2.1 Overall System Architecture:

System architecture for Remote Document Vault describes the top level architecture by mentioning higher level activities of a Remote Document Vault. After close analysis of the requirements of the system, tasks can be divided into following high level parts:

- File Insertion
- File Extraction
- Text and Meta-data Search
- Category search
- Meta-data file editing.
- Parent-Child map builder

As the remote document vault application is a client-server based model, there will be multiple concurrent users of the whole system. So, here critical part of the architecture is that how system handles the concurrent requests from multiple clients and how the responses of these requests are generated. A generic view of system model between clients and server is shown in the below diagram.

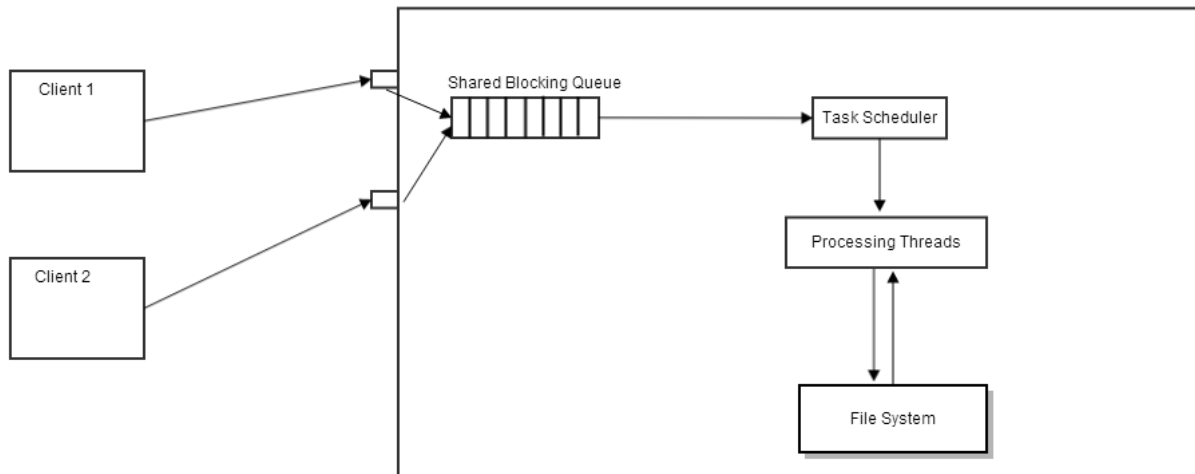


*Fig.1.1: Overall architecture of Remote Document Vault*

As shown in the figure, there are three important components of the system which are used to respond to client requests.

- Shared blocking queue
- Task scheduler
- Processing Threads

Clients will be connected to the server using their unique communication channel and in order to communicate, both server and client will use the same communication channel to send the request and response. Whenever a client will fire a command, it will be queued in the shared blocking queue. Shared blocking queue will store the request commands from all the clients. After a request is stored in the queue, task scheduler will take the first task from the queue and allocate a separate thread for the processing of request, and then it will take the next request and so on.

Here all the requests will be asynchronous with call-back. So, whenever a thread is finished its execution the response will be provided through the same communication channel. The advantage of choosing asynchronous over synchronous is that when a request will require large processing of data, client won't freeze.

Similarly, in order to process the requests threads will communicate to the server file system where documents are stored.

## 2.2 Client View:

Design of Remote Document Vault client also plays a critical role in the success and failure of the overall system. If cache mechanism is not implemented at client side, it may happen that there will be large redundant server calls for already downloaded file when user will select a file. This situation becomes worse when the number of documents and concurrent users are large for the entire system. Performance of the system is greatly affected by the unnecessary network calls. So, keeping in mind that number of documents and concurrent users in a vault can be very large, we should support the cache mechanism at the client side. So, that overall network congestion can be reduced.

An effective caching mechanism is the foundation of any distributed computing architecture [1].

At the time of start-up client makes a server call to get the various settings related to cache. These setting are defined at the server end and client uses these values to maintain the cache properly.

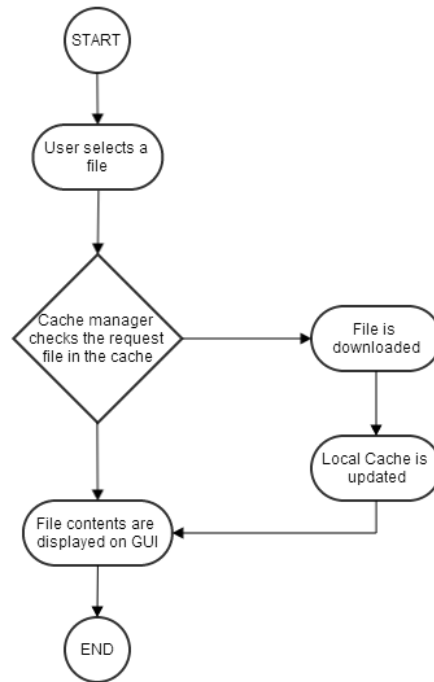


*Fig.1.2: Flowchart for cache manager*

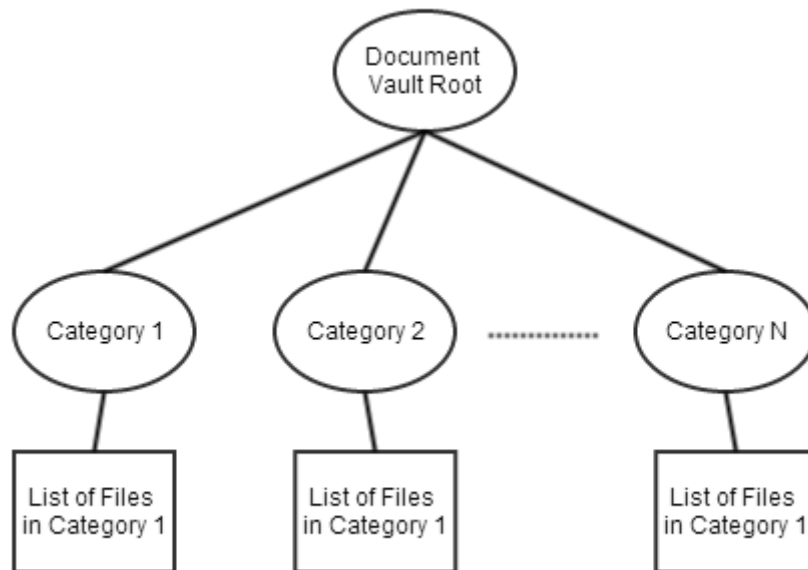
In current remote document vault client implementation, client uses the lazy loading and caching approach to improve the performance and availability of the server. Whenever a user selects a file from the graphical user interface, client checks that the copy of that file in the local cache. If the copy is there in the cache and it is valid, it is displayed on the client. Otherwise, a new request is fired to the server and after successful downloading of that file, local cache is updated.

Additionally, whenever user performs a text search or meta-data search, only file names and required values are retrieved from the server, Instead of retrieving the file contents of each and every file. Network calls to download a file is only sent when a user selects a file from any GUI pane.

Thus, by the help of effective caching and lazy loading approach a significant performance and availability improvements can be achieved. However, maintain a client cache has some implications like cache coherency and how long the cached contents will be there in the local cache. These issues are discussed in detailed in critical issues.

### 2.3 Organizing Principles:

For a large distributed system, it is critical how low level components are organized internally. So, the organization of documents is very critical in the remote document vault server. A remote document vault may have millions of documents contained within its file system. So, for each and every user query, it is not an efficient solution to scan the entire file system. Instead all the files should be organized in such a way that only related portion of the document based is searched. For example, if a user searches a particular text in a specific category, only files related to that category should be scanned.



So, to organize the whole system, document base will contain the parent XML file at the top level which will contain all the categories that are there in the document base. An individual category file will point to all the files that are there in that category. It is also possible that a category may have a sub category. So, in this case, it will contain the reference of sub directories. Each and every category file will be represented by an XML file which will store the file references of all the files in that category.

### 3. Users and Use-Cases

---

#### 3.1 Potential Users:

Remote document vault provides the flexibility to adapt itself according to a wide range of user needs. In remote document vault system, user can define the parent and child relationship between files. So, Remote Document Vault can be used in various fields.

This section identifies the users of the document vault system and discusses about the features that are supplied by remote document vault for its users. Users of the system can be categorized into following categories:

1. Content providers
2. Software Developers
3. Administrators
4. Architect
5. Quality Assurance
6. Manager
7. Customer

#### **Content Providers:**

Content providers deals with publications which are useful for users. Content providers can use the remote document vault system to insert and extract the documents in the system. Remote document vault gives the ability to define the children and parent relationships which can be used by content providers to make the software useful for its customers.

#### **Software Developers:**

Remote document vault system can be very useful for the software developers who work on large set of files. With the help of remote document vault, they can insert and view the source code files to/from the system. Additionally, software development life cycle of software contains documentation of components. So, developers can define the parent-child relationship between various software design documents and finding the relationship between files can be easy.

#### **Administrators:**

Administrators can use remote document vault software to manage the structure and organization of vault. They can view the access control lists and govern that how the system will be used by individuals.

#### **Architects:**

Architects can use the document vault to improve the source code management and software documentation management. They can categories the project into several categories like: Source Code, Design documents etc. So, that the any file or document can be easily tracked by text search or meta-data search. Additionally, architects can use the vault's ability to create categories and parent child relationships for selected files.

#### **Quality Assurance Analysts:**

Quality Assurance Analysts ensure that the system is in correct state and it works exactly according to the requirements. Quality assurance analysts will use Remote Document Vault to check if there is any inconsistency or bug in the system. They follow the complete system

testing process so that before the software reaches to its customers, it should not have any bugs or inconsistencies.

#### Team Manager:

Team managers can use the remote document vault to track the project documentation and file dependencies. They can search the document base for the particular file and keywords. They can manage the whole structure of the vault by creating and managing categories.

#### Customer:

Customers will use the remote document vault to find the documents relevant to them by providing search keywords and attributes. They will search desired keywords and system will return all the files which contain those keywords in that category. Additionally, in very large software systems like Document Vault or Document Repository, customers will have the ability to upload the documents and their meta-files.

### 3.2 Use Cases:

A use case is a list of steps, typically defining interactions between a role (known in UML as an "actor") and a system, to achieve a goal [3].

