

# **AUTO-GENERATE THE DOCUMENTS OF THAI QUALIFICATIONS FRAMEWORK FOR HIGHER EDUCATION BY CLOUD COMPUTING.**

**Tanaporn Pantuprecharat**

*Faculty of Industrial Technology*

*Suan Sunandha Rajabhat University, Bangkok, Thailand*

*E-Mail: tanaporn.pa@ssru.ac.th*

## **ABSTRACT**

This paper proposes a cloud-based solution to auto-generate the document about TQF by using the Google Apps Script to automate documentation tasks across G Suite. Google Apps Script is a cloud based scripting language for extending the functionality of Google G Suite and building lightweight cloud-based applications.

This method uses Google Apps Script to pull data from the rows and columns of a Google Sheet and generate individualized Google Docs by getting only the information needed into the defined template. The proposed system, from user prompts, let users auto-generate TQF document from data stored in Google Sheet file.

**Keywords:** Cloud Computing, SAAS, Google App Script, G Suite for Education, TQF.

## **INTRODUCTION**

Since the emergence of e-learning, that facilitate student to utilize electronic technologies to access educational curriculum outside of a traditional classroom anywhere and anytime, every teacher is required to do e-learning. Does anyone realize that it increase workload aside from teaching and researching task of teacher. That means, teacher has to create instructional media in the form of electronic whether it be document file, multimedia file, audio file, etc. They will create those files by using the application on their own local PCs and then they have to upload those files to a web server after finished. Teachers must learn how to create webpages. Although, later there is LMS (Learning Management System), the web application to support work about education; such as Moodle, teachers can create their own webpage much more easily by using LMS. However, when users have finished creating the electrical media file, they still need to be uploaded into web server.

In 2009, Office of the Higher Education Commission: OHEC announced to enforce Higher educational institutions to comply to the Thai Qualifications Framework for Higher Education (TQF: Hed) [2]. Since then, teachers must create the TQF 3 and TQF 5 document of every subject they teach in each semester, that's even more an additional burden. As we know, fifty percent of content in TQF document are the same as its previous year and can be got from another manual of TQF document. So, when teachers create the TQF 3 or TQF 5, they always copy the file of previous year and make changes only some parts such as semester year and teaching plan. Sometimes, teachers may face the problem about incompatibility between versions of applications installed in personal computer, then the document may be distorted from previous setting. Or, if there are changes about format of document, the TQF of all subjects must be effect even the essence of content still be the same.

Fortunately, later, cloud technology emerges. Google, one of the leaders in cloud technology, developes G Suite that is SAAS comprised of many cloud applications such as google docs, google sheet, google slide, google drive, google classroom, google site etc.

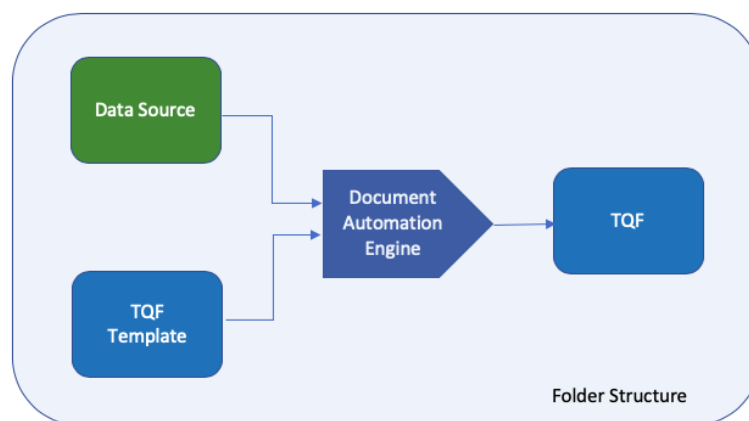
Those cloud application in G Suite can be utilized for creating the educational media and made it on e-learning for all task about education. Moreover, cloud applications have some advantages over desktop applications. They can be less expensive, simpler to manage, and easier to update and use. Since a browser is the most often-used tool for end users, many find it convenient to work with cloud-based applications that are delivered through a browser. All users use same version of application on cloud server, without installation needed on their personal computers, so no problem about incompatible between version installed on various PCs. [3] Data and files can be stored on cloud storage, so it has no need to required huge storage in PCs. Since, most of processing are run on cloud computing server, the PC is not require high specification of CPU, but requires only web browser installed. Moreover, when users want to publish file on webpage, they do not need to upload any files to web, because it's created on cloud storage, it's already on the internet. If user want to share that educational media on their webpage user just share that file and create link to that file on webpage. Cloud storage has features for user to set access permission easily.

