



# คู่มือมาตรฐานการพัฒนาซอฟต์แวร์

---

Software Development Standards (SDS)

## ทีม 1

สาขาวิชาวิศวกรรมซอฟต์แวร์ Software Engineering

---

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา



## การควบคุมการเปลี่ยนแปลงเอกสาร

ตารางที่ 0-1 การควบคุมการเปลี่ยนแปลงเอกสาร

เวอร์ชันปัจจุบัน		3.9.1		
วันที่แก้ไขเวอร์ชันล่าสุด		24 มกราคม 2565		
ผู้จัดทำเอกสารนี้		ฝ่ายพัฒนาซอฟต์แวร์ทีม 1		
เวอร์ชัน	วันที่	รายการที่แก้ไข	ผู้แก้ไข	หมายเหตุ
1.0.0	30 สิงหาคม 2563	สร้างเทมเพลต	ธนเกียรติ หอมหวล	ต้นฉบับ
2.0.1	20 สิงหาคม 2564	สร้างเทมเพลต และแก้ไขเนื้อหา	พงศ์ธร และ ชัญญาพัชญ์	
2.1.1	9 กันยายน 2564	ปรับแก้ แพลตฟอร์ม	อภิญญา	
2.2.1	15 กันยายน 2564	ปรับแก้เนื้อหา	ชัญญาพัชญ์ และ อภิญญา	
2.2.2	17 กันยายน 2564	ปรับแก้เนื้อหา	อภิญญา	
2.2.3	18 กันยายน 2564	ปรับแก้เนื้อหา	อภิญญา	
3.8.1	20 มกราคม 2565	ปรับแก้เนื้อหา	อภิญญา	
3.9.1	24 มกราคม 2565	ปรับแก้เนื้อหา	พงศ์ธร อภิญญา และภูมิพัฒน์	



## สารบัญ

หน้า

การควบคุมการเปลี่ยนแปลงเอกสาร .....	ก
สารบัญรูปภาพ .....	ค
สารบัญตาราง .....	ง
ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (CODING STANDARDS).....	1
1. การตั้งชื่อไฟล์และคลาส .....	1
2. การตั้งชื่อฟังก์ชัน .....	3
3. การตั้งชื่อตัวแปร.....	4
4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล.....	5
5. การตั้งชื่อตัวแปรของ CONFIG .....	6
6. การเขียนคอมเมนต์ .....	7
ส่วนที่ 2 มาตรฐานส่วนติดต่อผู้ใช้ (UI STANDARDS).....	12
1. การแสดงสีปุ่ม (BUTTON COLOR) .....	12
2. การจัดวางตำแหน่งปุ่ม (BUTTON POSITION) .....	13
3. การแสดงกล่องข้อความยืนยัน (CONFIRM BOX).....	15
4. การแสดงผลอื่น ๆ.....	15



## สารบัญรูปภาพ

ภาพที่

หน้า

2-1 แสดงรายการปุ่ม และสีปุ่ม (BUTTON COLOR).....	12
2-2 แสดงการจัดวางตำแหน่งปุ่ม (BUTTON POSITION).....	13
2-3 แสดงการจัดวางตำแหน่งปุ่ม (BUTTON POSITION).....	13
2-4 แสดงการจัดวางตำแหน่งปุ่ม (BUTTON POSITION).....	14
2-5 แสดงการแสดงกล่องข้อความยืนยัน (CONFIRM BOX).....	15



## สารบัญตาราง

ตารางที่

หน้า

0-1 การควบคุมการเปลี่ยนแปลงเอกสาร .....ก



## ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (Coding Standards)

มาตรฐานการเขียนโปรแกรมนี้เป็นมาตรฐานที่กำหนดขึ้นในบริบทของการพัฒนาระบบ โดยใช้ CodeIgniter Framework ซึ่งเป็นสถาปัตยกรรมแบบ MVC (Model View และ Controller) โดยส่วนที่มีการกำหนดมาตรฐาน ประกอบด้วยไฟล์ในโฟลเดอร์ Controller, Model, View, Config ดังนั้นจึงมีการกำหนดมาตรฐานการเขียนโปรแกรมแบ่งตามหัวเรื่องและ MVC รวมถึง Config และมาตรฐานเกี่ยวกับฐานข้อมูล ดังนี้

### 1. การตั้งชื่อไฟล์และคลาส

เกี่ยวกับชื่อไฟล์ Controller, Model และ View รวมทั้งชื่อคลาสของ Controller และ Model

#### 1.1. การตั้งชื่อไฟล์และคลาสของ Controller

##### หลักการตั้งชื่อไฟล์และคลาส

1.1.1 ตั้งชื่อไฟล์ขึ้นต้นด้วยพินท์ใหญ่ตามด้วยอักษรพินท์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง ( \_ ) เช่น Admin\_config.php

1.1.2 การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับไฟล์ และขึ้นต้นด้วยอักษรพินท์ใหญ่เช่น Admin

1.1.3 ควรตั้งชื่อตามมอดูล หรืองานของระบบนั้นๆ เช่น การจัดการแฟ้มข้อมูล ควรตั้งชื่อ Folder\_management.php

1.1.4 ส่วนของรายงานของระบบ ควรตั้งชื่อลงท้ายด้วย \_report เช่น Member\_report.php

1.1.5 คอนโทรลเลอร์สำหรับการเชื่อมโยงข้อมูลกับระบบสารสนเทศอื่น ควรตั้งชื่อด้วยชื่อระบบหรืองาน\_controller เช่น DQS\_controller