Following is a list of general use-cases:

#### 1. User inserts a file in the remote document vault:

| Application | User inserts a file in the remote document vault                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User        | User who uses the Document Vault Client                                                                                                                                                                                                                                                                                                                                                                                                 |
| Applies to  | <i>Remote Document Vault System</i>                                                                                                                                                                                                                                                                                                                                                                                                     |
| Steps       | <ol style="list-style-type: none"> <li>1. User runs the graphical user interface of Document Vault Client.</li> <li>2. User goes to the Document Management Pane.</li> <li>3. User selects the Category and a file to be uploaded from the local file system.</li> <li>4. User enters the child dependencies.</li> <li>5. User clicks on upload button.</li> </ol>                                                                      |
| Events      | <ol style="list-style-type: none"> <li>1) File is successfully uploaded to the server.</li> <li>2) If the error occurs during uploading, error message is displayed on the user interface.</li> <li>3) An XML file is generated for the text file in the server and child dependencies and other file attributes are filled in the XML file.</li> <li>4) Corresponding xml category file is updated for the every insertion.</li> </ol> |

#### 2. User performs a search on Category Tags

| Application | User performs a Category Search                                                                                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User        | User who uses the Remote Document Vault                                                                                                                                                                         |
| Applies to  | <i>Remote Document Vault System</i>                                                                                                                                                                             |
| Steps       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault Client.</li> <li>2. User goes to Category pane and selects the category to search.</li> </ol>                                            |
| Events      | <ol style="list-style-type: none"> <li>1) All the files names in the user specified categories are loaded.</li> <li>2) If more than one file categories are selected, categories should be displayed</li> </ol> |

|  |                                                                      |
|--|----------------------------------------------------------------------|
|  | with the file name.                                                  |
|  | 3) Category description should be updated according to new category. |

### 3. User selects a file from Category Search Result:

|                    |                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application</b> | <b>User selects a file from Category Search Result.</b>                                                                                                                                                                                                                                                                                                      |
| <b>User</b>        | User who uses the Remote Document Vault.                                                                                                                                                                                                                                                                                                                     |
| <b>Applies to</b>  | <i>Remote Document Vault System</i>                                                                                                                                                                                                                                                                                                                          |
| <b>Steps</b>       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault client and search files in category.</li> <li>2. User selects a file from the search results.</li> </ol>                                                                                                                                                                              |
| <b>Events</b>      | <ol style="list-style-type: none"> <li>1) All the panes should be updated according to the current selected file.</li> <li>2) Text and Meta Data contents should be fetched for the new file and should be displayed on the graphical user interface.</li> <li>3) Parent and Child relationships should be updated according to new selected file</li> </ol> |

### 4. User updates tags of current meta-data XML file.

|                    |                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application</b> | <b>User updates the meta data tags for the current meta-data file</b>                                                                                                                                                                         |
| <b>User</b>        | User who uses the Remote Document Vault.                                                                                                                                                                                                      |
| <b>Applies to</b>  | <i>Remote Document Vault System</i>                                                                                                                                                                                                           |
| <b>Steps</b>       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault client and navigates to the information pane.</li> <li>2. User updates the meta-data tags for a file.</li> </ol>                                                       |
| <b>Events</b>      | <ol style="list-style-type: none"> <li>1) Meta file update request goes to the server and the selected file is updates.</li> <li>2) If the file is locked, a user friendly error message is shown on the graphical user interface.</li> </ol> |

### 5. User navigates to the Parent-Child relationship pane.

|                    |                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application</b> | <b>User navigates to the Parent-Child relationship pane.</b>                                                                                                                                                                                                                                                                                                                |
| <b>User</b>        | User who uses the Remote Document Vault.                                                                                                                                                                                                                                                                                                                                    |
| <b>Applies to</b>  | <i>Remote Document Vault System</i>                                                                                                                                                                                                                                                                                                                                         |
| <b>Steps</b>       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault client and navigates to the parent-child relationship pane.</li> </ol>                                                                                                                                                                                                                               |
| <b>Events</b>      | <ol style="list-style-type: none"> <li>1) A request goes to server to fetch the parent and children of currently selected files.</li> <li>2) In case of successful response, all the parents and children of currently selected file are displayed in the pane</li> <li>3) In case of error message, user friendly error message is displayed on the GUI client.</li> </ol> |

### 6. User performs a text query by providing search options.

|                    |                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------|
| <b>Application</b> | <b>User performs a text query and gives keywords and category as search options.</b>                                           |
| <b>User</b>        | User who uses the Remote Document Vault.                                                                                       |
| <b>Applies to</b>  | <i>Remote Document Vault System</i>                                                                                            |
| <b>Steps</b>       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault client and navigates to the Text Search Pane</li> </ol> |

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <ol style="list-style-type: none"> <li>2. User provides one or more keywords in the keyword text box.</li> <li>3. User selects one or more categories from the category item.</li> <li>4. User selects one or more switches (AND or OR)</li> </ol>                                                                                                                                                                                                                                                                                                           |
| <b>Events</b> | <ol style="list-style-type: none"> <li>1) A request goes to the server to search all the text files in user specified category which contains the user specified keywords.</li> <li>2) In case of success response, graphical user interface populates the search results with file name and categories.</li> <li>3) In case of error response, a user friendly error message is shown on the client.</li> <li>4) If user selects any of the files from search results, all the information panes are updated according to the current selection.</li> </ol> |

## 7. User performs a meta-data search.

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application</b> | <b>User updates the meta data tags for the current meta-data file</b>                                                                                                                                                                                                                                                                                                                                                       |
| <b>User</b>        | User who uses the Remote Document Vault.                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Applies to</b>  | <i>Remote Document Vault System</i>                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Steps</b>       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault client and goes to meta-data search pane.</li> <li>2. User enters some meta-tags in the corresponding field.</li> <li>3. User selects one or more categories from the category drop down.</li> <li>4. User clicks on search button.</li> </ol>                                                                                                       |
| <b>Events</b>      | <ol style="list-style-type: none"> <li>1) A Meta-data search request goes to the server with user specified tag names and categories.</li> <li>2) Server searches all the meta-data files in the user specified categories and return the file name and corresponding category in results.</li> <li>3) If an error occurs at the server, a user friendly error message is shown in the graphical user interface.</li> </ol> |

## 8. User creates a new category:

|                    |                                                                                                                                                                                                                                                                                                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application</b> | <b>User created a new category.</b>                                                                                                                                                                                                                                                                                        |
| <b>User</b>        | User who uses the Remote Document Vault.                                                                                                                                                                                                                                                                                   |
| <b>Applies to</b>  | <i>Remote Document Vault System</i>                                                                                                                                                                                                                                                                                        |
| <b>Steps</b>       | <ol style="list-style-type: none"> <li>1. User runs the Document Vault client and goes to Category pane.</li> <li>2. User enters a new category and supplies the required field description for the category and clicks on submit.</li> </ol>                                                                              |
| <b>Events</b>      | <ol style="list-style-type: none"> <li>1) A request goes to server to create a new category under the root tag and server creates an XML file for the new category with the user specified description and other fields.</li> <li>2) In case of error response, a user friendly message is shown on the client.</li> </ol> |

## 4. Views

Document Vault client provides the graphical user interface for the users to interact with the system. Document vault client displays the information related to files and categories using the graphical user interface and gives the ability to user to change the view according to his needs. Graphical user interface of Document Vault client contains following panes:

- Content View
- Dependency View
- Document Management View
- Text Search
- Meta-Data Search
- Categories

Each view is explained below:

### 1. Content View:

Content View provides the file content of current selected text file and meta-data file. View is divided into two parts:

- **Text View:** Text view displays the text contents of the currently selected file
- **Meta-Data View:** Meta-Data view displays the contents of meta-data file for current selected file.

Content View of remote document client is shown below:

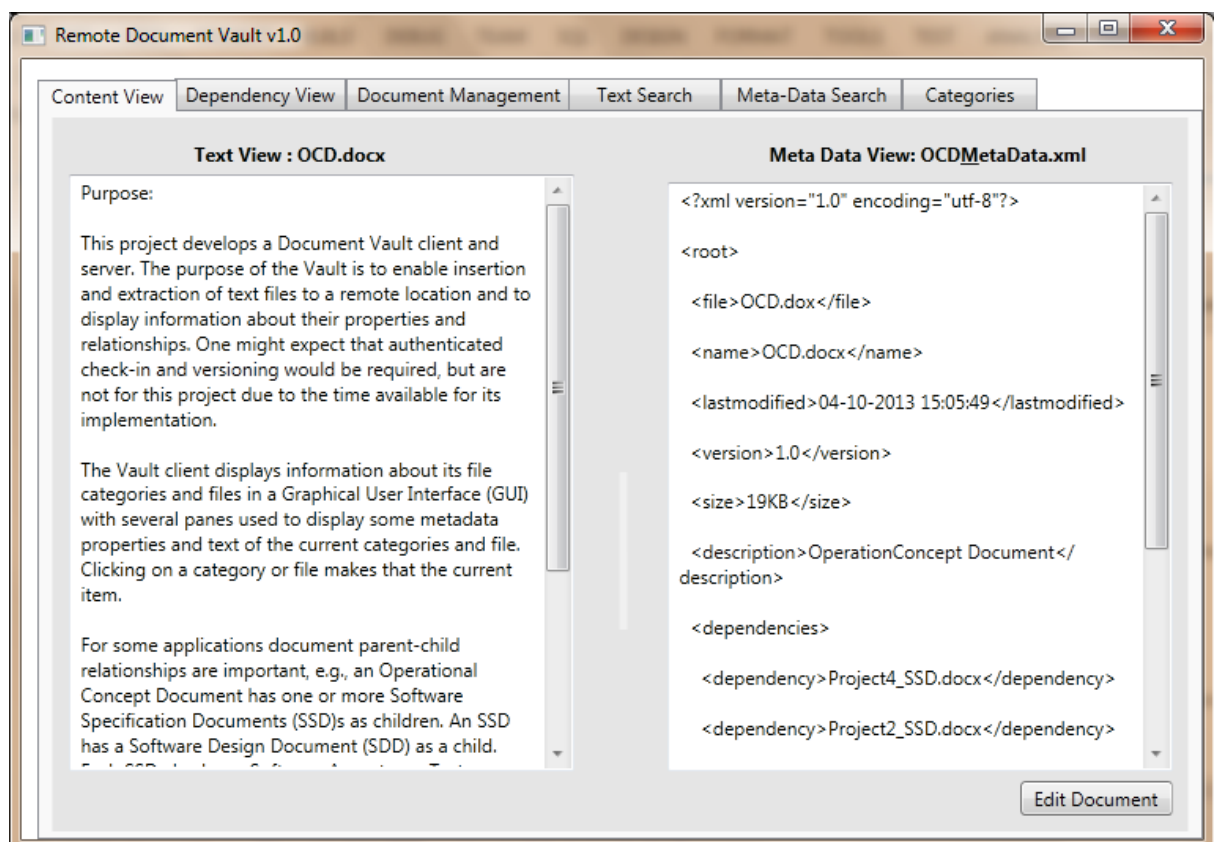


Fig. 3.1: Content view of the document vault client.

Whenever another file is selected from other views, contents of Text and Meta-Data views are updated according to the newly selected file.

Apart from the above mentioned functionality, Meta-Data view also provides the feature to edit currently selected meta-data file. When edit document button is clicked, it shows a modal window to enter the tag names and corresponding values of tags to the user. When user clicks on OK button on modal dialog window, values of tags are sent to the server and current meta-data file is updated. However there are two policies regarding the changes in meta-data file:

- If the meta-data tag already exists in the file, its older value is overwritten by the new value.
- If the tag is not there in the meta-file, a new tag is created with the user defined value.

## 2. Dependency View:

Dependency view displays the file dependencies for currently selected file. It shows the parents and children of current selected file. Additionally, it shows the information regarding the current selected file and category.

Dependency view is divided into three columns: First column shows the file properties of currently selected file and category information for the selected file. Second View shows the parent list of currently selected file and the third view shows the children of currently selected file.

