VIRTUAL PERSONAL BOOKSHELF SYSTEM

BY

YEOH KEAT LIANG

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONS)
Faculty of Information and Communication Technology
(Perak Campus)

MAY 2015

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: VIRTUAL PERSONAL BOOKSHELF SYSTEM

Academic Session: MAY 2015

I ____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:
1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes

Verified by,

__________________________ __________________________

(Author’s signature) (Supervisor’s signature)

Address:

____

__________________________ __________________________

__________________________ Supervisor’s name

Date: ______________________ **Date**:_____________________

VIRTUAL PERSONAL BOOKSHELF SYSTEM

BY

YEOH KEAT LIANG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled "**VIRTUAL PERSONAL BOOKSHELF SYSTEM"** is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : __________________________

Name : __________________________

Date : __________________________

ACKNOWLEDGEMENTS

Sincerely thanks to everyone who helped to complete the report. Special thanks to the project supervisor – Mr. Yong Tien Fui who had given the golden opportunity to do the project on the topic "**Virtual Personal Bookshelf System**". The project supervisor had given a lot of encouragement and support to overcome problems and cope well when challenges are faced.

Finally, special thanks to Dr. Liew Soung Yue and Mr. Ooi Joo On who provided knowledge of problem solving skill, presentation skill and project planning, given guidance and idea of writing proposal.

ABSTRACT

Virtual Personal Bookshelf System (VPS) is a Document Management System developed to provide individual level of user document management. This project develops VPS to overcome user problems and fulfill user requirements in document management. The modules achieved in this project are document Categorization, Recommendation, Search and Sharing. The application utilizes document processing, calculation algorithms and cloud computing system storage to achieve its modules. The system was innovated by reviewing benchmarked system development and designed by integrating benchmarked algorithm. The system structure consists of Client side application and Server side application where both sides requires connection to system database and file system storage. Client side application functions to provide user connectivity with the system, user side document categorization, display processed recommendation from the server side application, user document searching and document sharing though cloud system. Server side application works as a system database management application, performs entire system document categorization and processing and process recommendation for every system users. The system is implemented and tested to gain user evaluation compared to the closest commonly used personal document management system, Windows Explorer. Lastly, the project was concluded based on the system testing result and stated for its future improvements.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

VPS	Virtual Personal Bookshelf System

Chapter 1: Introduction

1.1 Problem Statement

This project is innovated based on a few reviewed problem statements. The Problem Statements are listed below,

Problem 1: Required Flexible storage options in enterprise organization (Top Five Requirements for Secure Enterprise File Sync and Sharing, Citrix 2015)

Need to meet compliance requirements and to simplify management in enterprise organization or for personal usage. In current advancing society, sharing and file synchronization is required to ease user activities. This is because users require instant file storage and file transfer in interdependent society activities among users.

Problem 2: Need to index and organize large amount of documents of users (Why Document Management: a White Paper 2008)

Users, especially in enterprises possess huge amount of documents. These documents are required to be organized and managed to ease organization transactions. Besides that, organizations are able to save costs and reduce expenses to categorize documents.

Problem 3: Required to increase productivity and efficiency in daily activities with efficient document management. (Document Management Overview 2007)

Users are required to increase productivity and efficiency in daily activities. Therefore with efficient documentation management included recommendation functionalities and filtering index document management is required to improve user productivity.

1.2 Background and Motivation

Text Documents represents the basic structure of information and data storage. Since the beginning of human civilization, text document had been utilized for information sharing and storage. However over few decades, Physical text document had been digitized into virtually stored data information. With the help of computer systems, mankind had been transforming classical written text documents into digitized documents. According to a reviewed report, an average amount of 250 megabytes of data per person per year is produced and only 0.003 percent of the total amount is printed form. (From Measurement to Management Using Data 2004). Furthermore, these numbers are predicted to increase drastically (Cutting the clutter tackling information overload at the source 2009).

This phenomenon triggers a problem to manage these increasing amounts of digital documents. Therefore systems to manage digitized documents are required to solve these problems. This project proposes a solution by developing a system application with problem solving algorithms in handling the mentioned problems.

1.3 Objectives

This project objective proposes a solution by developing a document management system specifically in managing users personal documents and provide feasibility to users of the system. The objectives of the project are as follows:

Objective 1: To provide file system sharing between users and instant document storage

The development of document management system attempts to provide document allocation between users of the system while facilitating instant document storage beyond user`s personal local disk drive.

Objective 2: To facilitate document organization among amounts of document owned by users

The Categorization functionality of this developed application contributes basic document organization. The solution strives to help users save time and expenses to reorganize important document in their local machines.

Objective 3: To assist user in improving daily productivity by minimizing hectic processes of document managing

This project aims to handle this matter by providing documents to users using recommendation and document searching algorithms. Improving methods to bring document to users is one of main goal of this project.

1.4 Proposed Approach

The Proposed Solution illustrates a System Application named as Virtual Personal Book Shelf (VPS). The system consists of client side application and server side application as shown in Figure 1.1.

The overall system flow requires the two sides of application. On the Client side application determines documents held or provided by the user. If changes detected by the system, temporary categorization will occur. After that, the system will determine if there are new recommended documents or notifications from other users. On the Server side application, the administrator of the system detects changes on the database and performs Categorization and Recommendations upon detected changes. The brief descriptions of the system flow are as follows:

Client Side Application

1. Client Application Login
   This process determines the client user who starts the system and determines the document possessed by them.
2. Local Document Categorization
   If some new documents detected on a client user computer, categorization will occur based on the newly acquired documents. The system will temporary determine categories for the new documents and update database for the new changes
3. Determine Recommendations
   System determines recommendation to the user by retrieving database information
4. Determine Notifications
   System determines notifications from other users updated sometime before the current user starts the system. These notifications consists of shred documents or updated user network.
5. Document Search
   The system provides search functionalities to users to ease document retrieval on the local machine side.

6. Share Document

   Users are able to share documents among themselves using the system

Server Side Application

1. Whole Database Document Categorization

   Server side application determines changes and performs overall document categorization. This procedure collects the documents on the whole system and process to determine the actual categories of each documents.
2. Process Acquired Recommendation

   After collecting categories for each documents, the system will calculate and produce document allocation for every concerning users.
3. Manage Database

   Server side application provides database management to the administrator user. The admin is able to forcefully make changes to the database if required.
4. Update Stopwords, Stemwords and Synonyms

   On the server side application, admins of the system are able to update Stopwords, Stemwords and Synonyms changed on the world dictionary. This is to facilitate accuracy to the system on processing categorization and recommendation.

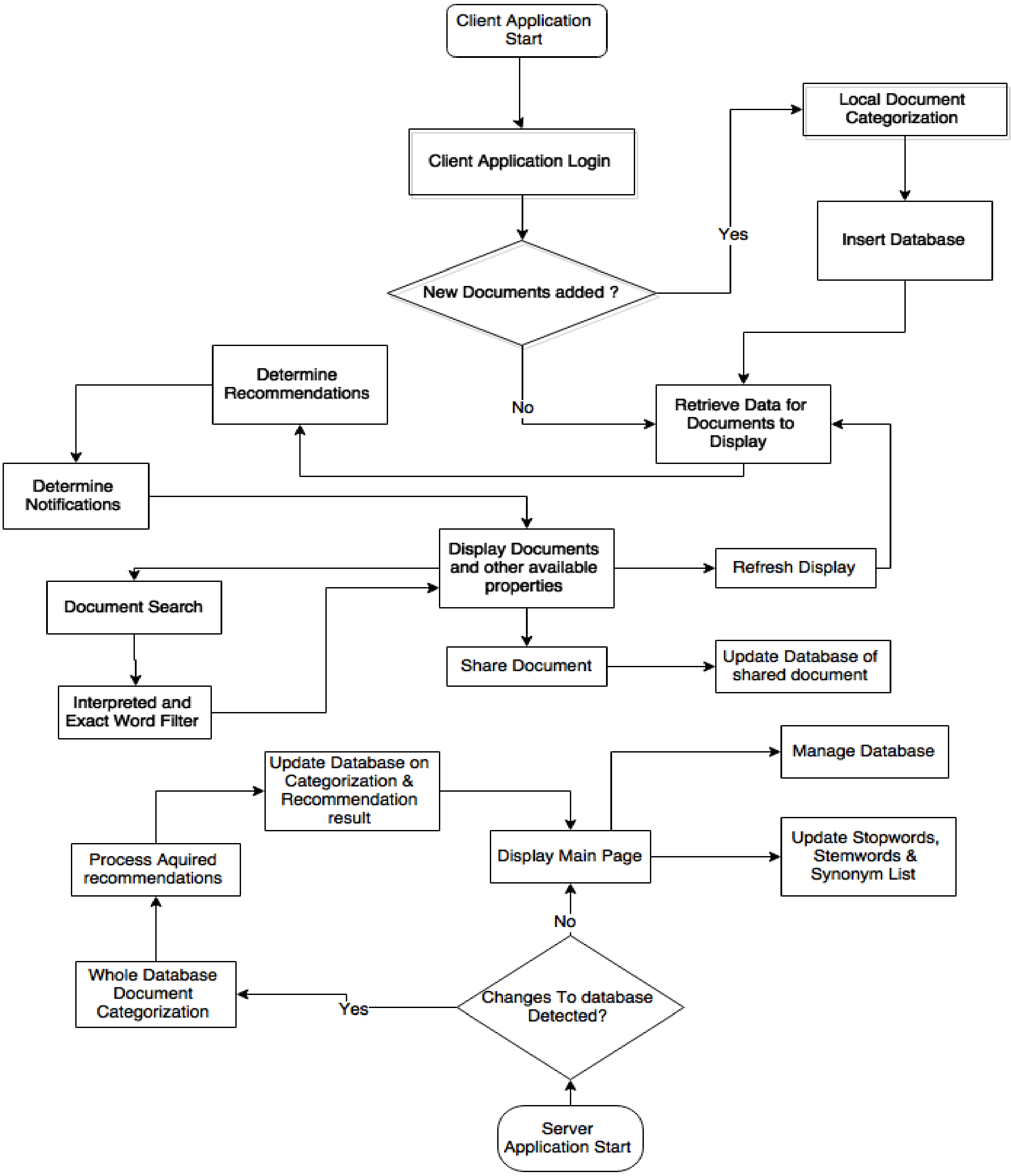

Client Application Start
Client Application Login
New Documents added ?
Yes
Local Document Categorization
Insert Database
No
Retrieve Data for Documents to Display
Determine Recommendations
Determine Notifications
Display Documents and other available properties
Refresh Display
Document Search
Interpreted and Exact Word Filter
Share Document
Update Database of shared document
Manage Database
Update Database on Categorization & Recommendation result
Display Main Page
Update Stopwords, Stemwords & Synonym List
Process Aquired recommendations
No
Whole Database Document Categorization
Yes
Changes To database Detected?
Server Application Start

Figure 1.1: VPS Flow Chart

1.5 Project Achievement

The achievement of this project consists of a few modules of the system. Each module utilizes studied algorithms and innovated solutions. The modules are as follows,

Module 1: Categorization

This module applies the algorithm of clustering. The steps and methods working with this module are as follows.

