**CIS 681- Software modeling and analysis**

**“REMOTE DOCUMENT VAULT”**

*Operational Concept Document*

*VERSION: 0.1*

**Syracuse University**

**(October 21st, 2013)**

***Instructor: Jim Fawcett, PHD***

PRIYANKA B.TAYADE

SUID: 42687-3258-1

**TABLE OF CONTENT:**

1. Introduction………………………………………………………………………….….. 04
2. Users .…..………………………………………………………………………….……. 05
3. GUI…..…………………………………………………………………………….……. 09
4. Design…..………………………………………………………………………….……. 14
5. Operation………...…………………………………………………………………….... 21
6. Startup document vault…..………………………………………………………….........22
7. Startup document vault..…………………………………………………………...……. 24
8. Text search operation…...…………..……..………………………………………….…. 26
9. Metadata search operation.………..………………………………………………….…. 28
10. Edit metadata operation….…….…………………………………………………...……. 30
11. File upload operation ……...………………………………………………………….…. 32
12. Critical issues………… …..………………………………………………………….…. 34

**LIST OF FIGURES**

1. Network diagram of document vault……..………………………………………….….. 06
2. Graphical user interface for text search………...………………………………….……. 09
3. Graphical user interface 2 for text search ………...……………………………….……. 11
4. Secondary file management view…..………………………………………………….... 12
5. Graphical user interface for metadata search…..…...……………………………….........13
6. Category structure.……….………………………………………………………...……. 14
7. dependency map ………..…………………………………………………………….…. 16
8. Package Diagram for Remote Document Vault……….……………………………….... 18
9. Activity diagram for startup document vault ………............……………………….........22
10. Activity diagram for startup client..…………………...……………………………...…. 25
11. Activity diagram for text search operation………...………………………………….…. 27
12. Activity diagram for edit metadata operation…...….……………………………...……. 29
13. Activity diagram for edit metadata operation …...…..……………………………….…. 31
14. Activity diagram for upload file operation..………...……………………………...……. 33
15. **INTRODUCTION**
    1. **PURPOSE**

Remote document vault is a repository to store documents. The purpose of this vault is to provide facility to store large volume of documents in an organized way. Each document has its associated xml file holding attributes of the document such as keywords, dependency, description, version number etc. called as metadata. The Document vault has document vault client which is a standalone application on client machine. The purpose of the vault is insertion and extraction of documents from a remote document vault and also to display file information and their properties i.e. metadata. The vault has added functionality of search. Search can be text search in text files or metadata search in xml files. The result of the text search is displayed as the list of files in which match for the search text was found. Metadata search return metadata files content of the xml file for which the match was found. This information is displayed in information view. During insertion of documents the vault maintains the properties of the documents in xml file which is associated with the uploaded file.

The vault classifies each file in one or more categories. The vault provides a facility to navigate through categories and get the list of all the files in the selected category. The vault also provides the list of dependency for each file. Category is one form of dependency. Dependency is if one file is some way dependent or is derived from other file/files then the file/files in which one file dependent on is called its parent and vice versa then that file is called child. Vault has the functionality to map such kind of association.

We can summaries the functionality as follows.

1. Store documents in the document vault.
2. Help to search for files.
3. Upload new document.
4. Maintain metadata for each document
5. Edit metadata for already saved metadata files.
6. Categorize document based on category.
7. Maintain parent and child relationship between files.
8. **USERS**
   1. **CLIENT**

The Client is someone who is a registered user. Client uses the application which is managed by an owner. There can be various people with same or different requirements using this document vault to manage their documents.

* 1. **JOURNALIST**

Journalist writes facts, articles or stories for a newspaper. Therefore a lot of documentation is done by a journalist. To assemble all these documents in a meaningful way, journalist may need a document vault for that purpose. Therefore a journalist can find an article very easily by performing text search or metadata search. He does not have to go through all the files to find his required article or need not rely on the name of the file to find a particular content. Journalist may also need to find related articles. Therefore he can find such articles very easily using document vault category functionality or can find parent or child files for that article as well.

* 1. **WRITER**

A writer may need to do research on various texts he can do it easily with the help of search functionality. Writer may also need to save various drafts of his written material. These drafts can be saved in the document vault in form of versions. The metadata files of these drafts will change in version each time the new draft is saved. The writer can find for a particular string in any draft if he has deleted the data from next draft. Therefore writer will not need to remember which file holds what data by seeing the metadata for that draft he can easily get that information and also with the help of search functionality.

* 1. **DEVELOPER**

These developers are client. In a source code many class files can be using functions from other class files or may be inheriting the properties of the other class files. Pulling out all the files which are depend on the given files will be a tiresome task as the developer will need to go through each class file code. They can need store their source code files in the document vault as the vault given a functionality to organize the files in hierarchical way in form of parent child relationship. Therefore the developer will be able to pull the required data very easily without going through any code each time he need to find a related file.

* 1. **TESTERS**

Testers/ QA team can be client using the remote repository. Testers do a lot of documentation in form of test cases. There are various rounds of testing few of the bugs may get resolved and few of them may get reopened. So they need to keep a track of all issues resolved and unresolved. This may create a lot of files which are depended on each other like a test case file may be dependent on the source code file. If the tester is doing automation testing then there can be interdepended test case code files for the automated test cases and a test cases version can also be maintained for each round each round of testing. Therefore with the use of document vault the tester will be able to find all its documents very easily. This will save them a lot of time.



*Fig: 1 Shows the how the vault is used*

Fig1. Shows a network diagram that shows the implementation of remote document vault. In the figure many client over the network are using the service (remote document vault) provided by the organization. The organization maintaining a remote document vault can be cloud based. Users all over the internet are able to use this vault. There is a vault client sold/distributed to each user and with the help of this vault client application the user is able to connect to the remote document vault. The vendor can be someone different from the person who is providing the service of the vault to the users or they may be the same person. We will understand users within the company or organization providing the vault service.