A snapshot of the Dependency View is shown below:

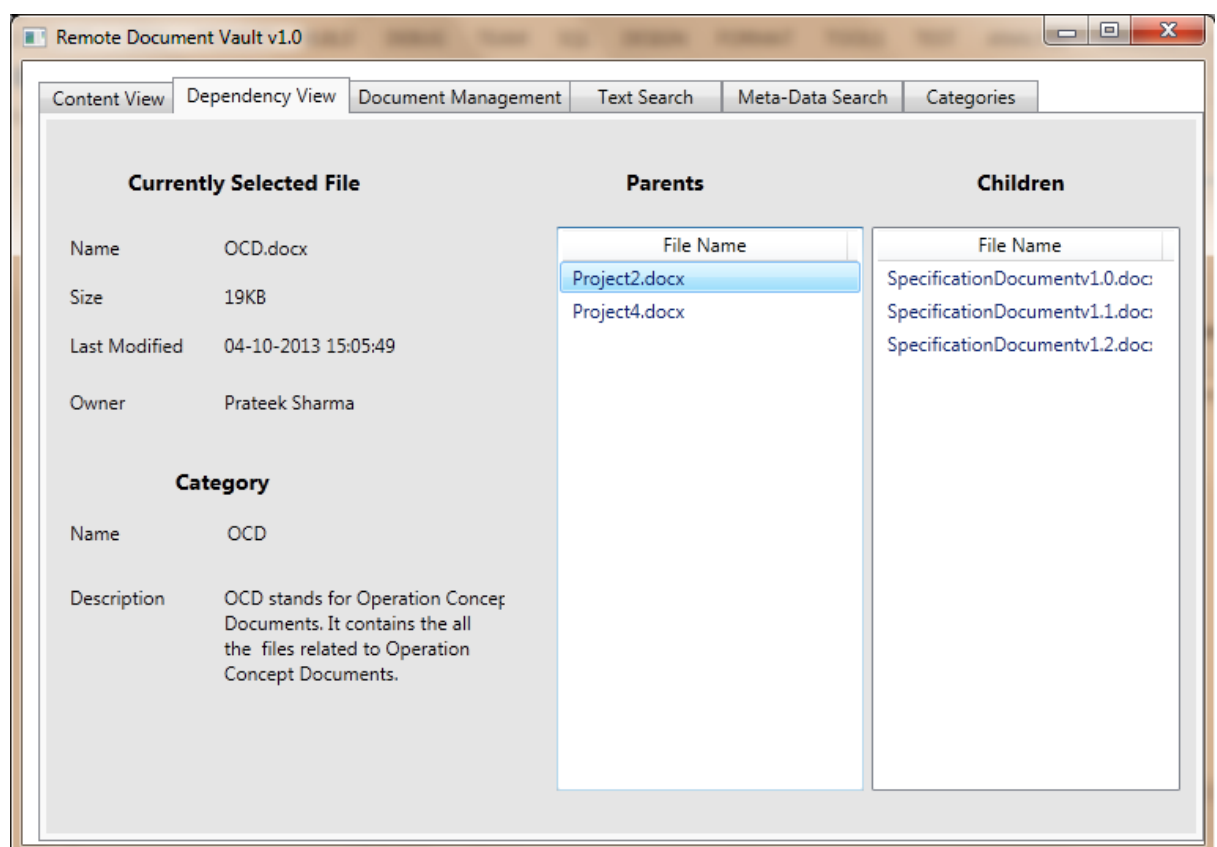


Fig. 3.2: Dependency view of the document vault client.

Both parent list and children list are selectable list and user can select any file from the any of the list. If a file is selected from any of the lists, other views in the application are updated accordingly.

### 3. Document Management View:

Document Management View provides the ability to insert files into the remote document vault. User can use the document management view to insert files to remote document vault by providing the required fields like description text, category, keywords, and file dependencies.

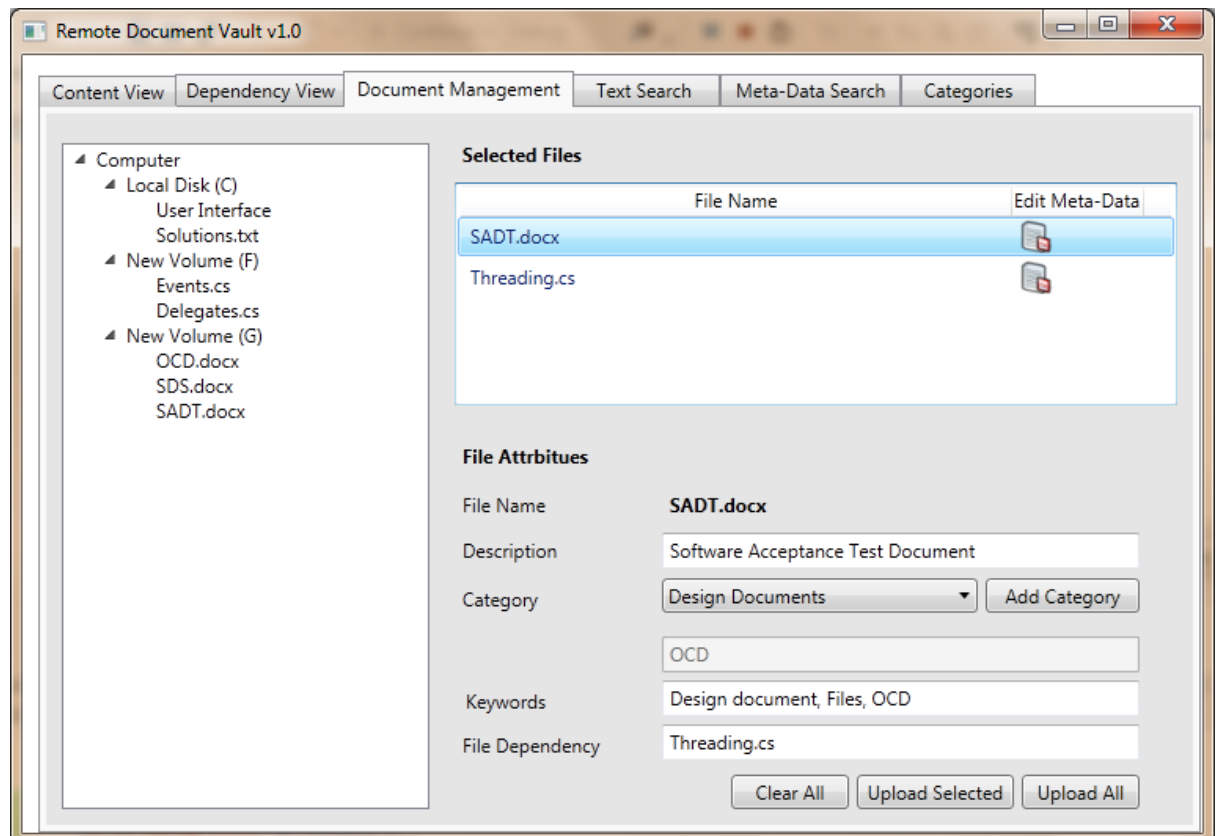


Fig. 3.3: Document management view of the document vault client.

A snapshot of the document view is shown above.

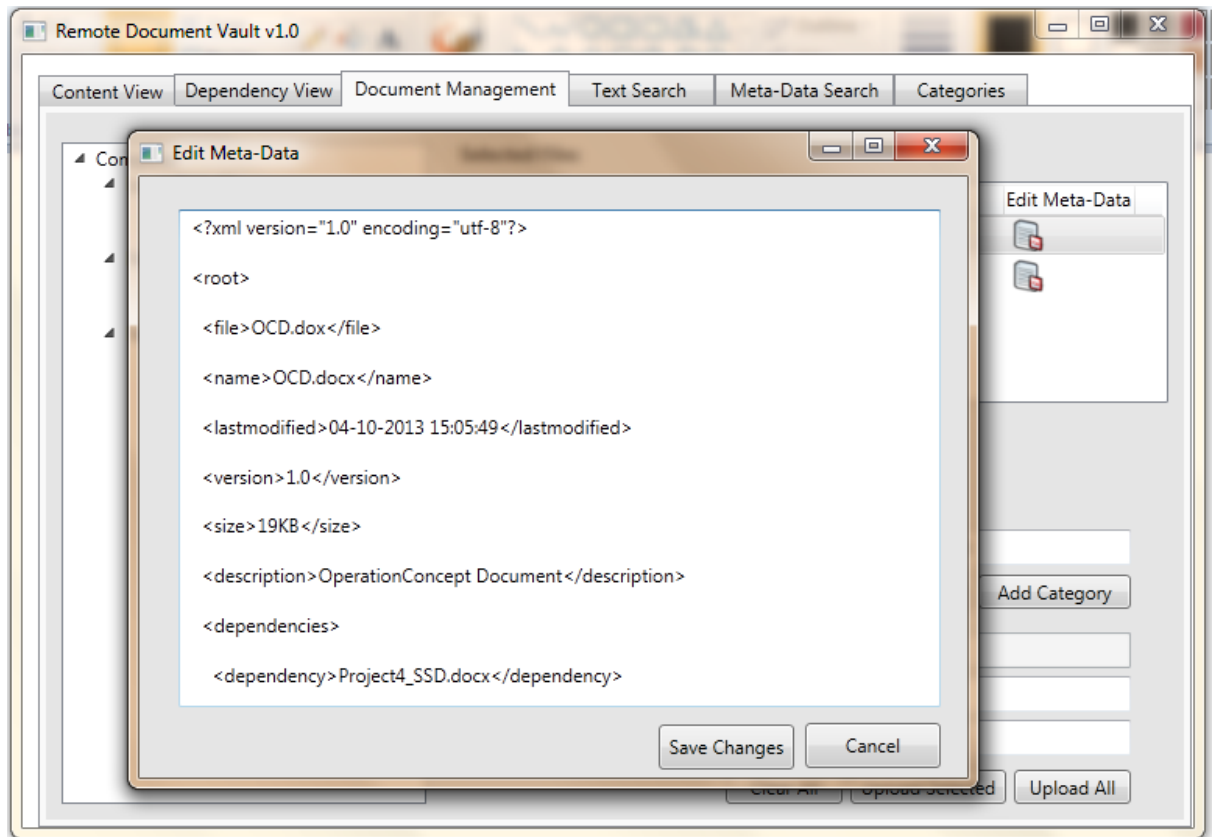
As shown in the picture, user can browse the files in local file system to insert the documents into the vault and selected files will be added in the lists. After adding the files, user can enter the file's meta-data fields in the File Attributes. User can give the description text, one or more category, keywords and Child dependencies in the file attributes. After clicking the upload button, files are inserted in the local file system and a meta-data file is generated with the user defined fields.

Document management view gives the ability to the user to select the child dependencies according to his needs and user can upload the child and parent at the same time.

Here with the help of edit XML button provided with uploaded document files, user can edit the XML which is generated after the insertion of document. Initially these

buttons will be disabled but after uploading the documents these will get enabled and when user will click on the edit xml button of any file. He will have the opportunity to edit the XML meta-data contents of the selected file. These files are not downloaded from the server; instead client generates the files from user defined categories and dependencies.

A snapshot of edit xml view is shown below:



#### 4. Text Search View:

Text Search view extends its functionality from the *Project 2: Composite Text Analyzer*. It provides the basic functionalities of *Composite Text Analyzer* but also provides the ability to select one or more categories from the graphical user interface to search the keywords.

