DOCUMENT MANAGEMENT SYSTEM FOR INTERNATIONAL COLLEGE, KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

Kittichai Lerdkitphaiboon

Duangporn Temcharoen

Apipol Niyomsak

Thesis – Academic Year 2015
B.Eng. in Software Engineering
International College, King Mongkut's Institute of Technology Ladkrabang

Thesis — Academic Year 2015

B.Eng. in Software Engineering
International College, King Mongkut's Institute of Technology Ladkrabang

Title

Document Management System for International College, King Mongkut's Institute of Technology Ladkrabang

Members

Mr.	Kittichai	Lerdkitphaiboon	55090006
Mrs.	Duangporn	Temcharoen	55090022
Mr.	Apipol	Niyomsak	55090052

Approved for submission

Isara Anantavrasilp — 31.05.2016

Dr. Isara Anantavrasilp (Advisor) — Date

31/5/2016

Assist.Prof.Dr. Chaiwat Nuthong (Co-advisor) — Date

31/5/2016

Dr. Nattaphong Jungteerapanish (Co-advisor) — Date

Abstract

Mr.	Kittichai	Lerdkitphaiboon	55090006
Mrs.	Duangporn	Temcharoen	55090022
Mr.	Apipol	Niyomsak	55090052
Dr.	Isara	Anantavrasilp	Advisor
Assist.Prof.Dr.	Chaiwat	Nuthong	Co-advisor
Dr.	Nattaphong	Jungteerapanish	Co-advisor

Academic Year 2015

Nowadays, in the international college area have the among hudge problems about the information physical documents and the workflows is very complicate to following so the electronic system which is Document management system(DMS) will help to replace the physical documents into electronic documents and the system can manage and store all the files on the cemter, IC staff also can retrieve, share, tracking and view the detail of documents on our system. Our project can improve efficiency of IC staff working and use the papar when necessary.

Acknowledgement

At first, we would to express my great appreciation to our project advisor, Dr. Isara Anantavrasilp for the great valuable suggestion in developing our a web-based document management application for software project. We are grateful for his advice, not only the project methodologies but also many methodologies in real life. With report, it is a great honor to get the suggestion and the guideline from him in writing an excellent thesis. Without his help and suggestion, our report would have been completed.

In addition, I proudly like to say sincerely thank you to all professors from the faculty of International College, KMITL who always encourage us to do the project and give us the best teaching and knowledge in software engineering during 4 years. We are also particularly grateful for Dr. Nattapong Jungteerapanich and Dr. Chaiwat Nuthong for the recommendation to create the nice feature for the project

Finally, We gratefully acknowledge our parents who always support in learning and studying in this faculty during 4 years. We wish to thank our friends who always help us and support us throughout the software project period.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

IC is a faculty of KMITL has about 25 employees, there are have two types of employees which are academic staff and adminstrative staff.

Their main objective is to provide educational services to students. Currently, they are having three major problems involving accumulating documents. First, the organization takes countless hours to store and retrieve documents even though they have organized documents into categories. Simply retrieving a document can take hours or even weeks. This problem hinders employee's productivity when they need to retrieve many documents. Second problem is losing track of documents, when the documents have to pass through many departments its often get lost along the way of workflow; A workflow is a series of repeatable steps performed in a sequential manner to complete a task.

Third problem is that some IC's workflows are so complicated that they are difficult to keep track of. There are many actions and conditions to execute. Many people from both inside and outside of the organization may be involves during workflow's execution. Supplementary documents, or so called attachments, may need to be attached with original document in order to pass some certain guidelines. As a result, IC can't certainly identify who is currently involved and what actions they must do with that document. They can not track attachments effectively causing delay in archiving and error-prone.

We are working on document management system for IC, King Mongkut's Institute of Technology Ladkrabang, called Monkey Office. Because we want to provide a system to manage documents within IC. Our proposed solution is to manage, store, and retrieve documents digitally using a computer software. The employee will use less time to search and be less error-prone. We will create a system that provide ability to search and to track documents. When we have developed search and track documents, IC can

manage their own files and track documents to get increase organization's productivity.

The similarities between Document Management System (DMS) and file server is to provide a shared storage of computer files that can be accessed by other computers. DMS aims to provide archive documents digitally with access controls while file server aims to store and retrieve files. The differences between DMS and file servers are file to file relationship and predetermined workflow. A file to file relationship is a built-in DMS feature that shows additional documents inside an organization. Usually, a single document relates to other documents. For example, a fire insurance contract refers to property insurance contract and fire policy. Predetermined workflow is an already created workflow assigned to one or multiple documents. DMS automatically assigns access rights to documents based on current workflow's task. One of these two features suppose to have in order to be DMS. Some DMS provide additional features such as version control or document assembly to satisfy organization's requirements.

1.2 Objective

Our primary objectives are

1. To design IC's workflow model[1] associated with each type of documents.
2. To track and to report the current task during workflow's execution. To store and retrieve digital documents corresponding to outgoing or finished workflows from and to IC's server.

1.3 Terminology

We would like to clarify these two words—workflow and document—in terms of our project. A "workflow" is a collection of repeatable tasks performed to achieve a goal [13]. It can execute multiple tasks sequentially or simultaneously depending on how it is created. For instance, there are four tasks namely 'P', 'Q', 'R', and 'S'. The workflow executes P first followed by Q. Then, it executes R and S at the same time. The execution of P and Q are sequential because Q have wait for P to finish first. On the other hand, the execution of R and S are simultaneous because R and S execute in parallel. In organizations, the goal of the document is get them approved, distributed, or archived. Each task may perform with specific conditions. A person, machine, group of persons or machines are responsible to carry out the task [1]. A "document" is a

[1] Although workflow is not our main focus, we work closely with other researchers on "web-based automatic form generated system" [14]. Not only we share a similar objective, but also we have to link our system with them. We need their general workflow models to achieve this objective.

computer's executable file associates with each user's workflow. For example, a user executes workflow 'X'. One of the X's task requires user to upload a Portable Document Format (PDF) file as an input. The task will produce a reply letter as an output. There are two documents in X—the PDF report and the reply letter. So, how workflow and document are related together? A document can be an input or end-product of the task. The task may require a certain document to execute. Other documents may be produced by the task.

1.4 Scope

The similarities between DMS and file server is to provide a shared storage of computer files that can be accessed by other computers. DMS aims to provide archive documents digitally with access controls while file server aims to store and retrieve files. The differences between DMS and file servers are file to file relationship and predetermined workflow. A file to file relationship is a built-in DMS feature that shows additional documents inside an organization. Standalone organization's documents are rare. Usually, they relate to other documents. For example, a fire insurance contract refers to property insurance contract and fire policy. Predetermined workflow is an already created workflow assigned to one or multiple documents. DMS automatically assigns access rights to documents based on current workflow's task. One of these two features suppose to have in order to be DMS. Some DMS provide additional features such as version control or document assembly to satisfy organization's requirements.

We are going to develop a web-based application hosted on an IC's server. Only people within IC and granted external users can access the system. Our primary focus is storing digital documents produced from tasks in the workflow. The system has to manage workflows which are created by the staff of IC and resulting documents. Only documents are doing on the workflows, in the scope of this project. This thesis conducts only on the following two types of documents.

1. Form
   A formatted document with blank fields that prompts user to fill in which generated the form by the admin on Web-based workflow management system. Forms are created according to IC's work procedure and instruction. Work procedure provides details on working procedures, departments, or persons who hold responsibilities for documents. Work instruction indicates employee's roles and responsibilities. IC defines non-filled form as 'Form' and already filled form as 'Record'. In this project conduct with three sub-types forms.

   (a) Absence form
   For IC's employee and students to take days off due to personal or business reasons.

(b) Student internship form
A form for IC students to take an internship.

(c) Conference outside KMITL form
For lecturers to request going to a conference outside KMITL.

Because all of these forms are in electronic format, we have to complete the tasks of workflow then it can be convert them into the document file.

2. Physical document
Documents received from outside or physical document from IC's departments in order to proceed some tasks in a workflow. If received document is in paper form, it has to be scanned by a scanner first. The task includes these documents as attachments.

Chapter 2

Literature Review

This chapter introduces some existing document management system. Section 2.1 gives lists of required features of Document Management System for Monkey Office. Section 2.2 explores some existing document management applications.

2.1 Integrated features

Document management is the process of applying policies and rules to how documents are created, persisted, and expired within an organization. We analyse each service based on the following features.

Access Control
: This feature controls how much user can interact with the document. The system should use role-based access control to authorize the user because IC classify employees by their responsibilities. A role-based access control defines what operation user can perform based on their roles.

Archiving and Retention
: How the system stores documents on a permanent storage. How the system search and retrieve files. This feature manage the files and how to access them.

Collaboration
: This feature enables users to cooperate with other individuals on the same document.

Revision Control
: Revision control is the management of changes to documents. It stores changes' history for later recovery. In this case, the system has to have a periodic document backup when user is editing a document. Lastly, user should be able to create new revision of document when they ready to publish the document.

Form Management

This feature let user creates and manages document forms. The system treats form as another type of document. The difference is user can fill in the blank fields provided in the document.

Document Searching Service

When user request a document, the system will look in its storage based on given keywords. User does not have to look within the storage by oneself. The system provides user with various searching criteria for user to get what they need.

2.2 Related Works

We pick nine popular existing document management applications with related to our project requirements to investigate. Each application has its interesting functionalities and characteristics. and review them based on seven characteristic features as stated above.

Alfresco One

Alfresco One allows organizations to manage any type of content from small office documents, photographs, to large video files. The system allows users to choose how to access their content — via the Web, desktop or email — while the server enforces access controls and security.

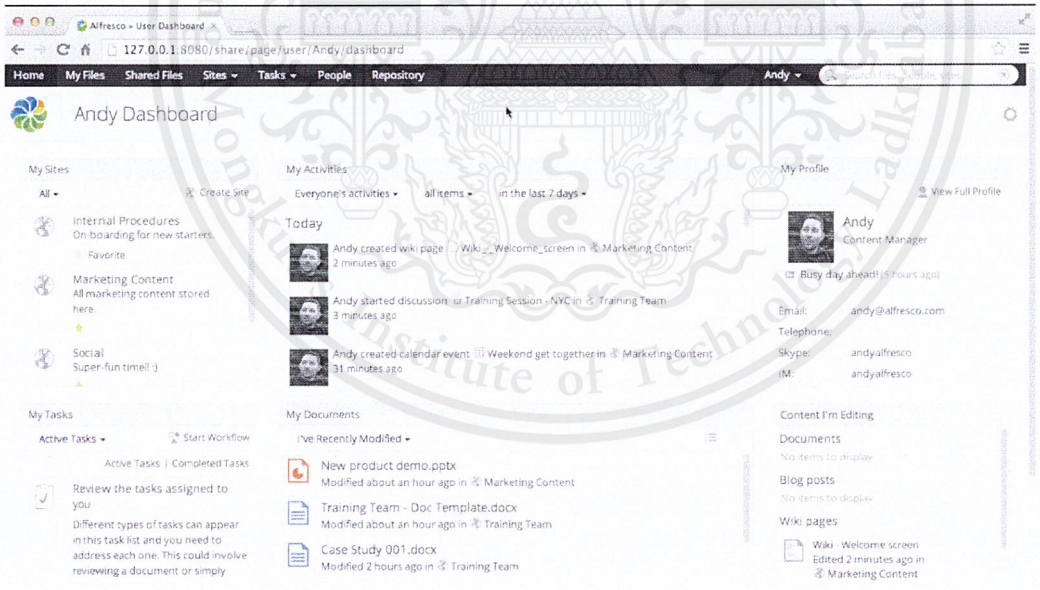


Figure 2.1: A screenshot of Alfresco One (Community Edition) 5.0.0d [5]

Dokmee

Dokmee has multiple editions targeted at companies of all sizes. It can run in a Windows-based Intranet network, as a Web-hosted system or as an individual software on a personal computer. User can organize files into folders and can store unlimited number of files.

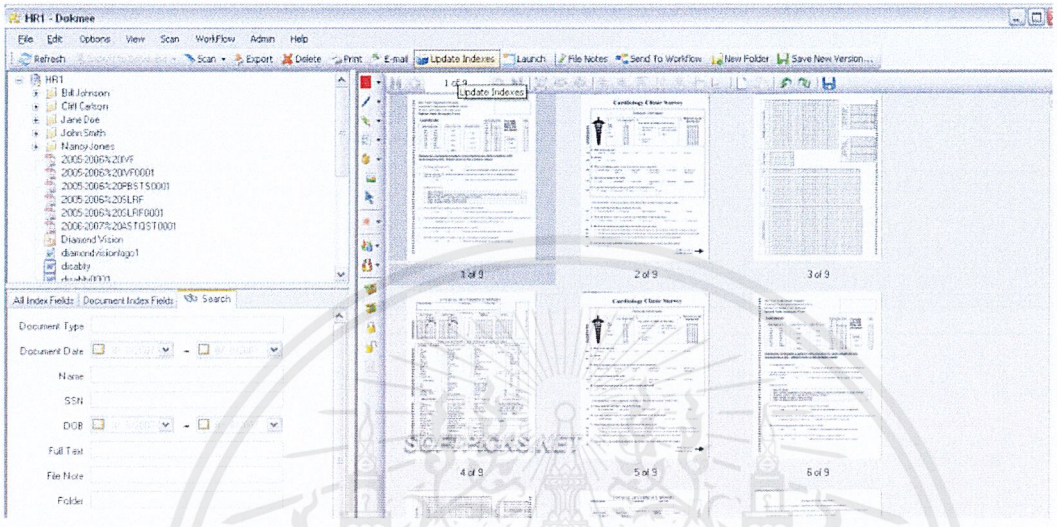


Figure 2.2: A screenshot of Dokmee 3.0.2.4 [21]

Asite Document Manager

Asite Document Manager provides an organization with a centralize storage with folder-like structure. They claim that doing so saves user's time and effort of each party managing their own filing system.

Eclipse Document Managment

This program, developed by docSTAR, offers solution for storing and scanning electronic document files. User can grant access to documents based on user permission. It has a centralized storage accessible from other locations. It searches documents by text, field, annotation, and name with auto-complete feature. The program also claims that it can locate documents that have a similar pronunciation too.

eFileCabinet

eFileCabinet hosts documents on their cloud server. It searches document by filename and tags. The program saves search history every time user perform a search for auto completion. It can look for word and phrases inside a document. User can sends document directly to eFileCabinet from Windows File Explorer or Microsoft Office Products (Add-ons).

