MOBILE DOCUMENT MANAGEMENT SYSTEM

INDIVIDUAL FINAL REPORT

SUPERVISOR: DR. VINCENT LAU
SECOND EXAMINER: DR. BETA YIP
STUDENT: CHAN KA CHUN TONY (3035094830)
# Table of Contents

Purpose ................................................................................................................................. 3  
Contributions to the project .............................................................................................. 3  
Project Overview ............................................................................................................... 4  
Requirement gathering and Scope Definition .................................................................. 4  
User interface prototyping ............................................................................................... 5  
System Interaction (Employees, managers, admin).......................................................... 8  
Framework adopted for implementation ........................................................................... 9  
Add Document Page .......................................................................................................... 12  
History of Document ......................................................................................................... 15  
Admin Panel (company customization) ............................................................................. 16  
Users ................................................................................................................................... 17  
Department ....................................................................................................................... 19  
Category .............................................................................................................................. 20  
Choice of admin .................................................................................................................. 21  
Workflow function ............................................................................................................. 21  
  Functional Design ........................................................................................................... 21  
  System Paper Prototyping ............................................................................................. 22  
  Office workflow situation ............................................................................................... 24  
Calendar function .............................................................................................................. 24  
Feedback / Enquiry Page .................................................................................................... 27  
Cloud Hosting ...................................................................................................................... 30  
  Google Cloud Platform .................................................................................................. 30  
    App Engine.................................................................................................................. 31  
    Approach: Google Cloud App Engine ....................................................................... 31  
    Compute Engine ......................................................................................................... 32  
    Approach: Google Cloud Compute Engine ............................................................... 32  
    Google Compute Engine Architecture ....................................................................... 35  
Limitations and possible solutions.................................................................................... 36  
  Concurrent editing of document is not supported ......................................................... 36  
  Camera Upload is not supported ................................................................................... 36  
  Admin authorization of document upload ...................................................................... 36  
  Workflow Function deficiency ...................................................................................... 37  
Future Development ......................................................................................................... 37  
Reference ............................................................................................................................. 38
Purpose

In this report, I will describe my main contributions to our Final Year Project- Mobile Document Management System. Please refer to our group report for project background, project objectives, project details and contribution of other teammates.

Contributions to the project

In this project, a web application document management system is implemented for managing document in a company. We focus on providing a tailor-made application for SMEs and we try to suit every needs of a company, which in turn achieve streamlined business workflow, efficient document management and collaboration within a company. I have participated actively in every stage of the whole software development cycle.

For requirement gathering, I was responsible for arranging meetings with supervisor to have discussion on the mobile document management and let my team gather some basic requirements on the topic. Research has also been carried out to analyse the existing applications. We also gather the information on what is the most essential part of document management in SME. We came up with the idea that focus has to be put on “customization for company” and “enhanced workflow management”

For design, I was involved in user interface prototyping, functional design, early database schema design and system use case design. I also took part in designing situations that the business workflow definition can be applied to and I did paper prototyping to design a workflow function framework for my teammates to build on.

For implementation, I am involved in both some front-end layout implementation and some backend functions implementation. I have assisted with Felix about the business workflow layout. I am the one who do the cloud hosting and deployment. Different cloud platforms are compared. Finally, I successfully deploy the application into Google cloud (Google Cloud Compute Engine Instance) and upload document into cloud.

Meanwhile, I also participated in testing in our application, report framework formatting and I am the one who take the initiative to organize meeting with groupmates regularly and contact with our supervisor. I also took part in poster design to showcase the most valuable and unique part of our project to outsiders.

Details of contribution in each stage of software development will be discussed in depth in the upcoming sections respectively.
**Project Overview**

The following are the main functions that have been implemented in our mobile document management system. The following diagram aims at providing a better understanding of the project functions before the introduction of the upcoming sections.

![Diagram of project functions](image)

**Requirement gathering and Scope Definition**

Before the start of the project, actually all of our members have carried out research on analysing and evaluating the existing document management system. For me, I have taken into account Capterra, a website that lists the top 20 MOST POPULAR Document Management System. At last, I have gone in depth with five system, namely eFileCabinet, Alfresco One, Document5-File manager, Zoho Docs, File Manager App.