##### ข้อห้าม

1.1.8 ห้ามตั้งชื่อขึ้นต้นด้วยตัวอักษร c เช่น c\_index, con\_index และ controller\_index

1.1.9 ห้ามตั้งชื่อขึ้นต้นด้วย ชื่อระบบ\_ ยกเว้นคอนโทรลเลอร์หลักของระบบเท่านั้น

#### 1.2. การตั้งชื่อไฟล์และคลาสของ Model

##### หลักการตั้งชื่อไฟล์และคลาส

1.2.1 โมเดลต้องประกอบด้วย 2 ไฟล์คือ Da และ M

1.2.2 ตั้งชื่อไฟล์ด้วยตัวอักษรพินท์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง ( \_ ) ได้เช่น Da\_DQS\_folder.php, M\_DQS\_folder.php



1.2.3 การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับชื่อไฟล์และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่ เช่น

Da\_DQS\_folder, M\_DQS\_folder

1.2.4 ควรตั้งชื่อตามชื่อตารางในฐานข้อมูลเท่านั้น

1.2.5 โมเดลหลักของระบบ ควรตั้งชื่อด้วย ชื่อระบบ\_model เช่น DQS\_model

ข้อยกเว้น

1.2.6 กรณี CodeIgniter เวอร์ชัน 3 ตั้งชื่อไฟล์ขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่

### 1.3. การตั้งชื่อไฟล์ View

หลักการตั้งชื่อไฟล์

1.3.1 ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีด ( \_ )

ในกรณีที่ชื่อไฟล์มีหลายพยางค์ เช่น v\_member\_upload\_file.php

1.3.2 ชื่อไฟล์ View ที่เป็นการทำงานของหน้าหลักควรตั้งชื่อไฟล์ต้นอักษรตัวเล็กทั้งหมด และคั่นคำด้วยเครื่องหมายขีดล่าง ( \_ ) ตัวอย่างเช่น v\_member\_home



## 2. การตั้งชื่อฟังก์ชัน

เกี่ยวกับฟังก์ชันของ Controller และ Model

### 2.1. การตั้งชื่อฟังก์ชัน Controller

#### หลักการตั้งชื่อฟังก์ชัน

2.1.1 ตั้งชื่อฟังก์ชันด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง ( \_ )  
ได้เช่น show\_admin\_config()

#### หมวดของฟังก์ชัน

2.1.2 ฟังก์ชันสำหรับการบันทึกในฐานข้อมูล ตั้งชื่อขึ้นต้นด้วย insert คั่นคำด้วยด้วย  
เครื่องหมายขีดล่าง ( \_ ) เช่น insert\_member()

2.1.3 ฟังก์ชันสำหรับการแก้ไขฐานข้อมูล ตั้งชื่อขึ้นต้นด้วย update คั่นคำด้วยด้วย  
เครื่องหมายขีดล่าง ( \_ ) เช่น update\_folder()

2.1.4 ฟังก์ชันสำหรับการลบข้อมูลในฐานข้อมูล ตั้งชื่อขึ้นต้นด้วย delete คั่นคำด้วย  
เครื่องหมายขีดล่าง ( \_ ) เช่น delete\_folder()

#### 2.1.5 ฟังก์ชันสำหรับการแสดงผล

1) หน้าหลักสำหรับแสดงข้อมูลตั้งชื่อขึ้นต้นด้วย ตัวอักษรพิมพ์เล็ก show และคั่นคำ  
ด้วยเครื่องหมายขีดล่าง ( \_ ) เช่น show\_member\_home()

2.1.6 ฟังก์ชันสำหรับการนำเข้าและอ่านข้อมูลจากไฟล์ ตั้งชื่อขึ้นต้นด้วย upload  
และตามด้วยเครื่องหมายคั่น ( \_ ) เช่น upload\_file()

#### 2.1.7 ฟังก์ชันสำหรับส่งออกข้อมูลในรูปแบบต่าง ๆ

1) ส่งออกข้อมูลในรูปแบบไฟล์ภาพ ตั้งชื่อลงท้ายด้วย \_image  
เช่น save\_qrcode\_image(),

2.1.8 ฟังก์ชันสำหรับรับ - ส่งค่าในรูปแบบ AJAX ตั้งชื่อลงท้ายด้วย \_ajax เช่น  
get\_dept\_list\_ajax()

### 2.2. การตั้งชื่อฟังก์ชัน Model

#### ฟังก์ชันในไฟล์ Da

2.2.1 ประกอบด้วยฟังก์ชันหลัก 4 ฟังก์ชันเท่านั้น โดยตั้งชื่อด้วยตัวอักษรพิมพ์เล็ก  
หรือคั่นคำด้วยเครื่องหมายขีดล่าง ( \_ ) ได้แก่ insert(), update(), delete(), get\_by\_key()





### ฟังก์ชันในไฟล์ M

2.2.2 ฟังก์ชันอื่น ๆ นอกเหนือจากฟังก์ชันในไฟล์ Da เช่น หารคิวรีข้อมูลต่าง ๆ การอัปเดตบางฟิลด์ การลบโดยไม่อ้างอิงหลัก เป็นต้น

2.2.3 ตั้งชื่อฟังก์ชันด้วยอักขรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (\_) ได้ เช่น get\_all()

2.2.4 โครงสร้างของชื่อฟังก์ชัน action\_data\_by\_condition(for\_something)