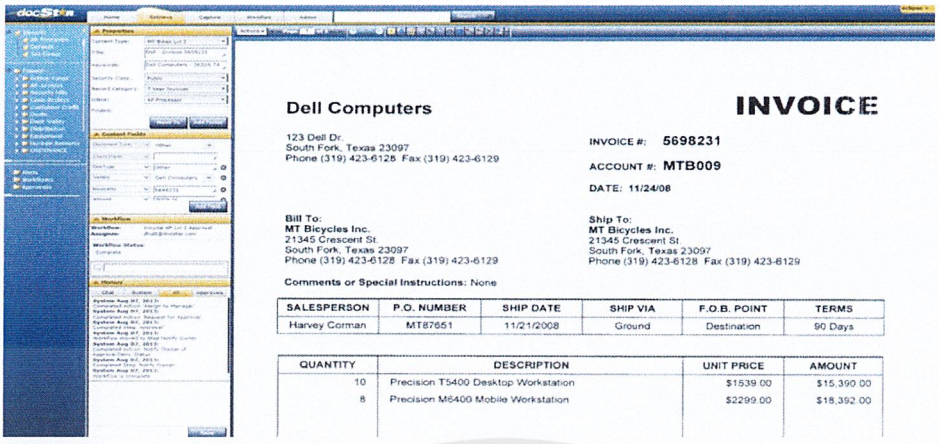


Figure 2.3: A screenshot of Eclipse Document Managment 3.1 [8]

Contentverse

Contentverse has file naming and storing similar to cabinet and folder structure, indexed for immediate full-text search and retrieval. The program track every changes – every version of any file available for reference. The program allows user to assign permission to users based on their specific roles.

SmartFile

SmartFile is a file management solution that enables everyone to securely share, manage and control files both offline and online. SmartFile offers Optical Character Recognition (OCR) search. User can find content within their documents. The program narrows search result by let user specify date interval and file size interval. User can view PDF, JPG, PNG, GIF, and other image formats with its own viewer. The characteristics of this program is its permanent activity logs storage. An activity log contain file's access information, who access the file, what user did to the file. These logs are kept in the program permanently.

isoTracker Document Control

The software encrypts documents and stores safely in folders. User can set access permission on files or folders themselves. The program search documents by metadata and tags. Documents can be reviewed, commented on and modified. The software also features a complete usage log for each document.

Zoho Docs

Zoho Docs is a document management system on a cloud server. User can access the file anywhere on their smart phone device. The software encourages real-time collaboration among other team members. The program also has its own tools for creating word documents, spreadsheets, and presentations. Search feature of this

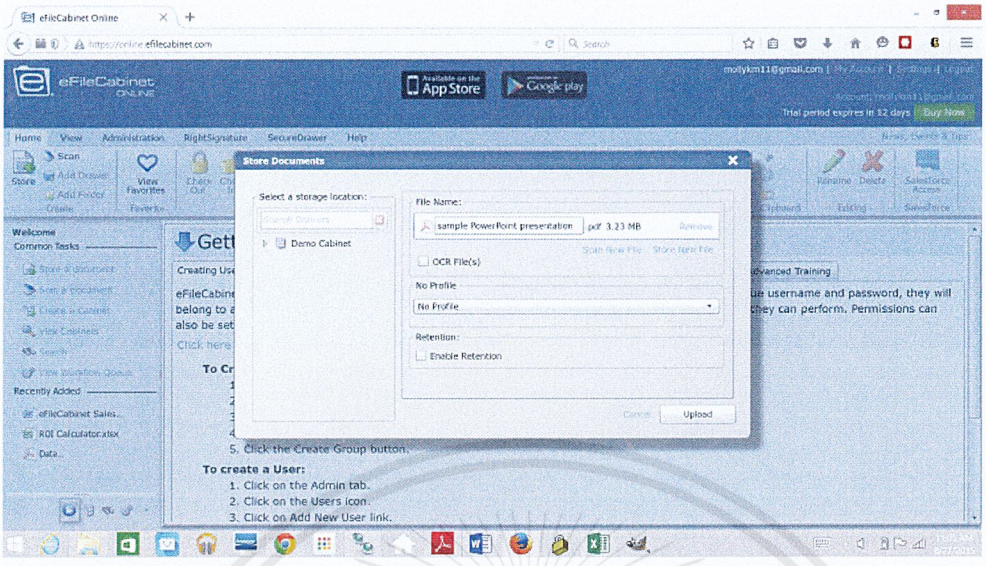


Figure 2.4: A screenshot of eFileCabinet 2014 [16]

program can search by filename only. It can not search for contents within the file. Zoho Docs has a predefined folders separate by categories such as accounts, sales, and marketing. User can create a multi-level folder structure to organize their files.

Table 2.1 shows a comparison application platform available. Table 2.2 shows a comparison of features that we interested. "✓" symbol inside the table indicates a yes.

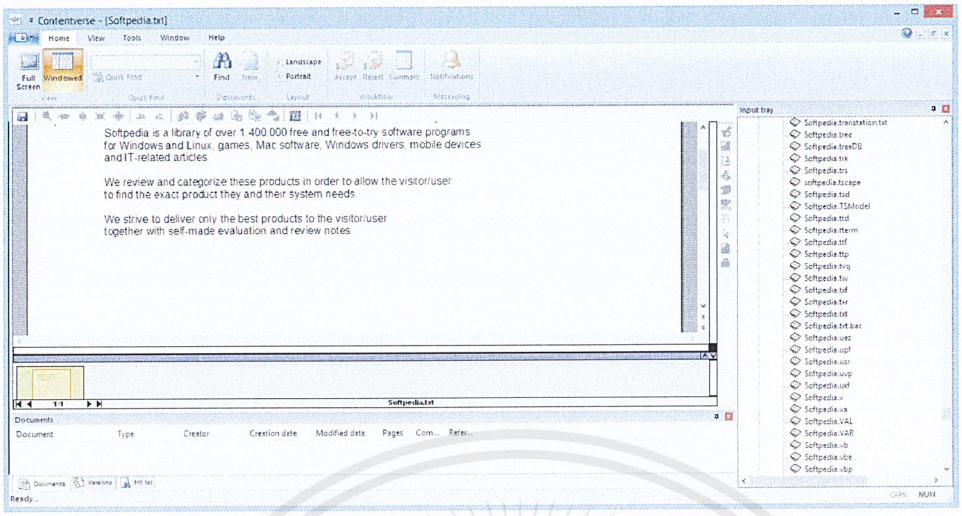


Figure 2.5: A screenshot of Contentverse 8.0 [20]

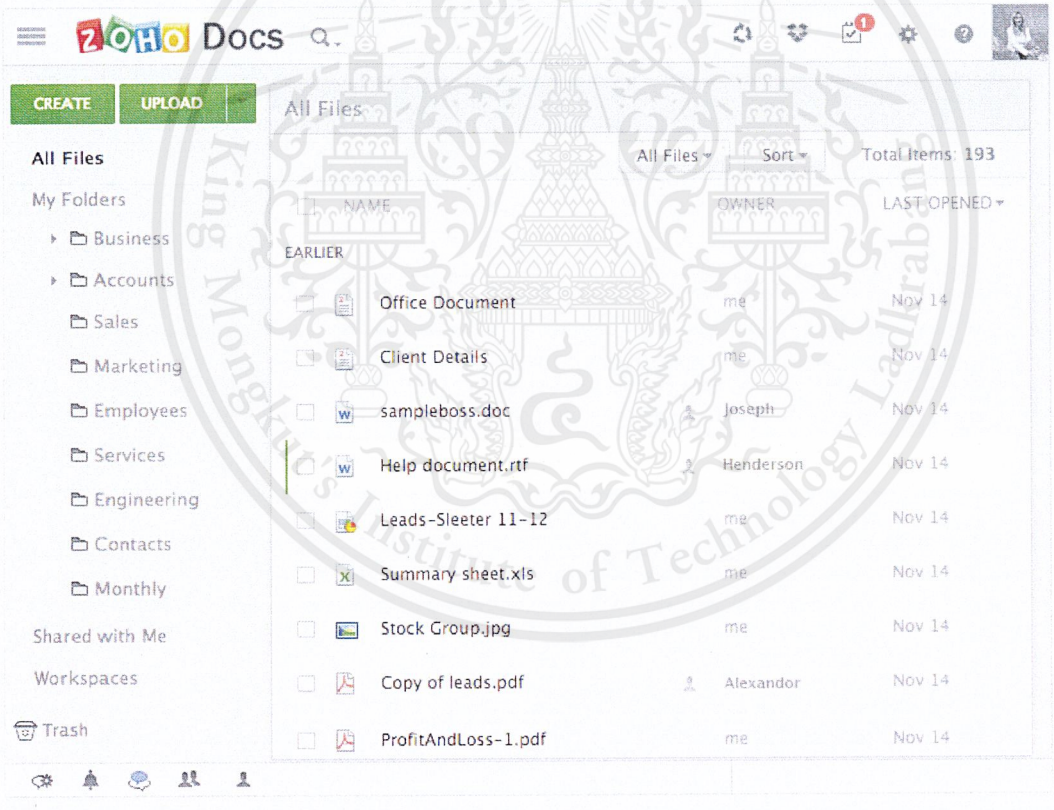


Figure 2.6: A screenshot of Zoho Docs [24]

Table 2.1: The comparison of all reviewed program by their deployment

	Deployment		
	Require a program to be installed on a computer	Has mobile application	Access via a web server
Alfresco One		✓	✓
Dokmee	✓	✓	✓
Asite Document Manager		✓	✓
Eclipse Document Management	✓		✓
eFilecabinet	✓	✓	✓
Contentverse	✓	✓	✓
smartFile			✓
isoTracker			✓
Zoho Docs	✓	✓	

Table 2.2: The comparison of all reviewed program and their features

	Features						
	Access Controls	Archiving & Retention	Collaboration	Document Assembly	Form Management	Document Searching Service	Revision Control
Alfresco One	✓	✓		✓		✓	✓
Dokmee	✓	✓				✓	✓
Asite Document Manager	✓	✓	✓		✓	✓	✓
Eclipse Document Management	✓	✓	✓	✓	✓	✓	✓
eFilecabinet	✓	✓	✓			✓	
Contentverse	✓	✓	✓	✓	✓	✓	
smartFile	✓	✓				✓	
isoTracker	✓	✓	✓				
Zoho Docs	✓	✓	✓	✓	✓	✓	✓

Chapter 3

Background Knowledge

3.1 An overview of Document Management System (DMS)

In the 1980s, printers, scanners, and household computers started to gain popularity. Organizations start to take actions on managing their information records and assets seriously. Document Image Processing (DIP) systems were the only available software to satisfy their needs. DIP is the electronic version of filing cabinet where documents need to be scanned, indexed, and store in the system [2]. DIP can also analyse figures, texts, and handwriting using various image processing techniques [4].

Later on in the 1990s, Electronics Document Management System (EDMS) is developed targeting large enterprises with high volume of documents. It is the improved version of DIP with workflow functionality. Workflow functionality enables organizations to passed around scanned document throughout the organization to designated employees. EDMS also has its own document repository allowing documents to be indexed and tracked using version control. There are various sub-types of EDMS such as Electronic Record Management Systems (ERMS) deals mainly with record keeping. Enterprise Content Management (ECM) is a suite of applications that manages documents and records. Now, EDMS acronym is shortened to DMS. Figure 3.1 shows a basic components of DMS. Most DMS applications have these components implemented within.

Document Repository
: Document Repository is a place where indexed documents are stored, usually on a server.

Integration with desktop applications
: Allowing user to save documents straight to application after the document is

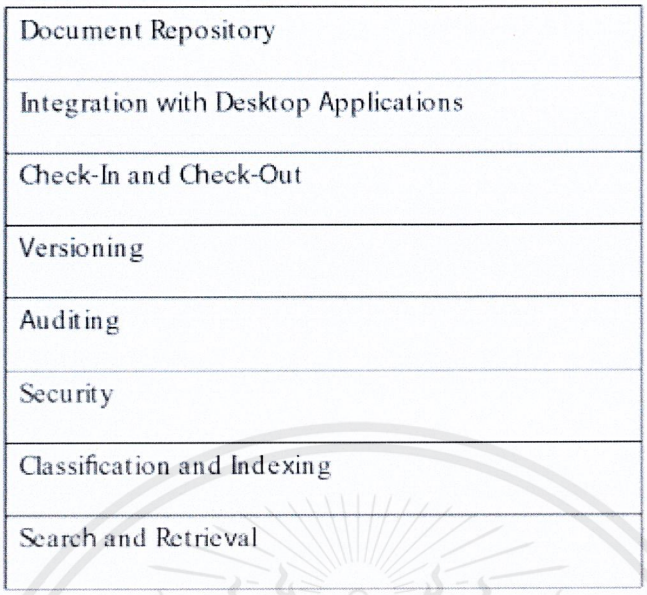


Figure 3.1: Components of EDMS (Adam 2008)

created. It is usually a 3rd party add-on embedded in popular office applications such as Microsoft Office.

Check-in and check-out
This feature controls who is allowed to make changes or to read documents. Basically, it is a user permission control system.

Versioning
Keeping track of changes by assigning a version number to a document. The number is incremented after the document passes major revisions. User can access the previous versions of the document.

Auditing
Track who changes document, when, and where. Only authorized user can read these information.

Security
Controls how documents should be stored in the server to prevent unauthorized users from attacking the system.

Classification and indexing
Metadata and tags provide more information to documents. It helps to search and retrieve documents easier.

Search and Retrieval

Allow user to retrieve documents according to keywords. Keywords can be a metadata or contents within a document. A system may offer advance search criteria by looking for individual fields and combine with other fields using basic boolean operations (AND, OR, NOT).

3.2 Metadata

Metadata literally means "data about data" [6, p. 1].

There is no clear definition on what metadata is because this term is used differently in different communities. For librarians, it refers to information in the library catalog that help users to find the right book in the library. For search engines, it means descriptions of web page's contents and keywords used to rank relevant websites. For others, it refers to a descriptive information of resources in human readable format. Whatever metadata refers to, they share the following similar usage.