1. DOCUMENT VAULT SERVICE PROVIDERS.
   1. **ADMIN/ MANAGER:** These are the users who have special rights to configure the system. That is deciding the category name or number of categories etc. Their job is to take decision and monitor the system. The manager/admin control panel has several/all functionality which is not provided to any general user. For example can have access to all the files in the system.
   2. **OWNER**: Owner is the one who has purchased the product from the vendor for his or her own use. The nature of use decided by the owner makes the system public, private or protected or for personal or organizational use. Hey may wish to have admin panel and have access to all the settings and functionality of the system which will be discussed in length in coming sections.
   3. **VENDOR**: Vendor is the one who has manufactured and has sold the product in the market. Vendor many not have any special or rather any right to access the system once it has been sold. But they are interested in keeping a track of the error log or other statistic report. This can be used for bug solving and further development/ R&D of the project. Therefore the vault can have an error reporting tool.
   4. **DEVELOPER TEAM:** These are the people who have designed the document vault. Developer might be given some temporary admin or user id so that developer can build or redesign the tool and perform his unit test cases. Developer might also need access to the error log so that they can keep track of bugs and make the system efficient.
   5. **QA TEAM (QUALITY ASSURANCE TEAM):** Testers might need access to the error log so that they can keep track of bugs and run their test cases accordingly. Tester might be given some temporary admin or user id so that tester can run his/her test cases.
   6. **DATABASE MANAGER (DBA TEAM):** Some ofsystem data such as user details can be stored in the database of the organization. Therefore the tool needs to communicate with the database. And to maintain the database the DB team needs to have access to both tool and database of the system.
   7. **NETWORK MANAGER:** Network manager has a crucial role when the system is cloud based or used in an organization. The system should able to record all the details necessary for network admin to manage the system.
   8. **POLICY MANAGER/ BA:** These people decide the policy of the company and this might largely change as how the tool is implemented. For example if they are allowing user to view all the files in the system or the user can only view his or her files. They can decide to allow users to edit categories etc.
   9. **CUSTOMER CARE:** If any issue arises with the software or service. Customer care representative might be able to solve such issue. They can have some special rights such as viewing all the user data and having access to the system. They may be able to block certain files in case of license, expiration issue or other issues and even at times register complaints of violation of privacy or copyrights or in case of abuse. They may be able to reset user password.
2. **GUI**

GRAPHICAL USER INTERFACE FOR TEXT SEARCH

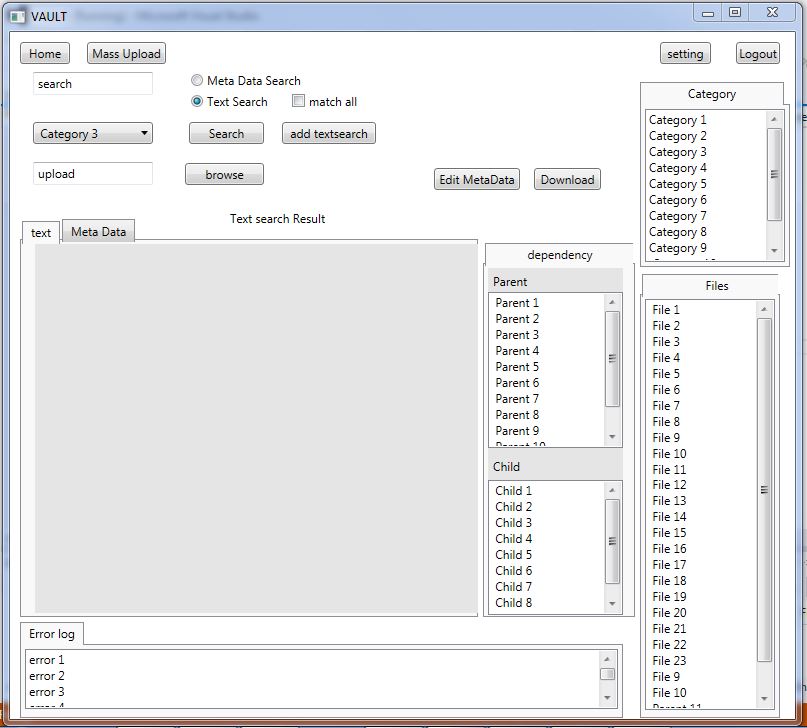
****

Fig 2: Graphical user interface for text search

The text search result will be displayed as shown in the GUI above.

1. Home button helps us to navigate from any point of operation back to the user home screen.
2. Text box having text holder as search is used to perform text search or metadata search.
3. User will have to select metadata search or text search radio button to notify which search the user wants to perform.
4. By default text search radio button will be selected.
5. If the text search radio button is selected then user can add one or more text box by click on add textsearch button to perform multiple search operation.
6. User can check match all radio button to notify that he wants to perform an ‘and’ operation for text search i.e. the result must include all text search string
7. If user selects on metadata radio button then match all check box will be disabled.
8. User can select any category/categories in which he wants to perform the search operation from the select box which is on the left side of screen. It allows user to select multiple inputs by clicking on each one of the element of select box and the selected elements will be populated.
9. If the user hits search button then the list of all the files matching the search result will be displayed in file section on the right most side of the screen.
10. In the figures above user has selected category 3.
11. If the user clicks on any of the files listed on right side which are the result of text search then that file becomes the current file and text and metadata for that file is displayed.
12. File text and metadata is displayed in tabs.
13. The list of parent and child will also be displayed in dependency section.
14. If the user clicks on any of the parent or child file from the list then that file becomes the current file and text and metadata for that file will be shown, also the parent child list for that new current file will be updated.
15. If user clicks on edit metadata button then a popup containing all editable metadata fields will be shown. These fields will be holding previously held values.
16. User is able to edit the data in these files and can press submit to start the operation of editing. If user clicks on reset button then these metadata fields will rest to their original values.
17. User can also click on download button to download the document and xml for the current file to their machine.
18. There is an error log which is present at the bottom most part of the screen. This error log informs the user about all the error occurred while processing via customized error messages.
19. error log may contains all the file names which do not have associated metadata data files or the files it couldn’t read or any read write operation error.