- 1) action คือ การกระทำ ตัวอย่างเช่น get, search, count, update
- 2) data คือ ข้อมูลที่ต้องการ ตัวอย่างเช่น project, projectname, projecttype
- 3) by\_condition คือ เงื่อนไขการค้นหา เช่น by\_id, by\_name

### หมวดของฟังก์ชัน

2.2.5 ฟังก์ชันสำหรับคิวรีดึงข้อมูล

1) สำหรับดึงข้อมูลทั่วไป ไม่มีการค้นหา หรือค้นหาแบบมีเงื่อนไขไม่ซับซ้อน ได้แก่ ดึงข้อมูลทั้งหมด (get\_all) ข้อมูลที่ขึ้นต่อ

## 3. การตั้งชื่อตัวแปร

### 3.1 ตัวแปรสำหรับรับค่าจาก Fetch Array และ Object

ความแตกต่างระหว่าง array และ object ต้องตั้งชื่อตัวแปรรับค่าคนละแบบ

#### หลักการตั้งชื่อตัวแปร

3.1.1 ให้ตั้งชื่อตัวแปรด้วยตัวอักษรพิมพ์เล็กทั้งหมด

3.1.2 กรณีรับค่าจากการ Fetch array ให้ตั้งชื่อว่า arr\_ข้อมูลนั้นๆ เช่น \$arr\_fol

3.1.3 กรณีรับค่าจากการ Fetch object ให้ตั้งชื่อว่า row\_ข้อมูลนั้นๆ เช่น \$row\_fol

### 3.2 ตัวแปรแทน Object ของ Model

ตั้งชื่อด้วยตัวอักษรพิมพ์ใหญ่ หรือตัวอักษรพิมพ์เล็กทั้งหมด เช่น M\_DQS\_folder

ใช้ชื่อตัวแปรว่า fol, FOL

### 3.3 ตัวแปรที่รับค่ามาจากรฐานข้อมูล

#### หลักการตั้งชื่อตัวแปร

3.3.1 ให้ตั้งชื่อตัวแปรเป็นตัวอักษรพิมพ์เล็กทั้งหมด

3.3.2 กรณีรับค่าหลาย record ให้ใช้ขึ้นต้นด้วย rs\_ชื่อย่อหรือชื่อเต็มของข้อมูล เช่น

\$rs\_ps, \$rs\_person



3.3.3 กรณีรับค่า record เดียว ให้ใช้ขึ้นต้นด้วย uq\_ชื่อย่อหรือชื่อเต็มของข้อมูล เช่น  
\$uq\_ps,\$uq\_person

### 3.4 ตัวแปรทั่วไปหรือ Array

3.4.1 ให้ตั้งชื่อตัวแปรด้วยตัวอักษรพิมพ์เล็กทั้งหมด แล้วคั่นด้วยเครื่องหมายขีดล่าง ( \_ )

### 3.5 ตัวแปรนับรอบของลูป

3.5.1 ใช้ตัวแปร \$i เพื่อนับบรรทัดของลูป

3.5.2 ใช้ตัวแปร \$i, \$j และ \$k หรือ \$x, \$y และ \$z ร่วมกัน กรณีมีลูป มากกว่า 1 ลูป  
ได้ตามความเหมาะสม

## 4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล

### 4.1 การตั้งชื่อฐานข้อมูล

ให้ตั้งชื่อฐานข้อมูลให้ตั้งว่า devteam1\_db เท่านั้น

### 4.2 การตั้งชื่อตาราง

หลักการตั้งชื่อตาราง

4.2.1 ขึ้นต้นด้วยชื่อระบบ ต้องเป็นตัวอักษรพิมพ์ใหญ่ทั้งหมด เช่น DQS

4.2.2 ต้องคั่นด้วยเครื่องหมายขีดล่าง ( \_ ) เป็นต้น เช่น DQS\_Folder

### 4.3 การตั้งชื่อฟิลด์

หลักการตั้งชื่อฟิลด์

4.3.1 ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด

4.3.2 หลักการตั้งชื่อฟิลด์ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมดต้อง แล้วคั่นด้วย  
เครื่องหมายขีดล่าง ( \_ )

4.3.3 ต้องขึ้นต้นด้วยชื่อย่อของตาราง ความยาวไม่เกิน 7 ตัวอักษร เช่น ตาราง  
DQS\_Member ชื่อย่อเป็น mem เป็นต้น

4.3.4 หลังชื่อย่อของตาราง ให้ระบุชื่อฟิลด์นั้นๆ โดยมีชื่อฟิลด์ที่ต้องบังคับใช้ในรูปแบบ  
เดียวกัน ดังนี้

1) ชื่อฟิลด์ที่เป็นคีย์หลัก ต้องลงท้ายด้วย id เช่น mem\_id

2) ชื่อฟิลด์ที่เป็นความหมายหรือข้อมูลหลักของตาราง ต้องลงท้ายด้วย name  
เช่น mem\_firstname, mem\_lastname

3) ชื่อฟิลด์ FK จากตารางอื่น ให้ใช้ชื่อเดิมมาต่อท้าย เช่น fol\_mem\_id



#### 4.4 การเขียนคอมเมนต์ (Comment) ของตารางและฟิลด์

ทุกตาราง และทุกฟิลด์ต้องมีการคอมเมนต์หรือนิยามความหมายกำกับไว้ให้ครบถ้วน ไม่มีข้อยกเว้น