1. Eliminate stop words

2. Eliminate stem words

3. Form matrix for frequent words of each document

4. Compare similar frequent words between documents

5. Find the words with maximum frequency of pair of documents

6. Group documents into cluster

7. Identify topics and label them with topic cluster.

Module 2: Recommendation

Recommendation module applies of Collaborative Filtering algorithm. The following shows the brief procedure and formula for this algorithm.

1. Determine number of documents in each user`s document categories

2. Calculate similarities (sim(i,j)) between each user pairs(i and j). Formula below shows the calculation for similarities and average similarities between users

$$S=|\text{User A Doc}_i \,/\, \text{User B Doc}_i| * 100\%$$

User J Doc_i= Number of User J`s document in category i

S = Similarity percentage between users

Figure 1.2 Similarity percentage formula

$$\text{Avg}S = (\text{Total}S/\text{Total No. of Categories})$$

AvgS = average similarity

Figure 1.3 Average similarity formula

3. Obtain total list of pairs of similarities

4. Group users in cluster and perform recommendation using average % similarity to do recommendation. Numbers of documents to be recommended are as follows.

No of Doc to recommend = Round(*AvgS* * No. of Doc in a category)

Figure 1.4 Number of Documents to recommend formula

Module 3: Search

Search module of the system consists of 2 integrated algorithms namely, Indexing algorithm and Search algorithm. The system requires document indexing with the indexing algorithm before performing Search algorithm. The procedure summarized for the both algorithms are as follows,

Indexing Algorithm

Indexing algorithm is the composite of a few sub algorithms. The algorithm takes in the conditions to perform word indexing. The conditions are listed as follows.

1. Word count algorithm
   Calculate maximum number of terms in a document (top 3 most frequent words are used in this project)
2. Unique Word Algorithm
   Word which appears least documents among all documents. This algorithm does not applies only for a document each but considers a collection of document before performing the procedure.
3. Bold algorithms
   Take in considerations of bold words within documents. (Applies only for documents that possess bold words properties)
4. Italics algorithms
   Take in considerations of all italic words within documents. (Applies only for documents that possess italic words properties)
5. Heading or subheading algorithms
   Take in consideration of words of headings within documents. (Applies only for documents that possess heading and subheading properties)

Search Algorithm

Search algorithm applies 2 sub-algorithms in performing query search. The algorithms are described as follows:

1. Exact Query Search algorithm
   Search documents based on exact word of user query. This algorithm is the basic compare and match words for each document indexed before producing search result.
2. Interpreted Query Search algorithm
   Search on synonym of the word of user query. Interpreted query matched user query words with similar meaning of indexed words. Besides the exact word the user queries, the words that is in the synonym list will be selected.

Module 4: Sharing

Sharing module of this system utilizes cloud storage and file transfer features among users. Client users are able to share documents through internet connection within the system application. For this project, Dropbox api was used as server file storage and file transfer for client users.

1.6 Report Organization

This report is organized as follows, on the current chapter, **Chapter 1: Introduction** describes the problem statements, motivation of the project study, objectives and proposed approach. The rest of the report organizations are **Chapter 2: Literature Review**, **Chapter 3: System Design**, **Chapter 4: Methodology and tools**, **Chapter 5: Requirement**.

Chapter 2 reveals the benchmark studies and previous system reviews compare to the proposed system. The chapter also describes the algorithms used in to develop the software. On chapter 3, the system design is described. The chapter consists of diagrams and figures to demonstrate the detail explanation of the system structure and data flows.

Meanwhile, the rest of the software development methodology and tools are described at chapter 4 and chapter 5 states the requirement to deploy and execute the system application.

Chapter 2: Literature Review

The article review for this project consists of a few similar system application reviews and algorithm reviews. System review provides reference and comparisons to proposed system while algorithm reviews provide guidance in the system development.

2.1 System Review

Article 1: Generation and Maintenance of Semantic Metadata for Personal Multimedia

The article presents the K-IMM system, an joined approach for ontology-based personal multimedia document management. The system manages documents by obtaining document metadata. Document properties like date stored in the hard drive, source of the documents were obtained from, title for the document, type of document stored and etcetera are used in this system to form organization of documents. The system is efficient for managing documents in personal computers, details of user`s documents are effectively available however the system does not analyse document`s contents or hidden information. Therefore the system is unable to organize documents correctly if the document does not contain sufficient specified information.

Article 2: Semantic Based Personal Computer Resource Management System

The benchmarked model proposed in this literature review is the Ontobook system. The model utilises semantic approach to manage user`s personal document where user`s daily behaviour on the computer are being considered. The advantage of this system is that it actively tracks user`s action on the computer and able to perform query search based on user`s input. In other words, searching document with the system provides more accuracy with the system considers user preference. The disadvantage of this model however it is limited with management of user documents on a specific computer. No document sharing and the system do not cope well with new irregular added documents to the system.

Benchmarked System and Proposed System Comparisons

Features	K-IMM system	Ontobook system	VPS system (Proposed Solution)
Categorization	Document metadata management based	User behaviour on document management based	Document content and user preference categorization management
Recommendation	None	Based on user behaviour to generate user preference for document filtration	Collaborative filtering among users` behaviours on the system

Search	Document metadata indexing with simple search query	None	Indexing with various word properties and Search query with exact and interpreted word query
Sharing	None	None	Utilizes cloud file storage and file transfer system

Table 2.1 Benchmarked System and Proposed System comparisons

2.2 Algorithm Review

There are a few algorithms reviewed to provide inspiration and guidance for the development of this project. However, a number of the reviewed algorithm are selected to facilitate the system development. The article reviews are as follows

Article 1: A Frequent Term Based Text Clustering Approach Using Novel Similarity Measure

The benchmarked algorithm proposed clustering method in document categorization. The categorization process begin by eliminating stop words and stem words of documents, then form matrix for frequent words of each documents and compare similar frequent words between documents. The documents are then paired up with maximum words frequency and group into cluster, thus identified topics with topic cluster. The algorithm is analysed and proposed to implement in the project system`s document categorization module.

Article 2: Collaborative Filtering Recommendation Algorithm Based on Cluster

This article proposed collaborative filtering method in conducting recommendation. The process involves determining number of documents in each user`s document categories, calculate similarities between users of number of documents and obtain total list of pairs of similar users. Finally the users are grouped into clusters and perform recommendation. This algorithm is implemented to support the recommendation module of this project.

Article 3: Intelligent Search Engine Algorithms on Indexing and Searching of Text Documents using Text Representation

The document search module consists of algorithm proposed by this article. The algorithm consists of indexing and search query section. In indexing section, the algorithms are:

1. Word count algorithms
   Calculate maximum number of terms in a document in determining document index. In this article the maximum word count 15 obtained is the top 3 most frequent words in a document.

2. Unique word algorithms
   This algorithm considers words which appears in the least documents among all documents. The words are considered unique for document indexing.

3. Bold algorithms
   Bold words are taken for document indexing in this algorithms

4. Italic algorithms

   Italic words are taken for document indexing in this algorithm.

5. Heading or subheading algorithm

   All headings in a document are taken in documents indexing.

Indexing section algorithm integrates the above algorithms and when user perform search, search query section comes in. Search query section algorithm of this article consists of:

1. Exact query search algorithm

   This algorithm conducts search on exact word of user query

2. Interpreted query search algorithm

   Synonym of user query`s word is search to match documents index in this algorithm.

Chapter 3: System Design

VPS system is composed of client and server application interconnected with each other through database and cloud file storage system. To further elaborate the system design, the following diagrams and descriptions tends to furnish detail overview of the system.

3.1 Use Case Diagram

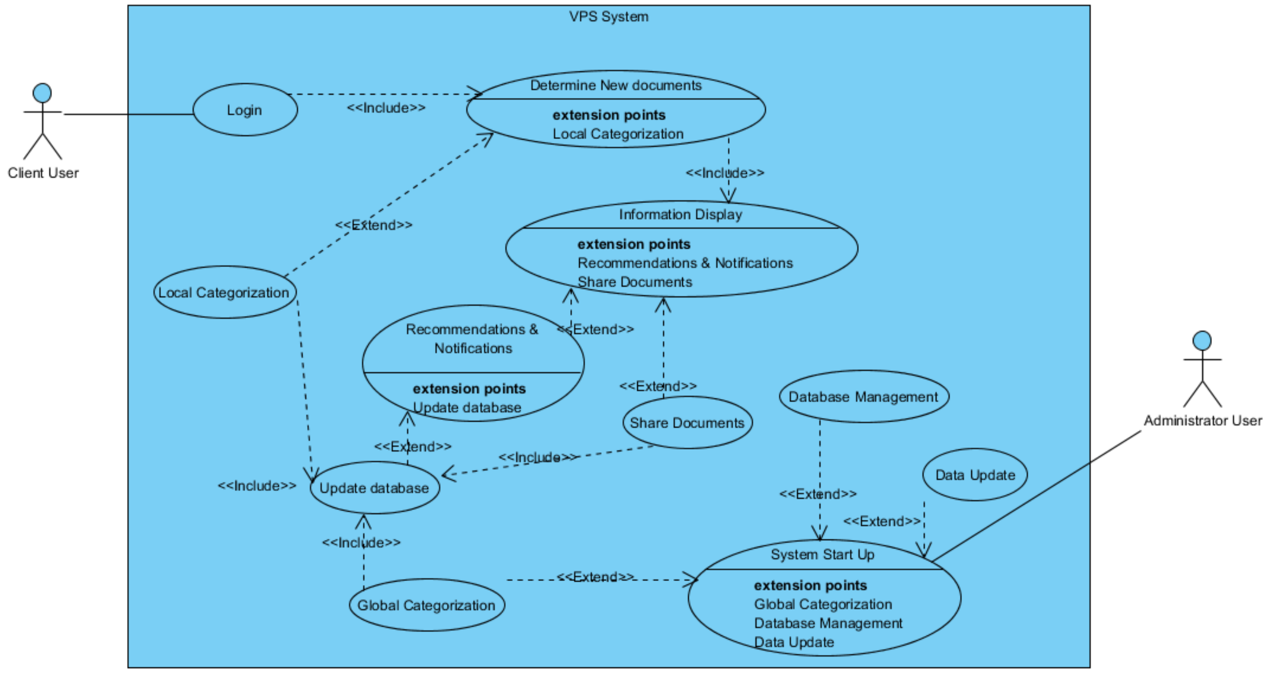


Figure 3.1: VPS Use Case Diagram

Figure 3.1 shows the use case diagram for VPS system. The system consists of 2 actors, the client user and admin user. Both actors of the system interact using various different use cases and join interactions with a single use case involving system database. Client user and admin use cases are further describe as follows,

Client Use Cases

1. Login (include)

   Client enters credentials for the system to determine individual users. This use case determines the data to retrieve and display to current interacting user.

2. Determine New Documents (include)

   In this uses case, the system determine whether the current user had supplied new documents. Other than that, the system also determines the documents the user currently possesses and prepares data values for display.