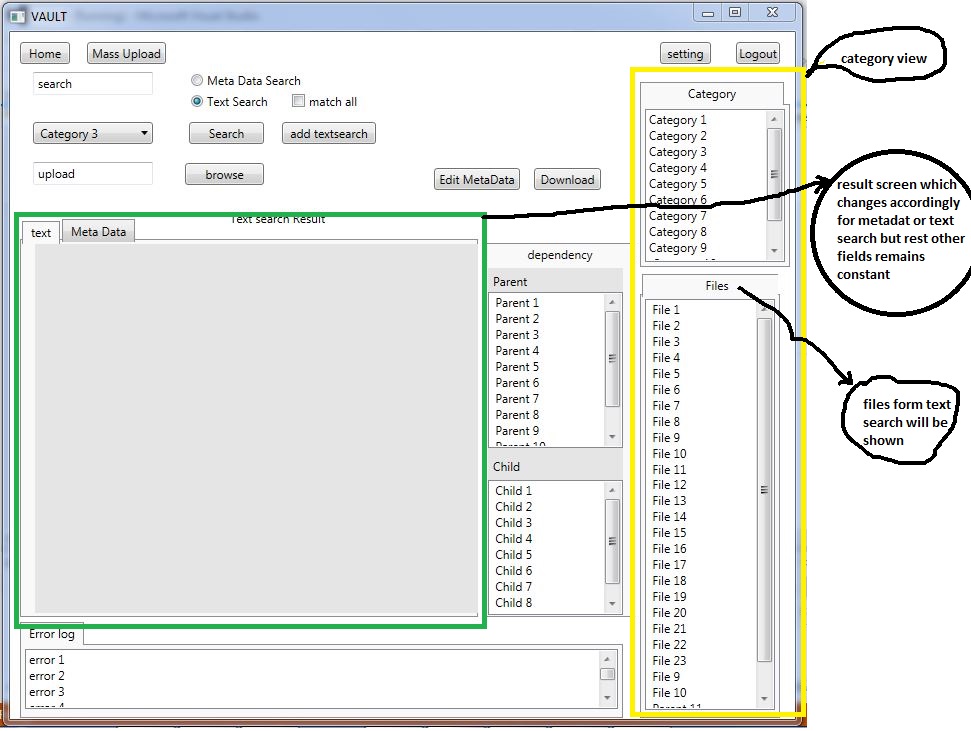


Fig 3: Graphical user interface 2 for text search

1. Category and files section which are on the right most part of the screen which is shown in yellow in figure above makes up category view.
2. The green box in the above figures changes accordingly for text search or metadata search respectively.
3. If user clicks on any category from the category view then all the files in that category will be listed. User can click on one or more category.
4. Mass upload buttons allows us to perform upload file operation for large chunk of files at a go. The functionality of mass upload is explained in critical issues.
5. Logout button ends user session.
6. Browse button helps user to browse local files. The figure below shows it.

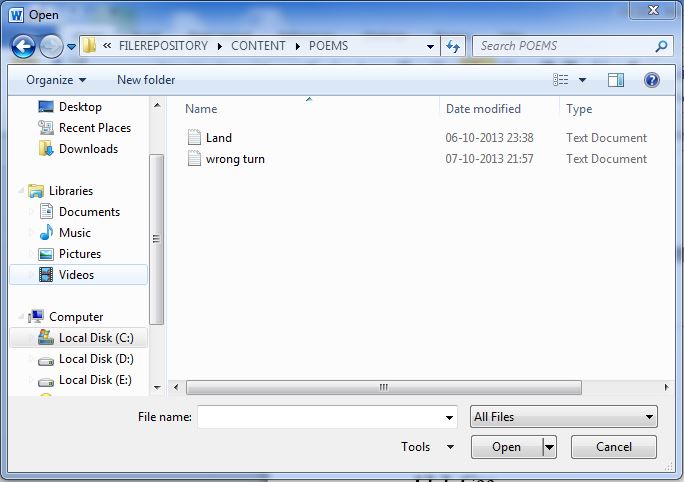
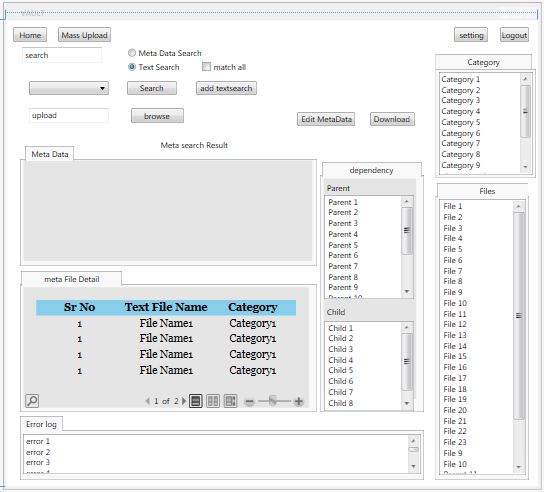
****

Fig 4: shows secondary file management view.

The secondary file management view is for local browsing of files. This acts a popup when user clicks on browse button. User can navigate and select their desired file for upload. After clicking on open button the file will be loaded and another popup to input metadata values will appear. User will input the metadata values and will hit submit.

GRAPHICAL USER INTERFACE FOR METADATA SEARCH



*Fig5: Graphical user interface for metadata search*

GUI for metadata search is shown in the figure above. The screen contains a metadata section which will display metadata content. The resultant file names of the metadata files will be shown in metadata file section for metadata search operation. If user clicks on any of the file the metadata content for that file will be shown. User can download the content of the files of metadata search result.

1. **DESIGN**

The vault consists of a vault client and a remote document vault located on server side. Vault client is a standalone application on client side on client machine. Vault client communicates with the document vault over the network or internet. Client needs to first establish a connection with the remote document vault before using it. Client establishes the connection by authentic login using their user account. The document vault is accessible to client after the successful user login. In this section we will see how the vault is designed.

* 1. **REMOTE** **DOCUMENT VAULT**

Remote Document vault acts as a file Repository. To understand the working of document vault we need to familiarize our self with the following vault functionalities.

1. Category
2. Dependency
3. Directory structure
4. Dependency map
5. Text analysis tool
6. Metadata tool
   * 1. **CATEGORY:** 

*Fig: 6 Category structure*

The figure above shows the classification of files based of categories. This will allow the system to better maintain the document vault. Root is the main xml file which contains information of all the categories. The categories are xml file which contains detail of that category and also maps to their sub categories. The sub categories may again have their sub categories if not then they have text file and xml file mapped to them. Each file has its associated xml mapped to it.

A file can belong to one or more category or subcategory.

CATGEORY NAME: at first is system defined but can be edited by the user once the document vault is installed.

CATGEORY NUMBER: Number of categories by default will be 5 or per defined but the user can increase the number of categories as per their needs.