##### หลักการเขียนคอมเมนต์

4.4.1 ฟิลด์ให้นิยามความหมายว่าใช้เก็บข้อมูลอะไร ตัวอย่างเช่น

fol\_id คือ รหัสโฟลเดอร์

fol\_name คือ ชื่อของโฟลเดอร์

4.4.2 การระบุตัวอย่างของข้อมูล หากฟิลด์นั้นมีตัวอย่างของข้อมูลชัดเจน ให้ใส่ต่อท้ายในเครื่องหมายวงเล็บด้วย ตัวอย่างเช่น DQS\_Department คือ สถานะของแผนก (1=ใช้งาน, 0=ไม่ใช้งาน)

#### 5. การตั้งชื่อตัวแปรของ Config

##### ข้อบังคับ

5.1. ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด

5.2. ต้องขึ้นด้วยเครื่องหมายขีดล่าง (\_)

##### หลักการตั้งชื่อฟิลด์

5.3 ขึ้นต้นด้วยชื่อย่อของระบบ (สอดคล้องโฟลเดอร์ของระบบ ใช้คำว่า folder เช่น \$config ["folder"]

5.4 ที่อยู่ไฟล์ที่อัปโหลดของระบบ ใช้คำว่า upload\_path เช่น \$config["upload\_path"]

5.5 ที่ตั้งไดเรกทอรีของระบบ ใช้คำว่า root\_path เช่น \$config["root\_path"]

5.6 ชื่อฐานข้อมูลของระบบ ใช้คำว่า db\_name เช่น \$config["db\_name"]

5.7 ที่อยู่รูปภาพต่างๆ ของระบบ ใช้คำว่า image\_ ชื่อข้อมูลนั้นๆ เช่น \$config["image\_header"]

5.8 ที่อยู่ไอคอนต่างๆ ของระบบ ใช้คำว่า icon\_ ชื่อข้อมูลนั้นๆ เช่น \$config["icon\_add"], \$config["icon\_edit"] และ \$config["icon\_delete"]

##### ข้อห้าม

5.3. ห้ามตั้งชื่อ Config ซ้ำกับชื่อที่มีอยู่แล้ว

5.4. ห้ามแก้ไขหรือลบ Config โดยพลการ หรือหากต้องการตั้งชื่อคอนฟิกเกี่ยวกับระบบอื่นเอง ให้ตั้งชื่อคอนฟิกขึ้นต้นด้วยชื่อระบบของตัวเองก่อนเสมอ เพื่อป้องกันการเขียนทับคอนฟิกของระบบอื่น



## 6. การเขียนคอมเมนต์

### 6.1. คอมเมนต์คลาสของ Controller และ Model

ในคลาสของ Controller และ Model ให้เขียนคอมเมนต์รูปแบบเดียวกัน

#### ข้อบังคับ

6.1.1 ให้เขียนคอมเมนต์ทุกไฟล์ ไม่มีข้อยกเว้น

6.1.2 ให้เขียนคอมเมนต์ฟังก์ชันกำกับทุกฟังก์ชัน ไม่มีข้อยกเว้น

6.1.3 เขียนคอมเมนต์คลาสไว้บรรทัดแรกของไฟล์

6.1.4 เขียนคอมเมนต์คลาสด้วยภาษาอังกฤษ หรือภาษาไทยเท่านั้น

#### หลักการเขียนคอมเมนต์ส่วนของ Class

6.1.5 บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*

6.1.6 บรรทัดที่ 2 ระบุชื่อคลาส เช่น Admin\_config

6.1.7 บรรทัดที่ 3 ระบุชื่อคลาส และข้อความเกี่ยวกับการทำงานคร่าว ๆ

เช่น show admin home

6.1.8 บรรทัดที่ 4 ระบุชื่อผู้สร้างไฟล์คลาสหลังหัวข้อ @author เช่น @author Ashirawat

6.1.9 บรรทัดที่ 5 ระบุวันที่สร้างไฟล์คลาสหลังหัวข้อ @Create Date เช่น @Create Date

2564-09-23

6.1.10 บรรทัดที่ 6 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/

#### หมายเหตุ

แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งก่อนเขียนข้อความเสมอ (ยกเว้น

บรรทัดที่ 1 และบรรทัดที่ 6)

#### ตัวอย่างการคอมเมนต์คลาส

/\*

\* Admin\_config

\* show admin home

\* @author Ashirawat

\* @Create Date 2564-09-23

\*/

## 6.2 คอมเมนต์ฟังก์ชันใน Controller, Model และ View

ในฟังก์ชันของ Controller, Model, View (ส่วนที่เป็น Javascript) ให้เขียนคอมเมนต์รูปแบบเดียวกัน

### ข้อบังคับ

6.2.1 เขียนคอมเมนต์กำกับทุกฟังก์ชัน ไม่มีข้อยกเว้น

6.2.2 เขียนคอมเมนต์ฟังก์ชันไว้ด้านบน ก่อนประกาศฟังก์ชันนั้น ๆ

6.2.3 เขียนคอมเมนต์ฟังก์ชันด้วยภาษาอังกฤษ หรือภาษาไทยเท่านั้น

### หลักการเขียนคอมเมนต์ส่วนของคลาส

6.2.4 บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*

6.2.5 บรรทัดที่ 2 ระบุฟังก์ชัน เช่น show\_changepass()

6.2.6 บรรทัดที่ 3 ระบุชื่อข้อความเกี่ยวกับการทำงานคร่าว ๆ เช่น show change password member on member management