1. To identify resources.
2. To distinguish or bundle similar resources.
3. To find resources based on search criteria.

Electronic documents contain descriptive metadata providing additional information on the document such as Document name, Author name, Date etc. The reason for having metadata associate with the document is that they can be indexed in the system. Index is a systematic arrangement of entries designed to enable users to locate information in a document [12]. User can use metadata to quickly locate documents after they are archived in the system. The electronic document contains information such as those in figure 3.2, 3.3, and 3.4. Figure 3.2 shows the metadata of a docx file created by Microsoft Word 2013. User can edit some metadata fields directly from properties view in Windows 10 or in Microsoft Word. figure 3.4 is the metadata of a Portable Network Graphics (PNG) file. It has information on image dimension and bit depth. User can specify date taken in "Date taken" field. "File" metadata category in figure 3.2 and 3.4 are generated automatically by the operating system for every type of file. The operating system can use this metadata to index the file within the hard disk. Figure 3.3 shows metadata in a PDF file. This metadata needs a PDF reader software (e.g. Adobe Reader, Foxit Reader, etc.) to show it. Windows 10 can't display this metadata because Windows doesn't have native support for PDF. However, there is a third-party software available[1] that displays PDF's metadata on a file explorer.

[1] `http://coolsoft.altervista.org/en/pdfpropertyextension`

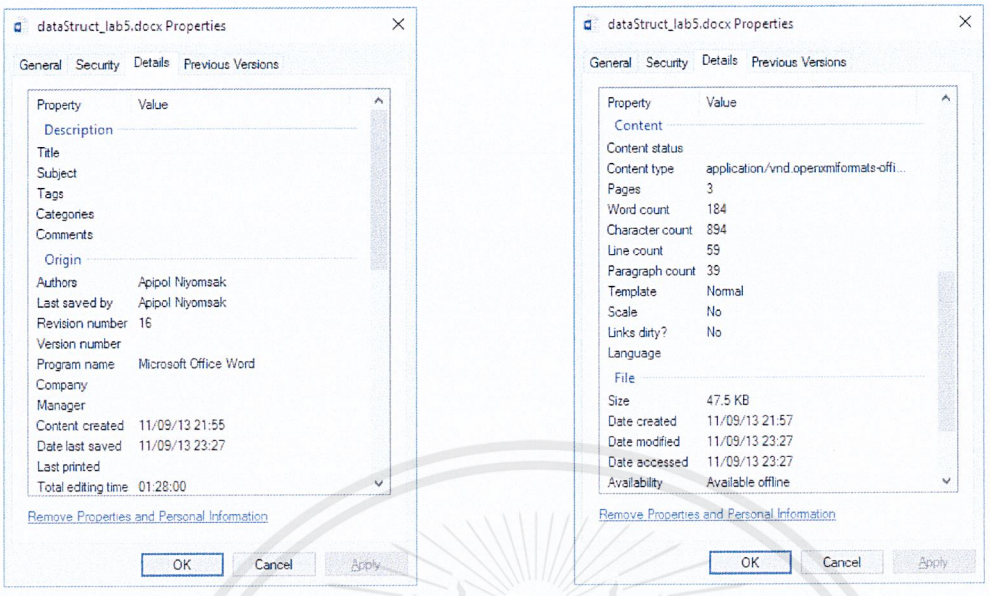


Figure 3.2: Microsoft Word Document (docx) metadata viewed Windows 10

3.3 NoSQL

Not only SQL (NoSQL) is a database that does not use Structured Query Language (SQL). It refers to any database that does not follow the traditional Realtional Database Management System (RDBMS) model. SQL is designed to be a query language for relational databases which are usually table-based. Records are stored in rows and columns represent fields. On the other hand, NoSQL allows to define fields while creating a record. Nested values are common in NoSQL databases because NoSQL is aggregate oriented. Hashes, arrays, and objects can be nested with themselves.

The main characteristic separating NoSQL databases from RDBMS is that they do not use query languages derived from SQL. The following list shows common features of NoSQL [10, p. 12-16].

Schema agnostic

NoSQL databases do not require schema to be defined explicitly. Any type of records can be stored without having to know how database stores them internally. This improves flexibility when designing document's metadata model. Some constraints such as field type and field length do not have to be defined beforehand. NoSQL can store mixed data type in the same field. If there is a new type of form, we can reuse the native form model by extending it instead of defining a new schema.

Nonrelational

Figure 3.3: Portable Document Format (PDF) metadata viewed on Adobe Reader DC

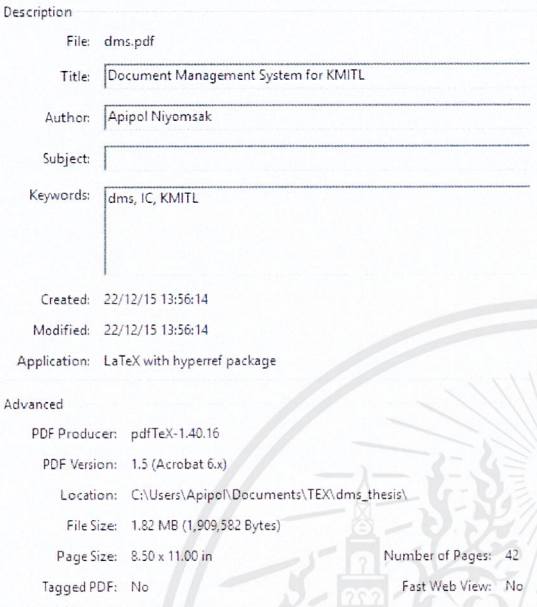
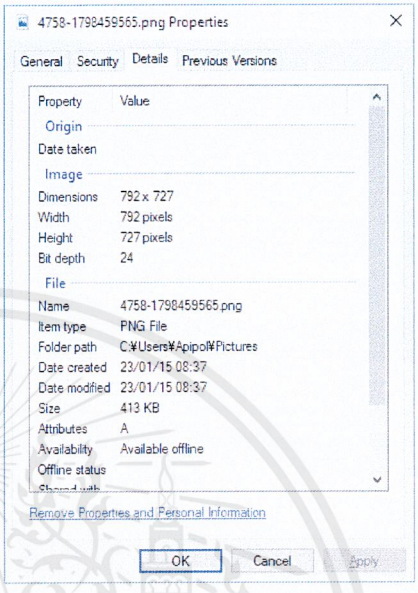


Figure 3.4: Portable Network Graphics (PNG) metadata viewed on Windows 10



A relational database needs relations to describe how tables relate to each other. Unlike RDBMS, NoSQL databases do not have any relation concept. They don not store record's relation. Meaning that the metadata model does not have to go through database normalization process to minimize data redundancy. Programmer can focus on modelling metadata without having to worry about modelling table relations in RDBMS.

There are many NoSQL storage types available to model the content. For example, "Column-oriented database" stores data as columns instead of rows in RDBMS. "Graph store" represents data as nodes and relationships as edges in a graph. NoSQL databases with "Document Store" storage type organized data as a hierarchical model with parent-child relationships(Document Store will keep every documents and any related with documents on database). DMS for IC use "Document Store" storage type to manage document's metadata. "Document store" is a hash table in which there exist a unique key. A unique key points to a particular data. The data in which the key pointed to is called "value". The value can also contains a key or a an array of keys. For example, Using 3.4 to show metadata as key-value in table 3.1.

The reason to use this storage strategy is that can query values from keys of keys in any depth. Another reason is that programmer can store objects to NoSQL database

Table 3.1: Figure 3.4 shown as "Document store" format

Key	Value	
Dimension	Key	Value
	Width	792
	Height	727
	Bit depth	24
File	Key	Value
	Name	4758-1798459565.png
	Item type	PNG File
	Folder path	C:/Users/Apipol/Pictures
	Date created	23/01/15 08:37
	Date modified	23/01/15 08:37
	Size	413 KB
	Attributes	A
	Availability	Available offline

easily by converting them to JavaScript Object Notation (JSON) objects. "Document store" use JSON, Binary JSON (BSON), or Yet Another Multicolumn Layout (YAML) to manipulate data. Client have to use REST API provided by the database to query the data. REpresentational State Transfer (REST) is a network-based software architectural style of the World Wide Web [9, 15]. It takes care interactions between client and server. Client program communicates with the database using Hypertext Transfer Protocol (HTTP) through exposed APIs. "Document store" databases can import JSON file directly. Querying from "Document store" databases will return JSON results. Some web programming languages such as PHP and Javascript can parse JSON results by converting them to associative arrays. Thus, programmer can use them to develop their systems.

3.4 Business Process Modelling Notation (BPMN)

BPMN is a graphical notation that depicts the steps in business process [19]. The goal is to represent business processes using standard graphical notation. BPMN targets business users and process implementers who need a standard model to communicate their business process. Business users create, manage, and monitor processes while process implementers turn processes into a physical implementation. BPMN does not focus on why, when, and how a process is performed. But rather what processes, which are the steps to achieve that process, and who should do them. With BPMN, organizations can understand, improve, and control their business processes. Table 3.2 summarizes BPMN symbols that appear in this report along with their descriptions.

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011) [18]

Element	Description	Notation
Start Event	Indicates where a process begins	

Continued on next page

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011)

Element	Description	Notation
Intermediate Event	Intermediate Event represents a status reached in a process, and it is modelled explicitly.	
End Event	Indicates where a process ends	
End Event Terminate	Terminate all running activities in a process immediately without any error handling.	
Message	A Message represents information communicating between two participants.	See two rows below.
Start Message Event	An event triggers a process to start when a Message arrives from a participant.	
End Message Event	A message received from a participant concludes the end of a process.	

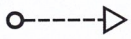
Continued on next page

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011)

Element	Description	Notation
Activity	An activity performs work within a business process. It can be single or compound. It is an executable element of BPMN.	See three rows below.
Task	A Task is a unit of work performs in a process. It has a task name inside a rounded rectangle notation. The task is used when process can not be broken down any further. It has no internal parts, representing a single action.	
User Task	A subtype of Task in which human is the one who perform a task with the help of a software application.	
Sub-Process	A Sub-Process composes of a single activity or compound activity. It refers to a process that can be broken down to a finer detail. The collapsed Sub-Process hides activity details from the diagram indicated by a plus sign. The expanded one shows activities inside the diagram indicated by a minus sign.	 Collapsed Sub-Process  Expanded Sub-Process

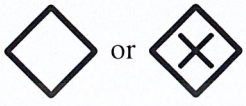
Continued on next page

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011)

Element	Description	Notation
Activity Looping	Activity Looping loops itself continuously until its boolean condition becomes *true*. It evaluates condition every time before/after it performed for every loop iteration.	
Normal Flow	Shows activity order performed in a process	
Conditional Flow	It will perform the outgoing sequence flow if and only if conditional expression evaluates to *true*.	
Default Flow	Execute this flow if all other outgoing conditional expression evaluate to *false*.	
Message Flow	Indicates a Message sent between two participants.	
Gateway	Controls branching, merging, forking, and joining multiple sequence flow paths.	See four rows below.

Continued on next page

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011)

Element	Description	Notation
Exclusive Gateway	Creates multiple alternative paths in a process. If one of the conditional expression evaluates to *true*, it executes that path without evaluating others. It has the same functionality as conditional flow. The difference is that Exclusive Gateway allows multiple branching from the same node.	
Event-based Gateway	Evaluates which event occurs first, not which condition is met. If that event occurred, go to that outgoing flow and discard other conditions.	
Parallel Gateway	Parallel gateway is used to create and to combine parallel flows. It represents multiple concurrent activities performing at the same time. It doesn't evaluate any condition or event before execution. Parallel gateway will wait for all incoming flows to finish before proceeding to the next outgoing flow.	
Inclusive Gateway	Evaluates all conditional expression unlike the exclusive gateway. It is used to create alternative parallel paths. If there are multiple conditional expressions evaluated to *true*, those flows will be executed independently in parallel.	

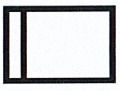
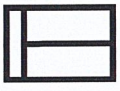
Continued on next page

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011)

Element	Description	Notation
Data Object	Provide information about what activity requires to perform and/or what they produce. Data Object can also refer to a collection of Data Objects. Data Input have a white arrow on the top corner. It is placed before activity starts indicating what that activity requires. Data Output have a shaded arrow on the the corner. It is placed after activity finished indicating what activity produces.	 Data Object  Data Input  Data Output
Text Annotation	Text Annotation provides additional information to a BPMN diagram reader. Its notation composes of a opening square bracket with a dashed line attached on the left. The dashed line can connect to any BPMN element. The bracket contains a descriptive text on the right. It scales vertically to cover multiple lines of text.	

Continued on next page

Table 3.2: Summary of BPMN symbols (Object Management Group, Inc. 2011)

Element	Description	Notation
Pool	A Pool represents a collaboration between participants in a process. It can be partitioned to multiple smaller Pools called Lane. Each Lane assigns to one participant. The Participant is a person, machine, group of persons or machines responsible for the process execution within the Pool. The Pool is a container of activities. All activities inside the Pool can cross boundaries to any Lane but not outside the Pool. There are two types of Pool—horizontal and vertical. Both of them has the same functionality. The difference is their alignment. A horizontal one expands horizontally while a vertical one expands vertically.	 Horizontal Pool  Horizontal Pool with Lanes  Vertical Pool  Vertical Pool with Lanes.

Chapter 4

Requirement and Analysis

4.1 Introduction

The overall purposes of this project is to create a system to improve efficiencies of IC's digital document management strategies. The study involves investigating internal IC's working procedures and regulations. How IC execute strategies on their digital document assets. What are advantages and disadvantages of their strategies. Gaining such information is valuable to this study because plausible solutions could be proposed and implemented. Solutions that could further improve flows of documents inside IC. This chapter presents requirement gathering methods to address the problem.

4.2 Gathering Requirements

To understand what user expects the system to do is an essential part of developing a software. The more system operates close to their expectation, the more satisfaction they get. User's expectation on the system can be described as requirements. A descriptive text describes the system's functionality. What they are and why the user need them. The following sections discuss these two strategies in order to gather requirements from the user.