* + 1. **DEPENDENCY:** If the file is a sub file or a continuation of some other file or is depend on some file or bunch of files. Such association of one file with other file is known as dependency. Grouping files in categories is also a dependency. Dependency is very useful in managing files meaningfully. We get a parent child relationship applying dependency concept.

Parentis the file on which other file/files are dependent on. This dependency can be of any form for example a file ‘Othello’ (play written by Shakespeare) can be dependent on some files named ‘Work of William Shakespeare’. ‘Work of William Shakespeare’ is the parent of child ‘Othello’. There again can be some other file named ‘Commentaries on play-Othello’ which is dependent on file Othello. Thus in this case ‘Othello’ becomes the parent and ‘Commentaries on play-Othello’ becomes the child and so on. We can have one or more child to file. We can have one or more parent too. A file can be dependent on one or more files. Thus a file can have more than one dependency. For example file ‘annual revenue’ can be dependent on files ‘agriculture sector’, ‘Industrial sector’, ’Banking sector’, ‘Tourism sector’ and so on. Therefore file can have one or more parents. Likewise a file can also have one or more child example file ‘Romeo and Juliet’ can also be dependent on file ‘Work of William Shakespeare”.

From the examples above one must have come to know the use of dependency and how dependency would help extract the related files easily. All the information of dependency is stored in the directory map which will be discussed later.

* + 1. **DIRECTORY STRUCTURE:** The directory structure follows category structure which is the logical classification of files i.e. categories. Folders are named against their category name. Sub folders are named against sub categories. Text files and xml file are maintained in their respective category folders. Figure 4 above demonstrates the category structure and the directory structure as well. The root represents the main directory. Each category represents folders and each sub categories represents folders within them.

The root folder contains the dependency map xml files. Directory structure is the replica of category structure a file can belong to one or more categories. But we cannot store each copy of that file in each category folder it belongs to. As this will create redundancy of data therefore we store this file in only one of the category folder which has the least category id. Category id is just the number assigned to the categories instead of taking their full names. Thus in this was physically the file will be present in only one category folder but logically it will be mapped to its respective categories with the help of associated metadata file.

* + 1. **DEPENDENCY MAP:** Dependency map helps to store parent child dependency for a file. This parent child dependency information is stored in an xml file named dependency map. This file is located in the project root directory. The dependency map xml file is updated each time the document vault is launched and also when a new document is uploaded in the document vault.

****

*Fig 7: dependency map*

Figure above shows a demo dependency map. The map will have category node and then the file node which is present in that category. The parent and child nodes if parent and child exist for that file. Therefore the number of category nodes will be number of category present. Number of file node will be number of files present in that category. The figures also shows clearly how the file id is make up of.

* + 1. **TEXT ANALYSIS TOOL**

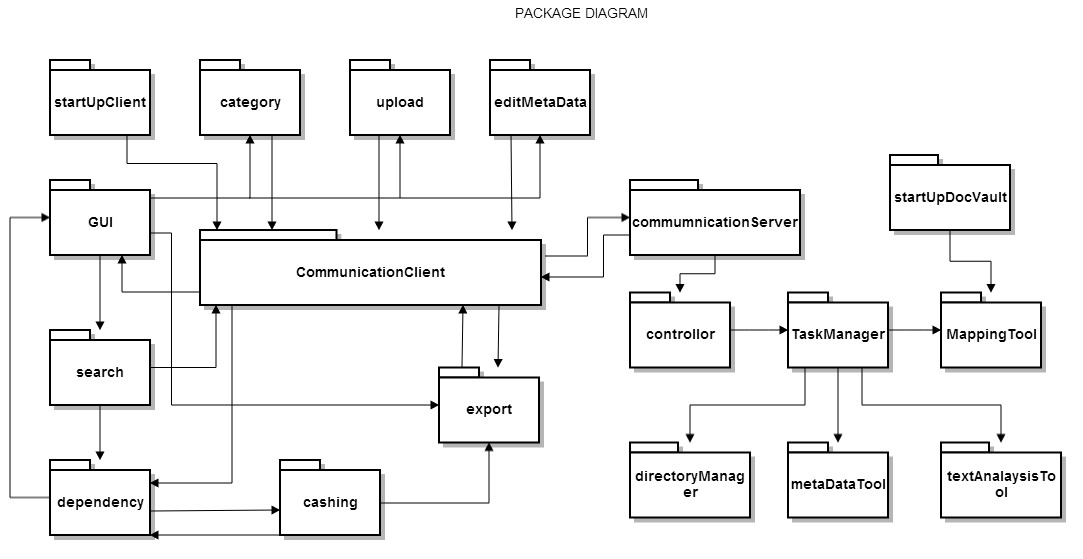
Text analyzer is a tool which takes input in form of a query and generates result for that query. There are two forms of input query one is text search query and other is metadata search query. User is able to match the required text sting/strings from the bunch of files in the document vault. The search result will return the fully qualified name of the text file matching the search criterion.

* + 1. **METADATA TOOL**

Metadata tool also takes input in form of a query and generates result for that query. The metadata tool performs generation of metadata and help to edit previously generated metadata. Metadata information is stored in xml file which is associated to the document whose metadata is being stored.

* 1. **PACKAGE DIAGRAM**

Figure below describes the working ok remote document vault different packages are related to each other.

****

*Fig 8: Package Diagram for Remote Document Vault*

The figure above can be explained as follows. The communicationClient is on client side and it deals with communication with server. Likewise communicationServer package is dedicated to communicate with clients. Both communicationClient and communicationServer uses windows communication foundation (WCF) to communicate over the network. WCF is use to communicate with remote standalone application. It can understand many different protocols and because of the ease of communication as WCF follows standard SOA (Service-Oriented-Architecture) it is widely used.

StartupClient is dedicated to launch vaultClient on the client machine. It establishes the connection with the server. The server has document vault which is located at remote site. I also perform user authentication and saves user session. Until the session does not close user can access document vault.

Events are generated with the help of graphical user interface by click on buttons, links, tabs or elements. With the help of event we perform search or file upload or edit metadata operation by calling their respective functions.

If user hits text search then event will call function from search package which will perform decision making i. e text search or metadata search and will send the search criterion to communicationClient where client will send this data in form of messages. This messages will be received at server side though communicationServer.