3. Local Categorization (extend)

   New documents detected will be processed for indexing and categorization. The categorized and indexed documents is recorded in the system database for further usage.

4. Information Display (include)

   This use case brings users to the main page with the provided features and information displayed to them. This use case gathers all relevant information and provides ease visibility to the user.

5. Recommendations & Notifications (extend)

   Before displaying information to users, this use case determines if there is new recommendation or notifications for the user. Users are able to accept the notifications or download recommended or shared documents to them.

6. Share Documents (extend)

   If required, users are able to share documents in this use case. The system updates database upon determining the targeted user and documents.

7. Update Database (include)

   This use case enables update, insert and delete on the system database. Whether the data produced from processes of categorizations, recommendations or sharing goes through this use case. This is also where the server side application and client side application meets on data transfer.

<u>Admin Uses Cases</u>

1. System Start up (include)

   Admin user start up this use case which allows the system to detect changes made on the system database. If changes detected, the user is prompt to perform categorizations.

2. Global Categorization (extend)

   This use case is invoked when there are changes on the system database. The system downloads all the documents managed and process the documents with indexing and categorization processes. After this use case, the system updates the database and notify client users on the changes.

3. Database Management (extend)

   Admin users are able to manage database using the system. Any database modification is permitted to authorized admin users. This is to facilitate management on client user`s data if required.

4. Data Update (extend)

   This use case enables admin users to update required fixed data in the system. The data required are Stop word list, Stem word list and synonym list. These data are the prerequisites for the system algorithms.

3.2 Data flow Diagram

3.2.1 Context Diagram

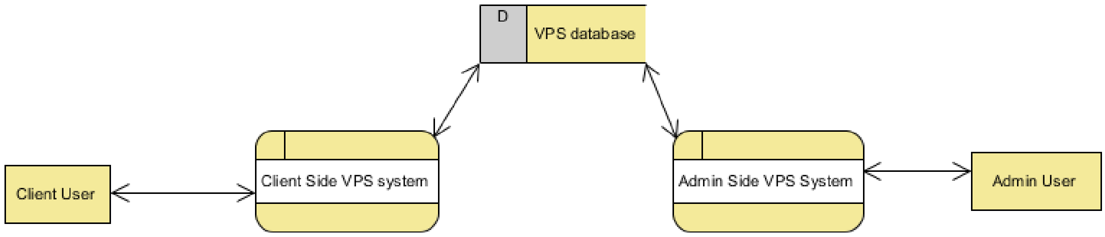


Figure 3.2 VPS System Context Diagram

The context diagram shows the brief data transfer and basic structure of VPS system. As shown in the diagram, the system consists of client side application and admin side application. Both system processes interacts join and interact with each other at the center system database.

3.2.2 Client Side Data Flow Diagram

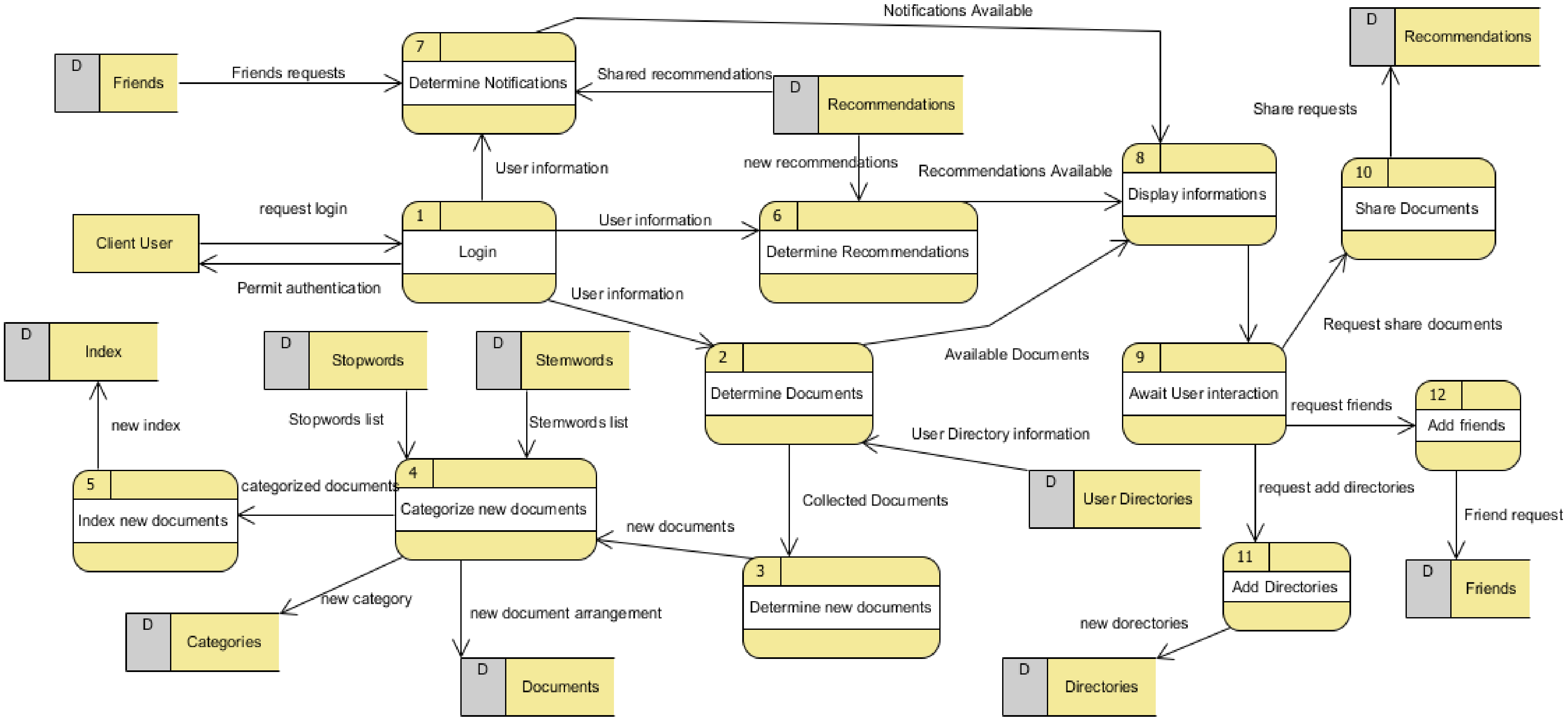


Figure 3.3 Client Side DFD

Client Side Data flow diagram shown at Figure 3.3 illustrates the data flow throughout the system structure of VPS. At **process 1: Login**, the system performs Login for client users. Client users provide login credentials and request for system login. The system then permit authentications if user credentials is valid. After the login process, there are 3 processes each performs different data transfers. **Process 2: Determine Documents** determines the documents owned by client users.

Process 2 retrieves defined user directories from recorded data in the database and performs document extracting on each obtained directories. At this point, the system will determine new documents included by the user at **Process 3: Determine New Documents**. These new documents is then processed and categorized (**Process 4: Categorize New Documents**). This process take in predefined data of stop words and stem words from database and produce arranged documents and categories. Other than that, the system also performs **Process 5: index new documents** after obtaining the processed documents. This process produce word index to the database for further usage.

After branching from Process 3, the system then displays the required documents on **Process 8: Display information**. This process is carried out even the system does not detect new documents. On the other hand, before the system displays the overall information to the client user, **Process 6: Determine Recommendations** is performed which obtains system generated recommendations to the current user and display to the user. Besides recommendations, the system process notifications at **Process 7: Determine Notifications**. This process is made up of processing friend request notifications and shared document recommendations. Upon available notification, client user shall be alert.

After reaching the main page of the system application, the system idles for user operation, **Process 9: Await user interaction**. If the user requires to share documents, **Process 10: Share Documents** handles user share request by populating the database before alerting the targeted users. **Process 11: Add Directories**, enables users to add new directories while **Process 12: Add Friends** let users to add new friends. The two processes achieve their procedures by retrieving and updating system database.

3.2.3 Server Side Data Flow Diagram

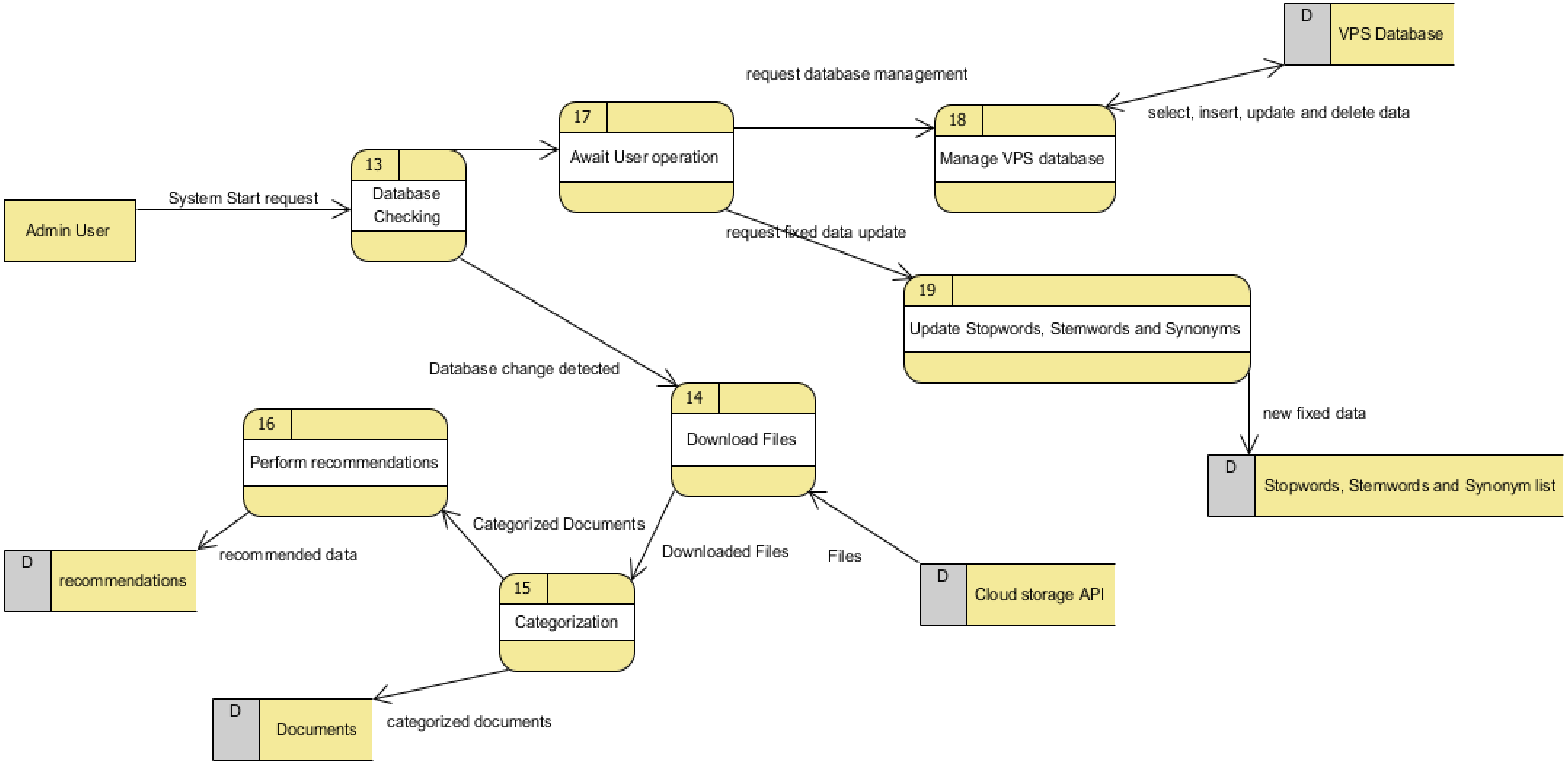


Figure 3.4 Server Side DFD

Figure 3.4 portrays the server side data flow diagram. On the beginning of system start up, the system check for changes on the database. (**Process 13: Database Checking**). If detected changes in database, the system download required documents (**Process 14: Download Files**).