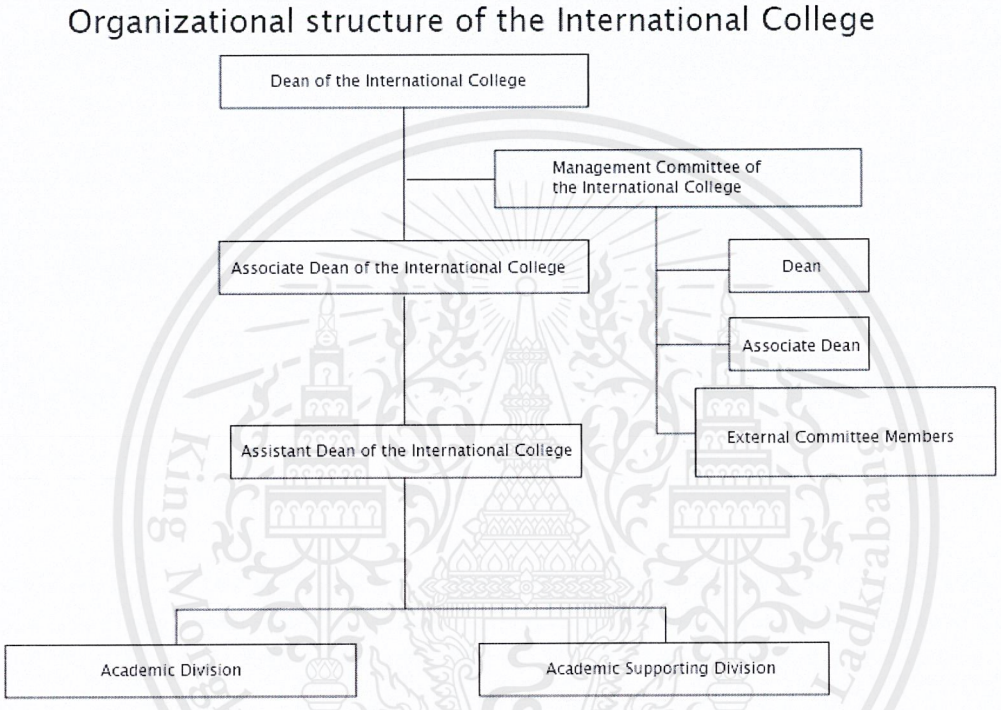
4.2.1 Interview

Gall (2003) states that interview is the spontaneous generation of questions in a natural interaction, typically one that occurs as part of ongoing participant observation field-work. Brady (2011) points out that what interviewer and salesman have in common is potential customers whom one could hold their attention to talk. Getting an interview means making an appointment to see the subject, identifying questions related to the research topic, and showing on time for the interview. The purpose of interview is to

gain information from interviewee by having interviewer asking questions. Researchers can gain useful insights from the subject who is expertise in one's field.

Figure 4.1 shows IC's organization structure. The top-most is the dean followed by deputy dean then assistant of deputy dean. On the right hand side are committee members. Only academic and academic support located at the bottom-most hierarchy involves archiving documents.

Figure 4.1: IC organization's structure



In this study, IC staffs are the subject of this study divided into two focus groups:

Academic staffs as a primary focus group because they are responsible for keeping records of all IC's documents. Transfer documents around the organization according to IC's workflow specifications. Managing day-to-day operations within the organization. They also provide academic advices and guidances to IC undergraduate and postgraduate students.

Administrative staffs as a secondary focus group because they are directly involve in keeping and organizing documents. Some of them have privilege to issue, review, and approve documents. In order to know more about their procedures, deputy dean and deputy dean's assistant are selected as a sample for this study.

After conducting the questionaire interview, IC staffs reveal that they would like a system that manage their organization's documents automatically. What they mean by managing is that they would like to know what are they suppose to do with received documents. Things they have to do could be approval, signing, or printing documents. Then, forwarding the document to another person who is involved in the workflow. They also would like to know any additional documents required before forwarding documents so that they can prepare them beforehand.

For every official IC's documents, IC assigns an identification number internally. There are eleven types of documents separated into three categories based on head of the department who is responsible for those documents as shown in table 4.1.

Table 4.1: Document type with identification code separated by category

Category	Document Type	Identification Code
General Management 1	Archive	AA
	Human Resources	AB
	Quality Assurance	AC
	Parcel	AD
	IT/KM	AE
General Management 2	Plan and Risk Management	BA
	Accounting	BB
	Research	BC
	Academic Management	BD
Academic	Academic Administration	CA
	Student Affairs	CB

To generate an identification number for a document, it has to confine to the following format: $TTYYYYDDDD$. Where TT is the identification code from table 4.1. $YYYY$ is a four-digit such as 2016 or 2015. $DDDD$ is a four-digit running number. It starts at one and must increment by one each time as the new document is created. If $DDDD$ is less than one thousand, it must fill with leading zeros until there are four digits presents. If the number is greater than 9999 which is a maximum limit, it has to

continue counting normally but without any leading zeros. Although, having the same type of document more than ten thousand documents has never happened to IC before. If new year has started, the $DDDD$ must reset back to one (0001).

To clarify how the identification number is generated and combined together, assume that IC already have two documents of type "Academic Administration"(CA). Mr. X would like to take a vacation on May, 2016. So he requests an "Absence Form" and put in an information. Then, he submits the form to the system. The absence form is a part of academic administration so TT is "CA". The document is created at 2016 so $YYYY$ is "2016". Because there are two documents already in the system. The next number is three so $DDDD$ is "0003". Combining TT, $YYYY$, and $DDDD$ together yields CA20160003.

4.2.2 Case Study

The figure 4.2 shows a case study of IC's documents in 2015. The case study is divided into three cases: 1. storing digital documents; 2. retrieving digital documents; 3. tracking digital documents. Each case comprises of four types of documents; 1. absence form; 2. student internship form; 3. conference outside KMITL form; 4. external document. They come from the scope stated in section 1.4.

These mentioned forms emerge from a specific type of IC's official documents. External digital documents are unofficial documents because IC isn't the one who issue the documents. Thus, figure 4.2 can be rewritten for simplicity as shown in figure 4.3.

The workflow of each type of forms are represented as an activity digram shown in figure 4.4, 4.5, and 4.6. The common operation that appears on all diagram is that someone has to either approve or reject the document. User is the person who initiate the document, filling forms, and submit to the staff for approval. "Absence form" and "Conference outside KMITL form" requires user, staff, and dean to sign the document. If someone reject the document, user has to be notified about the rejection. The final process is to "finalize document" as shown in figure 4.7. Staff has to upload attachments as an evidence acquired from scanning to the system. Then, the system will notify user that the submitted document has completed successfully.

4.3 Requirement Summary

This section summarizes software requirements into itemized lists. There are two subsections, user requirement and system requirement.

Figure 4.2: Case study of this project

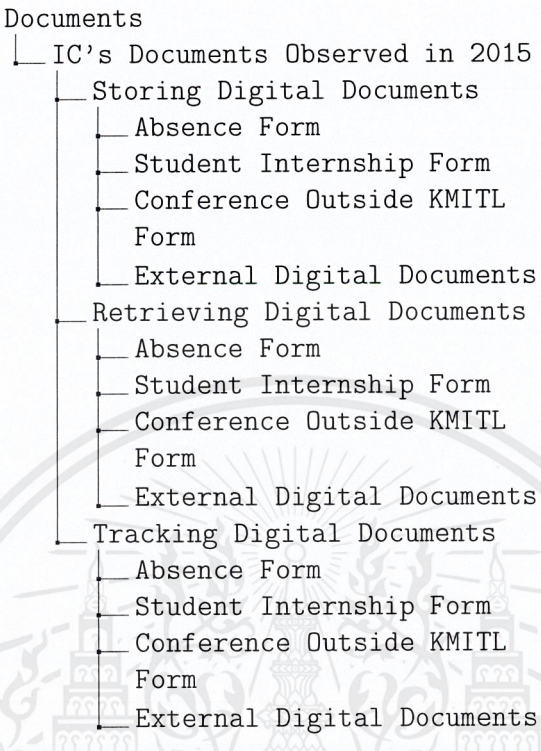
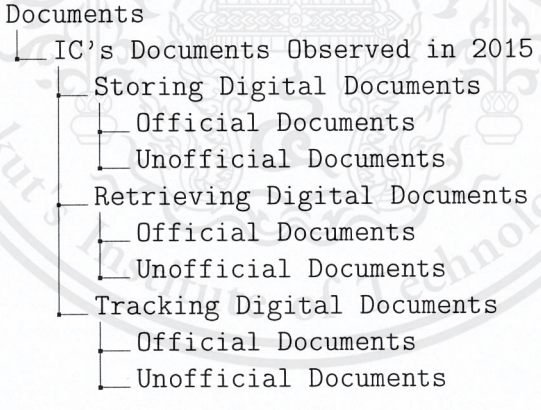


Figure 4.3: Simpler version of figure 4.2



4.3.1 User Requirement

User requirement is a goal or task that specific classes of users must be able to perform with a system, or a desired product attribute [23].

1. User shall be able to includes digital attachment files to the document as it states.
2. Staff shall have read-access to documents.
3. Staff shall be able to print out paper documents from digital documents including attachments.
4. Any IC personnel who have a write-access to official documents shall have read-access to all previous versions of those documents.
5. User shall be able to search their documents by the following criteria:

   A document identification code given by the system.

   A Form's name.

   A date when the document is created and modified.

   A Document's status indicates whether the document is rejected, pending, or completed successfully.

   Any mentioned criteria can be mixed together to create more complex search queries.
6. User should be notified when their documents are rejected or completed successfully. The notification only happens inside the system. Meaning that user require to access the system in order to get notified. The system notify user by updating document's status to a new one.
7. Staff should be notified when there is a new pending document forwarded to them. The notification method is the same as 6.

4.3.2 System Requirement

System requirement is a top-level requirement for a product that contains multiple subsystems, which could be all software or software and hardware [23].

1. The system shall operate as a web application hosted on a web server so that all users can access it in one place.
2. The system shall not disclose internal workflow to any user for all associated documents except for an administrator.
3. The system shall assign an identification code to official documents in the format $TTYYYYDDDD$.

4. The system shall contain the REST API so that it can communicate with front-end of this system. The API shall not be publicly available. The API shall be in URI "/api/document/<interface >/" where <interface >is the name of the interface. URI, input parameter, output, and description are shown in table 4.2.

Table 4.2: A list of API of this system

URI	Parameter(s)	Return Value (JSON)	Description
/api/document/read	User ID, Document ID	A document's metadata on a given ID.	Query a document.
/api/document/ref	User ID, Document ID	All previous version of the document	Query all previous version of the document on a given ID.
/api/document/attach	User ID, Document ID	Document's attachments	Query all document's attachment files on a given ID.
/api/document/attach	User ID	Document's attachments	Query all user's attachment files in all user's documents.
/api/document/delete	User ID, Document ID	HTTP response	Delete user's document on a given ID. The system must not delete other documents owned by a different user.

4.3.3 Quality Attribute

Quality attribute is a kind of non-functional requirement that describes a service or performance characteristic of a product [23].

1. The system shall encrypt all documents and attachment files.
2. The system shall use HTTPS communication protocol to exchange data between a web server and a client.
3. A user interface should have an English localization available.
4. A user interface should be intuitive enough that user don't have to rely on a user manual. Meaning that user doesn't have to go through many processes to achieve their tasks. Big buttons and large self-explanatory texts are also preferable.

Figure 4.4: An activity diagram for "Absence form"

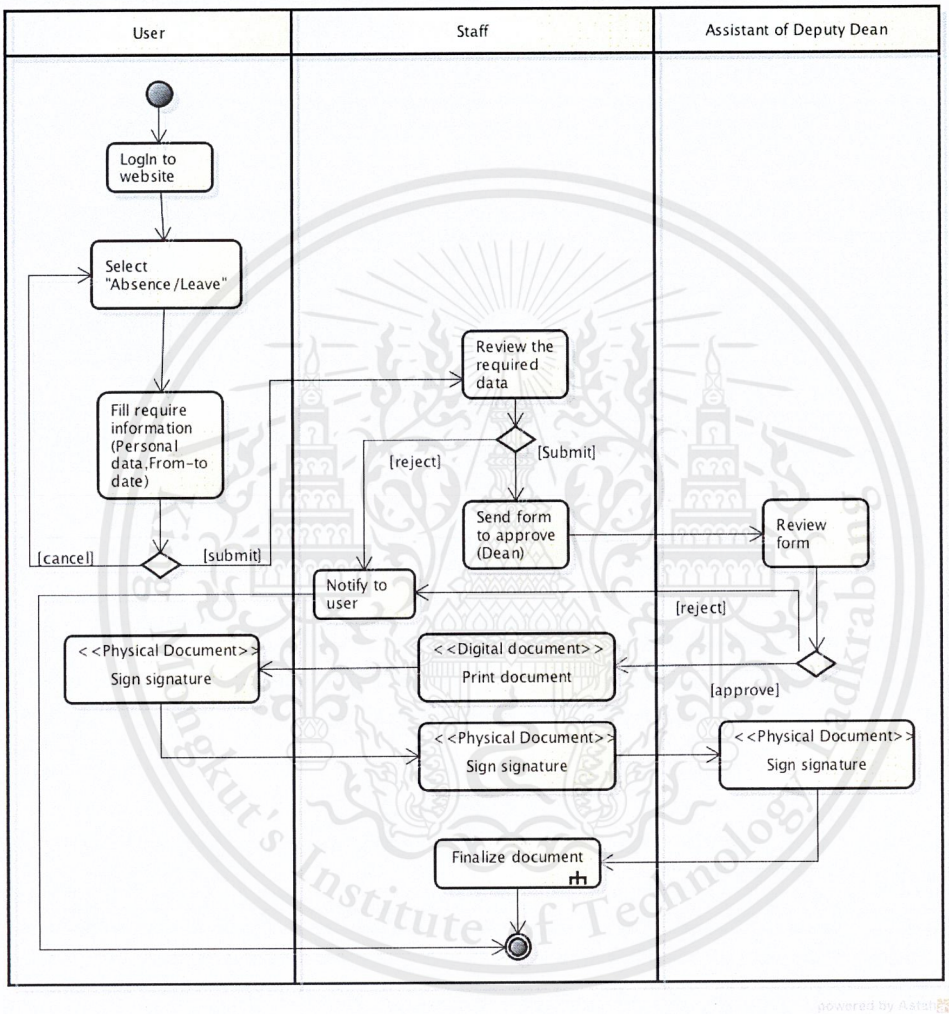


Figure 4.5: An activity diagram for "Student internship form"