Search package also communicates with dependency package dedicated to handle dependencies i.e. parent child relationship for a particular file. This package will get the dependency data from cache which is present at client side using cache package. Then dependency package will refine data for the current file and send it to GUI for display.

The parent child mapping information is stored on client side by cashing. Cashing package keeps this information updated. This will be further discussed in detail in critical issues.

If the user clicks on any category in category view then the files contained in that category will be displayed. In order to accomplish this task an event will be raise from GUI which will call a function from category package. Category package will pass on that data to the communicationClient to form a request to the server in form of message. The server will in turn provide all the files in that category to communicationClient which will again pass the information for display to the GUI.

In the same way as search package and upload file package edit metadata package is responsible for editing metadata for the text file.

Export package is responsible for all download operations. User is able to download the search result in form of pdf or excel or doc file.

Startup doc vault run each times the server is restarted or the remote document vault is launched. Is responsible to create and update dependency map which stores parent and child relation for all the files.

Controller is responsible to manager multiple user request and process them i.e. understand what task is to be performed. For example search or file upload etc.

Task manager assigns threads for each task which controller has decoded on basis of user messages. It makes the decision of task to be performed by assigning thread to each task.

Mapping tool package is responsible to make dependency i.e. parent child map of the system. The process for this is explained in startup client. Dependency map updated each time the system the document vault is restarted or each time user edit metadata or upload a new document in the remote document vault.

Directory manager package deals with directory operation such as create folder to save file. As explained in the directory structure each category will represent a directory. If the user creates a new category we need to create a new directory for the same.

Text analyzer is a tool which takes input in form of a query and generates result for that query. There are two forms of input query one is text search query and other is metadata search query.

Metadata tool also which takes input in form of a query and generates result for that query. Metadata tool helps in creation and editing of the metadata file.

1. **OPERATION**

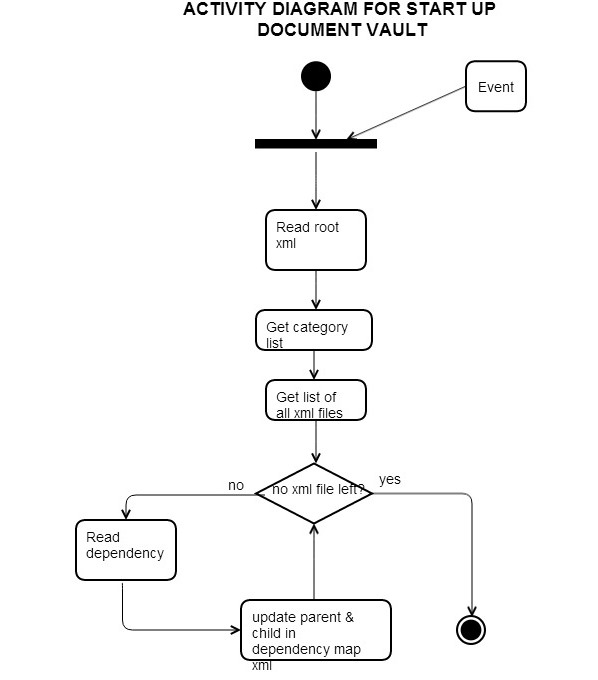
Operation of the vault can be divided into several activities. Each activity represents a functionality provided by the vault. They are listed as follows

1. Startup document vault
2. Startup vault client on client machine.
3. Text search operation
4. Metadata search operation
5. Edit metadata
6. Upload file

We will go through all the functionality one by one in detail in next following sections.

1. **STARTUP DOCUMENT VAULT**

Startup Document vault is responsible for doing some initial processing before using the document vault. The initial processing is to make a parent child map for each file in the vault as explained earlier in design section. The process for startup is given in the activity diagram that is shown in the figure below.

****

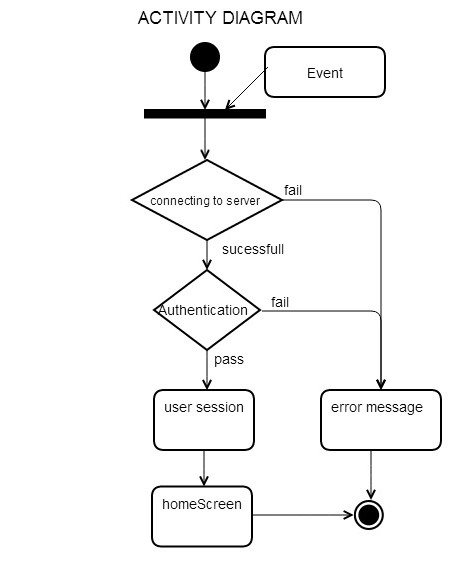
*Fig 9: Activity diagram for startup document vault*

* 1. Mapping tool is triggered with a launch event of the vault.
  2. The tool reads the root xml file which is present in the root directory of the vault
  3. The root xml contains details of categories. In this way the tool can know which all categories to scan.
  4. The tool gets the list of xml files from all the categories and their sub categories one by one.
  5. After which the tool scan each file one by one for dependency data and updates it in the dependency xml file.

1. **LAUNCH VAULT TOOL ON CLIENT MACHINE.**

Startup Client as the name suggests runs each time the client is launched on client machine. Startup client will establish the connection with remote document vault and also saves user session.

The server connection details are stored in the system by default. But can be reconfigured by the user in case the server is changed