For the purpose of this project study, simple cloud storage API is used to provide file storage. On the downloading document process, documents are downloaded into server machine from the cloud storage api. Follow up with next process, **Process 15: Categorization**, categorizes the downloaded database and produces organized documents. The algorithms used in this process are similar to client side algorithm.

The next process that the system performs is the recommendation process. (**Process 16: Perform Recommendation**). This process resembles client side application performing calculations for user recommendations by updating database and invoking the client side for recommended documents.

Besides VPS automated processes, the system also equipped with database configurations. **Process 18: Manage VPS database**, let admin users modify and manage system database, while **Process 19: Update Stopwords, Stemwords, and synonyms** facilitate admin to add required fix data for the system.

3.3 Entity Relationship Diagram

VPS system requires database structure for data transactions in and out of the system. The major entities involves in this system are:

1. CATEGORY
2. DOCUMENT
3. USER
4. User_DOCUMENT
5. DIRECTORIES
6. FRIEND
7. RECOMMENDATION
8. ChangeLog
9. STOPWORDS
10. STEMWORDS
11. SYNONYMS
12. SYNONYM_PAIR
13. WINDEX
14. DOCUMENT_WINDEX
15. GROUPS
16. USER_GROUPS

Figure 3.5 Shows the Entity Relationship Diagram for the above mentioned entities.

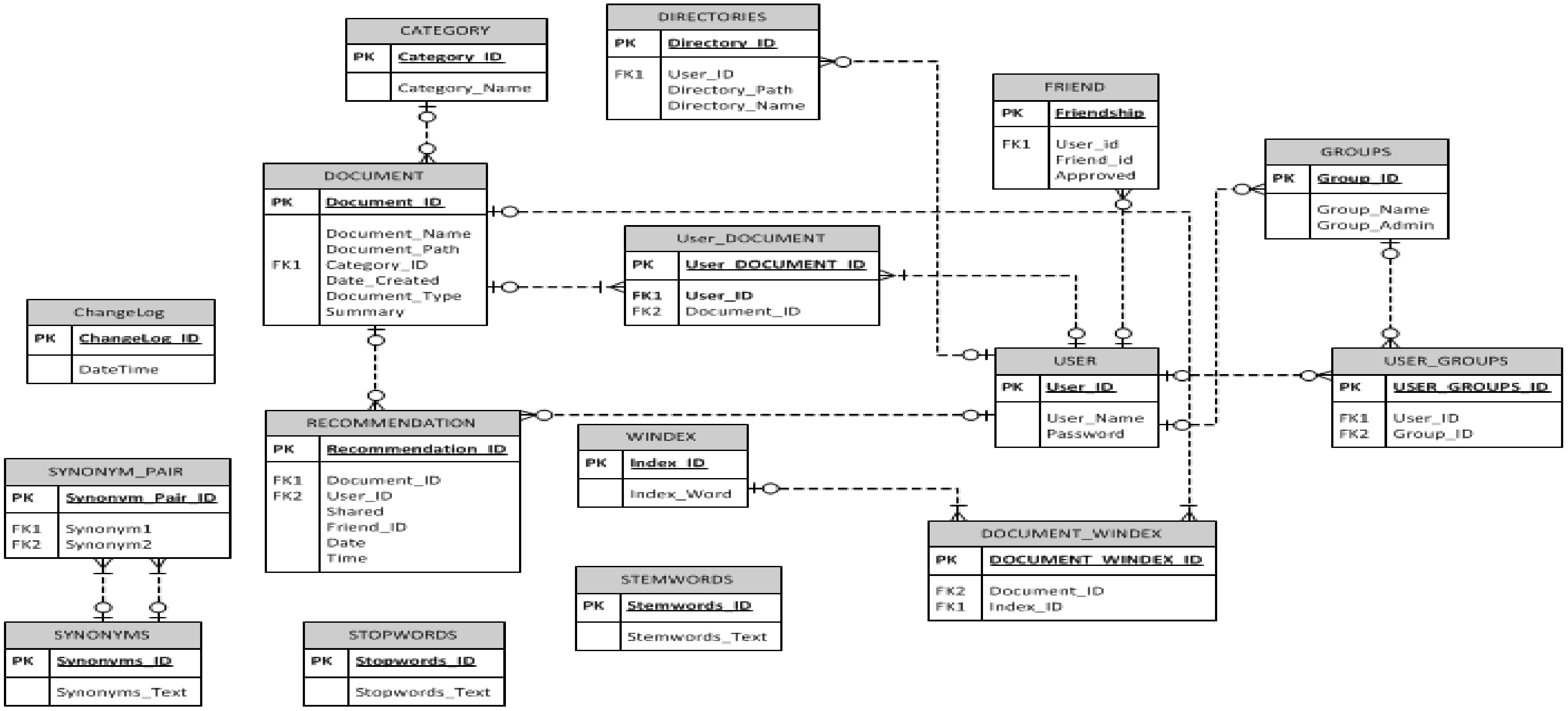


Figure 3.5 VPS Database ERD

As shown in Figure 3.5, the entity table portrays the data elements required for the system. The description for the functionalities of each entities are,

1. CATEGORY
   This entity is used to record the document categories generated by the system. Any new generated document category is recorded in the table and reference with its primary key

2. DOCUMENT
   Document table records all the required document properties by the system. Each row of the table represents a document identified by the system.

3. USER
   This table records all the users interacting with the system with each of the user details required by the system. The properties in this table are usually required by the login process to identify each unique user.

4. User_DOCUMENT
   User_Document keeps track of every user and document relationships. A user may own many documents whereas a document may belong to many users. Therefor this table is to keep track every ownership of the users and documents.

5. DIRECTORIES
   This is the directory entity need by the system to identify all the directories defined by the users of the system. The directories enable the system to look

for user`s documents in their local machines to be extract and process by the system.

6. FRIEND

   This table records the friendship established by the users of the system. Whenever a user is connected to another user through the system, data is recorded for the system to perform further operations like sharing of documents.

7. RECOMMENDATION

   This entity tracks every sharing of documents between users and recommendations generated by the system. Every time a user logs into the system, VPS will retrieve data from this table to look for required notification for the user. On the other hand when the user requests for a recommendations by the system, this is the table that the system look for information.

8. ChangeLog

   ChangeLog is the table that records required changes to the documents, users or ownership between users and documents in the system. This is to indicate whether or not the system is required to do categorization. This table data is mainly required by the server side application of the system.
9. STOPWORDS

   As mentioned in the entity title, this table records the stop word data required by the system. The data needed by the system to filter stop words when reading documents for categorization.

10. STEMWORDS

    Stem words table records the stem words supplied by admin users for the system document reading process for further categorization process. This table has the usage that resembles the STOPWORDS table.

11. SYNONYMS

    This table tracks the words in the dictionary that has potentials for similar meaning as another. Synonyms are required for indexing process of the Search module of the system.

12. SYNONYM_PAIR

    This entity records every pair of synonyms identified by the system. The data links every record in the synonym table to form relationship for similar meaning referred on the dictionary. The data of this table is usually facilitated by the admin users as well.

13. WINDEX

    This table is the conceptual definition of word index that is recorded to index every document in the system. This is to facilitate the Search functionality of the system.

14. DOCUMENT_WINDEX

    Associate with the table WINDEX, this table links the documents and index produced upon every indexing process. Every documents that is indexed with a word in the indexing process are recorded in this table.

15. GROUPS

GROUPS is the table of records of groups defined by users. The purpose of this table record is to ease users to share documents. Instead of sharing documents to many users one by one, users able to share documents to a number of users associated to the groups.

16. USER_GROUPS

The existence of this table in the database is to provide relationships of users to the groups defined. Every record indicates a user belong in the group.

3.4 Data Definitions

Data types of the table entities mentioned in the previous section is crucial for the system database to perform is purpose. The data definitions of each of the tables are as shows.

Table 1: CATEGORY						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Category_ID	Category ID	INT		Yes		
Category_Name	Category Name	NVARCHAR	50			

Table 3.1 CATEGORY Table Definition

Table 2: DOCUMENT						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Document_ID	Document ID	INT		Yes		
Document_Name	Document Name	NVARCHAR	MAX			
Document_Path	Document Path	NVARCHAR	MAX			
Category_ID	Category ID	INT			Yes	CATEGORY
Date_Created	Document date created/ modified	Date				
Document_Type	Document file type	NVARCHAR	MAX			
Summary	Short preview of document content	NVARCHAR	MAX			

Table 3.2: DOCUMENT Table Definition

Table 3: USER						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
User_ID	User ID	INT		Yes		
User_Name	User Name	NVARCHAR	MAX			
Password	User Password	NVARCHAR	MAX			

Table 3.3: USER Table Definition

Table 4: User_DOCUMENT						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
User_DOCUMENT_ID	Identification of document and user relationship	INT		Yes		
User_ID	User ID	INT			Yes	USER
Document_ID	Document ID	INT			Yes	DOCUMENT

Table 3.4: User_DOCUMENT Table Definition

Table 5: DIRECTORIES						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Directory_ID	Directory ID	INT		Yes		
User_ID	User ID	INT			Yes	USER
Directory_Path	Directory Path	NVARCHAR	MAX			
Directory_Name	Directory Name	NVARCHAR	MAX			

Table 3.5: DIRECTORIES Table Definition

Table 6: FRIEND						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Friendship	Friendship Identification	INT		Yes		

User_id	User ID	INT			Yes	USER
Friend_id	Friend with User ID	INT				
Approved	Approved status	BIT				

Table 3.6: FRIEND Table Definition

Table 7: RECOMMENDATION						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Recommendation_ID	Recommendation ID	INT		Yes		
Document_ID	Document ID	INT			Yes	DOCUMENT
User_ID	User ID	INT			Yes	User
Shared	Shared Status	BIT				
Friend_ID	Target User ID	INT				
Date	Date recommended	NVARCAR	50			
Time	Time recommended	NVARCHAR	50			