6.2.7 บรรทัดที่ 4 ระบุข้อมูลที่ระบบต้องรับเข้ามาหลังหัวข้อ @input เช่น @input : old password and new password

6.2.8 บรรทัดที่ 5 ระบุข้อมูลที่ระบบแสดงหรือข้อมูลส่งออกมา หลังหัวข้อ @output เช่น @output : password has changed

6.2.9 บรรทัดที่ 6 ระบุชื่อผู้สร้างไฟล์คลาสหลังหัวข้อ @author เช่น @author Natruja

6.2.10 บรรทัดที่ 7 ระบุวันที่สร้างไฟล์คลาสหลังหัวข้อ @Create Date เช่น @Create Date : Date 2564-08-05

6.2.11 บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/

### หมายเหตุ :

6.2.12 แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งก่อนเขียนข้อความเสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 8)

6.2.13 กรณีไม่มีข้อมูลรับเข้าให้ใส่เครื่องหมายขีด (-)



ตัวอย่างการคอมเมนต์ส่วนของฟังก์ชัน

/\*

\* show\_changepass()

\* show edit member

\* @input edit data of member

\* @output data has changed

\* @author Natruja

\* @Create Date 2564-08-05

\*/

### 6.3 คอมเมนต์ส่วนของ View

ข้อบังคับ

6.3.1 ให้เขียนคอมเมนต์ทุกไฟล์ View ไม่มีข้อยกเว้น

6.3.2 หากมีฟังก์ชันในหน้า ต้องเขียนคอมเมนต์ฟังก์ชันกำกับทุกฟังก์ชันไม่มีข้อยกเว้น

6.3.3 เขียนคอมเมนต์ส่วนของ View ไว้บรรทัดแรกของไฟล์

6.3.4 เขียนคอมเมนต์ส่วนของ View ด้วยภาษาอังกฤษ หรือภาษาไทยเท่านั้น

หลักการเขียนคอมเมนต์ส่วนของ View

6.3.5 บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*

6.3.6 บรรทัดที่ 2 ระบุชื่อไฟล์ View เช่น v\_member\_login.php

6.3.7 บรรทัดที่ 3 ระบุชื่อข้อความเกี่ยวกับการทำงานคร่าว ๆ เช่น Display login

6.3.8 บรรทัดที่ 4 ระบุข้อมูลที่ระบบต้องรับเข้ามาหลังหัวข้อ @input เช่น @input : error, username, password

6.3.9 บรรทัดที่ 5 ระบุข้อมูลที่ระบบแสดงหรือข้อมูลส่งออกมา หลังหัวข้อ @output เช่น @output : login

6.3.10 บรรทัดที่ 6 ระบุชื่อผู้สร้างไฟล์คลาสหลังหัวข้อ @author เช่น @author : Krisada

6.3.11 บรรทัดที่ 7 ระบุวันที่สร้างไฟล์คลาสหลังหัวข้อ @Create Date เช่น @Create Date : 2564-08-05

6.3.12 บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/





#### หมายเหตุ

6.3.13 แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งก่อนเขียนข้อความเสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 8)

6.3.14 กรณีไม่มีข้อมูลรับเข้าให้ใส่เครื่องหมายขีด (-)

6.3.15 กรณีที่มี @author มากกว่า 1 คน ใช้เครื่องหมาย (,) คั่นระหว่างชื่อ เช่น Ashirawat, Krisada

#### ตัวอย่างการคอมเมนต์ส่วนของ View

/\*

\* v\_member\_login.php

\* Display login

\* @input error, username, password

\* @output login

\* @author Ashirawat, Krisada

\* @Create Date 2564-08-05

\*/

#### **6.4 คอมเมนต์บรรทัดเดียวหรือตัวแปรต่าง ๆ**

กรณีต้องการคอมเมนต์เพื่อนิยามความหมายของตัวแปร หรือส่วนการทำงานบรรทัดนั้น ๆ หรือคอมเมนต์เพื่อระบุวันที่แก้ไข ผู้แก้ไข หรือหมายเหตุสำหรับกรณีที่มีการปรับแก้หรือเพิ่มเติมโปรแกรม (ไม่บังคับ)

#### หลักการเขียนคอมเมนต์

6.4.1 ขึ้นต้นด้วยเครื่องหมายคอมเมนต์ก่อนที่จะประกาศตัวแปร 1 บรรทัด เช่น //

6.4.2 เว้นวรรค 1 ครั้ง ตามด้วยคอมเมนต์ที่ต้องการ โดยสามารถคอมเมนต์เป็นภาษาไทย หรืออังกฤษได้ตามความเหมาะสม

6.4.3 ให้คอมเมนต์ด้านบนก่อนประกาศตัวแปร หรือก่อนบรรทัดนั้นๆ

#### ตัวอย่างการคอมเมนต์

// กำหนดค่าเริ่มต้นของ i โดยใช้สำหรับการนับรอบ

\$i=0;

// ส่วนของ Input Password

<input type="password" name="password">

### 6.5 คอมเมนต์สำหรับการแก้ไขไฟล์หรือฟังก์ชัน Controller, Model และ View

กรณีต้องการคอมเมนต์ส่วนของการทำงานที่มีคำสั่งมากกว่า 1 บรรทัด เพื่อระบุขอบเขตการทำงานนั้นๆ (ไม่บังคับ)