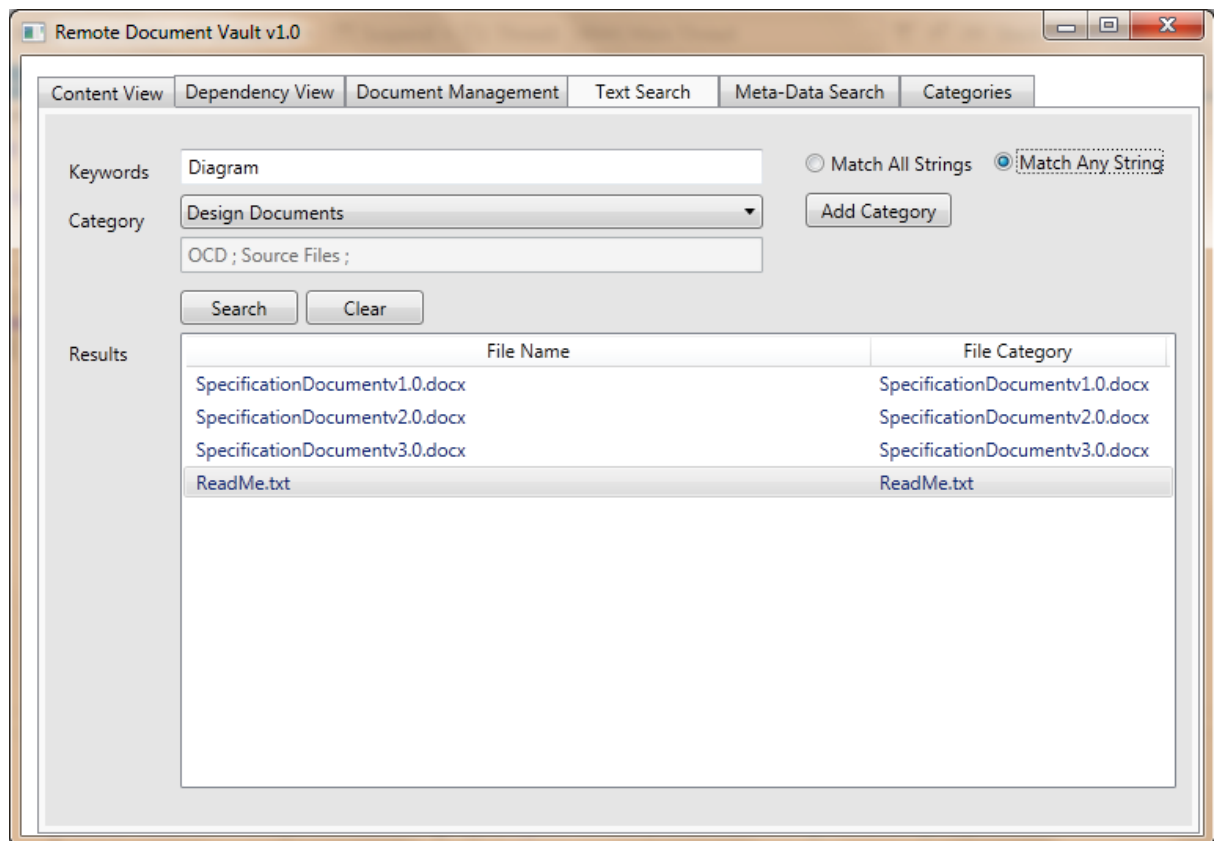


Fig. 3.4: Text search view of the document vault client.

A snapshot of the Text Search view is shown above.

As shown in the picture, various fields are provided to perform the text search in the document vault. These fields are following:

- **Keywords:** User can provide the keywords for the search query in the keywords text box. If there is more than one keyword in the keywords field, they should be entered in a space separated list and if a keyword contains spaces, it should be entered in the quotes.
- **Switches:** User can select any of the radio buttons shown in the user interface. Match all string searches all the keywords in a file and Match Any String searches at least one keyword in the files.
- **Category:** User can select one or more category from the graphical user interface in which the search will be performed. User can add the categories from the drop down list provided in categories and after clicking the Add Category button, category is added in the search category.

Results of the text query are shown in the table with file name and file category. The result list is a clickable list and user can change the currently selected file by selecting any file from the result. If user does so, other views are changed according to the newly selected file.

## 5. Meta-Data Search View:

Document vault client extends the meta-data search functionality provided by the *Project 2: Composite Text Analyzer* and adds the ability to perform the meta-data search in the user specified categories.

A Meta-Data search view is shown below in the figure:

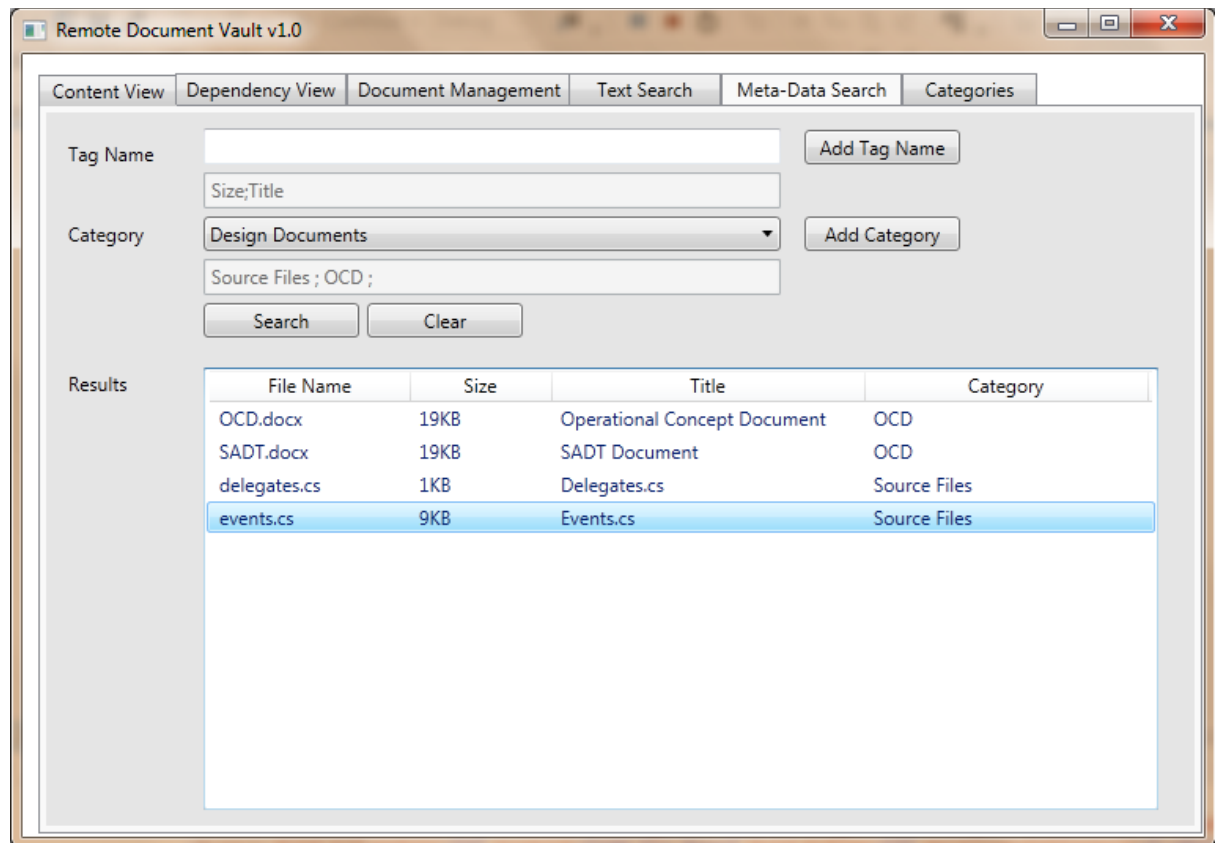


Fig. 3.5: Meta-Data search view of the document vault client.

Meta data view provides the following fields to perform the meta-data search:

- **Tag Name:** User can enter one or more tag names in this field. After entering a tag name, user clicks on the add tag name field to enter the tags in the query. All the tag names are displayed in the below disabled text box and query will be performed on the tags displayed in the disabled text box.
- **Category:** User can provide one or more category for the search by selecting the category from the drop down list and by clicking on Add Category Button. All the user selected categories are shown in the disabled textbox and search queries will be performed on these categories.

When user clicks on the search button a meta-data query is performed on the server and results are displayed in the results table. Results table lists all the file names and corresponding categories with the user defined tag names and values. The file list in the result table is a clickable list and if a user selects a file from the lists, it makes the selected file as current file and all the views are updated accordingly.

## 6. Categories View:

Categories view lists all the categories in the document vault with the files in each category and provides the ability to the user to enter a new category in the document vault.

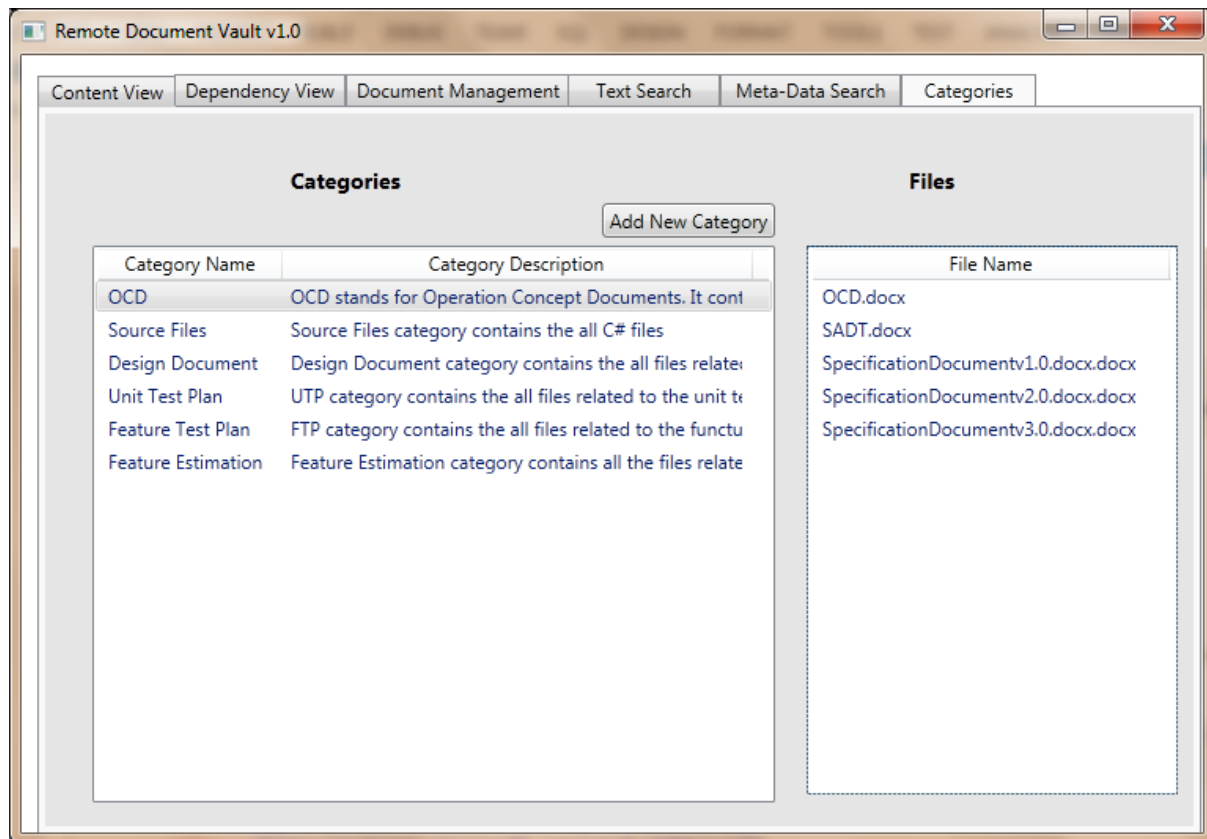


Fig. 3.6: Categories view of the document vault client.

A snapshot is shown for the Categories view above.

Categories view shows all the categories in the remote document vault and displays all the files for each and every category. Initially only first category file name list is downloaded from the server and as the user selects other categories, a request is sent to the server to provide file names for the user selected category. Categories table are shown with the category name and description columns. Similarly, file list for the selected category is shown with file names of every file in that category.

Files table also provides the ability to make any file as current file from the list. When a user double clicks on a file, current selected file for the system is changed with the newly selected file and contents of other modules are changed accordingly.

Apart from this, categories view also provides the ability to the user to create a new category in the remote document vault server. When user clicks on the Add New Category button, a modal dialog window is shown to enter the category name and category description values. After filling this information, when user clicks on Add button, a request is sent to the server to create a new category.