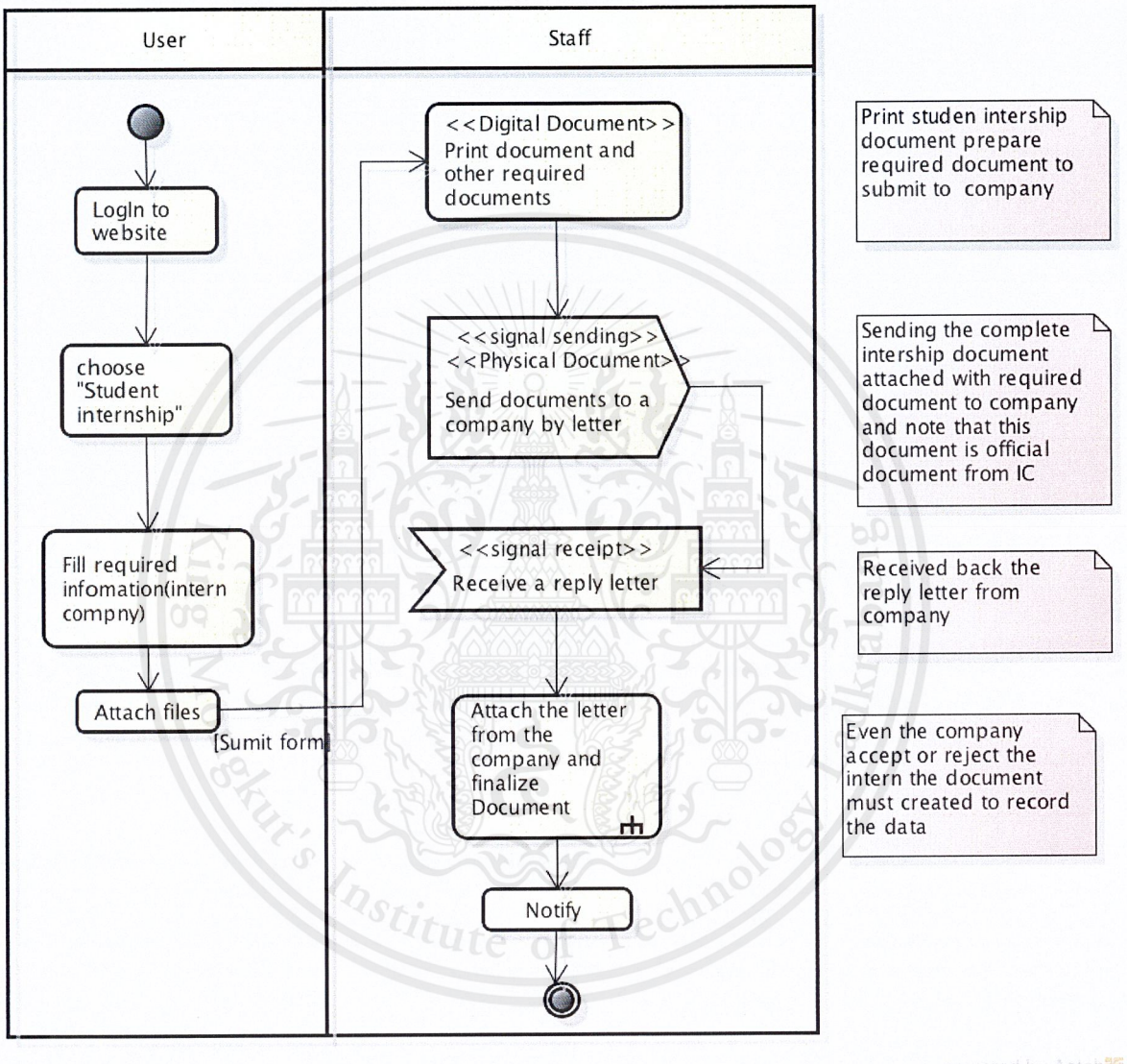


Figure 4.6: An activity diagram for "Conference outside KMITL form"

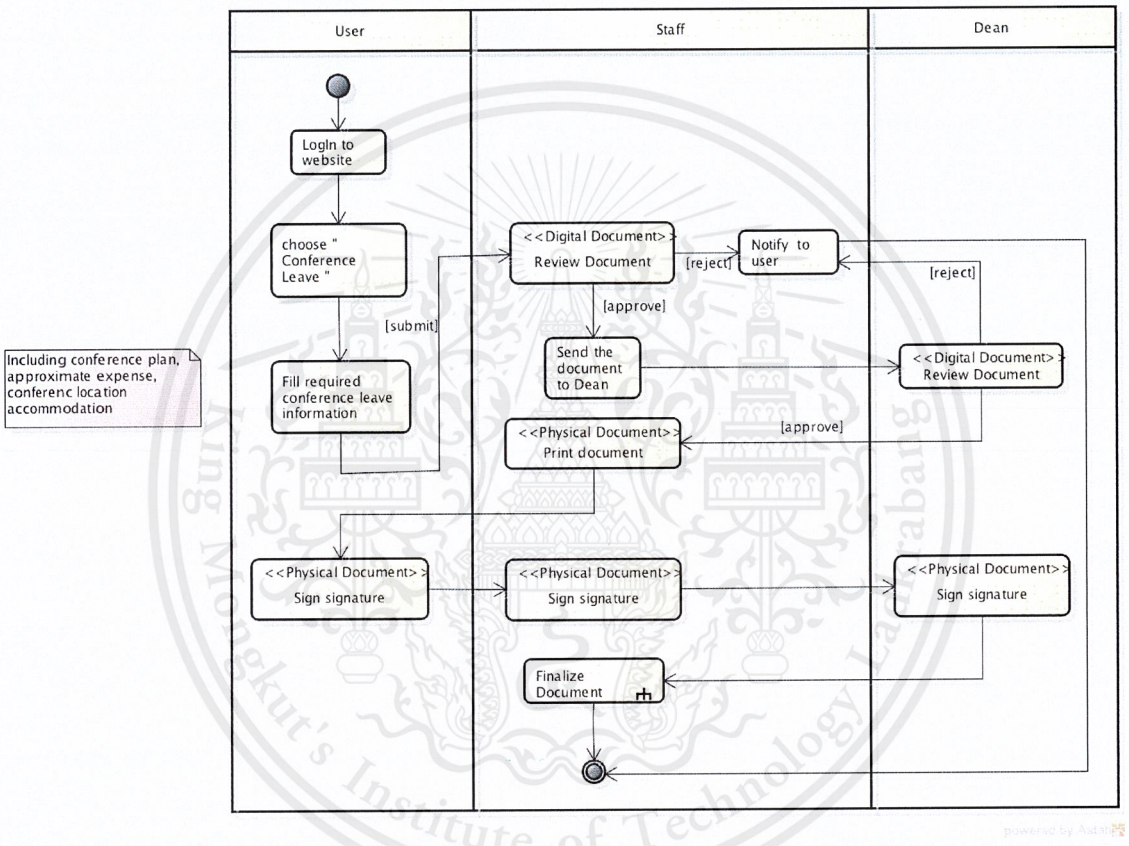
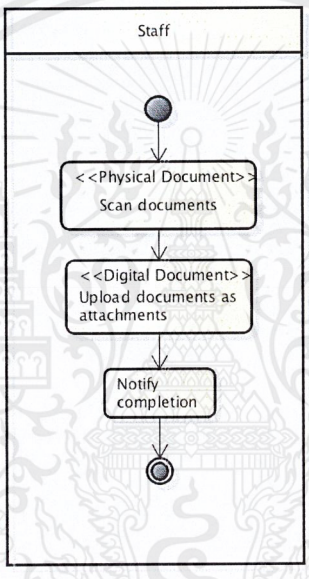


Figure 4.7: A sub activity diagram of task "Finalize document"



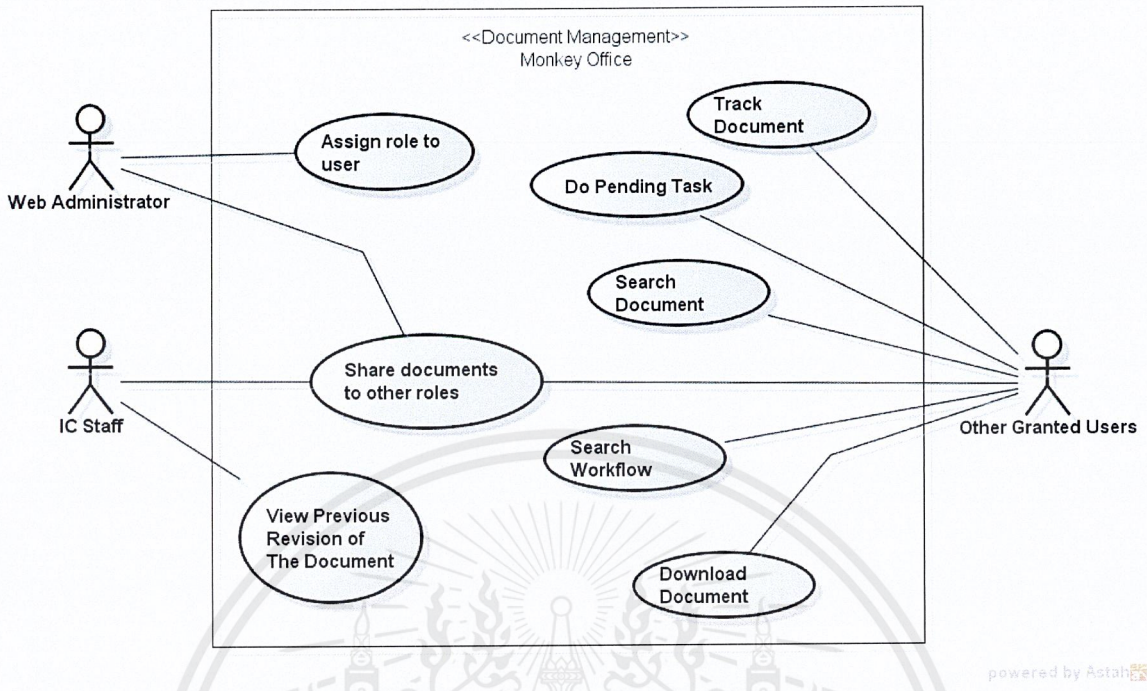


Figure 4.8: A usecase diagram for Monkey Office

4.4 Use Case

Figure 4.8 captures system functionalities and requirements represented as Universal Modelling Language (UML) use case diagram. Table 4.3 describes who these type of users are. From table 4.4 to table 4.11 elaborates each use cases in greater detail.

Table 4.3: Type of user and description

Actor	Description
Web administrator	A website maintainer who has privilege to grant access to user by providing a username and password. They can see what is going on within the system. For example, they can access any documents and workflows. They know many workflow is executing by who and when. A web adminsitrator can also manage user within the system such as create, update, delete, and assign roles to a user
IC staff	IC's employees who is in IC's organization structure from figure 4.1
Other granted users	Users who is neither all of the above, but got access permission from the web administrator. They can be students or employees from a different department of KMITL

Table 4.4: Use case: Assign role to user

Use Case Name	Assign role to user
Description	Assigning roles of IC such as student or staff to the user.
Primary Actor	Web administrator
Pre-condition	1. Have already logged in to the system. 2. There is at least one user registers to the system.
Post-condition	The system assigns a role to user and notify him/her.
Trigger	1. A user has registered to the system. 2. A user request web administrator to change a role.
Basic flow	1. On the top menu-bar, click on the "Admin" tab. 2. Navigate to "Settings" and click on the "Role Management" 3. Click on the desired role and select which user to be assigned to. 4. Click "save" button.

Table 4.5: Use case: Share documents to other roles

Use Case Name	Share documents to other roles
Description	Let a group of users categorized by roles to have read-access to the document
Primary Actor	Web administrator, IC staff, and other granted users
Pre-condition	1. Have already logged in to the system. 2. There are at least two users in the system. 3. There is at least one document owned by a user. 4. The document has to complete all tasks defined by a workflow successfully.
Post-condition	User(s) who are selected by the document's owner have read-access to the designated owner's document
Trigger	This use case can happen at any time
Basic flow	1. On "My Documents" tab, click the "Edit" button located on the "visibility" column. 2. Tick a checkbox on a desired role to be able to see the document. 3. Click "save" button.

Table 4.6: Use case: View previous revision of the document

Use Case Name	View previous revision of the document
Description	Find out more information about the previous revision of the same document
Primary Actor	IC staff
Pre-condition	1. Have already logged in to the system. 2. At least two documents are already exist in the system.
Post-condition	The highlights differences such as file name, author, or attachments between the previous revision and the current revision
Trigger	This use case can happen at any time
Basic flow	1. On "My Documents" tab, locate the document and click on "More Details" button 2. Click "Compare with previous revision" button.

Table 4.7: Use case: Track document

Use Case Name	Track document
Description	Track a document executing in a workflow
Primary Actor	Other granted users
Pre-condition	1. Have already logged in to the system 2. User executes a workflow that is going to produce at least one document.
Post-condition	The system display an executing task defined by the workflow
Trigger	The document is forward to the next task in the workflow
Basic flow	1. On "My Documents" tab at "status" column, click on the status (created, in progress, or done). 2. The system displays the entire workflow and highlights the executing task in which will produce the document.

Table 4.8: Use case: Do Pending Task

Use Case Name	Do pending task
Description	Given a task from the workflow, a user has to perform an action upon the task as instructed by the workflow such as upload attachment(s), fill a form, or sign a document
Primary Actor	Granted users
Pre-condition	1. Have already logged in to the system 2. User executes at least one workflow and the workflow is not complete its process yet
Post-condition	1. The task disappear from the view 2. The task is carried out to the next task in the workflow
Trigger	The task requires user to take an action
Basic flow	1. On "Pending tasks" tab, select a desired task and click on "More detail" 2. The system show instructions on what user has to do with the task. User has to perform accordingly. 3. Click "submit" to complete the task

Table 4.9: Use case: Search Document

Use Case Name	Search Document
Description	Ask the system to find documents.
Primary Actor	Granted user
Pre-condition	Have already logged in to the system
Post-condition	The system lists documents according to search query
Trigger	This use case can happen at any time
Basic flow	1. Enter the document's identification code or the name of the form. 2. Adding specific searching criteria such as date creation or document's status. 3. Click "Search".