#### ข้อบังคับ

6.5.1 เขียนคอมเมนต์ด้วยตัวอักษรภาษาอังกฤษเท่านั้น หรือตัวอักษรภาษาไทยเท่านั้น

6.5.2 ต้องระบุคอมเมนต์ไว้ทั้ง 2 ส่วน คือ ส่วนบนและส่วนท้ายของการทำงานนั้นๆ

หลักการคอมเมนต์ส่วนบน

6.5.3 เปิด - ปิดคอมเมนต์ไว้ด้านบนก่อนเริ่มการทำงานนั้นๆ โดยใช้เครื่องหมายคอมเมนต์แบบ PHP คือ /\* และ \*/ ตามลำดับ หรือเครื่องหมายคอมเมนต์ใน HTML ก็ได้

6.5.4 ส่วนของข้อความให้เว้นวรรค 1 ครั้ง แล้วขึ้นต้นด้วย Start แล้วตามด้วยอธิบายส่วนการทำงานนั้นๆ

#### หลักการเขียนคอมเมนต์

6.5.5 ให้เขียนต่อจากคอมเมนต์ในส่วนของหัวไฟล์หรือฟังก์ชันนั้น โดยระบุครั้งที่แก้ไข วันที่ทำการแก้ไข ชื่อผู้แก้ไข (กรณีไม่ใช่ผู้สร้างไฟล์ หรือฟังก์ชันนั้น) และอธิบายสิ่งที่ทำการแก้ไข เช่น @Update Date 2: 2564-9-17 Ashirawat แก้ไขส่วนของปุ่มสมัครสมาชิก

#### หลักการคอมเมนต์ส่วนท้าย

6.5.6 เปิด - ปิดคอมเมนต์ไว้ด้านล่างสุดหลังการทำงานนั้นๆ โดยใช้เครื่องหมายคอมเมนต์แบบ PHP คือ /\* และ \*/ ตามลำดับ หรือเครื่องหมายคอมเมนต์ใน HTML ก็ได้

6.5.7 ส่วนของข้อความให้เว้นวรรค 1 ครั้ง แล้วขึ้นต้นด้วย End แล้วตามด้วยอธิบายส่วนการทำงานนั้นๆ

#### ตัวอย่างการคอมเมนต์สำหรับการแก้ไขไฟล์หรือฟังก์ชัน

```
<!-- Start View login --!>
```

```
--Login --
```

```
<!-- End View login --!>
```



## ส่วนที่ 2 มาตรฐานส่วนติดต่อผู้ใช้ (UI STANDARDS)

มาตรฐานส่วนติดต่อผู้ใช้เป็นส่วนเทมเพลต Bootstrap Framework เป็นต้นแบบในการกำหนดมาตรฐานที่ใช้ในการพัฒนากันอย่างแพร่หลายและมีลักษณะเหมือนกัน ดังนั้น จึงจัดมาตรฐานนี้ เพื่อให้การแสดงผลส่วนติดต่อผู้ใช้งาน (User Interface) เป็นไปตามมาตรฐานเดียวกัน

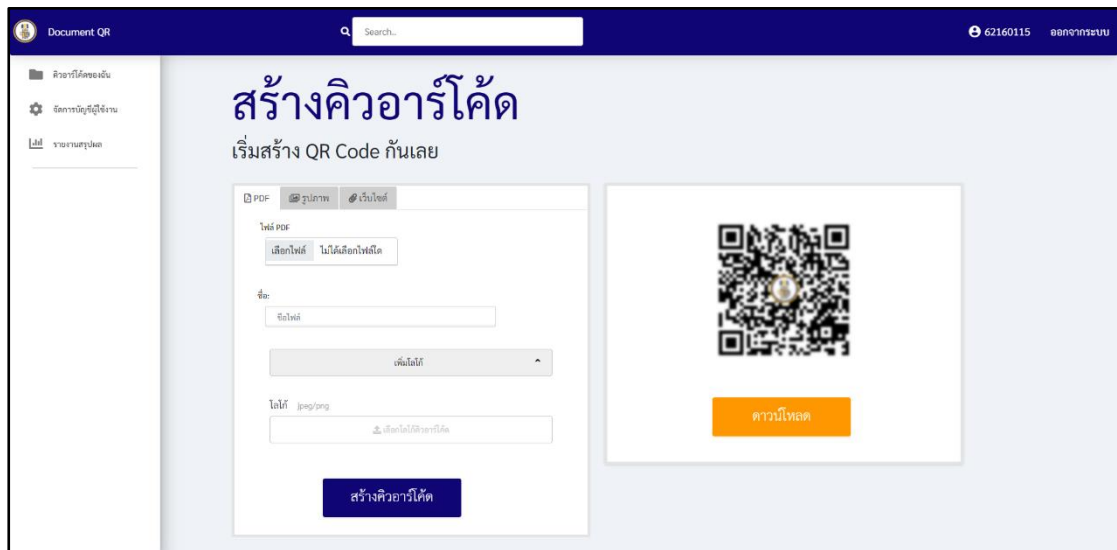
### 1. การแสดงสีปุ่ม (Button Color)



ภาพที่ 2-1 แสดงรายการปุ่ม และสีปุ่ม (Button Color)