In addition, google developed the powerful tools called google app script, that is a cloud based scripting language for extending the functionality of Google Apps and building lightweight cloud-based applications. Developer can use Apps Script to write small programs that extend the standard features of application in GSuite and make them can automatically work together seamlessly.

This research developed web application to auto-generate TQF3 and TQF5 document. Data and document template for TQF3 and TQF5 stores centralize as file system, on cloud storage dedicated for that department. To create the TQF3 or TQF5, teacher just modify some necessary information in prepared google sheet and press command to generate document and the system will auto-generate TQF3 or TQF5 rapidly.

## MATERIALS AND METHODS

TQF Document Automation System is comprised of 5 primary components, including TQF Template, Data Source, Document Automation Engine, result of generated TQF document and Folder Structure, as shown in figure 1.



**Fig.1.** System Components of TQF Auto-generate System

First, TQF template is the documents created to be the template of TQF document that must comply with the standard format defined by the university. The part of Document Template is implemented by google docs. The second component, data source, is the centralized TQF data stored in google sheet to be pulled and placed in document template to generate the TQF document. The third component works as the engine to pull data from

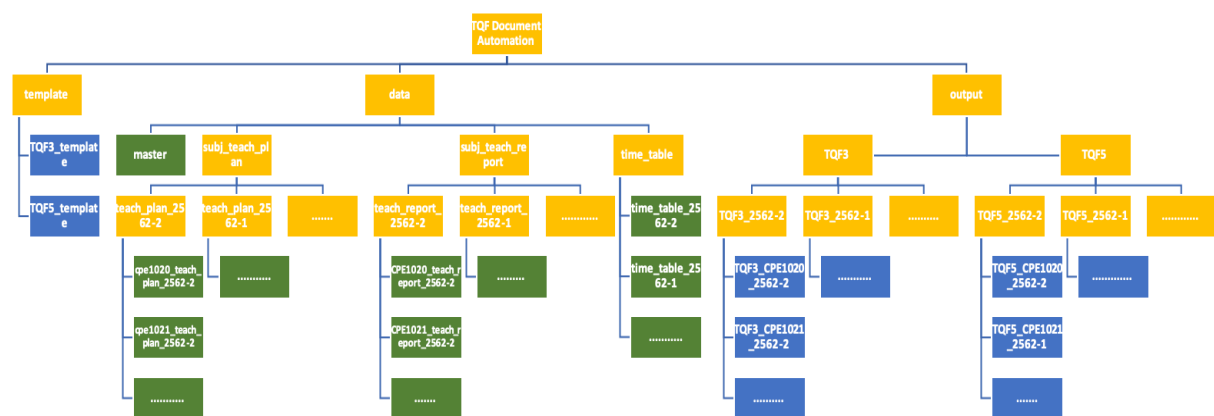
google sheet and placed in document template to generate the TQF document. This engine can work by the instructions coded by google app script, that is the tools that developed by google to extend productivity to other cloud applications in GSuite. Finally, the last component is Folder structure, implemented by google doc, that is cloud storage in G Suite. It is folder structure, well designed, responsible for the container of all files in this system; document template files, data source files, resulted TQF document files.

In Auto-generate TQF Document process, the Document Automation Engine pulls data from Data Source and place it into the right position defined in TQF Template to generate the result of TQF Document. All files, including data source, document template, Automation Engine and resulted TQF are contained in folder structure implemented by google drive.

### Folder and File Structure for TQF Document Automation

Folder structure is the files container implemented by google drive. All documents, including TQF document template and output document; TQF3 and TQF5, are stored in format of google doc file. The data stored in google sheet files.

In figure 2, presents folder structure of system. All files, including data store, program scripts for auto-process and resulted TQF totally stored in form of file system in google drive dedicated for that department. The blue objects mean files created by google doc.; including TQF Document template file and generated TQF. The green objects stand for the files that is google sheet format used for storing the data in the system. And finally, the yellows mean folder in google drive platform. Each file is well designed to group in the suitable categories and placed at the right location. Folder structure, initially, separates into 3 main folders, including template, data and output. The template folder is container for google docs document template file. The data folders are container for storing all files about data; that will be pulled to placed into document template. The output folder is the folder to contain result generated files.