It is found out that simple document features including downloading/uploading file, searching, categorizing, permission setting are all incorporated in nowadays’ document management system. Some include more advanced functions like concurrent editing, Wi-Fi file sharing, built in highlighting functions, notification functions. Although MDMS in the market are somehow multifunctional, most of the existing applications are designed for personal use and they are hard to put into practice in a company.

Taking into account two more powerful applications: Google Drive and Dropbox, it is impossible to stand out from them in terms of functionality. To make the application suitable for SMEs, we have to solve their real need instead of just implementing a series of functions that they may not benefit from. In light of this, we have found that lack of business workflow management support is one of the big problems in SMEs. In order to make our application being unique and at the same
time useful for SME, we have set up our scope that apart from the basic document management functions, our main focus was on business workflow path definition and some tailor-made functions for companies like providing options to companies to customize their document management. In this case, it will make our application suit the needs for SMEs.

**User interface prototyping**

I have involved in user interface design prototyping at the beginning of the project. With an aim to provide users with comfortable document management experience, providing a user-friendly and straightforward interface is one of the main focuses in this project. I have used the tools proto.io to make the design as follows to serve as a start for our project. The main difficulty is that how I can make use of simple and user-friendly interface to provide an informative and comprehensive document management service to users.

For login page, “remember me” option is removed due to security concerns. For the navigation bar, the functions are more or less the same. However, for my documents, shared with me and outgoing documents, it is grouped into a document repository for simplicity and higher understandability.
For search design, it is more or less the same. For document information, the document information and permission setting are separated into two pages.
For document design and permission setting design, the final product is to combine the two screens into one, which means permission is set during the creation of document.

**Notification design**

For notification function, it remains here but it is changed to email notification instead of in-app notification. Employees in a company are supposed to check email frequently. Email notifications are the most suitable in case they do not always open the application. For calendar function, it remains the same except the appearance of the calendar.

**Calendar design**

For workflow design, the “View workflow graph” function is implemented in the same way that a graphical tree will be used to implement the relationship. However, a simple circle node is used instead of a picture. Apart from this, we use colour of node instead of wordings to show the progress.
For all these user interface prototyping, most of them are adopted and used as a framework and base for implementation. Our final product is having a similar format and layout as the prototype although it is having a different appearance in terms of colour and methods of presentation due to changing requirement alongside our development. For user interface, instead of making it very visually appealing, we adopt a simple, user-friendly and formal layout to suit with the nature of SME document management.

**System Interaction (Employees, managers, admin)**

Concerning the use case diagram above, there are mainly three actors in our mobile document management system and they are employees, managers and admin. It summarizes the interaction of each role to the system and describes how people in the SME can be benefited from our system.
Employees

- Upload documents to share with teammates or a desired user.
- View the document shared to him.
- Some documents are shared to the whole department so that all members in that department have the permission to gain access to the file.
- Concurrent editing between employees is not supported. However, employees can check out a file for modification and the file is locked until he checks in the new version. There is a versioning control and history for each document.
- Define a business workflow path and view the progress (e.g. Claim form submission, proposal submission to manager, distribution of work to teammates, setting self-deadline)
- View the calendar to keep track of the deadline
- Receive email notification
- Gain access to documents at any time anywhere

Managers

- Managers can do all things that employees can do
- Review the document uploaded by team
- Define a business workflow path and view the progress (e.g. Assigning tasks to team members with a deadline, simplifying a promotion procedure.)

Admin

- Admin can do all things that employees can do
- Authorize and reject document in the repository
- Define document management preference (e.g. Category, department, user creation and alternation)
- Monitor the system for the company

Framework adopted for implementation

In this project, our group members are of different background. Some are particularly strong in backend programming while some are relatively talented and interested in frontend interface implementation and design. Before the start of implementation, I have done a series of research on finding a particular framework to follow for better division of labour and separation of concern. In this case, Smarty is the framework that we have chosen in this project.