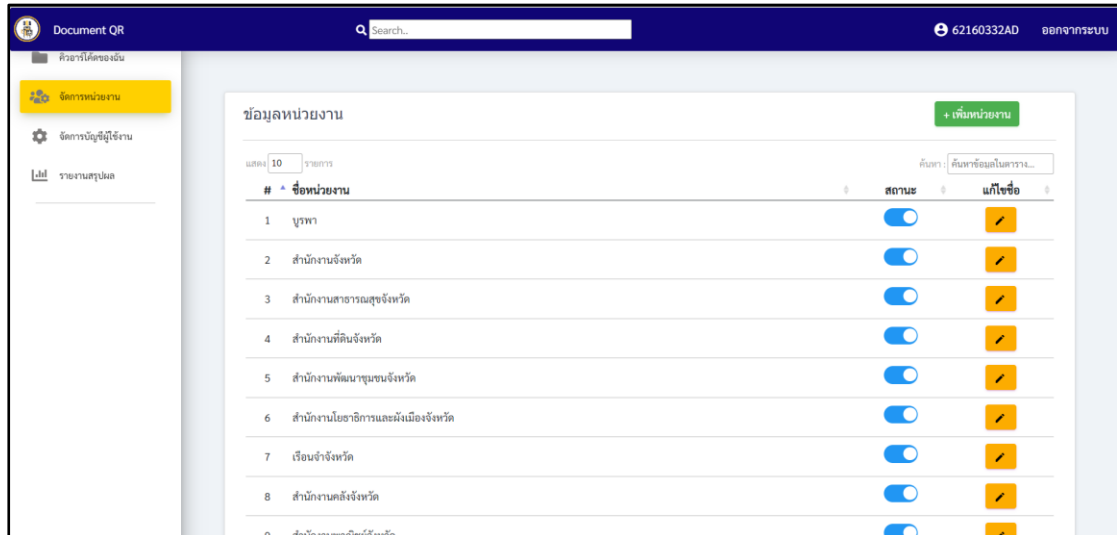
■ ปุ่มสร้างคิวอาร์โค้ด และแก้ไขชื่อไฟล์	แสดงเป็น	สีน้ำเงิน
■ ปุ่มค้นหา, ย้าย และบันทึก	แสดงเป็น	สีฟ้า
■ ปุ่มยืนยัน, เพิ่มแผนก และสร้าง	แสดงเป็น	สีเขียว
■ ปุ่มสร้าง และดูสถิติการเข้าใช้งาน	แสดงเป็น	สีขาว
■ ปุ่มลบ และยกเลิก	แสดงเป็น	สีแดง
■ ปุ่มดาวน์โหลด	แสดงเป็น	สีเหลือง

## 2. การจัดวางตำแหน่งปุ่ม (Button Position)



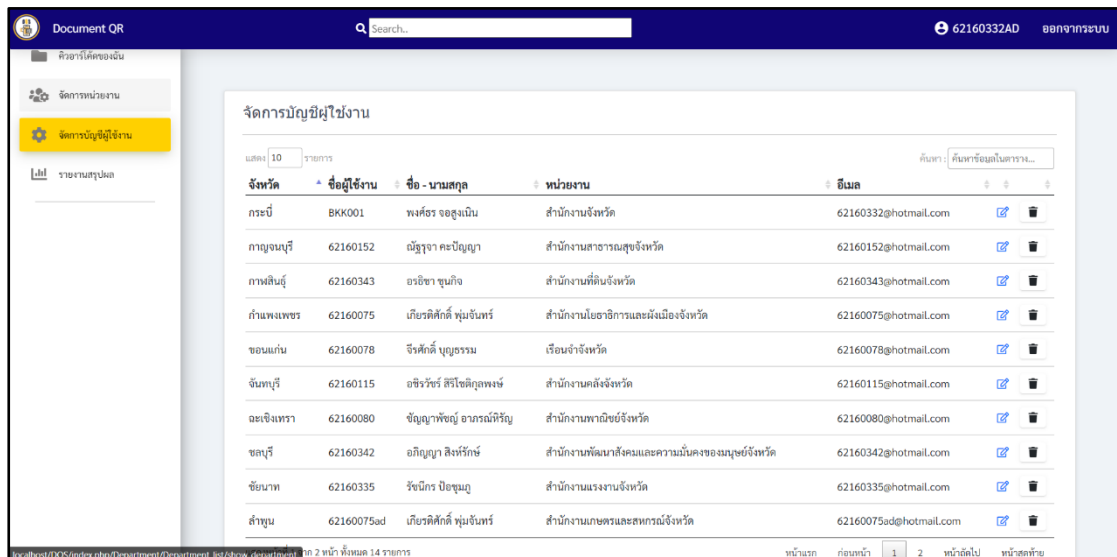
ภาพที่ 2-2 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

- ปุ่มสร้างคิวอาร์โค้ด                      จัดวางตำแหน่งกลาง
- ปุ่มดาวน์โหลด                                จัดวางตำแหน่งขวาล่าง