**Fig.2.** Folder Structure

Data folder composed of one google sheet file named master, and 3 folders; including subj\_teach\_plan, subj\_teach\_report and timetable, consequently. The subj\_teach\_plan folder is the folder for store data about course specification (TQF3) of each subject taught in each semester. Since, this data can be changed as needed in every semester, so this folder is

separated into sub-folder for each semester such as subj\_teach\_plan\_2562-2, subj\_teach\_plan\_2562-1, subj\_teach\_plan\_2561-2 and so on. And inside sub-folder of semester separated into sub-folder for each subject; for example, in subj\_teach\_plan\_2562-2 folder, composed of teach-plan files, in google sheet format, of all subjects taught in that semester; such as CPE1020\_teach\_plan\_2562-2, CPE1020\_teach\_plan\_2562-2 etc.

The subj\_teach\_report folder is the folder containing data about Report of Course Implementation (TQF5), that structure of folder will be the same as folder of subj\_teach\_plan. And last but not least, in data\_source folder, timetable folder is the container to store google sheet file of teaching timetable for each semester. So, this folder contains files with the name like timetable\_2562-2, timetable\_2562-1, timetable\_2561-2, timetable\_2561-1 and so on. The reason that these files are important is these files embed the script to run command to create folder structure and command to send the email to notify teacher to update and generate the TQF document of each semester.

The design concept of this folder structure is that, for the data that may be updated in every semester, the folder will be newly created individually to stores files of each semester. For the files that stored the kind of master data, the data rarely changed such as data about faculty basic information, subject or teacher, will be stored in one google sheet file named as “master”. As shown in figure 3, all these sheets; including subject, teacher, basic\_information and curriculum are in master\_data file. The sheets named subject\_objective, std\_dev\_method, teach\_plan, eval\_plan, teach\_resource and improve\_plan are in the file of subj\_teach\_plan for each subject.

### **Auto-generate TQF Document**

The Auto-generate TQF document process is the program written in JavaScript and google app Script in order to call built-in libraries of google docs, sheet and Drive to automate tasks. The written script will be embedded in each timetable and subject\_teach\_plan files.

For each semester, when TQF3 document required to be generated , the administrator of this system will access the timetable file of previous semester then make a copy of that file and change its name to be as new semester, open that file to input timetable of new semester, and then, in add-on menu, click “Prepare file for new semester” command, the script will create 2 folders into folder structure. First folder is teach\_plan folder of that new semester; for example teach\_plan\_2562-2 into data/subj\_teach\_plan folder. The second folder, TQF3 folder of that new semester; for example TQF3\_2562-2 will be created in /output/TQF3. After that, the script program will auto-generate the google sheet file named teach\_plan\_2562-2 for every subject that taught in that new semester. The content of those files will be copied from file of previous semester. The script, when finish creating of all teach\_plan file, the script will send the link for teach\_plan of each subject to responsible teacher by email. Teacher, when receive email, was expected to update teaching plan of responsible subject. When finish updating, that teacher will click at “TQF3” command in “Generate TQF” menu to generate google doc TQF3 file of that subject into /output/TQF3/TQF3\_2562-2 folder immediately.

<b>subject</b>							
subj_code	subj_name_thai	subj_name_english	subj_filename	subj_prerequisite	subj_Co-requisites		
subj_credit	subj_desc_thai	subj_desc_eng	subj_category	subj_group	subj_sub-group		
teach_std_semester	teach_std_year	std_capacity	teach_location				
<b>teacher</b>							
teacher_name	contact_room	contact_floor	contact_build	contact_faculty	work_phone		
mobile_phone	Website_url	email	Facebook	Line	Twitter		
<b>basic infomation</b>							
major	faculty	curriculum					
<b>timetable</b>							
semester	subj_code	teacher_name	responsible_person	day	time		
				building	room		
<b>curriculum mapping</b>							
subj_code	dev_1-1	dev_1-2	dev_1-3	dev_1-4	dev_1-5	dev_1-6	dev_1-7
	dev_2-1	dev_2-2	dev_2-3	dev_2-4	dev_2-5	dev_2-6	dev_2-7
	dev_2-8						
	dev_3-1	dev_3-2	dev_3-3	dev_3-4			
	dev_4-1	dev_4-2	dev_4-3	dev_4-4	dev_4-5	dev_4-6	
	dev_5-1	dev_5-2	dev_5-3	dev_5-4			
<b>subject objective</b>							
subj_purpose	obj_modify						
<b>std dev method</b>							
ethic_teach	ethic_eval	knowledge_teach	knowledge_eval	skill_teach	skill_eval		
social_teach	social_eval	analysis_teach	analysis_eval				
<b>teach plan</b>							
week	desc	hr	activity	teacher_name			
<b>eval plan</b>							
category	method	week	ratio	result			
<b>teach resource</b>							
book	doc	recommend_doc					
<b>improve plan</b>							
eval_by_student	eval_strategy	teach_improve	review_by_student	check_and_improve			