Php Smarty is chosen due to the following reasons:

- Smarty tags are easier to understand and customize than any PHP statement.
- Some members in our group are not too experienced with using PHP. A tag-based syntax for HTML management will be easier to read, understand and maintain.

- It maintains better separation of concern. No PHP knowledge is needed to manage smarty templates. Web developers can be responsible for HTML, CSS and JS while PHP developers can focus on implementing backend logic.

The framework involves the design pattern of Model-View-Controller. The model refers to the database’s data model. View is handled by smarty TPL, HTML and JavaScript to display the visualization of the data that the model contains. PHP serves as the role of controller to control the data flow into model object and updates the view accordingly.

For simpler functions like “Check out”, which is to allow users to download a document to local for editing and modification, html codes are embedded in the PHP file.
For complicated functions like “Add document”, which is to allow users to upload document to server, html are stored in tpl file, logic are stored in the php file. It is for easier collaboration, we can seek help from group members when we encounter problem. The part with problem can be separated and isolated out for analysis and discussion.
Add Document Page

To enter the “add document” page, what users have to do is to click the “Add Document” button in the navigation bar.

After filling in all the required fields and the submit button is pressed, document is uploaded in the form of HTTP POST Request.
Afterwards, using open source, the uploaded file is moved to the server directory for systematic storage and retrieval and saved as a .dat format. In this case, the document cannot be directly opened.

```php
$newFileName = $fileId . '.dat';
move_uploaded_file($temp_name[$count], $GLOBALS['CONFI'] ['dataDir'] . '/' . $newFileName);
```

Code from add.php
Document storage location

However, there should be a size limit for document upload. It cannot be set too low to avoid unsuccessful upload of common documents like word files, pdf files. However, the limit should not be set too large because the server has a limited size. To set the upload limit, it is done by configuring the php.ini as follows. Currently, we set the upload_max_filesize limit to 10Mb. At the same time, we have to ensure the memory_limit to be large enough.

```
; Maximum allowed size for uploaded files.
; http://php.net/upload-max-filesize
upload_max_filesize = 10M
```

Code from php.ini

```
; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 128M
```

Code from php.ini
**History of Document**

For each document, in the document repository, if users click on a document, he/she will be prompted to a page showing history of that particular document.

![Document Repository Page](image1)

![Document History Page](image2)
On the history page, latest information of the document including category, file size, creation date, owner, description, comment and revision will be provided.

For the lower part with “History” logo, it is a table storing the history( modification record) of the document. Attributes like version number, modification date, modified person and points to note are also attached. Users can check out a file to modify the content. Afterwards, he/she has to check in back the file into the system. At that moment he/she checks in, a history record will be created, showing that there are someone modifying the file since the creation of the document.

For retrieving the history of the document, referencing from opensource, it is to select all from the mdms_log database table by linking the document id to the mdms_log.id and linking the userid with id in mdms_user database table.

**Admin Panel (company customization)**

To enter the admin panel, click the “Admin” section in the navigation bar, you will be prompted to the section with all the admin privileges listed. For me, I propose and am responsible for the first three sections: Users, Department and Category. Admin has the right to add, delete, update information or display information for each of the group( Users, Department, Category ). Although I have done some research on the popular department and categories, it is better to include such flexibility so that the system can be catered to different company. In this case, unlike existing document management system or personal use, we can provide SMEs with a tailor-made office document management experience.
**Users**

Admin can create new users, register accounts for employees or create higher privilege account for seniors. For each creation, name, username, password, e-mail address have to be filled in and the new user will be assigned to a department. Special rights will be granted if specified. After creation of account, email will be sent to the user. These are database operation to add, delete or retrieve the rows with checking.
Add User Page

Last Name: 
First Name: 
Username: 
Phone Number (E.g. 98765432): 
Password: 
E-Mail Address: 
Department: 
Is Admin?: 
Dept. reviewer for: 
Can Add Documents?: 
Can Check-In Documents?:

Add User  Cancel

Delete User Page

User: Chan, Ka Chun - tonychan

Delete  Cancel

Update User Page

ID: 2
Last name: Chan
First name: Ka Chun
Username: tonychan
Phone number: 68241943
Password: Leave empty if unchanged
E-mail: wequ@gmail.com
Department: Information Systems
Admin: 
Reviewer for: 1 of 2 selected
Can Add Documents?: 
Can Check-In Documents?:

Update User  Cancel

View User Page

You are here: Admin > Choose User to View > Show User: tonychan

User information
ID: 2
Last name: Chan
First name: Ka Chun
Username: tonychan
Department: 1
E-mail: wequ@gmail.com
Phone number: 68241943
Admin: No
Reviewer: Yes

Back
Department

Similar to users, admin can create, delete, update or view the existing departments with a user list. There are some predefined departments but admin is still allowed to create or delete to meet with company’s operation requirement. These are database operation to add, delete or retrieve the rows with checking.
Category

Similar to department and users, admin can manage the existing category defined for documents. However, they can add new one for better categorization of office document. These are database operation to add, delete or retrieve the rows with checking

Add Category Page

Delete Category Page
Choice of admin

According to our system, first we will assign one person in the company to be the admin so that the admin can customize the system by defining their own required department, category. For user’s registration, our system allows both means of employee registration of admin creating the user for the company. It depends on the company itself to decide these options. The choice of admin is important. The admin should be a person that is familiar with company’s operation. He/she should also be a senior or management position in the company so that he knows well about the operation or structure of the company and he has the right to access documents to authorize. I personally think that the admin should be a Senior Administrative Officer or Administrative Manager. However, the company can still make their own decision to assign the position of system admin. The number of admin can be more than once.

Workflow function

Functional Design
All of our team members took part in designing the workflow function. I have carried out research and gathered information from Comindware and came up with some features during design as follows that should be incorporated into the workflow management function:

- Easy Graphical modeling of processes: It is better to visualize the workflow path definition so that process activity, resources, role of individuals or department can be manipulated properly. It is a easy and understandable way to define the workflow. Our system is making use of graph to visualize the workflow.

- Flexibility of workflow patterns: Some workflows are simple sequences while some are multiple sequences working in parallel. The system should allow flexibility and allow the workflow to be designed in various way. For example, in our system, parallel workflow, both backward and forward arrow are allowed.

- Options to Pre-fill workflow: Some of the workflow process are repetitive processes, pre-filled template can save users a lot of effort in defining workflow path. In our system, workflow graph can be uploaded and downloaded for use.

- Reporting features: It is better to include some notification methods to notify the users with the progress of workflow or include some tailor-made features or users to monitor their progress or deadlines. In our application, calendar is integrated with workflow function to serve as reminder.

**System Paper Prototyping**

After the first presentation, we gathered requirement and feedback from our supervisor and understand that workflow function should be a key feature in our system. Therefore, Charlae and I immediately started prototyping based on workflow function because we would like to have an early start. We did the prototyping independently and finalize the final interface after several discussions. I have used paper prototyping due to its simplicity and easy understandability. The following shows my paper prototype.

The above left diagram is the workflow prototyping before the presentation. At that time, only single workflow path was considered and the workflow definition is
troublesome and not user-friendly enough. After the presentation, according to the design criteria, the workflow path has to be visualized. Therefore, I come with the design on the right. The five buttons are create person, create document, upload template, download template, save and remove node respectively. In such design, users are allowed to create multiple workflow of more than one documents. Another value-added feature is that users are allowed to download the graph or upload the graph so that it saves time for path creation.

Apart from the six buttons, users have to define the user’s name, action, comment and deadline and also choose a desired document to create workflow. For information setting, the design is that users have to double click the node and the document to edit the information.

The following shows the final product of workflow path creation function. It is based on the design of the prototype. The seven buttons are confirm workflow, upload graph to local, download graph to local, add new node, delete node, rearrange the position and reset graph respectively. For this function, users can create nodes and double click to define users. Users can also use the “+” button to drag an arrow to create a forward link or backward link. The main difference is that workflow creation page is now confined to one document only. For each workflow definition, users can assign path based on one document. In other words, each document has its own workflow creation page. It is simpler to easier to implement. Also, the path can only start with the current user.
Workflow creation Page

After defining a workflow, users can tap the “Workflow” section to monitor and start the workflow. Details can be referred to Felix’s individual report. To track the progress, users may view the graph.