## 5. Context Diagram

---

Context diagram of a software system shows how software interacts with its surrounding environment. Remote Document Vault interacts with following environment components during its life cycle:

1. User Input
2. Client File System
3. Remote File System
4. Network resources

### User Input:

User initiates the execution of document vault by running the client interface. User can perform any one of the following tasks using the graphical user interface.

- User gets all the files listed in a category by using the category search pane.
- Currently selected text file and meta-data file will be displayed on the graphical user interface and user can edit the meta-data file by providing the meta-tags and values.
- User performs the text search by selecting one or more categories and by giving some keywords in the Search pane of graphical user interface.
- User performs the meta-data search by selecting one or more categories and by giving meta-data tags values and all the meta-data files which contains the specified tags will be displayed on the graphical user interface.
- User views the current child and parent relationship using the Document Navigation pane. User can select any file from the parent or child lists and views will be changed according to the specified file.
- User uploads a file using the document management pane and giving the keywords and child dependencies as values.

### Client File System:

Document Vault client interacts with the client file system to maintain the cache of downloaded file. Document Vault client uses the lazy loading of documents to reduce the network traffic and supports the mechanism of cached files. So, whenever a file is downloaded from the server, it is maintained in the cache, until the cached copy expires. Client interacts with the file system in following ways:

- When user wants to upload any text file on the server, client will communicate to local file system to get the contents of the file. Then, the file is transferred from client to server using message passing.
- When a file is downloaded from the server, a copy is maintained in the local cache.

So, client interacts with the local file system to maintain its cache. It may insert, delete and update cached files in the local file system.

### Remote File System:

Document vault server manages the complete document vault. So, it interacts with the local server file system various times during a client session. for the execution of following tasks:

- When a file is uploaded from a client, it is inserted in the repository and corresponding meta-data file is generated.
- When a meta-data file is updated from client end, the changes are reflected by modifying the file contents in the local file system.
- When a user selects a file, file is downloaded at client end. So, document vault server reads the contents of selected file and sends it across the network.

### Network Resources:

Network resources play the critical role of establishing a communication channel between a client and server. When a client wants to connect to server, it injects the session initiation message in the network resources and the messages are conveyed to server end where they are interpreted and suitable response is generated for the client. All the communication between a client and server passes through the network resources.

Additionally, to transfer a file from client to server or from server to a client, file contents are placed on same unique communication channel.

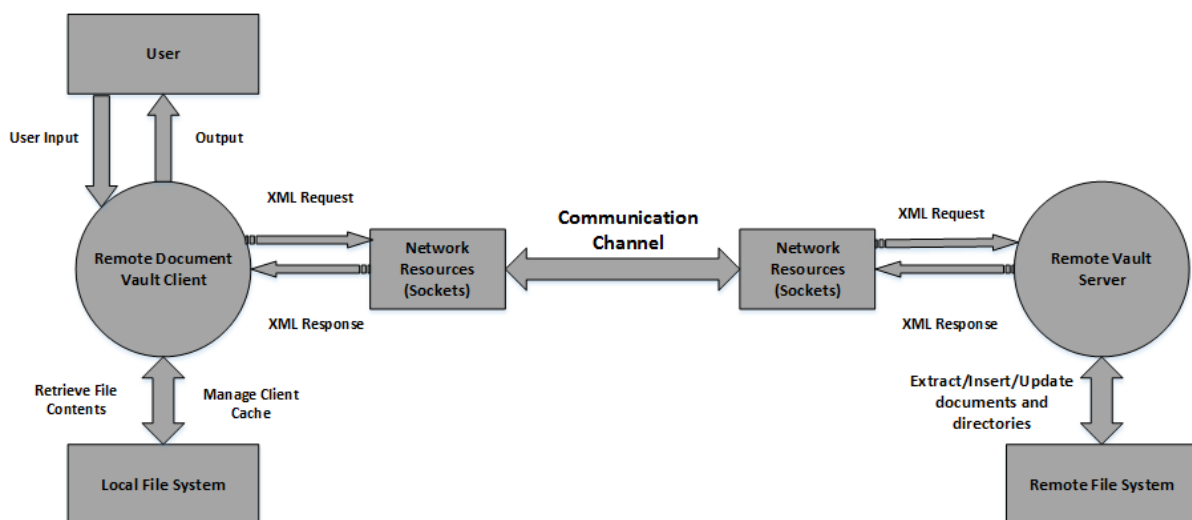


Fig 4.1: Context Diagram of Remote Document Vault

## 6. Partitions

---

### 6.1 Document Map Builder:

Document Map Builder tool is used by the document vault server to generate the parent list of all the files in the vault. Additionally, it can be used as standalone tool at server end. Below package diagram shows the standalone version of document map builder tool which is used at server end.

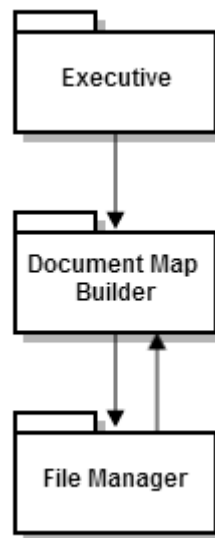


*Fig. 5.1: Package diagram for document map builder.*

Document Map Builder tool doesn't use any user interface. It creates a map of parent and child relationships of all the files in the vault and it is directly used by the document vault server as a DLL. Document Map Builder tool consists of three packages: Executive, Document Map Builder, and File Manager.

Executive Module starts the execution of Document Map Builder and it contains the main method for the execution.

Document Map builder uses the File Manager to scan each and every file in the document vault and it creates a map of parents for each and every file in the document vault. It reads every meta-data file and stores the child dependencies in the map. After scanning all the meta-data files in the entire vault, a map data structure is created. Document Vault Server uses this map data structure at the start-up to find parent relationships for all the files.

File manager module is used by the document builder to communicate with the local file system. It reads all the meta-data files in the file system and gives the contents to the map builder.

## 6.2 Remote Document Vault System Partitions:

Remote Document Vault client and server system can be divided into several high level partitions. Document Vault client and server modules are explained in details below:

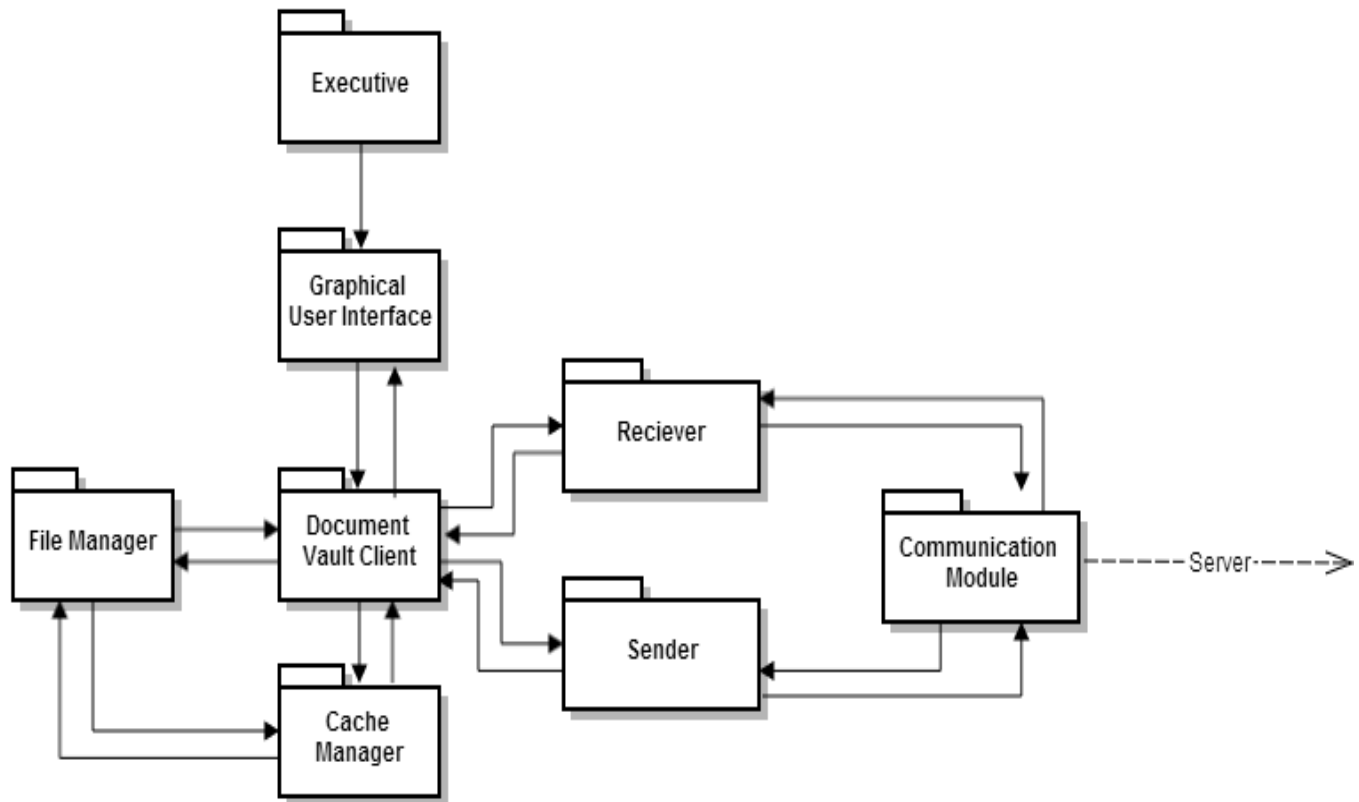


Fig. 5.2: Package diagram for document vault client.

### Document Vault Client Partitions:

Tasks associated to a client module can be divided into following parts:

- Present graphical user interface.
- Send user requested messages to the server.
- Receive and interpret the response received from the server.
- Support browsing of local files.
- Maintain a cache of downloaded files.
- Support editing of meta-data files.
- Provides ability to view categories and files in the category.

A client package diagram is shown above. On the basis of above mentioned tasks, remote document vault client can be portioned into following modules:

**1) Executive Module:**

Executive module is the main module for document vault client. Executive module is a very small package which contains the main method for the execution of client. It loads the graphical user interface of the client through which user can interact.

**2) User Interface:**

User interface module contains the main graphical user interface through which the user will communicate. It takes user input and show output of user requested commands. It will contain a Main window and many UI tabbed panes will be shown in the main window of user interface module and user will be allowed navigate to each and every module and perform any action on the data displayed on the user interface. Main window will consists of following panes:

- Categories Pane
- Text Search Pane
- Meta-Data Search Pane
- Document Management Pane
- Document Navigation Pane
- Document Content Pane

Each pane is discussed in detail in Views Section. Whenever a user performs any action on the graphical user interface, module sends the user specified action to the Document Vault client module and client module decides the processing required for the user specified action.

**3) Communication Module:**

Communication module handles the communication between client and server. It handles the three major communication tasks:

- It creates a channel between client and server through which client will interact to the server.
- When a request is received from the client, it encodes the message and sends it to the server.
- When a response is received from the server, it decodes the message and sends it to the client.

Here encoding means converting the requests to XML request messages that server can understand and similarly decoding means interpreting the received responses.

**4) Document Vault Client:**

Document vault client module handles all the interactions between user interface module, cache manager and communication module. Whenever user requests some file or data, associated command is sent to document vault client module. On the basis of command, it modifies the values in the command, sends it to the communication module which encodes the request.

Similarly, whenever a response is received from the server for a request. It is directed to the document vault client module by communication or receiver. After receiving the response, client parses the response and output is shown on the user interface.

When a client module is invoked, it communicates to the server to fetch the basic setting related to client like, MAX\_CACHE\_SIZE, MAX\_CACHE\_EXPIRY\_TIME etc.