Table 3.7: RECOMMENDATION Table Definition

Table 8: ChangeLog						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
ChangeLog_ID	Change Log ID	INT		Yes		
DateTime	Date and Time	DATETIME				

	Recorded					

Table 3.8: ChangeLog Table Definition

Table 9: STOPWORDS						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Stopwords_ID	Stop words ID	INT		Yes		
Stopwords_Text	Stop words text	NVARCHAR	50			

Table 3.9: STOPWORDS Table Definition

Table 10: STEMWORDS						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Stemwords_ID	Stem words ID	INT		Yes		
Stemwords_Text	Stem words Text	NVARCHAR	50			

Table 3.10: STEMWORDS Table Definition

Table 11: SYNONYMS						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Synonyms_ID	Synonym ID	INT		Yes		
Synonyms_Text	Synonym Text	NVARCHAR	50			

Table 3.11: SYNONYMS Table Definition

Table 12: SYNONYM_PAIR						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Synonym_Pair_ID	Synonym pair ID	INT		Yes		
Synonym1	1st Synonym to be paired	INT			Yes	SYNONYMS
Synonym2	2nd Synonym to be paired	INT			Yes	SYNONYMS

Table 3.12: SYNONYM_PAIR Table Definition

Table 13: WINDEX						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Index_ID	Word Index ID	INT		Yes		
Index_Word	Word Index Text	NVARCHAR	50			

Table 3.13: WINDEX Table Definition

Table 14: DOCUMENT_WINDEX						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
DOCUMENT_WINDEX_ID	Document word index ID	INT		Yes		

Document_ID	Document ID	INT			Yes	DOCUMENT
Index_ID	Word Index ID	INT			Yes	WINDEX

Table 3.14: DOCUMENT_WINDEX Table Definition

Table 15: GROUPS						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
Group_ID	Group ID	INT		Yes		
Group_Name	Group Name	NVARCHAR	MAX			
Group_Admin	Group Admin User	INT			Yes	USER

Table 3.15: GROUPS Table Definition

Table 16: USER_GROUPS						
Column Name	**Description**	**Data Types**	**Size**	**Primary Key**	**Foreign Key**	**FK reference table**
USER_GROUPS_ID	User Group ID	INT		Yes		
Group_ID	Group ID	INT			Yes	GROUPS
User_ID	User ID	INT			Yes	USER

Table 3.16: USER_GROUPS Table Definition

Chapter 4: Methodology and Tools

4.1 Methodology

Software development methodology implemented in this project development is the Evolutionary Methodology. Evolutionary methodology procedure involves improving software development stages by iterative and incremental approach to software development. The meaning of iterative and incremental development refers to the combinations of both iterative method and incremental build model for software development. The model is made up of essential process of modified traditional waterfall model.

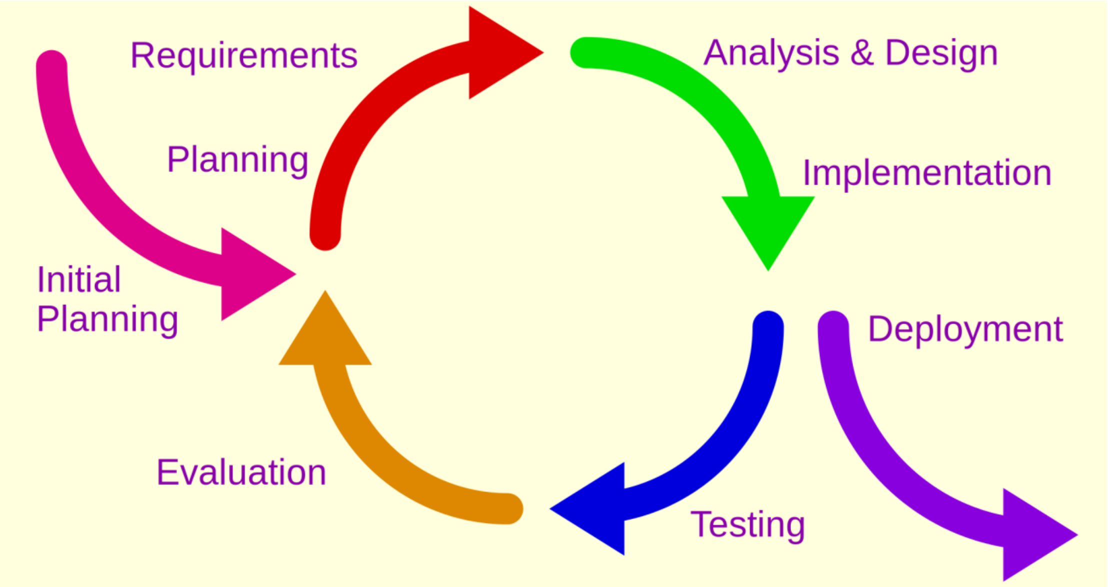


Figure 4.1 Iterative Development Model

Figure 4.1 is the iterative development model where the model portrays repetitive process of planning, requirement gathering, analysis and design, implementation, deployment, testing and evaluation. Each of the software development conducted within the same phase and iterates until software deployment.

Evolutionary methodology implies that other than forming a comprehensive artefact like requirements reviewed and accepts before creating comprehensive design model, the methodology allows evolutionary critical development over period in a repetitive manner.

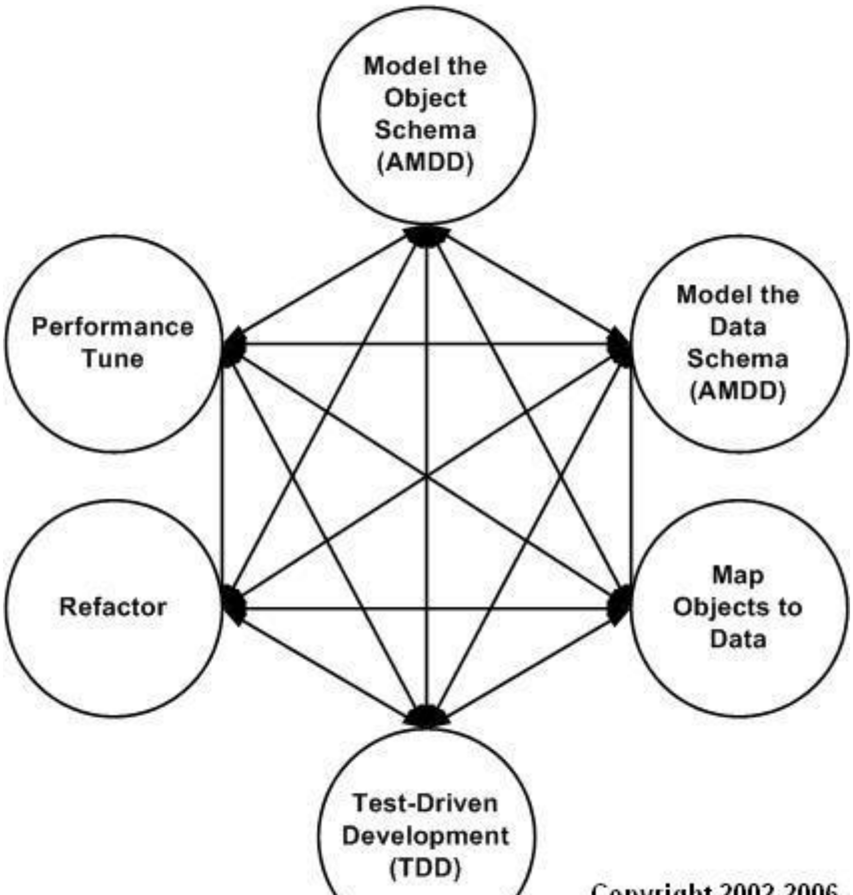

Model the Object Schema (AMDD)
Model the Data Schema (AMDD)
Map Objects to Data
Test-Driven Development (TDD)
Refactor
Performance Tune
Copyright 2002-2006 Scott W. Ambler

Figure 4.2 Evolutionary Model on Agile Project

Figure 4.2 displays the evolutionary methodology model. The phases of the software development methodology are as follows:

1. Object and Data Schema Modelling
   Usually involves class and data normalization techniques. The object and data modelling layouts the classes and data needed for the structure of the software.

2. Map objects to data
   Methods to solve impedance mismatch between objects and data is conducted at this stage. In evolutionary methodology, mapping is evolved over time and difficulties in mapping may motivate changes on objects and data.

3. Test-Driven Development
   This is the stage where software programs are tested and modified iteratively. This process continues until the program meets requirements.

4. Refactoring
   The stage where small improvements to the system in design without modifying the major module functionalities of the system are conducted. This process helps in evolve software design to achieve project objectives.

5. Performance tuning
   This is the phase where programmers fine tune the program to meet required performance. Other than that, this is the stage that the software program is adjust to a level compatible with the modern technologies.

Using the mentioned software methodology process, this project follows the iterative process as defined. Therefore the project realizations compatible with the development process are as follows.

1. Object and Data Modelling
   Classes of objects involving the project are designed on this stage. Object entities involved in this project are Documents and Users. Other than that, other classes required for program staging are also defined in this stage.

2. Maps objects to Data
   This is the software development process where layouts of the relationships and data are designed. For example for the user class, a user possess a number of documents and the document belongs in respective document categories.

3. Test Driven Development
   Programs are written in this phase and tested repetitively until the system meets analysis requirements with minimum error occurrence.

4. Refactoring
   Refactoring phase for this project development refers to the improvements of small functionalities to the software application. For example, a document contents preview functionality was added to provide users with ease of use of the system.

5. Performance Tuning

   This stage of the software development involves improving software performance. One of the improvements done is that adding multithreading to document processing algorithm to speed up document reading process.

4.2 Tools

Tools used in this project are mostly software development tools and data management tools. The following briefly describe the major tools used in each part the software development process.

1. Software program

   The software program written in this project is written in **c# programming** language. The program tools include packages provided by **Visual Studio 2013** compatible for c# programming language. For example File IO system package, Data Adapter packages and so on.

2. Database Structure language

   The database structure constructed for this software application are using MySQL and Transaction SQL database languages. For this project, online database are deployed. The database provider subscribed for this project is from the website www.db4free.net. The website uses MySQL language for database management and for Visual Studio to be able to connect MySQL database requires program reference package that is **MySql.data** package.

3. Cloud File Storage and Transfer

   The software application file storage and file transfer used is the **Dropbox Api** provided by Dropbox cloud storage. To connect to the dropbox api, Visual Studio includes package reference of **Nemiro.OAuth**.

4. Text File Management

   This project software application involves text document process. The reference package and file document types implemented in this software application are as follows.

File Type	Program Reference Package
Text Document File (.txt)	System.IO
Microsoft Words Document (.doc or .docx)	Microsoft.Office.Interop.Word
Adobe Acrobat Reader (.pdf)	itextsharp

Table 4.1: VPS Handling File Fypes and Reference Packages

Chapter 5: Requirement

The software developed for this project includes requirements for the software to execute accordingly. The types of requirements are as follows.