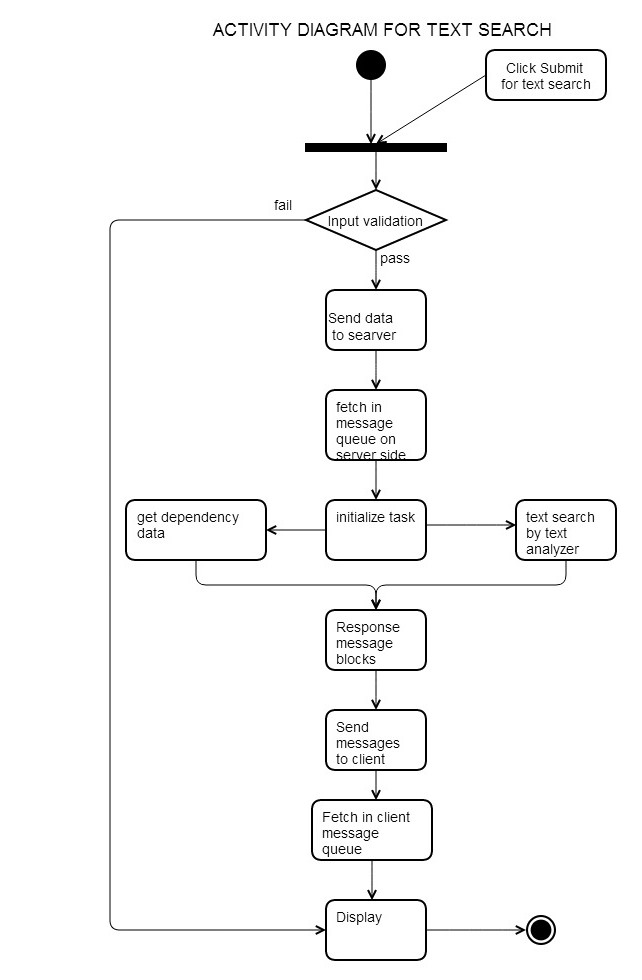
****

*Fig 10: Activity diagram for startup client*

* 1. The operation of startup Client will be triggered by launch event.
  2. Then the client will try to establish the connection with already saved server details such as IP, port number, server name, type of connection etc.
  3. If the client is able to reach to the server then the client will provide authentication details to the server such as client ID and password.
  4. If client is unable to reach to the server due to wrong server credentials or network problems etc. then a customized error message is displayed to user and the activity is ended.
  5. If the authentication is a success then client’s user session will be stored and client can access the server data till the session is present and user is directed to home screen.
  6. If the authentication fails then a customized error message is displayed to user and the activity is ended.

1. **TEXT SEARCH**

The vault provides text search functionality. The user can input text string and the text search functionality will help find the matching text string in the remote vault. The text search is performed on document files and not on xml files. The remote document vault contains a text analysis tool which accepts text strings and finds a match if any from the document files in the document vault. Thus the documents client need to take input from user for text search and communicate to the remote vault. The activity diagram explains the text search functionality in detail.

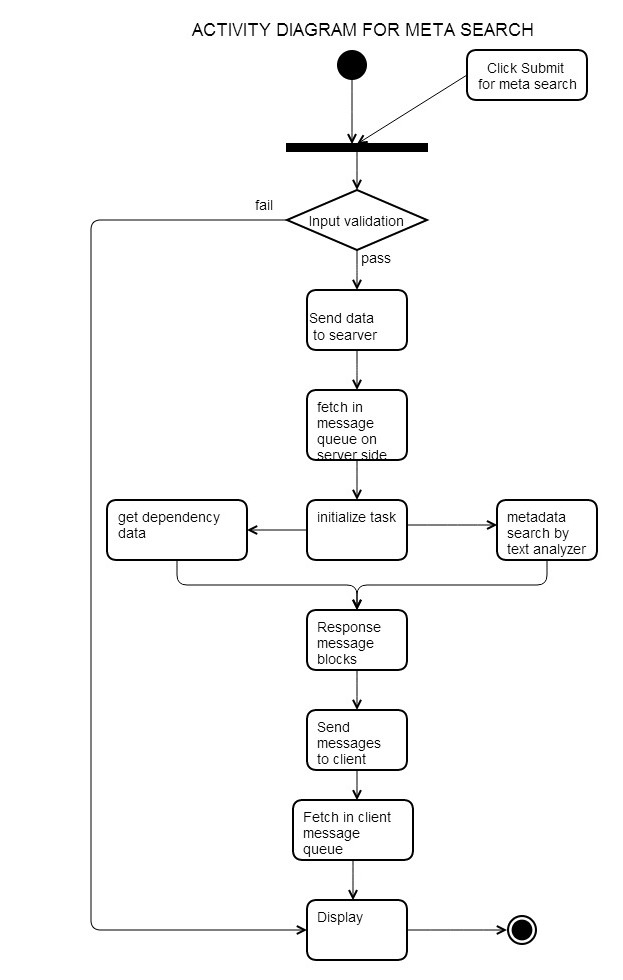
****

*Fig 11: Activity diagram for text search operation*

* 1. When submit is click by selecting text-search the event triggers the text search operation.
  2. First the user input is validated. Like text box field for text search should be nonempty etc. System also checks for user session. If the session has expired then user cannot send the data to server.
  3. This validated data is converted into messages and send over the network to the remote server with help of windows communication protocol.
  4. The server receives the messages and stacks them in message queue. Message queue is important as multiple users must be using the server. Therefore there will be a lot of messages coming in while the current message is still waiting for execution. Thus the remote vault will maintain a stack of these messages.
  5. Task will be assign for each message. Two threads for each message will be having the same priority. These threads will be executed simultaneously therefore each user text-search operation can be performed simultaneously. User may send multiple messages but task will not be assigned to the new message till the previous message is completely executed. This will allow equal time slice to multiple users.
  6. Task will be assigned on first in first out basis from message queue.
  7. One thread will run text analysis tool and the other will get the dependency i.e. parent child relation for the given category/categories.
  8. If category is not mentioned by the user in the search criterion then dependency of all the files should be sent as parent child data
  9. The result of the text analysis tool and the parent child data for the files will be converted into message blocks.
  10. These messages will be sent to the respective client and will be fetched in client message queue.
  11. Data from these messages are then displayed for the user.

1. **METADATA SEARCH OPERATION**

The vault provides metadata search functionality. The user can input tags and the metadata search functionality will help find the matching metadata tags in the remote vault. The metadata search is performed on xml files. The remote document vault contains a text analysis tool which accepts metadata tags and finds a match if any from the document files in the document vault. Thus the documents client need to take input from user for metadata search and communicate to the remote vault. The activity diagram explains the metadata search functionality in detail.