ภาพที่ 2-3 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

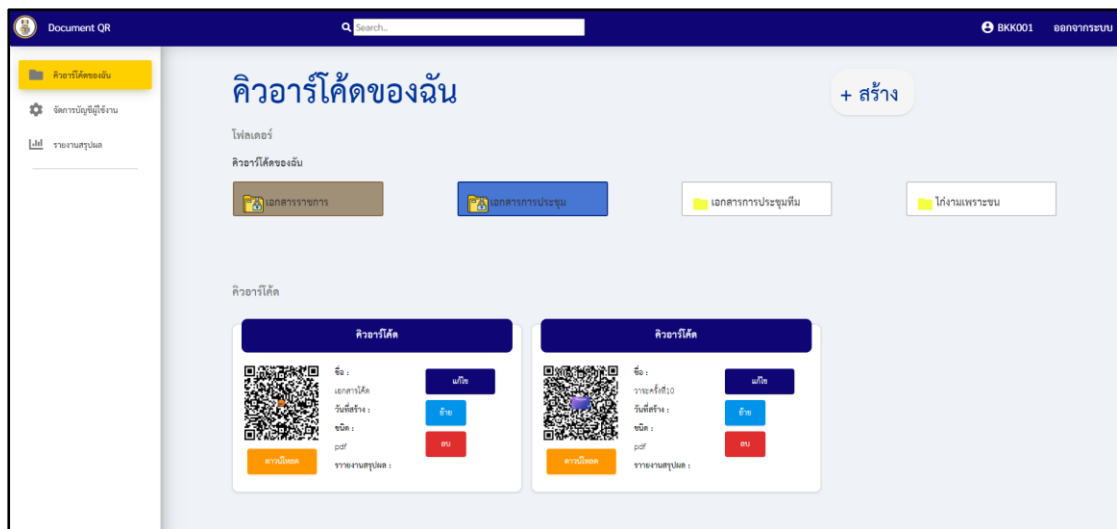
- |                 |                      |
|-----------------|----------------------|
| ■ ปุ่มเพิ่มแผนก | จัดวางตำแหน่งขวากลาง |
| ■ ปุ่มค้นหา     | จัดวางตำแหน่งขวากลาง |
| ■ ปุ่มแก้ไข     | จัดวางตำแหน่งขว      |



ภาพที่ 2-4 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

- ปุ่มค้นหา
- ปุ่มแก้ไข
- ปุ่มลบ

จัดวางตำแหน่งขวบน  
จัดวางตำแหน่งขว  
จัดวางตำแหน่งขว

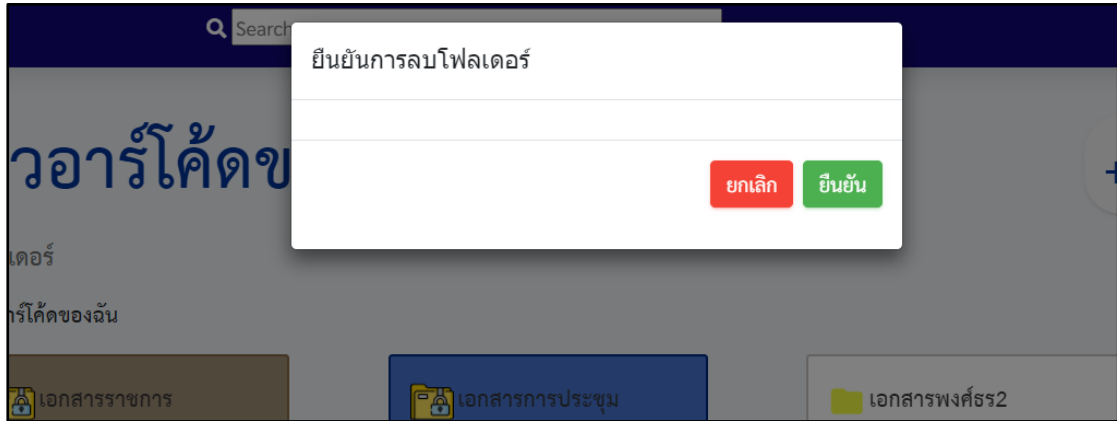


ภาพที่ 2-5 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

- ปุ่มสร้าง
- ปุ่มแก้ไข
- ปุ่มย้าย

จัดวางตำแหน่งขวบน  
จัดวางตำแหน่งขว  
จัดวางตำแหน่งขว

### 3. การแสดงกล่องข้อความยืนยัน (Confirm Box)



ภาพที่ 2-6 แสดงการแสดงกล่องข้อความยืนยัน (Confirm Box)

- ปุ่มยืนยัน จัดวางตำแหน่งขวา
- ปุ่มยกเลิก จัดวางตำแหน่งซ้าย

#### 4. การแสดงผลอื่น ๆ

กรณีเกี่ยวกับมาตรฐานในส่วนติดต่อผู้ใช้ที่กำหนดรูปแบบกันในทีมพัฒนา ให้เป็นรูปแบบเดียวกัน  
ภายในระบบ มีดังนี้

การแสดงความแจ้งเตือน Form Validation รูปแบบการแสดงผลจะเป็นในลักษณะของ Alert Box ทั้งระบบ

การแสดงผล Tooltip จะมีคำอธิบายเพิ่มเติมในส่วนเงื่อนไขต่าง ๆ ที่จำเป็นในการกรอกข้อมูลลงแบบฟอร์มซึ่งจะเป็นในลักษณะเดียวกันทั้งระบบ

การแสดงผล Placeholder ใช้ในการแสดงตัวอย่างของข้อความที่จำเป็นต้องกรอกภายในฟิลด์นั้น ๆ ซึ่งข้อความที่แสดงขึ้นมานั้นจะเป็นในลักษณะข้อความพื้นหลังของฟิลด์

การแสดงไอคอนหรือรูปภาพแทนการดำเนินการ เลือกใช้ในสิ่งที่เหมาะสมกับเทมเพลตของระบบ ซึ่งจะมีลักษณะเดียวกันทั้งระบบ

การแสดงผลวันที่ (Date Format) จะเป็นการแสดง วัน เดือน ปี และเวลา ซึ่งจะมีรูปแบบเดียวกันทั้งระบบ