Table 4.10: Use case: Search Workflow

Use Case Name	Search Workflow
Description	Find workflow(s) that user has executed
Primary Actor	Granted user
Pre-condition	1. Have already logged in to the system 2. User has executed at least on workflow
Post-condition	The system lists workflows according to search query
Trigger	This use case can happen at any time
Basic flow	1. Enter the name of the workflow. 2. Select workflow's execution status 3. Enter a sentence, word, or phrase to look for inside the task (optional) 4. Specify an execution date (optional) 5. Click "Search".

Table 4.11: Use case: Download document

Use Case Name	Download document
Description	Retrieve an electronic document file
Primary Actor	Other granted user
Pre-condition	1. Have already logged in to the system 2. User owns at least one document in his/her account
Post-condition	User retrieve the file from the system to his/her computer
Trigger	This use case can happen at any time
Basic flow	1. On "My Documents" tab, locate the desired document and click on a hyperlink at "Document name" column 2. Wait until the download completes

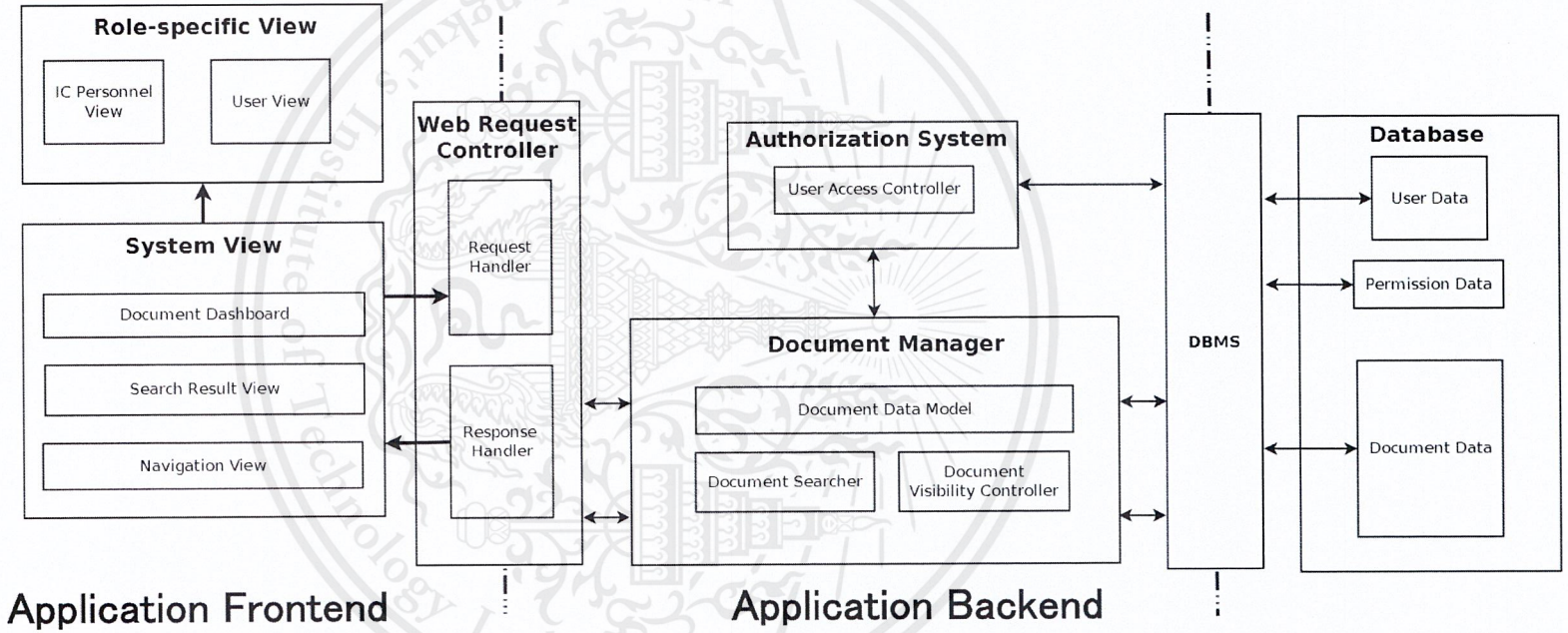
Chapter 5

Software Design

5.1 Architecture

Monkey office is a web application deployed on a web server. The application has two realms, front-end and back-end. The front-end interacts with users directly though provided interfaces. The system serves HyperText Markup Language (HTML) web pages to a client. The client who access the system by a HTTP request with their web browser will receive a web page as a response. The back-end is where the core of the system resides. Because the system needs permanent storage to store data on request. It needs to communicate with the database to store and retrieve data. Figure 5.1 shows overall subsystems and modules of the application.

Figure 5.1: Architecture of this system



5.1.1 Frontend

Frontend consists of user interfaces called view. It renders data received from application backend to users. User can only interacts with the system through views only. There are two types of view, system view and role-specific view. System view is a base view available for all type of users. It provides basic interaction functionality to the system such as search for documents, view documents, and navigating around the site. Role-specific view extends the system view with some extra functionalities available only to specific type of users. Those functionalities has already mentioned in section 4.4. IC personnel view available to staff, assistant of deputy dean, deputy dean, and dean only. User view available to user only. Table 5.1 shows all views in the system and their availability.

Table 5.1: System views and their availability

View	Availability			Description
	Web Administrator	Staff	Other Granted Users	
Document Dashboard	✓	✓	✓	View lists of documents
Search Result	✓	✓	✓	View search result
Navigation	✓	✓	✓	Menu on the top to navigate around the site
Document Visibility Settings	✓	✓		Configure which role(s) can access the document
Document Deatils	✓	✓	✓	View more information of the document
Role Management	✓			Assign, create, delete user's role(s)

Figure 5.2, 5.3, and 5.4 shows a mock-up for system view. The view name which has dark grey italic text indicates where its placement position in the web page.

Figure 5.2: Mockup for system view. This figure shows two views: 1. Navigation View. 2. Documennt Dashboard View. 3. Search Result View.

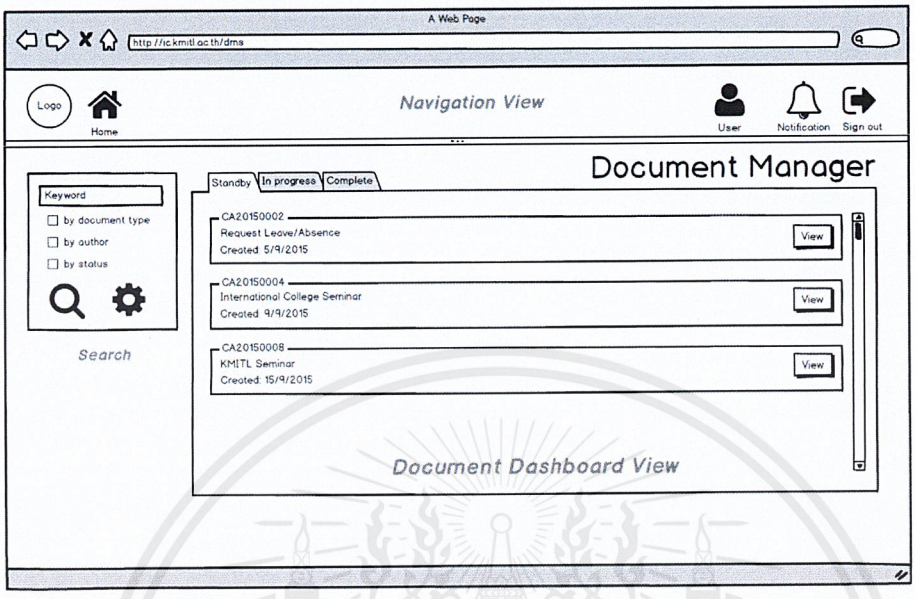


Figure 5.3: Mockup for search result view after performing a search operation

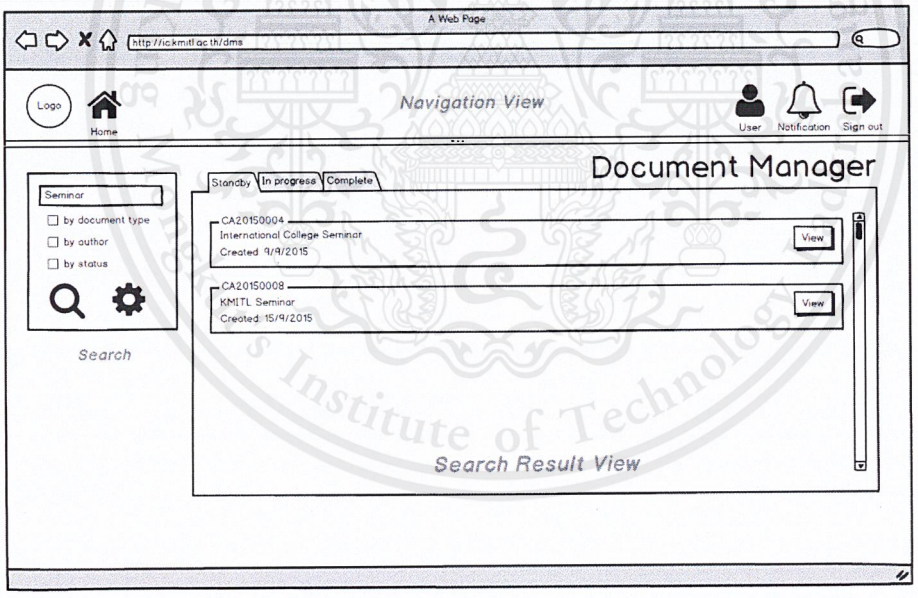
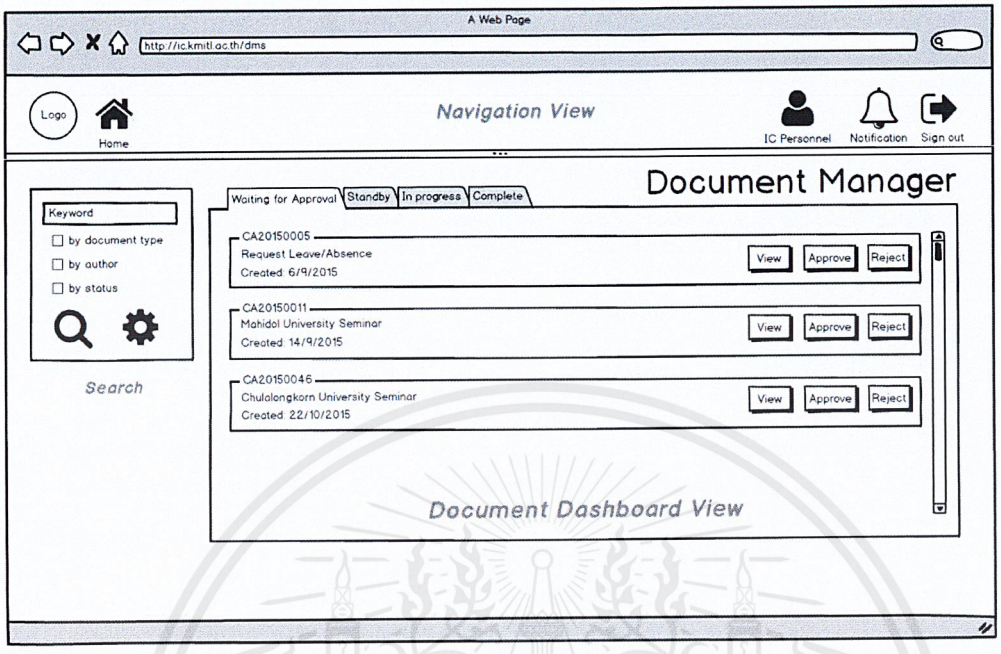


Figure 5.4: An extended view for IC personnel users



5.1.2 Backend

Application backend is the core of the system. It connects between frontend and a database. It processes input sent from the frontend and sends a response back for further processing such as displaying data to a user. There are two systems working on the backend, authorization system and document manager. The authorization system decides which user have authority to access and make changes to the document. It uses user data and permission data which are stored inside the database to make a decision. The document manager takes care a creation and a modification processes of documents. It has three subsystems:

5.1.3 Controller

There are two controllers placed between the frontend and the backend to control flow of data. Web request controller handles HTTP requests sent by the user through a system view and response sent by the system. It delegates requests to a designated system to process requests. Another controller located between the backend and the database is Database Management System (DBMS). It handles communication between the database and the application. It provides various functionalities to manage the database such as defining data and performing CRUD operations.

Subsystem	Description
Document Data Model	Define how each type of document should be created including assigning an identification code
Document Searcher	Manipulate document search results according to a user given query such as document name, document type, and author name
Visibility controller	Controls which user can access the document when a document owner reveal his/her document

5.2 Storing Digital Documents

Storing documents mean saving documents uploaded by users and create a metadata so that the system can access them. The purpose of storing metadata is to create an abstract concept of the document. According to requirements and use cases, the document should contain attachments and other related documents. The document can be official or unofficial documents. Both official and unofficial documents are the same except for one characteristic. Official documents has a customized identification code as mentioned in table 4.1. Unofficial documents has only regular identification number starting at one and increment by one for each document. Therefore, the parent document have to store a list of attachments. That is a list referring to itself. A class diagram in figure 5.5 shows relationships between each unit.

Document class represents a document. Table 5.2 describes what each attribute is, its data type, and its functionality from figure 5.5.

Table 5.2: Description of each attribute in *Document* class

Attribute	Data type	Description
id	String	Document identification
owner	String	The original user who create the document.
dateCreate	Date	A date when this record is created.

Continued on next page

Attribute	Data type	Description
name	String	The filename including extension that is going to appear on the page. It is the original filename that user uploaded. If it is a by-product from the workflow, the system automatically generates it.
is_auto_generate	Boolean	A boolean indicates whether this document is created by the system.
status	String	Indicates the current states document going through the workflow.
filepath	String	Absolute path to the file stored in a server.
relate2Docs	array of *Document*	Additional documents that involved with this document. If this document refer to other documents, they will be shown here.
attachments	array of *Document*	Other required documents or other dependent documents.