**Fig.3.** File and Record Structure

## RESULTS AND DISCUSSION

The development process of the auto-generate TQF document composed of 1.) Create the TQF Template document by google doc 2.) Create the folder structure on google drive 3.) Create datasource file by google sheet and input the data 4.) Write code with google app script to pull data from google sheet to place in document template on google doc. The figure 4 presents the partial of TQF Template. The Figure 5 presents some examples script of program. The figure 6 presents some part of the result generated TQF.

รายละเอียดของรายวิชา (Course Specification)	
รหัสวิชา	##subject_code##
ชื่อรายวิชาภาษาไทย	##subject_name_thai##
ชื่อรายวิชาภาษาอังกฤษ	##subject_name_english##
จำนวนหน่วยกิต	##subject_credit##
<b>๓. หลักสูตรและประเภทของรายวิชา</b> ๓.๑ หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ๓.๒ ประเภทของรายวิชา หมวดวิชาเฉพาะ กลุ่มวิชาเฉพาะ กลุ่มโครงสร้างพื้นฐานของระบบ	
<b>๔. อาจารย์ผู้รับผิดชอบรายวิชาและอาจารย์ผู้สอน</b> ๔.๑ อาจารย์ผู้รับผิดชอบรายวิชา ##teacher_responsible## ๔.๒ อาจารย์ผู้สอน ##teacher_name##	

Fig.4. a part of document template

```

var subject = Sheets.Spreadsheets.Values.get('1A3Z2U0ZpgdR4uI3-w1BGIEe2muWVc0WwXv4Jb5LTe5M', 'A2:M3');
var templateId = '1jJYIZffNU72LxdBswq31tv0fSTIGtBM0pvFXmkqah1c';

for(var i = 0; i < subject.values.length; i++){
  Logger.log(subject);
  var subj_code = subject.values[i][0];
  var subj_name_thai = subject.values[i][1];
  var subj_name_eng = subject.values[i][2];
  var subj_filename = subject.values[i][3];
  var subj_prerequisite = subject.values[i][4];
  var subj_credit = subject.values[i][5];
  var subj_desc_thai = subject.values[i][6];
  var subj_desc_eng = subject.values[i][7];
  var teacher_name = subject.values[i][8];
  var day = subject.values[i][9];
  var time = subject.values[i][10];
  var room = subject.values[i][11];
  var building = subject.values[i][12];

  //Make a copy of the template file
  var documentId = DriveApp.getFileById(templateId).makeCopy().getId();

  //Rename the copied file
  DriveApp.getFileById(documentId).setName('CourseSyllabus_' + subj_code + '_' + subj_filename + '_2562-1');

  //===== Insert body section =====
  //Insert the subject code
  body.replaceText('##subject_code##', subj_code);

  //Insert the subject Thai name
  body.replaceText('##subject_name_thai##', subj_name_thai);

  //Insert the subject English name
  body.replaceText('##subject_name_english##', subj_name_eng);

  //Insert the subject prerequisite
  body.replaceText('##subject_prerequisite##', subj_prerequisite);

  //Insert the subject credit
  body.replaceText('##subject_credit##', subj_credit);

  //Insert the subject Thai description
  body.replaceText('##subject_description_thai##', subj_desc_thai);

  //Insert the subject English description
  body.replaceText('##subject_description_english##', subj_desc_eng);
  body.replaceText('##teacher_name##', teacher_name);
  body.replaceText('##day##', day);
  body.replaceText('##time##', time);
  body.replaceText('##room##', room);
  body.replaceText('##building##', building);
}