Office workflow situation

- Claim form submission- An employee need to define a path: first, it is for manager to sign on it. Second, it is to pass to the officer in finance department to approve and pass back to him.
- Collaboration between team members- One team member initiate a report, pass to each team members to edit step by step, each person fills in his/her part, and finally pass to manager to review.
- Promotion procedure- Manager pass the promotion form to employee, employee sign on it and pass back to manager to review, manager pass to senior manager or approval.
- Manager assign task- A manager can assign some tasks like asking an employee to complete an income statement form. He can simply create a workflow and set a deadline.
- Self-defined deadline- Everyone in the company can define a workflow point to his/herself to create some self-target. Therefore, the deadline will be shown on the calendar. It is to foster employee’s self-management

Calendar function

To enter the calendar function, click the “Calendar” section in the navigation bar. It is an extension to the workflow function. At the same time, It is also a tailor-made function designed for each user to view and manage his/her business workflow deadlines. In this case, users may have a sense like how many days are there until deadline. It serves as a reminder and management tool.

Making use of different css format, today’s date, deadline date will be displayed with different format.
For retrieving the deadlines, I simply connect to the neo4j graph database, select all the deadline, action, docID for each of the deadline for a particular user. Afterwards, store them in a form of data_array and pass into data to function. Cases of multiple deadlines have been handled.

After entering the “Calendar” section, you will be prompted to a calendar showing the current month and date. Today’s date will be circled in green while the deadline will be indicated by a red dot.
If user touches the dotted date, information about the deadline, including the document id, document name, workflow action and exact deadline time will be shown.
This function serves as an integration to the business workflow function. Defining business workflow is a valuable feature. However, if people miss the deadline and do not manage the deadline properly, it is somehow not that useful to define the workflow. Therefore, I came up with the idea to include a calendar to further enhance the workflow function. Apart from this, users can create a workflow connecting to himself/herself to make a deadline and use the calendar to do some self-management.

**Feedback / Enquiry Page**

To enter the feedback page, users just have to click the “Feedback” section in the navigation bar. This page serves as a means for users in the company to provide feedbacks for our functions. As this is the first time we develop an application and
we would like to provide a tailor-made experience to users, users’ comments are valuable for us. For new users to our application, sometimes they may not fully understand all the functions within the app. As we do not have a user guide, this section is also designed to answer the enquiries and tackle the technical difficulties encountered by the users.

Feedback View

Code from Feedback.php

For this function, users are enforced to type in all the fields, so that we can follow up with both their feedback and enquiry by replying them email. It is also to prevent users intentionally sending junk mail. After filling in all the required fields and pressing the button “Submit”, an email will be sent to our email address: mdmshku@gmail.com in the format as follows.
Getting help from my teammate Charlae, I make use of the PHPMailer function to prepare the component for the email. For the email content, the name, email address and feedback are also appended, so that all detailed information has been collected for follow up.
Cloud Hosting

It is mentioned in the project that our application should be deployed to cloud platform for secure and huge document storage. Also, cloud hosting allows users to access our application and do their document management anytime anywhere. I am the one in the group that is responsible for this part. Therefore, taking advice from the previous presentation, well-known cloud providers are taken into account.

<table>
<thead>
<tr>
<th></th>
<th>Google Cloud Platform</th>
<th>Amazon Web Service Cloud</th>
</tr>
</thead>
</table>
| Advantages:          | - Fastest I/O, less access time  
- Well integrated with other Google services  
- “Pay per use”, Compute Engine pay in minute-level increments. | - The greatest breadth of new services are added all the time  
- Easy to seek help from books, experts from Internet  
- Choice of programming language is unlimited. |
| Disadvantages:       | - Most of the components are based on Google proprietary technologies  
- Choice of programming language is limited  
- It is not easy to transit away from Google Cloud Platform. | - Have experienced high profile outages in recent history  
- Expensive for continuous usage  
- Pay usage round up to hour |

Comparison between Google Cloud and AWS

- Words in red are attributes that are our key concerns that is beneficial to our project

Comparing the pros and cons of the two famous cloud hosting platform, Google Cloud Platform is taken into account for further analysis and studies due to its higher suitability. I chose to deploy our application into Google Cloud Platform.