A schema has to be defined first in order to record document's metadata to the database. The schema relies on a *mongoose.Schema* package. It provides utility for defining fields and data types suitable for saving to the database. As the requirement states that attachments are also document. *relate2docs* and *attachments* are fields that refer to additional documents. They should have *Document* data type.

Document class also defines methods to manipulate the schema which are its attributes. Table 5.3 describes each method in detail.

Table 5.3: *Document* methods

Method	Return data type	Description
created		Set document's status for newly created document.
inProgress		Set document's status as processing.
done		Set document's status as done.

Continued on next page

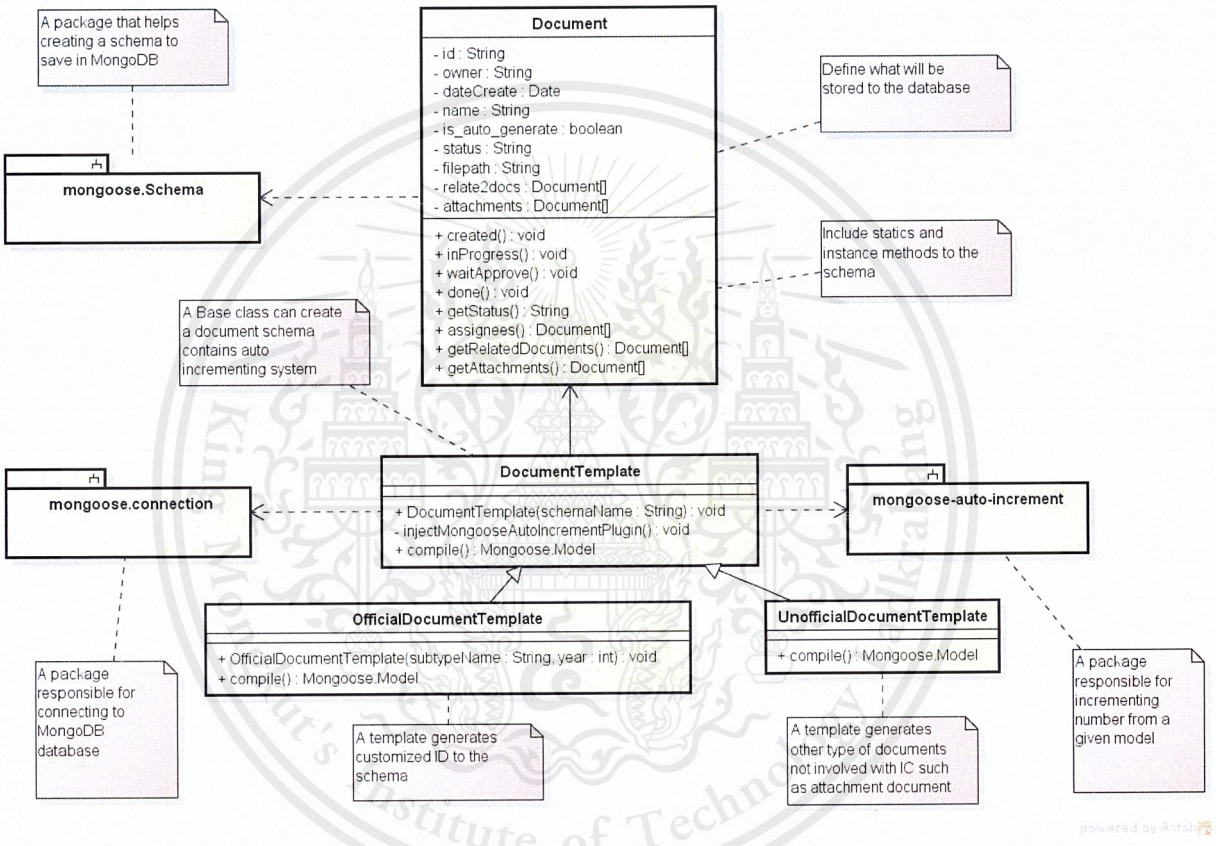
Method	Return data type	Description
waitApprove		Set document's status as waiting approval
getStatus	String	Get the current status on this document.
getRelatedDocuments	array of *Document*	Get all related documents.
getAttachments	array of *Document*	Get all attachments.

DocumentTemplate applies auto increment functionality to the schema. Every time the schema is saved, it invokes the auto increment functionality automatically. In design pattern, *DocumentTemplate* is a factory that creates instance object according to given parameters. There are two subclasses inherits *DocumentTemplate*, *OfficialDocumentTemplate* and *UnOfficialDocumentTemplate*. *OfficialDocumentTemplate* creates a document template based on a given type name and year. They are used to create a document with custom identification code as stated in table 4.1. The purpose of *UnOfficialDocumentTemplate* is to create external documents and attachments. It has the same functionality as *DocumentTemplate*. Table 5.4 explains what each method does.

Table 5.4: *DocumentTemplate* description

Method	Return data type	Description
OfficialDocumentTemplate		A class constructor that receives document type and year to generate an identification code.
injectMongooseAutoIncrementPlugin		Inject *mongoose-auto-increment* package to the schema as a plugin so that the schema can increment the number every time before saving the database.
compile	*mongoose.model*	Create a model from a schema because the *mongoose.Schema* package requires *mongoose.connection* in order to prepare the schema suitable for saving to the database.

Figure 5.5: Document template creation class diagram



5.3 Retrieving Digital Documents

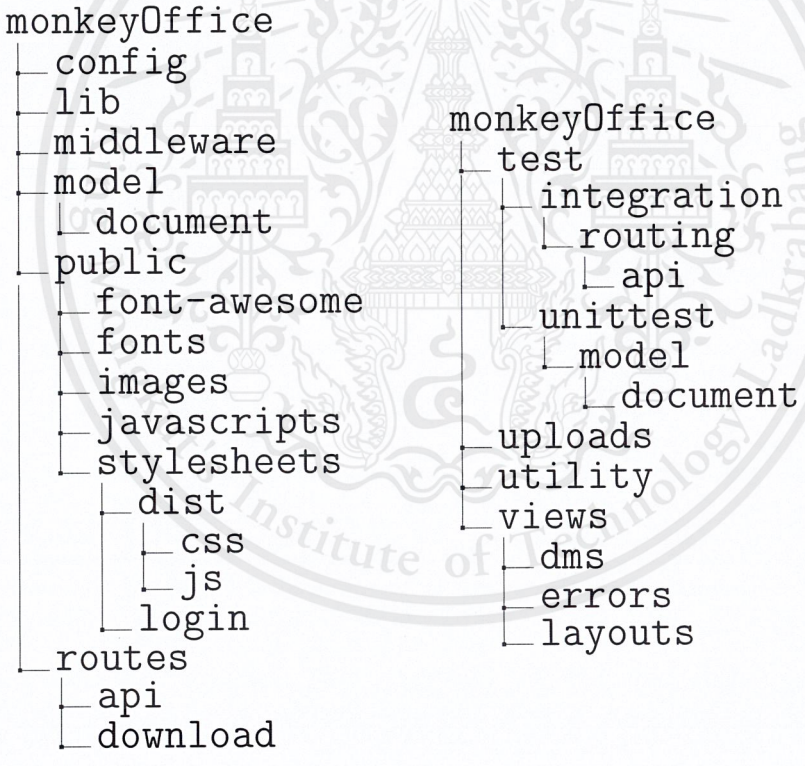
Retrieving digital documents means querying recorded metadata from the database. Referring to figure 5.5, the *id* field in *Document* class is a field with unique value. The system can query a specific document using the *id* field. The system can also use *owner* field to search documents owned by a specific user. Fortunately, *mongoose.Model* package encapsulates database layer. Each record in the database is treated as an object. After querying from the database, it assigns results to each field and instantiate an object from the *Document* class automatically.

Chapter 6

Results

6.1 Source Code Structure

Figure 6.1: Source code structure of this project.



From 6.2, at first time user have click the sign up link to create a new account for accessing into the main system which is another page 6.3

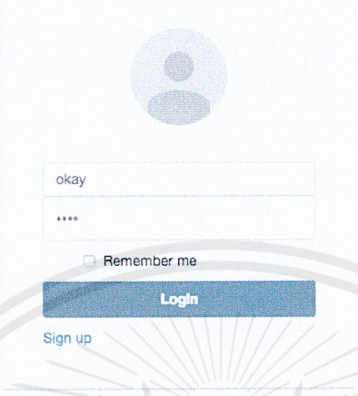


Figure 6.2: Log in page on website

After that when the user logged in user can see the tabs of documents, workflows, pending tasks and shared document. On my document tab, user can click view detail of document to get more information of document file and click to download the file on link name that user selected, user can searching the documents by name of document, author name and type of document. Moreover, user can edit the permission of each user's document 6.4, 6.5 and 6.6

On workflow tab user can search workflow by name, status, any content in workflow and execution date, user also can click to look more detail to go the workflow page that user selected file and pending task tab user can do only look more detail to do the task following workflow rule. 6.7, 6.8 and 6.9

At last, shared document tab is different from other because user can do only click to download file in any type of attach files or document files. 6.10

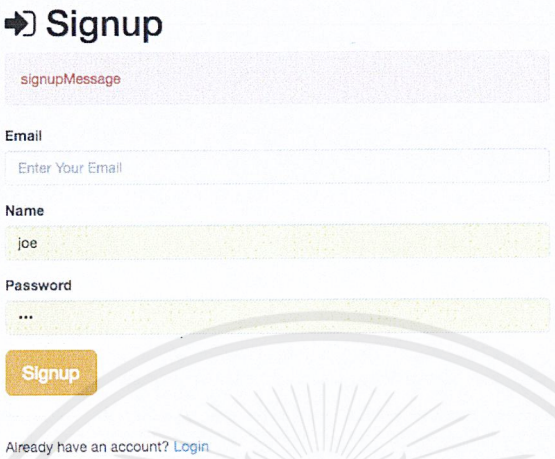


Figure 6.3: Sign up page on website

Figure 6.4: Home page on website

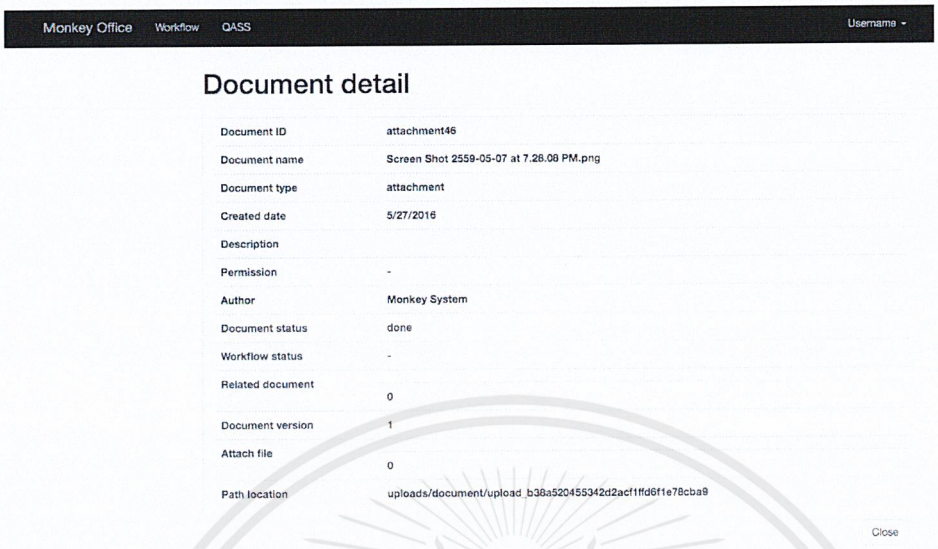


Figure 6.5: Document detail page on website

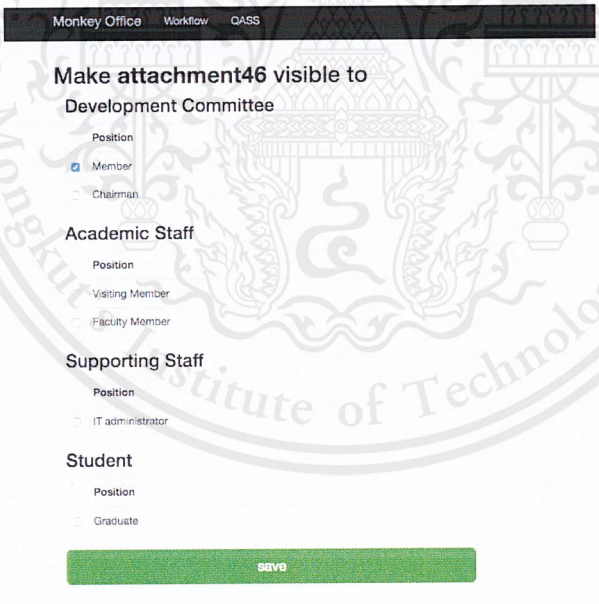


Figure 6.6: Permission page on website

Figure 6.7: Tab workflow in home page

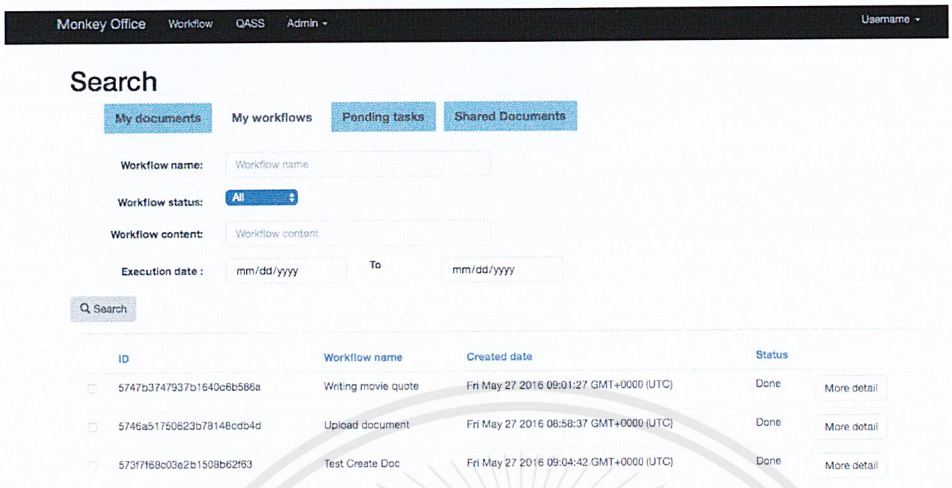


Figure 6.8: Workflow detail page on website

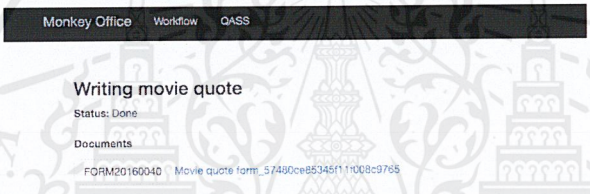


Figure 6.9: Tab pending task in home page

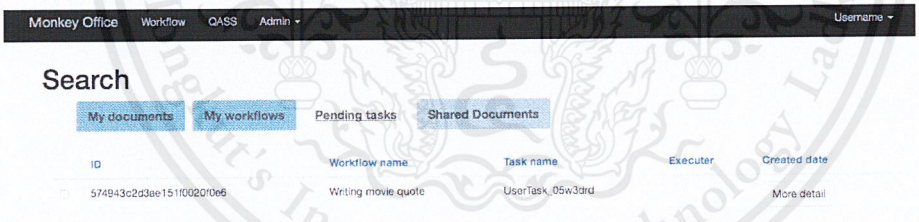
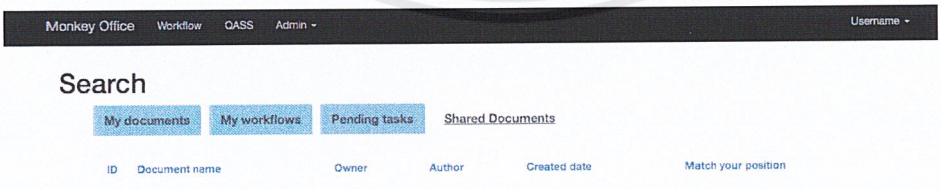