#### 5) Cache Manager:

Cache manager module handles the cache mechanism for document vault client. It caches the files that are used by Document Vault Client so that network traffic and overall performance of the system can be improved. Whenever a user requests a file from the server, document vault client checks whether the file is there in the cache or not with the help of cache manager. If the file is there in the cache, cache manager invokes the file handler module to read the files from local file system and file is transferred to client module.

On the other hand, if the file is not there in the cache, cache manager notifies this to document vault client and client sends the request to Receiver module to download the file from the server. After successfully downloading a file, client again invokes the cache manager to store the downloaded file in the cache. So, that it can be used in further requests.

Additionally, cache manager also handles that time period for which the file will be remained in the cache and it also handles the cache coherency issues.

#### 6) File Manager Module:

File manager module is used by the cache manager module. File manager modules interacts with the local file system to read the cached files. Similarly, it also stores the files received from cache manager into local client cache.

#### 7) Sender Module:

Sender Module is used when a client wants to send a file or message on the server. It contains following two class components:

- Sending Queue: A queue is maintained by the client to send the requests to server or to upload a file on the server.
- Sending Thread: Sending thread sends the requests to server using the queue. It retrieves the messages one by one and sends it to the server. To send the request messages to server, it interacts with communication module.

#### 8) Receiver Module:

Receiver Module is used when a response is arrived from the server. It maintains following two components inside:

- Receiving Queue: A queue is maintained by the client to receive messages from the server.
- Receiving Thread: Receiving thread will read the messages from queue one by one. When a response is completely downloaded, it is sent to the communication module to be parsed.

## Document Vault Server Partitions:

Document Vault Server will be executed separately at the server end and it will cater the client requests. Document Vault server will be responsible for the following tasks:

- Opens a communication between a client and server.
- Receives multiple client requests simultaneously.
- Uses the Documents map builder to map parent files to children
- In case of category query, lists all the files in a category and send it to the client.
- In case of parent and child relation query, it creates a list of parents and children and sends it to the client.
- In case of text search query, searches the specified category and return the file references to the client.
- In case of meta-data search query, searches the specified category and return the meta-data file references to the client.
- If a file is created, it generates a meta-data file.
- If a file is extracted, it sends the file block by block to the client.

Document vault server will be using multithreading for its processing of requests as there will be multiple simultaneous requests from the server. Thread level details are not discussed here because threading model will vary according to the implementation and underlying platform. Only basic level of explanation is provided for the threading in packages.

Module level details of diagram is following:

### 1. Server Executive:

Server Executive starts the execution of server. It contains the main method which starts the execution of Server.

### 2. Document Vault Server:

Document Vault Server is the main point of execution of server module. This is the first point where execution reaches after the executive runs. Responsibility of the Document Vault Server module is to ensure that when a server start, all the start-up activities works. For example, at the time of startup it ensures that map is built with the help of document map builder.

Additionally, at the start-up, it opens a socket on server with the help of communication module and all the client uses the same socket to communicate with server.

### 3. Communication Module:

Communication module is responsible for the communication between server and clients. It performs the following tasks in its lifetime:

- It opens a communication channel between server and a client whenever a new client joins.
- It maintains a map of active clients.
- It checks periodically to check whether a connection is alive or not.
- It receives the incoming messages from the clients and send them into shared blocking queue of Task scheduler module.

- It sends the response from various modules to the corresponding clients.

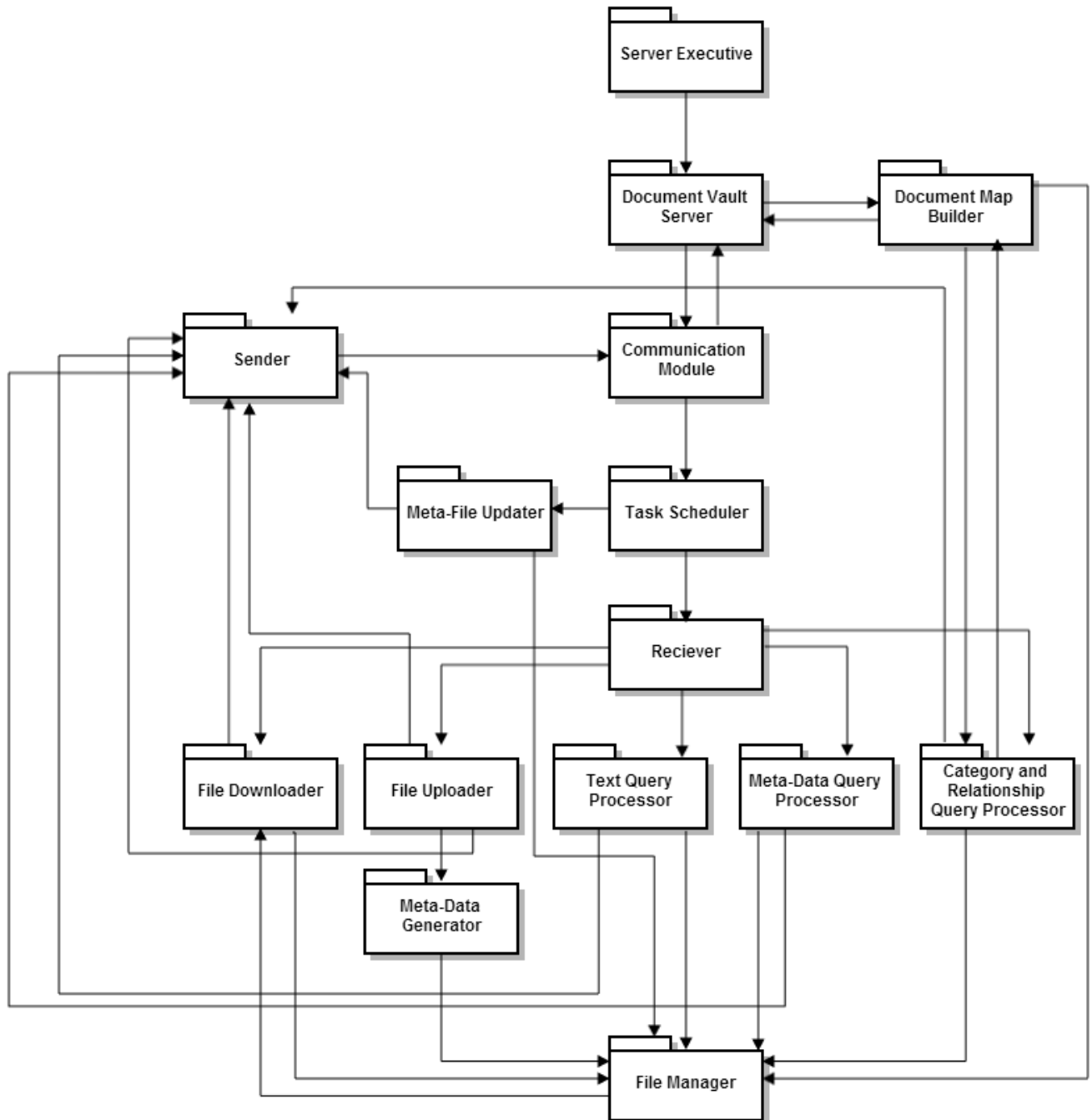


Fig. 5.3: Package diagram of document vault server.

#### 4. Task Scheduler:

Task scheduler contains the shared blocking queue and schedules the tasks as they arrive in queue. Whenever a request comes from a client, communication channel

puts the request in the shared blocking queue of Task Scheduler. Task scheduler takes the tasks from the shared queue and sends it to the receiver module until the queue becomes empty. If the queue is empty, it waits for the next incoming message in the queue.

## 5. Receiver:

Receiver Modules comes into pictures when a request is assigned for the processing. It analyses the request and gets the command from the request and on the basis of command, it creates a separated thread for execution. Receiver also ensures that if a command is coming in multiple parts, it waits till all the parts arrive and then it schedules the execution. According to the requirements, a user requested command can be categorised into following:

| Command Name                      | Command Description                                                  |
|-----------------------------------|----------------------------------------------------------------------|
| File Download Command             | Command is fired when a client double clicks on a file.              |
| File Upload Command               | Command is fired when a client inserts a document in the vault.      |
| Text Search Command               | Command is fired when a client searches a text                       |
| Meta-Data Search Command          | Command is fired when a client searches a meta-data tag in the vault |
| Category Search Command           | Command is fired when a client queries for the files in a category . |
| Parent Child Relationship Command | Command is fired when a client double clicks on the file.            |
| Meta-Data Update                  | Command is fired when a Meta-data file is updated,                   |

## 6. File Downloader:

File Downloader module is invoked by the Receiver module if the incoming request command is File Download. File Downloader module communicates with the File Manager and scans the user specified category for the file. When a file is found, it is transferred in 1024 byte blocks to the client using communication channel

## 7. File Uploader:

When a user inserts the file into the document vault using Graphical User Interface, a File Upload command is fired. When File Uploader module is invoked for a file, it uses the File Manager to create the file in the File System. At the same time meta-data file is generated for the inserted text file using the user specified keywords and dependencies. If file insertion successfully happens, a success response is placed in the sender module's queue. Otherwise, a failure response is placed in the sender's queue.

## 8. Text Query Processor:

Text Query Processor is invoked when a user searches keywords from the client in one or more categories. Text Query processor searches the user specified keywords using the File manager and places the response in the sender module's queue. For detailed description of text query processor module, refer OCD for Project1.

Some changes are done in the text query processor module so that it can support the multithreading environment and concurrent search on text files can be performed when the user searches a text in the vault.

#### **9. Meta-Query Processor:**

Meta-Query processor is invoked when a user searches meta-data tags in one or more categories. Meta-Query processor takes the meta-data tags and searches all the files in user specified categories. It places the result into the sender module's queue. Meta-Data query processor is described in detail in project 1 OCD.

To support the multithreading in meta-text processor, changes are done in this module. So that it can search the meta - data files concurrently. Additionally, there may be concurrent read and write issues to a file. So these changes are also done in the meta-query processor module.

#### **10. Category and Relationship Query Processor:**

This module is invoked when a client wants to see all the categories or all the files in a category or wants to see the parent-child relationship for a selected file.

For a category search command, module communicates with the file manager and create a list of all the files in the user specified categories.

Similarly, for a Parent-Child relationship command, module communicates with the document map builder and file system to find the parents and childs of a user specified file.

After a response is generated, module places the response in sender's queue.

#### **11. Meta-Data Generator:**

Meta-Data generator module generates a meta-data file when a file is inserted in the document vault. It recived inputs from the file uploader and creates a meta-data file with the help of file manager. Most of the functionality is defined in the Project 1 OCD.

#### **12. Meta-File Updater:**

Meta File updater module is invoked when user updates a meta-data file from the client. Meta-File updater modules receives the meta-tag names and values from the client and checks the user specified files for the supplied meta-data tags. If the tags are already present in the meta-data, corresponding values are overwritten.

Otherwise, a new meta-data tag is inserted into the respository. After processing the request, response is placed in the sender's queue.

#### **13. File Manager:**

File Manager module is not invoked directly from the Reciever module. However, this module is used by almost every other module for the processing of user requests. File Manager modules reads, updates and inserts file in the local file system based on the input. Additionally, File Manager module recieves the search category or keywords from the other modules and it scans the document vault for the user specified inputs.

There can be many concurrent file access issues at this layer because it may happen that two or more users a updating a file from their client. These issues are discussed in critical issues section in detail.

**14. Sender:**

Sender module tasks contains a shared response queue. Whenever a module is done with the processing of a request, it places the response in the sender's blocking queue. Sender dequeues the queue, encodes the response and sends the response to corresponding client using the communication channel.