Google Cloud Platform

To deploy our application to Google Cloud, I can choose to use Google App Engine or Google Compute Engine. In this project, both approaches have been tried and finally
the application is deployed to Google Cloud through Google Compute Engine instance.

**App Engine**

It is a Platform as a Service (PaaS). It is an engine to build and run code on Google infrastructure. No operating system needs to be managed. No servers. It is easier to build, maintain and scale. It supports the delivery, testing and development of software in a cloud-computing environment.

**Approach: Google Cloud App Engine**

**Project and bucket Setup**
First, I have created a project called mdms for hosting my application and a google cloud storage bucket such that API can be called to access the bucket for document upload.
The above picture shows the deployment of our application to the google cloud using App Engine. However, I cannot configure the application successfully in the App Engine because the application cannot directly connect to the local web server and local MySQL server. For our application, we actually need a virtual machine to include an always-on Apache Web server, a MySQL server connecting to the Google Cloud SQL and Neo4j server. Therefore, this approach is not that suitable and I have changed to use Compute Engine.

**Compute Engine**

It is an Infrastructure as a Service(IaaS). It delivers virtual servers running in Google’s innovative data centres and provide persistent storage. Compute Engine’s VMs boot quickly with persistent disk storage and deliver consistent performance.

**Approach: Google Cloud Compute Engine**

**System Architecture**

The Apache Server and Neo4j database are located inside the cloud. The Apache Server will be always-on to listen to the document handling request from the client. The neo4j graph database is for handling the client workflow creation request. The Google Cloud SQL is located outside the cloud with a unique IPv4 address. It is for retrieving the metadata of the document and updating information when needed. In such design, google cloud not only serves as a place to host the server, database, the documents uploading by our application is also stored in a directory in the cloud.
System architecture design

The Google Cloud Compute Engine instance is with the following configuration:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Ubuntu 16.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot disk and local disk size</td>
<td>10GB</td>
</tr>
<tr>
<td>External IP address</td>
<td>35.185.137.175</td>
</tr>
<tr>
<td>Software Installed</td>
<td>Php5.6 with all plugins, phpMyAdmin, MySQL 5.7, Neo4j graph database, JRE, Apache Server</td>
</tr>
<tr>
<td>Machine Type</td>
<td>1 vCPU, 3.75 GB memory</td>
</tr>
</tbody>
</table>

Table of VM configuration

I have created a Google Cloud SQL instance as follows so that I can generate a public IP address to connect to and create the required database through MySQL workbench.

<table>
<thead>
<tr>
<th>IPv4 address</th>
<th>104.199.193.71</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance connection name</td>
<td>mdms-163513:asia-east1:mdms</td>
</tr>
<tr>
<td>Database version</td>
<td>MySQL5.7</td>
</tr>
<tr>
<td>Location</td>
<td>Server</td>
</tr>
</tbody>
</table>

Table of Cloud SQL instance details
Afterwards, I made use of PuttyKeyGen to generate a public-private key pair and use WinSCP together with the private key generated to transfer the file to the VM with location: `var/www/html`.

To deploy my code into the Compute Engine VM, I used the command “`sudo su`” to grant root permission to install all the needed plugins and software in the LINUX command and grant suitable write permission to the file to allow my application function properly. I have also done a lot of configuration in my application so that it pointed to the right connection. I have to get familiar with the LINUX command that I have not learnt before. Finally, I successfully deployed my application with URL: [http://35.185.137.175/mdms_final/MDMS/](http://35.185.137.175/mdms_final/MDMS/) as follows.
Google Compute Engine Architecture

Google Compute Engine provides me with a Virtual machine with a private network. For the VM, it provides persistent disk, local disk and cloud storage. By deploying the code and installing the needed server in the VM, the document is uploaded successfully to the cloud.