Figure 6.10: Tab shared document in home page



Chapter 7

Conclusion

In conclusion, this thesis is about the Document Management System (DMS), which is a system proposed for the staffs and the lecturers in International College(IC), KMITL, to work together efficiently. Problems occur at the present, managing documentation's operation in the organization, is mostly manual and uses physical document to work on. When people keep their files in their own places. The problems include as following,

1. The waste of time when a person needs some information from other people and has to meet them directly to get the information he want.
2. The waste of time and manpower when people have to search for specific documents from many unrelated documents.
3. The difficulty of document tracking, in which people have to know others' progress and where they keep their works.
4. The complexity of workflows, in which people have to work together, follow specific sequences as well as deal with different places and different type of documents.

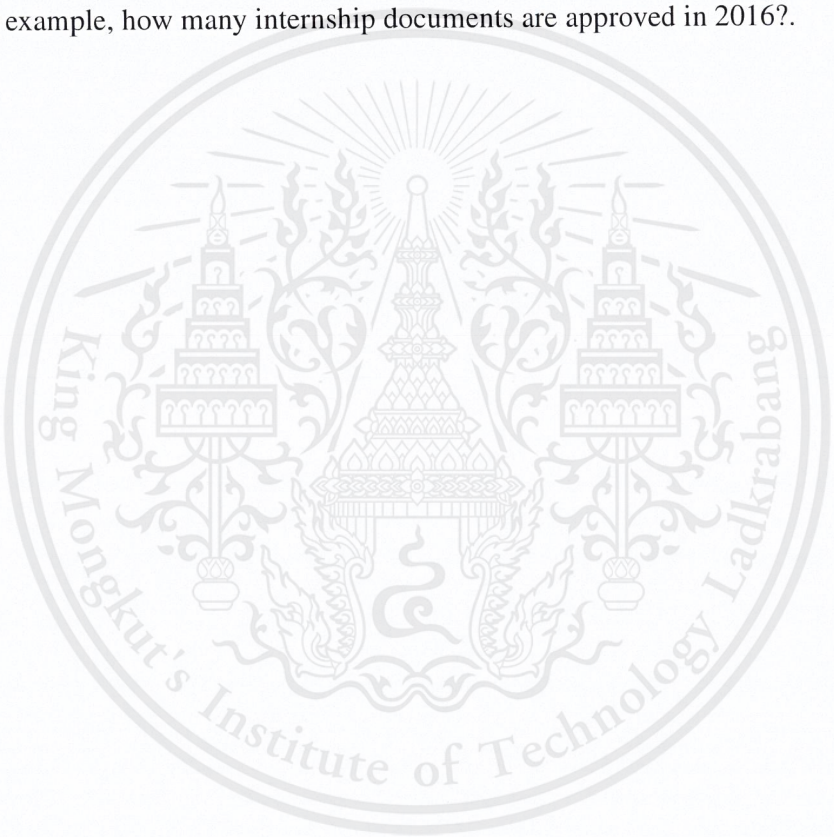
Monkey Office is the system to manage all documents in IC KMITL. It is created for solving all above problems. In the system, there are three parts which are the following,

1. QUALITY ASSURANCE SYSTEM(QA) is the systematic review of educational programmes to ensure that acceptable standards of education, scholarships and infrastructure are being maintained.
2. WEB-BASE WORKFLOW MANAGEMENT SYSTEM(WF) is the system that help people can track the documents as they follow a workflow or, even better, a system which could help automate entire workflows.

3. DOCUMENT MANAGEMENT SYSTEM(DMS) is helping to manage, store and retrieve documents.

This thesis is the last one, DMS. IC's staffs and lecturers can use it to manage, store, and retrieve the documents conveniently. DMS solves the problems by being a center of information. It lets people share their files. Moreover, users can search the documents: 1) document name, 2) document's author, 3) status and 4) created date. Furthermore, each public file, they can see all above details and the following details which are related works, document's version, attach files, and all description of the document.

Further development, DMS will available for other users and students can access to their course materials and other information online such as internship, announcement, and time table. Besides, the system can analyze all documents, then convert it to visual statistic. For example, how many internship documents are approved in 2016?.



Appendices

Appendix A

Project plan

The project starts at August 18th, 2015. We expect to deliver it on March 25th, 2016. For our project planing, we divide our work into 4 phases.

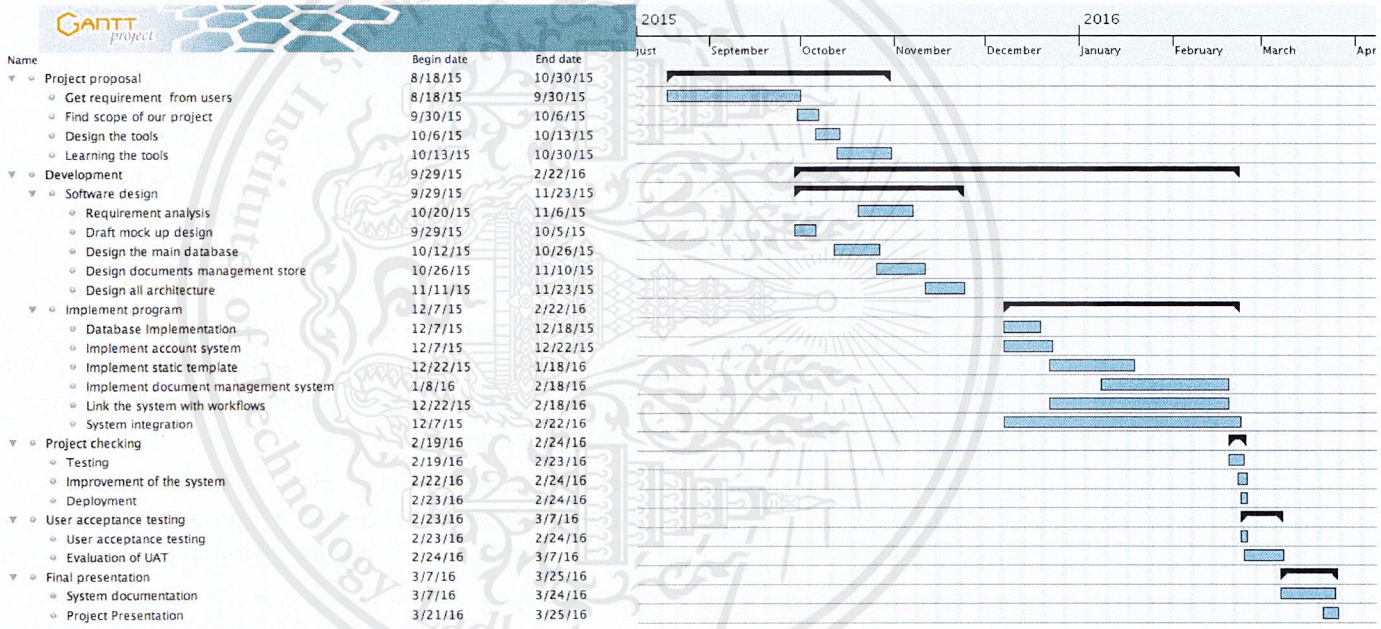
1. Plan and research (18/08/15 – 30/10/15)
   The first phase was to gather user's requirements by interviewing. We interviewed with IC's vice dean because he is a client who came up with this project. Then, we moved on to interviewing IC's employees about document related problem they encountered. Next, we discussed about software tools that solve the problem.

2. Design and architecture (29/09/15 – 23/11/15)
   On the second phase, we designed the software architecture. The architecture is the core of how software must behave, also to get an overview of system interaction. Then, we designed a Graphical User Interface (GUI) of this system.

3. System implementation (7/12/15 – 22/02/16)
   This phase began writing source codes based on user requirements, designed architecture, and technical specification. Later on, we connected each system's components together.

4. Testing (19/02/16 – 07/03/16)
   This phase ensured that systems ran correctly with user and system requirements. We also conduct user acceptance testing. Then, We evaluated testing result. If the result were satisfying, we would deploy the system to IC's server.

Figure A.1: Project's schedule shown as Gantt chart



Acronyms

API Aplication Programming Interface. 25, 46

BPMN Business Process Modeling Notation. 26–35

BSON Binary JSON. 25

DIP Document Image Processing. 15

DMS Document Management System. 14–16, 21

ECM Enterprise Content Management. 15

EDMS Electronics Document Management System. 15

ERMS Electronic Record Management Systems. 15

GUI Graphical User Interface. 69

HTML HyperText Markup Language. 54

HTTP Hypertext Transfer Protocol. 25, 46, 54

IC International College. 1–5, 21, 37–44, 54, 69, 70

JSON JavaScript Object Notation. 22, 25

KMITL King Mongkut's Institute of Technology Ladkrabang. 1, 4, 37, 41, 43, 44, 50

NoSQL Not only SQL. 19–21, 23–25

OCR Optical Character Recognition. 9

PDF Portable Document Format. 3, 18–20, 22

PNG Portable Network Graphics. 18, 20

RDBMS Realtional Database Management System. 19–21

REST REpresentational State Transfer. 25

SQL Structured Query Language. 19, 20

UML Universal Modelling Language. 22, 54

XML EXtensible Markup Language. 25

XMP Adobes Extensible Metadata Platform. 22, 23

YAML Yet Another Multicolumn Layout. 25

Bibliography

[1] Wil van der Aalst and Kees Max van Hee. *Workflow management.* MIT Press, 2002.

[2] Azad Adam. *Implementing electronic document and record management systems.* Auerbach Publications, 2008.

[3] Adobe Systems Incorporated. *Extensible Metadata Platform (XMP).* URL: `http://www.adobe.com/products/xmp.html` (visited on 12/21/2015).

[4] Shazia Akram. "Document Image Processing-A Review". In: ().

[5] Alfresco. *Alfresco 5.0 - Tour of Alfresco.* Dec. 4, 2014. URL: `https://youtu.be/F-HxaQ91lkk?t=17` (visited on 06/11/2015).

[6] Murtha Baca. *Introduction to metadata.* Getty Research Institute, 2008.

[7] John Brady. *The craft of interviewing.* Vintage, 2011.

[8] Capterra. Oct. 16, 2014. URL: `http://cdn0.capterra-static.com/screenshots/2050942/5376.png` (visited on 06/11/2015).

[9] Fernando Doglio. *Pro REST API Development with Node.js*, p. 3.

[10] Adam Fowler. *NoSQL for dummies.*

[11] M Gall Gall. "J. & Borg, W.(2003)". In: *Educational research: An introduction* 7 (2003).

[12] The American Society for Indexing. *What is Indexing.*

[13] Stefan Jablonski and Christoph Bussler. *Workflow Management: Modeling Concepts, Architecture and Implementation.* International Thomson Computer Press, 1996.

[14] Jaratrawee Deekhueankhan Kananek Atichartwiwat Kiattisak Kesornbua. "WEB-BASED AUTOMATIC FORM GENERATED SYSTEM". International College, King Monkut's Institute of Technology Ladkrabang, 2015.

[15] Mark Masse. *REST API design rulebook.* O'Reilly, 2012, pp. 5, 6.

[16] Molly K. McLaughlin. Sept. 10, 2015. URL: http://www7.pcmag.com/media/images/478079-efilecabinet-store-docs.jpg?thumb=y (visited on 06/11/2015).

[17] Misja.com. *Epoch Converter, Unix Timestamp Converter.* URL: http://www.epochconverter.com/ (visited on 12/22/2015).

[18] Object Management Group, Inc. *Business Process Model and Notation (BPMN).* Jan. 2011. URL: http://www.omg.org/cgi-bin/doc?formal/11-01-03.pdf.

[19] Object Management Group, Inc. *Object Management Group Business Process Model and Notation.* Nov. 2015. URL: http://www.bpmn.org/.

[20] Softpedia. *Contentverse - The main window of Contentverse enables you to view, organize and manage all your documents.* Nov. 2, 2013. URL: http://i1-win.softpedia-static.com/screenshots/Contentverse_1.png (visited on 06/11/2015).

[21] Softpicks.net. *Dokmee screenshots gallery.* Mar. 12, 2009. URL: http://img.softpicks.net/screenshots/Dokmee.jpg (visited on 06/11/2015).

[22] Dan Sullivan. *NoSQL for mere mortals.* Addison-Wesley Professional, 2015.

[23] Karl Eugene Wiegers. *Software requirements.* Microsoft Press, 2003.

[24] ZOHO. June 10, 2015. URL: https://www.zoho.com/docs/images/gallery/zoho-docs-overview-main-sm.jpg (visited on 07/11/2015).