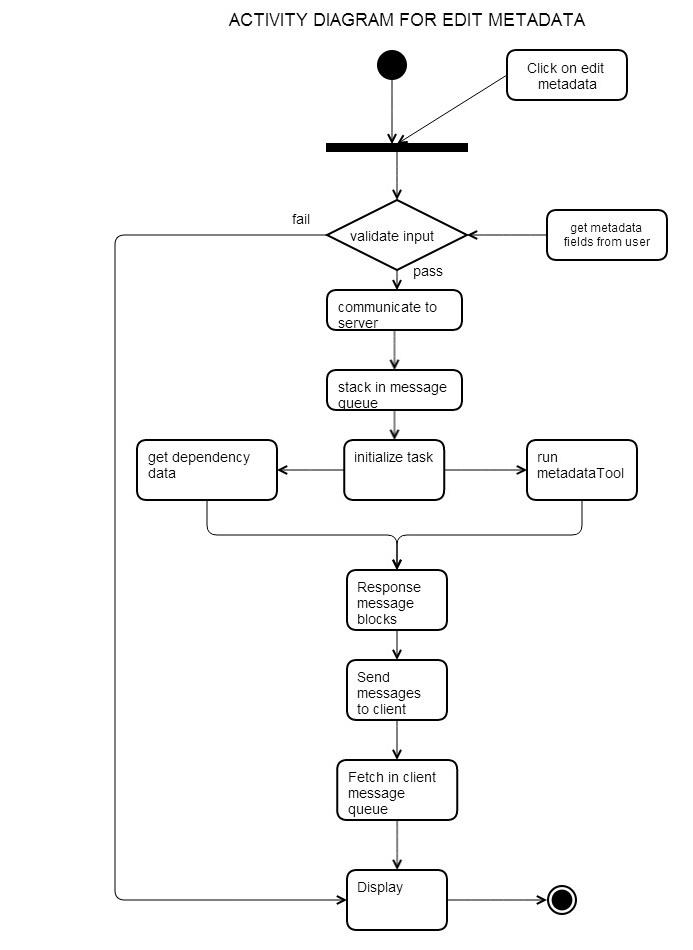
****

*Fig 12: Activity diagram for metadata search operation*

* 1. When submit is click by selecting metadata-search the event triggers the metadata search operation.
  2. First the user input is validated. Like text box field for metadata search should be nonempty etc. System also checks for user session. If the session has expired then user cannot send the data to server.
  3. This validated data is converted into messages and send over the network to the remote server with help of windows communication protocol.
  4. The server receives the messages and stacks them in message queue. Message queue is important as multiple users must be using the server. Therefore there will be a lot of messages coming in while the current message is still waiting for execution. Thus the remote vault will maintain a stack of these messages.
  5. Task will be assign for each message. Two threads for each message will be having the same priority. These thread will be execute simultaneously therefore each user metadata-search operation can be performed simultaneously. User may send multiple messages but task will not be assigned to the new message till the previous message is completely executed. This will allow equal time slice to multiple users.
  6. Task will be assigned on first in first out basis from message queue.
  7. One thread will run text analysis tool and the other will get the dependency i.e. parent child relation for the given category/categories.
  8. If category is not mentioned by the user in the search criterion then dependency of all the files should be sent as parent child data.
  9. The result of the text analysis tool and the parent child data for the files will be converted into message blocks.
  10. These messages will be sent to the respective client and will be fetched in client message queue.
  11. Data from these messages are then displayed for the user.

1. **EDIT METADATA OPERATION**

Edit metadata module edits the metadata information of the selected document. If the user clicks on edit metadata any point of time. The current file is taken as the file for editing its metadata. The following figure explains the operation of edit metadata functionality.

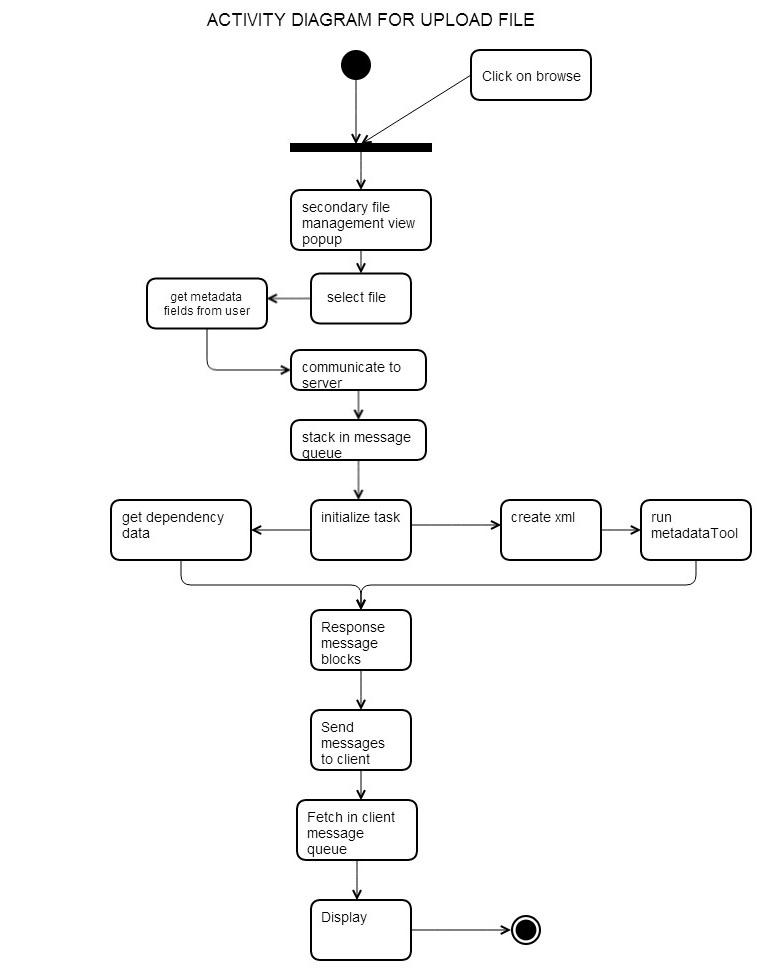
****

*Fig 13: Activity diagram for edit metadata operation*

* 1. Click on edit metadata will trigger the event for edit metadata operation.
  2. User will input new metadata tags which will be validated. All the mandatory fields should be field and user should input valid input such as only alphanumeric characters allowed. System also checks for user session. If the session has expired then user cannot send the data to server.
  3. This validated data is converted into messages and send over the network to the remote server with help of windows communication protocol.
  4. The server receives the messages and stacks them in message queue. Message queue is important as multiple users must be using the server. Therefore there will be a lot of messages coming in while the current message is still waiting for execution. Thus the remote vault will maintain a stack of these messages.
  5. Task will be assign for each message. Two threads for each message will be having the same priority. These thread will be execute simultaneously therefore each user metadata-search operation can be performed simultaneously. User may send multiple messages but task will not be assigned to the new message till the previous message is completely executed. This will allow equal time slice to multiple users.
  6. Task will be assigned on first in first out basis.
  7. One thread will run metadata tool and the other will get the dependency i.e. parent child relation for the given file.
  8. If file does not have any parent then category will be the parent for that file.
  9. The result of the metadata tool and the parent child data for the files will be converted into message blocks.
  10. These messages will be sent to the respective client and will be fetched in client message queue.
  11. Data from these messages are then displayed for the user.