GCE Architecture
Limitations and possible solutions

Concurrent editing of document is not supported

Currently, in our application, concurrent editing of document is not allowed. I understand that for easier and more efficient work collaboration, it is better to allow different users to edit at the same time so that works can be done in parallel, which in turn increase the working efficiency in a company.

In our application, although concurrent editing of document is not allowed, we have included “check-in” and “check-out” function to control the versioning of a document. Histories about who has modified the document or when the document is modified are all recorded. However, there are some limitations like when one user check-out a document, others are not allowed to edit that document until the users check in again so it may lead to long waiting time.

Camera Upload is not supported

Currently, for document upload function, it only allows users to upload document using a File Chooser to select files from mobile or computer. It is a limitation that mobile camera upload is not supported. If camera upload is supported, it will be more convenient for colleagues to upload a new paper document by simply capturing a photo from the mobile.

As our application is a web-based application, colleagues in a company not only can access our application through mobile application, they can also use our system using computers. For new paper documents, they can simply scan the documents into the computer and do the upload. Another concern is that mobile scanning often provides a lower resolution power. The document may not be clear enough. Office documents are supposed to be formal and neat. Therefore,

Admin authorization of document upload

In our system, for each document upload, before uploading to the document repository, admin’s approval is needed. Therefore, some privacy about a certain person/ a certain department will be reviewed by others during authorization of document. Therefore, if the admin is not assigned appropriately, it will be like invasion of privacy. However, if there is not authorization function, the system will not be regulated in terms of type of documents uploaded. There should be someone to scan the system regularly or any improperly stored documents.

For our application design, the admin is supposed to be a Senior Administrative Officer or Administrative Manager so that the admin is trustable and understand the inner operation of the company thoroughly. However, the company can still make their own decision to assign the position of system admin. The number of admin can be more than once.
Workflow Function deficiency

For the workflow function, currently there is no email notification when the users return backward to the previous users for amendment or when deadline is approaching. It will be hard to manage the progress if the user does not log in our application frequently to view the graph to see the progress.

In our application, I have tried to implement a calendar function to help users to manage their workflow deadlines. It is hoped that users can establish a habit for viewing the calendar regularly as a mean to do self-management and treat it as a part of work. However, it is better to include email notification as well in case users forgot the deadline and do not check the workflow status regularly.

Future Development

For our mobile document management system, we believe that the functions implemented are valuable to SMEs for their document management and company operation. Our application can be further developed to provide enhanced user experience for SMEs.

In terms of nature, the mobile document management system we developed is a web-based application. To use it in mobile application, we adopted a web view mobile application. It is better to further extend our system into a mobile application and make full use of mobile-friendly features.

In terms of functionality, we should adopt Google Realtime API to provide functions to load and work with Realtime documents. In this case, our application can handle all aspects of data transmission, storage and conflict when more than one user are editing a document. It will further enhance the collaborative nature of our application.

For the calendar function in our application, apart from storing deadlines defined from workflows, I think we can also create a function to let the system administrator set some company events deadline like general meeting, all hands meeting. Therefore, the calendar not only serves as a deadline reminder, but also a scheduler for that particular company. It will offer users with more customized experience.

For document handling in our application, all the documents available to one user will be displayed in the document repository. Although sorting and searching function is provided, it is better to some folder creation function in order to organize the files in a more systematic way as document organization is also an important part in a company.

In terms of scope, now our application targets at SMEs and our functions are somehow tailor-made to suit the needs for company. If our application is taken into practice and found to be useful, it is a possible solution to integrate with the core enterprise system in the company.
Apart from this, apart from SMEs, I think our system somehow suits the need for school as well. It is also a possible way to take our application into use in school. For example, using our application for students to upload their assignment, project presentation slides. The workflow function can also be used in school like defining a path for the signing of final year project form, in-class activity during lecturers. We may try to start with a small department or one faculty to see if it is effective.

Reference

Capterra

Comindware
https://www.comindware.com/blog/thought-leadership/7-features-you-need-to-have-in-your-workflow-management-system-in-2016/

Google Realtime API
https://developers.google.com/google-apps/realtime/overview

Google Compute Engine
https://cloud.google.com/compute/

Axero