## 7. Activity Diagram

---

This section discusses the main activities associated with Document vault client and server. Activity diagram are divided into two parts for the better understanding of the system.

### 7.1 Document Vault Client Activities:

As discussed in partitions section, top-level tasks for client can be divided into following parts:

- Connect to server.
- Show text-view of currently selected file.
- Show meta-view of currently selected file.
- Upload a file to server.
- Perform text-search.
- Perform meta-data search.
- Show all the categories and associated file.
- Show the parent and children of selected file.

Document Vault Client activity diagram is shown in below figure.

Document Vault client is started when the client exe is executed. After its execution, client tries to connect to the server and if the connection is established it goes to next activity. Otherwise, it retries for another time. After connecting to a server, it displays a main window in which there are several panes like Content view, document management view, text search view, meta-data view, categories search view and parent-child relationship view. User can navigate to any of the view and perform an action. These are shown as separate entities in the diagram.

In the content view, text and meta-data contents of the currently selected files are shown to the user. This view gets updated whenever user selects another file from another file and associated file and meta-data are downloaded from the server. In addition, in a separate activity, user can view the parents and children of currently selected file.

In a category pane, all the categories in the document vault are listed. When a user clicks on any of the category, all the file names are loaded in the list of files and now user can select any file to make it as current file.

When user views the text-search pane and enters the keywords and one or more categories as search parameters, a request is sent to the server for the text-search with user defined parameters and as a result, a list of files are shown on the graphical user interface which contains all the keywords in that category. User can select any file from the result list and can make it as a current file.

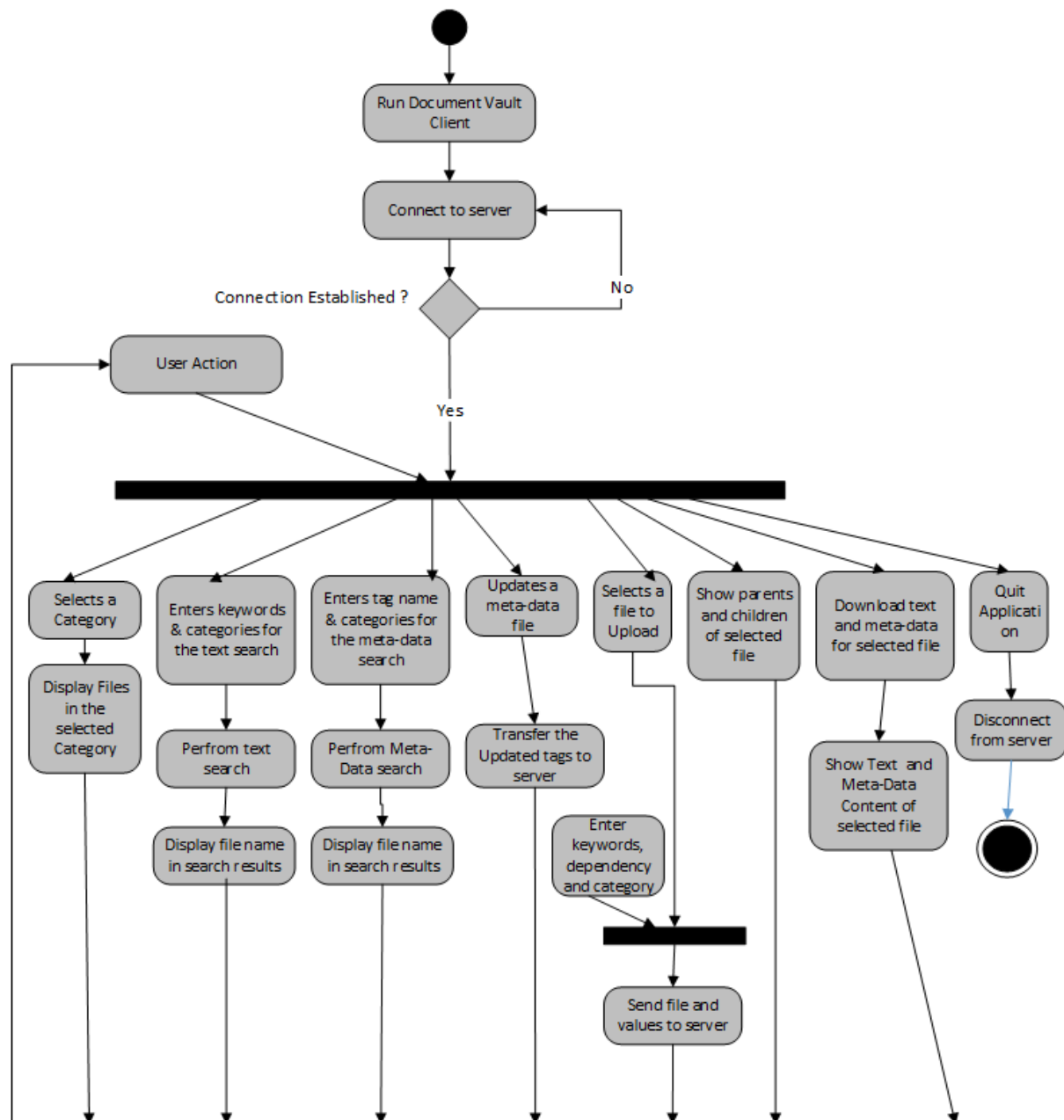


Fig 6.1: Document Vault Client Activity Diagram

In meta-data search activity, user enters meta-tags and one or more categories and request is sent to the server for meta-data search with user specified search parameters. After receiving the success response, result list is shown with user specified tag names and values in selected categories. User can select any file from this view and make this as current file.

When a user updates a meta-data file by giving the meta-data tags and values, a request is sent to the server. If user defined meta-data tag already exists in the file, the older value is overwritten by the new value. Otherwise, a new tag is created.

In file upload activity, user can insert a file to the remote document vault server. User selects a file to be uploaded and then it enters the keywords, one or more categories, file

dependency and description for the uploaded file. User request is sent to the server and file is inserted in the document vault.

In quit application activity, user exists from the main window and application disconnects itself from the server and main activity is ended.

## 7.2 Document Vault Server Activities:

As discussed in the Partitions section, top level tasks of document vault can be divided into following parts:

- Build parent map of every file.
- Accept client connections.
- Listen to client requests.
- Send the file to a client.
- Receive a file from a client.
- Update meta-data file.
- Send categories and file information to client.
- Perform text-search.
- Perform meta-data search.
- Return parents and children of selected file.

Document vault server activity diagram is constructed using above mentioned tasks and is shown below.

At the time of start-up, server uses the map builder module to create a map where every parent file is listed with corresponding children. After this, server opens a socket on which it accepts client connections and listen incoming requests from various clients.

Based on the incoming request, server handles it in separate activities. If the request is to download or extract a file, it checks the selected category and sends the file to corresponding client in blocks. Similarly, if the request is for insert a file into document vault, it checks the input parameters, sends the document to the vault, generates a meta-data file for the inserted file using parameters and updates the parent map with new file's child and parent entries.

If request is to update meta-data file in the vault, server reads the meta-data file from the vault and checks whether tags are present in the file or not. If the tags are already there in the vault, it replaces the old value with the new one. Otherwise, new tag with corresponding value is inserted in the file.

When incoming request is for the parent and children of current file, server checks the parent and children map to find the entries for the selected file and create a separate list for parents and children. After constructing the lists, server sends the lists in response to client request.

When request is for all the files in a category, server reads all the files in the specified categories and sends them to the client.

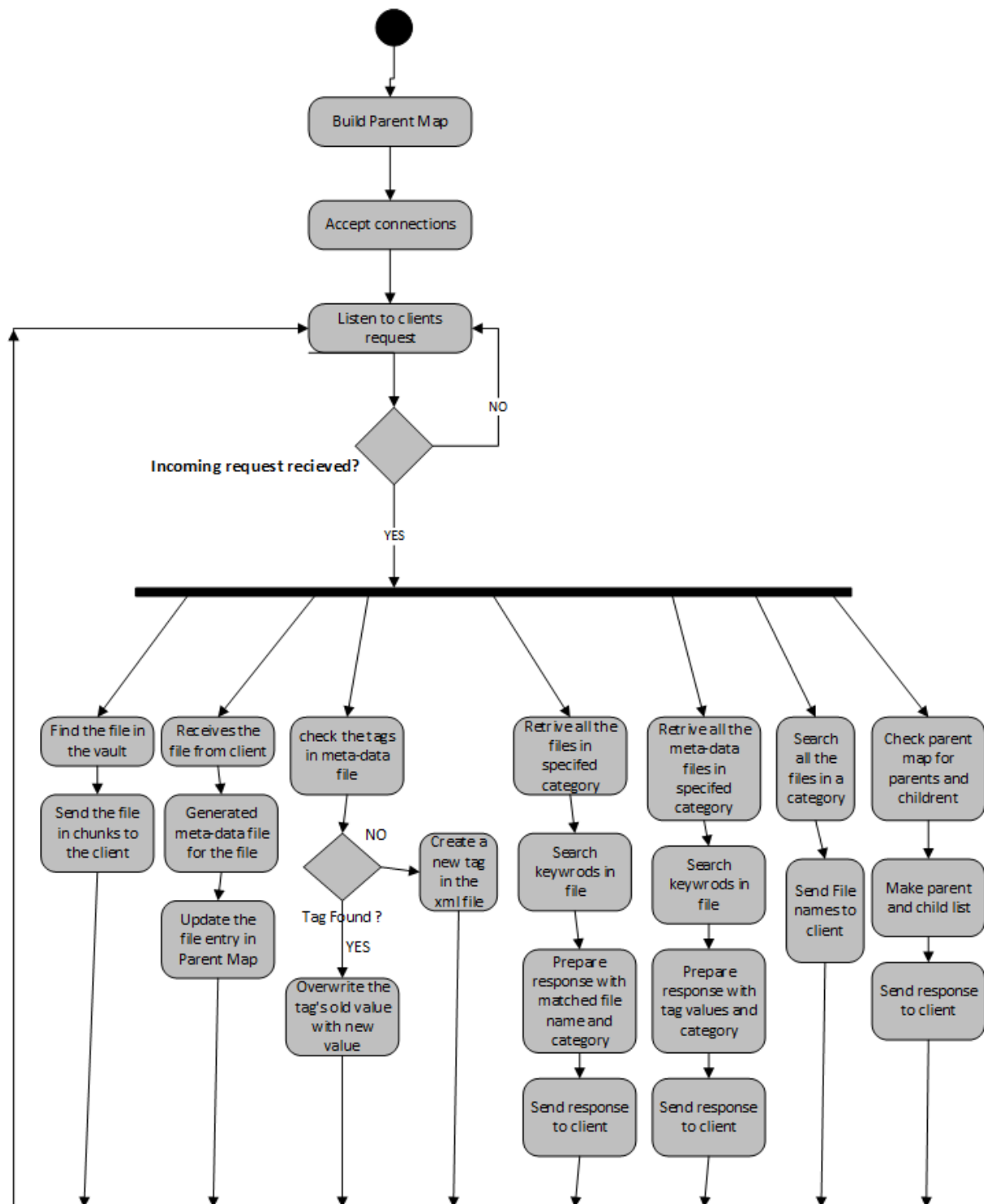


Fig 6.2: Document Vault Server Activity Diagram

When a request comes for the text search in a category, server reads the file contents of all the files in that category and checks whether the specified keywords are present in text or not. Then it makes the list of files with associated category names and sends it to the client. Similarly, if the request is for meta-data search, it reads all the meta-data files in the specified categories and tag names and values are stored in a map with file name as key. Then this list is sent to the client as response.