1. **UPLOAD FILE**

Upload file is similar to edit metadata operation. User will upload a new document and metadata for that file is created. Every document in the repository must have an associated metadata file. Therefore it is mandatory for the user to save metadata information while uploading the file. File upload operation is explained by following activity diagram

****

*Fig 13: Activity diagram for upload file operation*

* 1. Clicking on upload file button will trigger the event for upload file operation.
  2. A secondary file management view pops up which allows user of local browsing of files. The user selects a file and the client will upload that file.
  3. User will input metadata tags which will be validated. All the mandatory fields should be filled and user should input valid input such as only alphanumeric characters allowed. System also checks for user session. If the session has expired then user cannot send the data to server.
  4. This validated data is converted into messages and send over the network to the remote server with help of windows communication protocol.
  5. The server receives the messages and stacks them in message queue. Message queue is important as multiple users must be using the server. Therefore there will be a lot of messages coming in while the current message is still waiting for execution. Thus the remote vault will maintain a stack of these messages.
  6. Task will be assign for each message. Two threads for each message will be having the same priority. These thread will be execute simultaneously therefore each user metadata-search operation can be performed simultaneously. User may send multiple messages but task will not be assigned to the new message till the previous message is completely executed. This will allow equal time slice to multiple users.
  7. Task will be assigned on first in first out basis from the message queue.
  8. One thread will create xml and run metadata tool and the other thread will save the dependency data in the dependency xml. If the update is success then an update flag will be stored in dependency xml.
  9. This file will not have any dependency data saved previously.
  10. The result of the metadata tool will be converted into message blocks.
  11. These messages will be sent to the respective client and will be fetched in client message queue.
  12. Data from these messages are then displayed for the user.

1. **CRITICAL ISSUES**
2. **CONCURRENT USERS.**

There can be many users accessing the remote vault at the same time. This will increase the wait time for each user as the vault will process each user request at a time one by one on first come first serve basis.

**Solution:** All user requests should be processed more or less at the same time. Each request should be assignment separate thread. And therefore any uses request which takes a longer time will not make other users wait in queue for long.

1. **NETWORK TRAFFIC.**

Too many concurrent users will increase the load on the server.

Solution: in order to reduce this load repetitive data should be stored in cache on the client side. Large volume of parent child relation data will be sent each time. This data may not change as often. Therefor if is advisable to store the parent child relation data of each file i.e. the complete dependency map on client side.

1. **CHECKING FOR UPDATED DEPENDENCY DATA**

The parent child map which is stored at the client side in cache will be outdated if the dependency are changed by edit xml operation or upload file operation. Therefor user may be working on outdated data.

Solution: each time the dependency map containing the parent child relation is updated a flag is saved in that file containing timestamp representing latest update. This flag is different from all the previous flags. Every time a client request comes to the server, the server will check for the flag. If it is outdated then server will just send the updated information. And the client therefore will always work with updated data.

1. **CONCURRENT USERS ACCESSING THE SAME DOCUMENT.**

There can me two or more users accessing the same document at the same time.

The document vault should be able to resolve this issue.

**Solution:** The document vault gives the document to the user which requested first and a lock is maintained till the read operation is going on for that user request. Once the file is read it should release from the lock to be used by other users.

1. **LOSS OF DATA FROM SERVER TO CLIENT.**

If there was loss of packets or some network problem which had result to loss of data from server to client then client will have to send the request gain and the server will have to do the whole operation like text search and metadata search again. This increases the work for the server and will result into inefficient system.

**Solution:** the output buffer data which is being sent to a client should be maintained temporarily in a cache. Once the data is successfully passed on to the client then client should sent an acknowledgment message. If the server receives a positive acknowledgement message then it will empty the cache’s data. If the server receives a negative acknowledgement message then it will try to resend this data to the client. In this way it will not have to perform the operation such as text search etc. and generate the data again which was lost. Therefore the client and server will use connection oriented services in which data is passed to and fro followed by the acknowledgement message.

1. **MASS UPLOAD OPERATION:**

The tool would also be used by an organization because in the real life scenario large chunk of files can be uploaded by the user at a given time. It is not realistic to add metadata field manually for every single file upload. This process might require the user to spend hour and hour for file uploading.

**Solution:** The easiest solution to this problem is to give the user the option of mass upload. The system should specify the maximum number of files that the user can upload at a given time. After the mass upload the user can also upload a system readable file to generate metadata for the uploaded files. Therefor user won’t need to enter any information manually and the task will be efficient. System will have to incorporate excel reading class. User will have to upload the metadata in the excel sheet at the end of each mass upload.

1. **SYSTEM FAILURE.**

If some operation at the server throws an exception then it should not affect other operation for other clients and the vault should skip and try to continue with the rest of the operation. The system should not abruptly crash or stop working. It would be very in convenient for the users.

**Solution:** Proper error handling should be done by the developer. And proper testing should the done for the same.

1. **ERROR LOG.**

If there are any errors in the operation and the remote vault throws an exception then the exception will be properly handled and the operation will continue as normal. But several of the error will go unnoticed there for it is very important to have an error log so that the user can have the track of the degree of success and failure of the operation.

1. **RESUBMITTING OF THE FORM.**

If the user accidently clicks on the submit button twice then the search result will be sent to the server twice. Thus submission of the form twice before processing the first request should be avoided.

**Solution**: This can be done on server side by checking if the user input matches with previous input which is yet to be processed then the repeat message should be taken as the same.

1. **LARGE NUMBER OF FILES WHICH ARE NOT ACCESSED.**

There can be a large number of files which are not access from a long time. Such files which are lying redundant in the document vault will affect the vaults performance.

**Solution:** one solution to this problem is archiving the redundant files. These files will be stored in some other location. If any request arises for these archived files in future then user should be able to access them and these files should be restored in the vault.