```

get data from cell in google sheet and put in variables

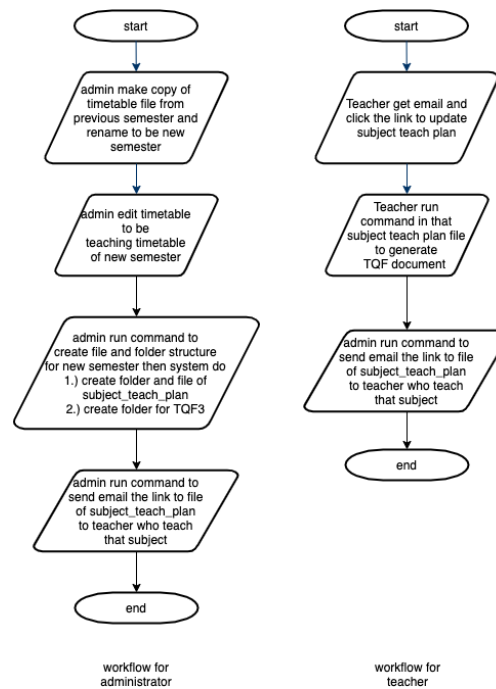
replace data got from google sheet and put to tag in google doc template

Fig.5. a part of script program that pull data from google sheet to map on google doc template

<b>รายละเอียดของรายวิชา (Course Specification)</b>	
รหัสวิชา CPE1020 รายวิชา การโปรแกรมคอมพิวเตอร์	
สาขาวิชา วิศวกรรมคอมพิวเตอร์ คณะ/วิทยาลัย เทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสวนสุนันทา	
ภาคการศึกษา 1 ปีการศึกษา 2562	
หมวดที่ ๑ ข้อมูลทั่วไป	
<b>๑. รหัสและชื่อรายวิชา</b>	
รหัสวิชา	CPE1020
ชื่อรายวิชาภาษาไทย	การโปรแกรมคอมพิวเตอร์
ชื่อรายวิชาภาษาอังกฤษ	Computer Programming
<b>๒. จำนวนหน่วยกิต</b>	3(3-0-6)
<b>๓. หลักสูตรและประเภทของรายวิชา</b>	
๓.๑ หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์
๓.๒ ประเภทของรายวิชา	หมวดวิชาเฉพาะ กลุ่มวิชาเฉพาะ กลุ่มโครงสร้างพื้นฐานของระบบ
<b>๔. อาจารย์ผู้รับผิดชอบรายวิชาและอาจารย์ผู้สอน</b>	
๔.๑ อาจารย์ผู้รับผิดชอบรายวิชา	อาจารย์ ธนภรณ์ พันธุ์ปรีชารัตน์
๔.๒ อาจารย์ผู้สอน	อาจารย์ ธนภรณ์ พันธุ์ปรีชารัตน์

**Fig.6.** The resulted TQF Document

This system has 2 types of user; admin user and teacher. At the beginning of the new semester, the admin user will make a copy of previous timetable file then rename it to be new semester and then update that copied file to be timetable of new semester. When finished, this user will run the add-on command in this timetable google sheet file to create the folder of new semester. Before the beginning of the new semester, the system will create a folder to contain the teaching plan of subjects, then create teaching plan files of each subject with google sheet and place them in that created folder. After that, the system will create a folder for TQF of new semester waiting for TQF to generate into. Finally, the email with a hyperlink to subject teach plan file will be sent to the teacher responsible. Responsible teacher who received the email will follow the hyperlink and update the teaching plan of that subject. When finished, teachers will run the command embedded in that file to generate TQF document. The TQF document will be successfully quickly generated in the TQF output folder of that semester. The process to use system is shown in figure 7.



**Fig.5.** User workflow to auto-generate TQF

The result of using this auto-generate the TQF document system to create the TQF document found that it much more quick and convenient compared to the previous manual process that slow, error-prone, and time-consuming.

## CONCLUSION

Developing program with Google Apps Script, the program can do the powerful tasks like automating repetitive tasks, creating documents, emailing people automatically and connecting the Google Sheets to other services as needed. This research can utilize the google app script to effortlessly develop the program to auto-generate the TQF document successfully as designed.

## REFERENCES

- [1] A. Fernández, D. Peralta, F. Herrera, and J.M. Benítez (2012). *An Overview of E-Learning in Cloud Computing*. Advances in Intelligent Systems and Computing, vol 173. Springer, Berlin, Heidelberg.
- [2] Tantiratanavong, J.Waijitragum, P. (2015). *TQF for Professionals*. The International Journal of Design in Society. Volume 2 Issue 1, January – June 2015.
- [3] Nookhong, Jarumon & Nilsook, Prachyanun. (2017). *System Architecture for Green University Resource Planning on Cloud Computing*. International Journal of the Computer, the Internet and Management. Volume 25 No.2, August 2017 pp. 68-75.
- [4] Cain, K. (2012, June 29). *Extending G Suite with Add-ons*. G Suite Developer. Retrieved from <http://socialmediatoday.com>