1. Hardware Requirements
   For the hardware requirement, this software is developed under desktop application platform. Therefore it is required for the software to run on **desktop computers** or laptops. Other than that, any devices that compatibles with the application execution is required to run this program.

2. System Requirements
   System requirements for this software application specifically defined as **Microsoft Windows** operating system. The Windows OS version required are **windows 7** and above, installed with **.Net Framework of version 4** and above.

3. Other Requirements
   Other crucial requirements of this application are **internet connection**. The software is required to connect to the world wide web in order to connect to the database provider and cloud storage api.

Chapter 6: Implementation and Testing

To implement and test the proposed system, a session of system application test and survey question and answering has been conducted. A number of 20 users with various age and gender were gathered to use the system and answer some simple questions on the system.

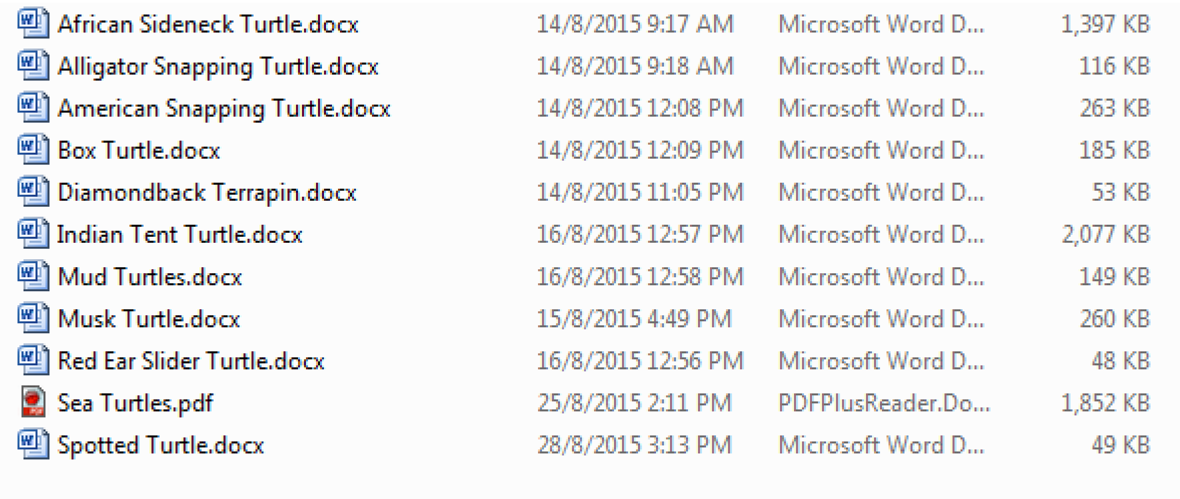
The survey question was constructed by two major parts, Scenario Testing and User Acceptance Testing. The scenario testing was formed to test the system`s categorization module or document display clarity and search module by testing the speed and accuracy to perform searching of documents compared with commonly used personal document management system, Windows Explorer. On the second part, User Acceptance testing requires users to answer questions based on their perception on the software application. The main modules tested in this part are the Recommendation and Sharing module.

6.1 Testing Procedure and Description

The following describes the testing and evaluation process carried out.

I. Scenario Testing

To test the document display clarity (Categorization module for VPS) of both VPS and Windows Explorer, a set of documents was provided to users. The Documents are made up of 2 categories, a general category of Turtles and another specific family of turtles called “Macrochelys”. This is to create an illusion to testing users of the given document categories as every document provided in the set contains the document name of “turtle”. Figure 6.1 shows the list of documents given to users for testing.

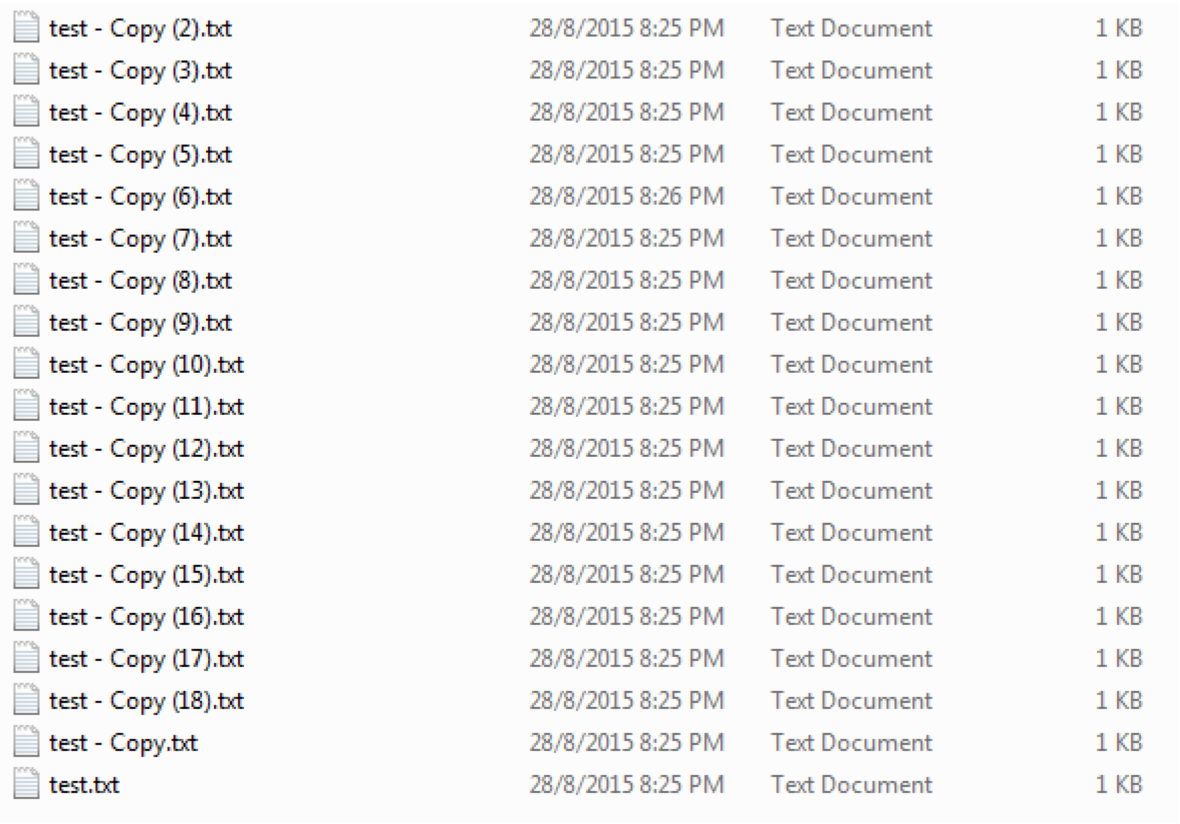


African Sideneck Turtle.docx	14/8/2015 9:17 AM	Microsoft Word D...	1,397 KB
Alligator Snapping Turtle.docx	14/8/2015 9:18 AM	Microsoft Word D...	116 KB
American Snapping Turtle.docx	14/8/2015 12:08 PM	Microsoft Word D...	263 KB
Box Turtle.docx	14/8/2015 12:09 PM	Microsoft Word D...	185 KB
Diamondback Terrapin.docx	14/8/2015 11:05 PM	Microsoft Word D...	53 KB
Indian Tent Turtle.docx	16/8/2015 12:57 PM	Microsoft Word D...	2,077 KB
Mud Turtles.docx	16/8/2015 12:58 PM	Microsoft Word D...	149 KB
Musk Turtle.docx	15/8/2015 4:49 PM	Microsoft Word D...	260 KB
Red Ear Slider Turtle.docx	16/8/2015 12:56 PM	Microsoft Word D...	48 KB
Sea Turtles.pdf	25/8/2015 2:11 PM	PDFPlusReader.Do...	1,852 KB
Spotted Turtle.docx	28/8/2015 3:13 PM	Microsoft Word D...	49 KB

Figure 6.1: Scenario Test Documents (Categorization)

The users are required to view the documents in Windows Explorer and VPS and provide answers according to the categories they found.

The other test, which tests the search capability of VPS was again compared with Windows Explorer. This time the users are given another set of documents which consists of undefined documents and categories. The documents are shown at Figure 6.2.



test - Copy (2).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (3).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (4).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (5).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (6).txt	28/8/2015 8:26 PM	Text Document	1 KB
test - Copy (7).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (8).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (9).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (10).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (11).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (12).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (13).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (14).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (15).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (16).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (17).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy (18).txt	28/8/2015 8:25 PM	Text Document	1 KB
test - Copy.txt	28/8/2015 8:25 PM	Text Document	1 KB
test.txt	28/8/2015 8:25 PM	Text Document	1 KB

Figure 6.2: Scenario Test Documents (Search)

The text documents given contain the word “double” in the content text. However there is only one document contains the word “single” in the content. Users are required to use the search bar provided in both Windows Explorer and VPS to find and locate the document specified and provide feedback of how they feel on the speed of both systems search functionality. This is to test the system`s content based search functionality where whether or not VPS`s search functionality works better than of Windows Explorer`s.

II. User Acceptance Testing

In User Acceptance Testing, users are introduced with some hidden functionalities available in VPS. Users are encouraged to try the Recommendation page where the system generates recommended documents automatically and the sharing function where users are able to drag and drop documents to be shared to friends they had included in the system. After users had satisfied with the try an error session with the system, the users are required to rate the accuracy and efficiency of the Recommendation and Sharing functionality of VPS.

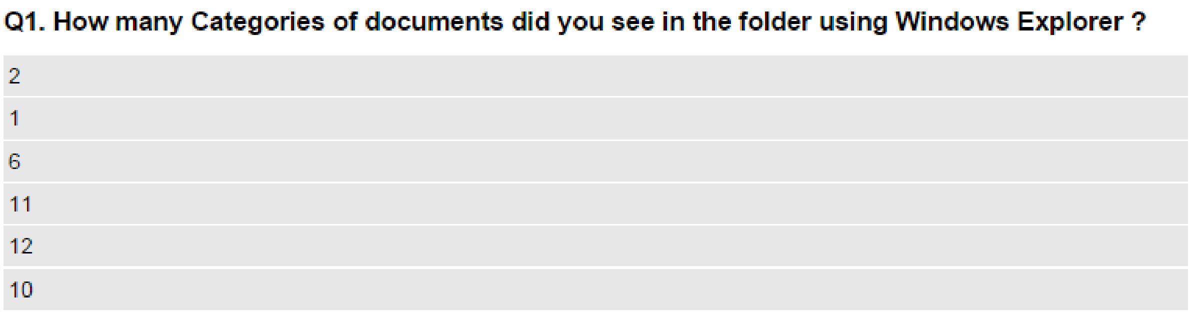
At the end of the question section, the users are also required to rate and elaborate on whether or not the system is easy to use, the attractiveness of the system`s graphical user interface and the most favourable functionality or controls of VPS. Finally, the users are asked a question of whether they accept the system as a market software application.

6.2 Data Collection and Survey Results

After the testing session, a number of results in data form had been collected. The results of every part of the test are as shows.