## 8. Critical Issues

---

While designing the architecture of a system, there can be several critical issues which need to be addressed. The lifetime of software is directly proportional to how the system handles these issues. In this section, some of the critical issues related to document vault client and server are addressed and they should be considered while implementing the design.

### 8.1 Document Vault Client:

#### 1. How cache coherency will be maintained in document vault client?

**Description:** Since the design uses the client side cache to reduce the network traffic and to improve the performance and availability of document vault server, there should be an optimal way to maintain the cache coherency so that the clients do not see the outdated files.

**Solution:** To overcome the cache coherency issue, client can send the list of file names and their <Last-Modified> tag value in the file's meta-data to server to check that these files are updated or not. At the server end, these files and tag values are mapped to vault's files and tag values. If a file's <Last-Modified> time is different than the vault's file. It is added in the outdated list. This sequence is done for all the files in the list and in the end final list contains all the files which are modified. This outdated list is then sent to the client.

Client's cache manager module takes this list and deletes all the files which are there in the list from the cache. Client doesn't attempt to download these files as client uses the lazy loading policy and whenever a one of these files is selected by the user, client will download these files from the server and a copy will be maintained in the server.

#### 2. What's the lifetime of files cached in the document vault cache?

**Description:** It may happen that there are several files in the vault client cache which are no longer used by the client from a very long time. There should be a way to keep the cache clean so that overall performance of a client can be improved.

**Solution:** Deciding the lifetime of files depends on the actual use of the software. If client is not using the cached documents from a very long time, the cache copies should be removed from the local client cache. To overcome this, server administrator can define the MAX\_CACHE\_PERIOD for documents in the cache. Cache manager should use an xml list where it should define all the cached files with <Last-Used> tag. This tag is used to hold the last used date and time of the cache file. Whenever cache manager is giving a cached copy to client for display purpose, it updates the entry for the file in the xml and now <Last-Used> tags contains the time stamp when the current file is used. This way at the start-up, client will check this xml file and all cached file's <Last-Used> tag values are compared to the server's MAX\_CACHE\_PERIOD. If any file is older than the MAX\_CACHE\_PERIOD, it will be removed by the cache manager.

**3. What will be the size of the client side cache or how many files will be there in the cache?**

**Description:** It may happen that client is browsing lots of files and as a result copies of these files are stored in the client cache. So, there should be a max cache size that should be defined so that if the cache size is exceeding the max value, an algorithm should be used to remove old files.

**Solution:** At the server end, we can define a MAX\_CACHE\_SIZE and at the start-up client will download this value in settings file. After that, cache manager will use this value at the client side whenever a file is inserted in the local document cache. If the local cache size is exceeding the MAX\_CACHE\_SIZE, typical least recently algorithm is used to remove the older files and size is maintained below MAX\_CACHE\_SIZE.

**4. What will happen if the multiple files are inserted in the vault that is mutually dependent (related)?**

**Description:** It is possible that client is inserting multiple documents in the vault and one document is dependent of the other in the uploaded file. For example, client is uploading two file File1.doc and File2.doc and File2.doc is the child of File1.doc. Now firstly client will send the request to upload the File1.doc and then File2.doc. In this case, server will throw an error response while insertion as the dependent file File2.doc is not there in the vault.

**Solution:** In this case, document vault client module can decide the uploading order of files. If multiple files are inserted at a time, client will analyze the document dependencies before sending the request and request will be sent from bottom-up fashion. It means first those files will be sent which are not dependent on the other files which are selected. Like in above example, client will send the request to upload the File2.doc first because it is not dependent on other uploaded files, the File1.doc will be sent to upload.

## 8.2 Document Vault Server:

**5. How to ensure that complete file is downloaded / uploaded?**

**Description:** In the current design, large files are sent in the 1024 Byte blocks. So, when a client is downloading or uploading a file, how to ensure that files are properly reached or received from the server.

**Solution:** The solution of this problem is adopted from traditional protocols. It uses the two types of messages to send a file to the server:

- a. Start Message
- b. File Contents

c. End message.

Initially, when the file upload will begin, START message will be sent to the server with the file name and content length of the file with a unique request id. After sending this message, server will assume that all the messages with the same request id are the file contents messages and server will store all the contents together while inserting a file. At the end, client will send the end message which will denote that file's contents are successfully transferred to the server. At the server end, server will verify the total file content size with the initial start message file content size. If the file content length matches the start length, file will be inserted into the server. Otherwise, a error message will be sent to the client denoting that file upload was not successful.

**6. How file concurrency issues will be handled in the document vault system?**

**Description:** Since, document vault is a client-server based architecture; there will be several users who will work on the same file or updated the same file at same file. For example, a client can send the update request for a document and other users are requesting the same document. So there should be an efficient way to deal with these file concurrency issues because impact of file handling will impact the overall performance of the system.

**Solution:** There can be following possible solutions of this problem:

**A) Locking Operations:** First solution to provide the concurrent read and write on documents are to provide the locking mechanism on basic operations like read and write. When a client will request to extract a document's content, system will ensure that document is not currently locked by some other thread. If it is locked by some other thread, the reading thread will go to sleep for a certain amount of time and then it will again try to read document. Similarly, write operations on a document will ensure that document is not currently locked. If the document is not locked, it will take a lock on the document, commit the changes and release the lock so that other requests can use this document.

**B) Concurrent reads and exclusive write access to documents:** To overcome the file concurrency issues, we can use the threading model where multiple threads can read the same document at a time but when a thread wants to write some data to document, it will take the lock on file and after it finishes writing, document will be available for the other threads to write. .NET library's ReaderWriterLockSlim class can be used in this purpose which allows multiple threads to read the shared resource but write access is grant exclusively to only one thread for shared resources.

**7. What will happen if the connection between client and server is terminated?**

**Description:** Since Document Vault is a network based application, it is possible that communication between client and server is terminated in between a transaction. Here transaction means file upload, file download or normal message passing.

**Solution:** There are two cases in this problem:

- A) If the connection is terminated while client is idle means no request is in pending state for the client, client tries to connect again to the server until the connection is successfully established. In this approach, if an attempt to connect to server is unsuccessful, thread will try again to connect to server but after taking some delay. This delay can be configured at the time on implementation.
- B) If the connection is terminated while a transaction is in process. Let's say client has sent a request to perform the text-search based on some keywords and before client receives its response, client's gets disconnected from the server. In this case when the server will send the response to client, it will not reach to the client. Solution to this problem is that when server is not able to send response back to the client, it will create a separate thread to ping the client after certain time delay. Server will try to contact to a client for a certain limit and after that connection will be considered as terminated. If client gets connected in between the try attempts, response will be sent to the respective client.

**8. If the vault contains thousands of documents, invoking map builder at the start-up will take considerable time. So, in this case, how the performance of the system can be optimized whenever server restarts?**

**Description:** In a vault of thousands of documents, to scan all the files and read the child dependencies will take significant time. It will impact the start-up time of the server and overall performance will be degraded.

**Solution:** In this case, it is better to keep the parent list of all the documents persistent. When the document map builder is invoked for the first time, it will scan all the files and parent list will be created. After that, it will create a separate xml file for this list. This way, whenever server restarts it will not have to scan all the files to create parent map list. Instead, it will use the existing xml file.

Similarly, at the time of inserting a file in vault, this xml file will be updated so that it will always be in sync with the latest documents.

**9. If users are allowed to create categories, there will be several categories in the vault. In this case, how the vault will be managed?**

**Description:** If users are allowed to create categories, there will be several categories in the vault which serves for only one or two users. In this case, category list will be huge and overall time to execute the queries will go up.

**Solution:** Solution of this problem is that document vault server should have an admin interface through which administrators of the system can manage the categories. Higher management people in an organization will work as an administrator of the server and whenever a new category is created by any of the

user, the approval will go to the administrator. So, that only relevant category will be there in the vault.

**10. What will happen if an inserted file's dependencies are not there in the vault?**

**Description:** It may happen that a client sends a file for the insertion in the vault with a dependent file which is not there in the vault.

**Solution:** In this case, server will check the whether the dependent file is there in the system or not. If the child is not there in the system, file will not be inserted in the vault and error response will be thrown to the client saying that "dependent file is not there in the vault".

**11. What will happen if a client is viewing a file and another client updates the file contents or dependencies?**

**Description:** It is possible that a client's currently selected file is updated by some other user. So, in this case, first client will see the wrong information as the file is modified by some other client.

**Solution:** Solution of this problem is that client can maintain subscription model on files that are currently used by active clients. When a client will request a file for the download, client will store the entry of that client with the file name in the subscription map with file name as key. So, whenever a file will be modified by the other client. A notification event will be fired to all the clients who are there in the subscription list of that file.

Although, this approach causes a very complex view of managing the subscription since there will be thousands of file in a vault so the system will have to manage thousands of subscription, however there will be only one selected file for the user at a time. So, whenever a client requests a new file from the server its previous subscription is deleted and new subscription is created with new file. This way number of subscriptions will be proportional to the number of active clients.

**12. If multiple clients are uploading a file with the same name but different content, how system will react in same category?**

**Description:** In a large distributed system, it is possible that multiple clients are uploading a file with same name with different contents.

**Solution:** System will use the file from the client whose request is processed first and it will give an error message to the other clients that file already exists.

**13. How security will be maintained within the system without any client authentication?**

**Description:** Since this is network application, same document vault client executable can be passed to other users who are not authorized to use the document vault application.

**Solution:** There can be two possible solutions for this problem:

- A) Using access control list:** Access to the document vault can be secured with the help of windows access control list. Windows access control lists [2] define the way to add trustee users in a network. Access control lists govern which users can receive messages from a URL. This feature can be used to give access rights of the system to only selected users.
- B) Server side admin interface:** A separate admin interface module can be created at the server side to define the list of allowed IPs and groups. Administrators can use this interface to configure the server and they can define the access privileges for the system.

#### 14. What will happen if the files in the document vault are not used from a long time period?

**Description:** In a document vault there will be a set of files which are not used from a very long period of time. For example, documents used in the development of a feature will be frequently used in that development cycle and later on the frequency will be reduced as the team will work on other features.

**Solution:** if a document is not longer used from a period time, it should be sent in the archived documents. There should be some policies defined regarding the max idle period of a document by the administrators of the vault system. We can use this approach while implementing the system. There will be a separate thread which will run in background and it will check the <Last-Modified> tags of documents and if the document is not used from the defined time, it will be archived.

## 9. Conclusion

---

This document described the architecture of the Document Vault system that can be used in implementing the system. Solutions of some critical issues are also discussed in this document so that the implementation will not contain serious flaws. In conclusion, Document Vault system provides following features:

- Powerful graphical user interface to insert and view documents in the vault.
- View categories and associated files.
- Gives the ability to user to decide the file dependencies and then shows the parents-children mapping of selected file.
- Search text using user defined queries.
- Search meta-data files for user specified tag values in the vault.
- Generates meta-data files for inserted documents.
- Ability to edit meta-data of a document by supplying tag names and values.
- Define a new category in the vault.

Additionally, Document Vault system can be very effective for medium and large industries where documents are frequently used in various phases. For example, document vault can be considered as code repository and it can keep the source code files and various design documents related to software features. On the top of existing system, layers can be added to customize the system according to the applications.

## References

---

1. <http://msdn.microsoft.com/en-us/library/dd129907.aspx> [Caching the Distributed Environment ]
2. [http://msdn.microsoft.com/en-us/library/windows/desktop/aa374872\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/aa374872(v=vs.85).aspx) [Access Control Lists]
3. [http://en.wikipedia.org/wiki/Use\\_case](http://en.wikipedia.org/wiki/Use_case) [ Use Cases ]