I. Scenario Testing

The first part of this system test, tests the **Categorization module**. The question and answers provided by users are shown at Figure 6.3 and Figure 6.4.



Q1. How many Categories of documents did you see in the folder using Windows Explorer ?

2

1

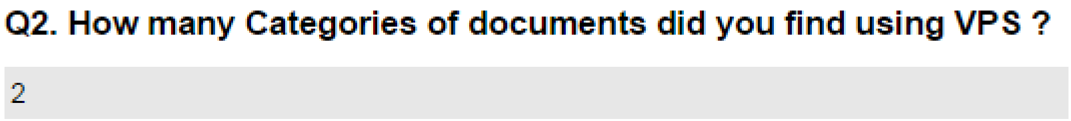
6

11

12

10

Figure 6.3: Question and Answers by users (Windows Explorer)



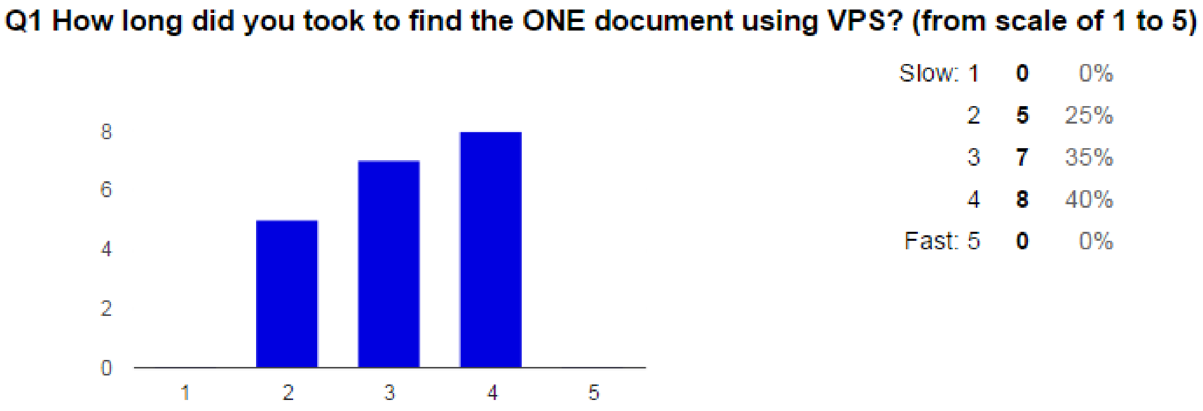
Q2. How many Categories of documents did you find using VPS ?

2

Figure 6.4: Question and Answers by users (VPS)

In Figure 6.3, answers obtained from users varied in number of document categories. Whereas the results obtain in Q2 clearly stated one single number by all users which matched the correct answer. Based on this result obtained, a conclusion can be deduced is that users are unable to identify how many categories of document handled by Windows Explorer as Windows Explorer does not show categories of documents without user indications. On the other hand, Users are able to find categories in VPS which are provided by the displayed application GUI.

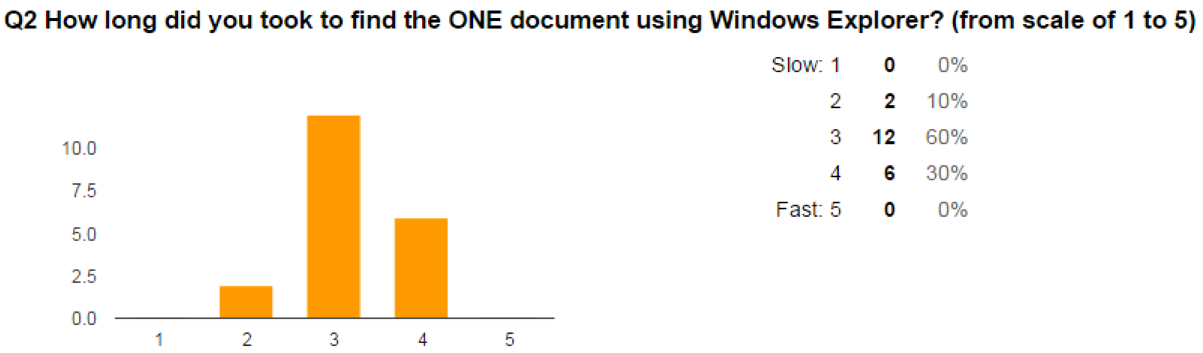
On the search testing, the results obtained are interpreted in bar graph as shows.



Q1 How long did you took to find the ONE document using VPS? (from scale of 1 to 5)

Slow: 1	**0**	0%
2	**5**	25%
3	**7**	35%
4	**8**	40%
Fast: 5	**0**	0%

Figure 6.5: Search Evaluation Result (VPS)



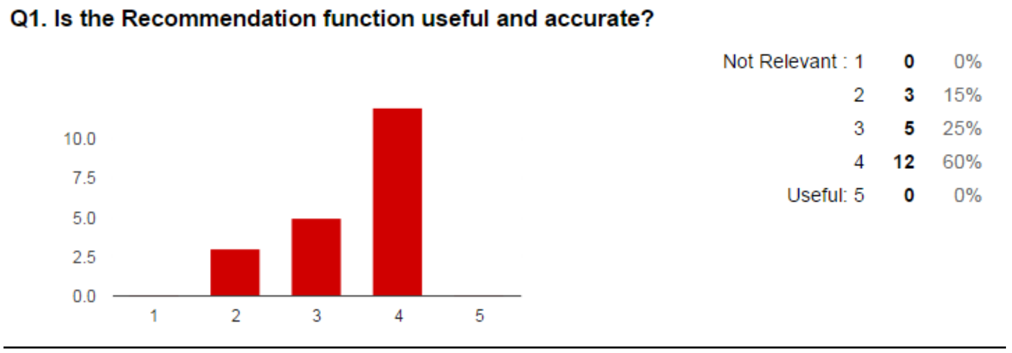
Q2 How long did you took to find the ONE document using Windows Explorer? (from scale of 1 to 5)

Slow: 1	**0**	0%
2	**2**	10%
3	**12**	60%
4	**6**	30%
Fast: 5	**0**	0%

Figure 6.6: Search Evaluation Result (Windows Explorer)

Results obtained in Figure 6.5 shows that there are more users who rate VPS search functionality slow than of Windows Explorer`s. However the maximum rate at VPS is at the rate fast while Windows Explorer is at the neutral between fast and slow. Therefore, it can be concluded that the speed of VPS search functionality are moderately equal with the speed of Windows Explorer. This is because VPS requires internet connection to perform the function while Windows Explorer requires random access memory (RAM).

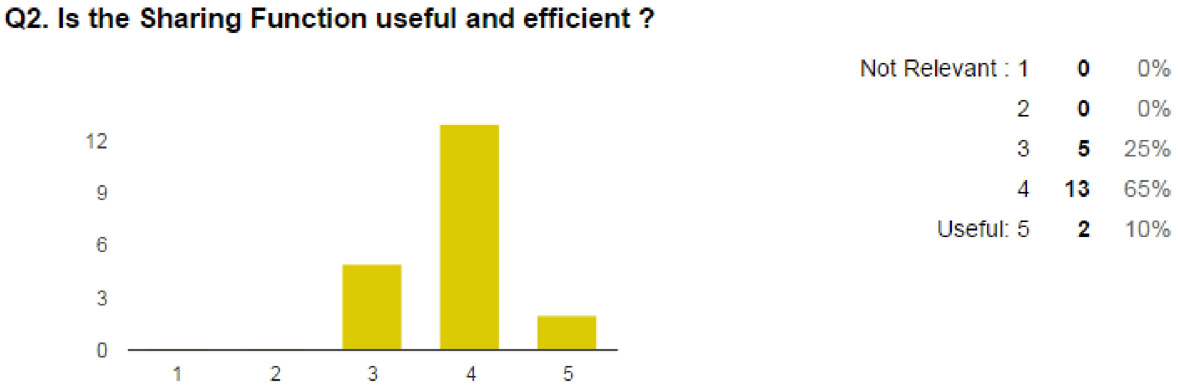
II. User Acceptance Testing



Q1. Is the Recommendation function useful and accurate?

Not Relevant : 1	**0**	0%
2	**3**	15%
3	**5**	25%
4	**12**	60%
Useful: 5	**0**	0%

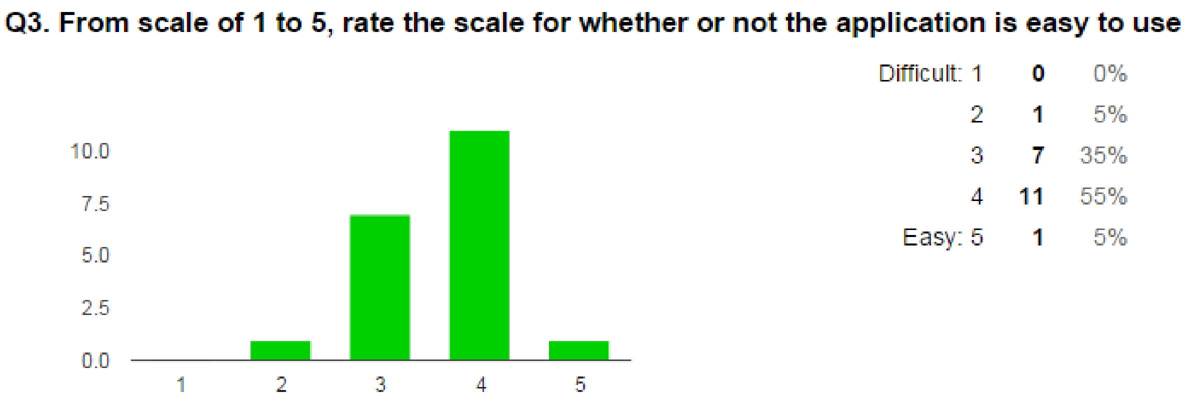
Figure 6.7: Recommendation Evaluation Result



Q2. Is the Sharing Function useful and efficient ?

Not Relevant : 1	**0**	0%
2	**0**	0%
3	**5**	25%
4	**13**	65%
Useful: 5	**2**	10%

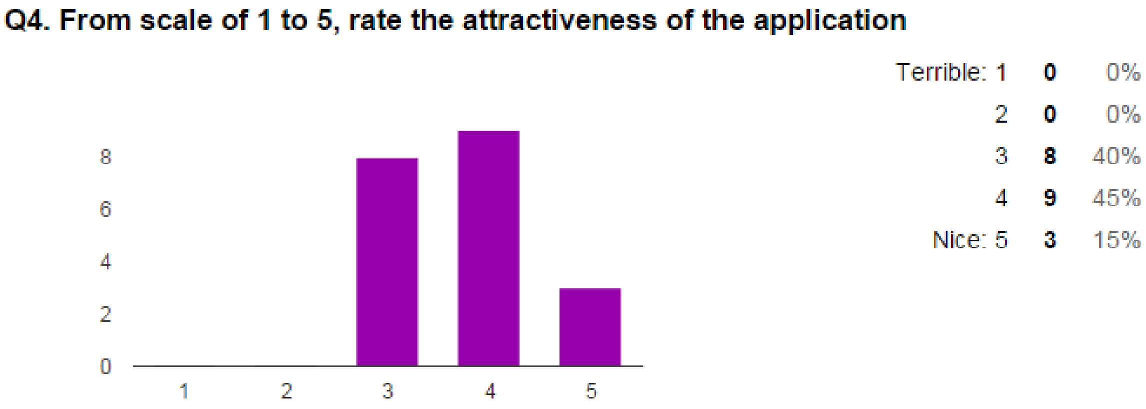
Figure 6.8: Sharing Evaluation Result



Q3. From scale of 1 to 5, rate the scale for whether or not the application is easy to use

Difficult: 1	**0**	0%
2	**1**	5%
3	**7**	35%
4	**11**	55%
Easy: 5	**1**	5%

Figure 6.9: Easy to Use Evaluation Result



Q4. From scale of 1 to 5, rate the attractiveness of the application

Terrible: 1	**0**	0%
2	**0**	0%
3	**8**	40%
4	**9**	45%
Nice: 5	**3**	15%

Figure 6.10: Application GUI Attractiveness Rating Result

Based on Figure 6.7, most users rate VPS`s recommendation function as usefulness of rate 4 which is 60% of the whole test users. At Figure 6.8, users also rate 4 as usefulness for the Sharing functionalities. The result continues with Figure 6.9 ease of use of the system and attractiveness of the GUI at Figure 6.10.

The last two questions are to determine the functionality or controls that users prefer and whether or not they accept VPS as a market software application. The results are as shows.

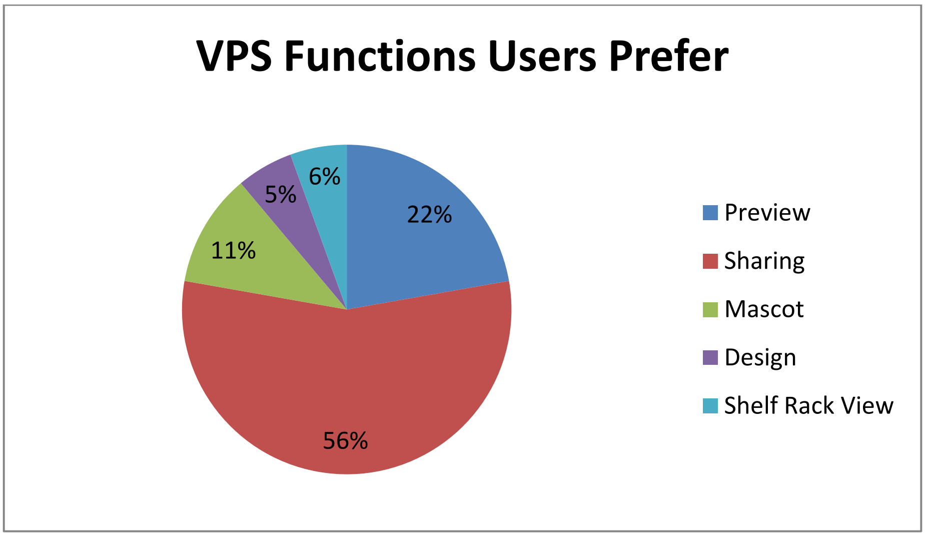


Figure 6.11: VPS Functions Users Prefer

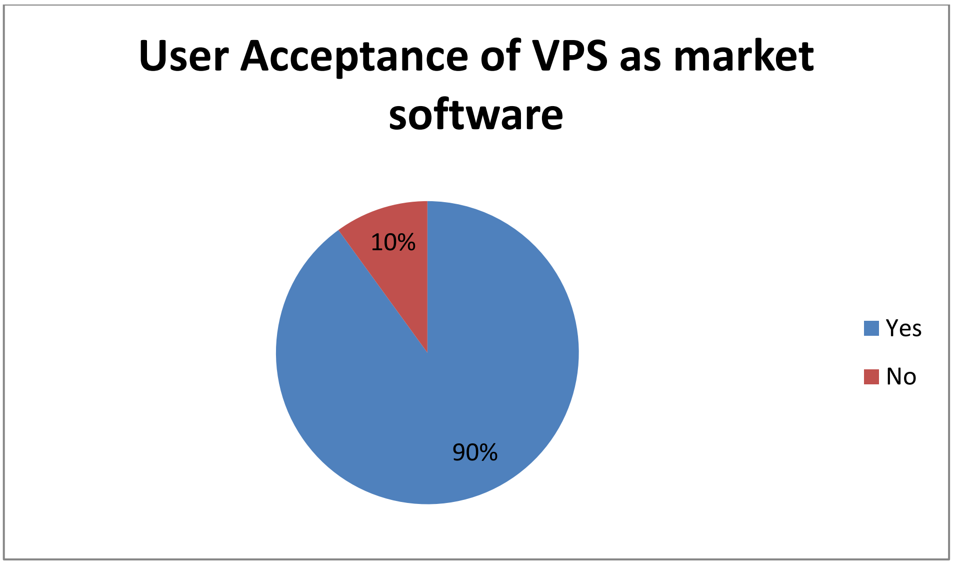


Figure 6.12: User Acceptance of VPS as market software

Figure 6.11 shows that Users prefers the Sharing capabilities most of 56% of the entire number of users and 22% for Preview functionality which comes second of the obtained result. For the last evaluation, 90% of the users accepts VPS as a market software as shown in Figure 6.12. Therefore as a conclusion from the result obtained, VPS is a publically acceptable software application.

Chapter 7: Conclusion

7.1 Project Review

This project develops a Document Management System, VPS with the objectives of to provide file system sharing between users and instant document storage, to facilitate document organization among amounts of document owned by users and to assist user in improving daily productivity by minimizing hectic processes of document managing. The modules and functionalities achieved are documents Categorization, Recommendation, Search and Sharing.

The Sharing module enables users to have a document sharing platform. This module is to achieve the first objective which provides users a file system sharing between users. The Categorization module strives to achieve the second objective which provides document organization to users. The last two module of Recommendation and Search aims to achieve the third objective which is to improve user`s daily productivity.

Based on the results obtained from the system testing, the project development is concluded a success as it archives most of the project objectives. The system contains all the minimum required functionalities. However there exists some limitations and weakness of the software application. The first limitation of the system is the document processing. VPS requires a large amount of fix data like Stop word lists and Stem word list to have accurate content processing of documents. Another limitation of the system is that the system requires stable internet connection to the cloud and database servers to utilize its functionalities. The limitation occurs during the system testing where users encountered system error when the application experienced low internet connectivity.

7.2 Future Work

The future developments of this system application contain various improvements to the software application itself. The first improvement that can be done is to alter document processing method of the system. The system can be programmed to learn words that can be included as a valid English term on the system. This is to handle the limitation of the system to require large fix data as mentioned. Besides, the system can be improved to implement more multithreaded work on document processing and image processing on images in documents.

Other than that, the system can be improved to embed with local database as backup to overcome low internet connectivity. The system application can be programmed to deal with local database on user interaction which does not require other user or system application connections. Finally, the system can be improved to have multiple cloud systems where users can have unlimited variety of cloud storage providers to maximize storage space.

Bibliography

Document Managemen Overview. (2007). Overview.

A Multi-attributed Hybrid Re-ranking Technique for Diversified Recommendations. (2014). *IEEE CONECCT2014* , 1-6.

Annett Mitschick, K. M. (2009). Generation and Maintenance of Semantic Metadata for Personal Multimedia. *2009 First International Conference on Advances in Multimedia*, 74-79.

Anuj Sharma, R. D. (2009). A Wordsets Based Document Clustering Algorithm for Large Datasets. *International Conference on Methods and Models in Computer Science, 2009*.

Citrix. (2012). *Top Five Requirements for Secure Enterprise File Sync and Sharing.* Citrix.

D. Minnie, S. (2011). Intelligent Search Engine Algorithms on Indexing and Searching of Text Documents using Text Representation. *2011 International Conference on Recent Trends in Information Systems*, 121-125.

D. Shivalingaiah, S. K. (2012). Applications of Cloud computing for resource sharing in academic libraries. *Proceedings of 2012 1ntemational Conference on Cloud Computing, Technologies, Applications & Management*, 34-37.

G.Suresh Reddy, D. ,. (2014). A Frequent Term BasedText Clustering Approach Using Novel Similarity Measure. *2014 IEEE International Advance Computing Conference (IACC)*, 495-499.

Ghasemi, G. Y. (2013). Geo-based Search Engine. *2013 5th Conference on Information and Knowledge Technology (IKT)*, 495-501.

Heckman, J. (2008). *Why Document Management:a White Paper.* Heckman Consulting.

HengSong Tan, H. Y. (2009). A Collaborative Filtering Recommendation Algorithm Based On Item Classification. *2009 Pacific-Asia Conference on Circuits,Communications and System*, 694-697.

Heon-min Lee, S.-b. H. (2014). A Query Recommending Scheme for an Efficient Evidence Search in e-Discovery. 1237-1241.

Hongqiang Wang, D. Z. (2010). Library Knowledge Sharing Based on Cloud computing. *2010 2nd International Conference on Software Technology and Engineering(ICSTE)*, V1-424 -V1-427.

Idilio Drago, M. M. (2012). Inside Dropbox: Understanding Personal Cloud Storage Services.

Khalifa Chekima, C. K. (2012). Document Categorizer Agent based on ACM. *2012 IEEE International Conference on Control System, Computing and Engineering, 23 - 25 Nov. 2012, Penang, Malaysia*, 387-391.

Leiyue Yao, X. X. (2013). Design a Teaching Resource Sharing System in Colleges Based on Cloud Computing. *2013 International Conference on Information Technology and Applications*, 374-378.

Li, X. (2011). Collaborative Filtering Recommendation Algorithm Based on Cluster. *2011 International Conference on Computer Science and Network Technology*, 2682-2685.

Mohammadreza Shams, A. M. (2013). Topic Word Set-Based Text Clustering. *IEEE 7th International Conference*, 1-10.

Pingsong Xia, J. X. (2013). An Application of Recommender System with Mingle-TopN Algorithm on B2B Platform. *2013 International Conference on Advanced Cloud and Big Data*, 170-176.

SUN Ning-ning, F. C. (2009). An Effective Three-step Search Algorithm for Motion Estimation. 400-403.

Woodward, J. R. (2010). The Necessity of Meta Bias in Search Algorithms.

Yi Zhou, Y. Q. (2010). Semantic Based Personal Computer Resource Management System. *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, V7-656-V7-660.

Ying-Wei Chen, X. X.-G. (2012). A Collaborative filtering recommendation algorithm based on contents` genome .

Yun-Ho Ko, H.-S. K.-W. (2012). Fast Motion Estimation Algorithm Combining Search Point Sampling Technique with Adaptive Search Range Algorithm. 988-991.

Zhisheng Li, K. C.-C. (2011). IR-Tree: An Efficient Index for Geographic Document Search. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 4, APRIL 2011* , 585-599.

Appendices

Please refer to "**VPS Testing Survey Form.pdf**"