**คู่มือมาตรฐานการพัฒนาซอฟต์แวร์**

***Software Development Standards (SDS)***

***เวอร์ชัน 2.2.0***

Icon

Description automatically generated

**ทีมพัฒนาซอฟต์แวร์ระบบ ทีมที่ 4  
*System Development Team 4***

**คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา**

ทีมบริหารจัดการคุณภาพ (Quality Management)   
ปรับปรุงล่าสุด พฤศจิกายน 2564

**สารบัญ**

หน้า

[ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (Coding Standards) 1](#_Toc87282131)

[1. การตั้งชื่อไฟล์และคลาส 1](#_Toc87282132)

[1.1. การตั้งชื่อไฟล์และคลาสของ Controller 1](#_Toc87282133)

[1.2. การตั้งชื่อไฟล์และคลาสของ Controller 2](#_Toc87282134)

[1.3. การตั้งชื่อไฟล์ view 3](#_Toc87282135)

[2. การตั้งชื่อฟังก์ชัน 4](#_Toc87282136)

[2.2 การตั้งชื่อฟังก์ชันของ Model 6](#_Toc87282137)

[3. การตั้งชื่อตัวแปร 7](#_Toc87282138)

[3.1 ตัวแปรแทน Object ของ Model 7](#_Toc87282139)

[3.2. ตัวแปรที่รับค่ามาจากฐานข้อมูล 8](#_Toc87282140)

[3.3 ตัวแปรสำหรับรับค่าจาก Fetch Array และ Object 8](#_Toc87282141)

[3.4 ตัวแปร Array 8](#_Toc87282142)

[3.5 ตัวแปร JSON 9](#_Toc87282143)

[3.6 ตัวแปรนับรอบของลูป 9](#_Toc87282144)

[4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล 9](#_Toc87282145)

[4.1. การตั้งชื่อฐานข้อมูล 9](#_Toc87282146)

[4.2 การตั้งชื่อตาราง 9](#_Toc87282147)

[4.3 การตั้งชื่อฟิลด์ 10](#_Toc87282148)

[4.4. การเขียน Comment ของตารางและฟิลด์ 11](#_Toc87282149)

[5. การตั้งชื่อตัวแปรของ Config 12](#_Toc87282150)

[6. การตั้งชื่อฟังก์ชันของ Helper 13](#_Toc87282151)

[7. การเขียน Comments 14](#_Toc87282152)

[7.1. Comment คลาสของ Controller และ Model 14](#_Toc87282153)

[7.2. Comment ฟังก์ชันใน Controller, Model และ Helper 15](#_Toc87282154)

[7.3. Comment ส่วนของ View 17](#_Toc87282155)

[7.4. Comment บรรทัดเดียว หรือตัวแปรต่างๆ 19](#_Toc87282156)

[7.5. Comment ระบุขอบเขตส่วนการทำงานนั้นๆ 19](#_Toc87282157)

[ส่วนที่2 มาตรฐานส่วนติดต่อผู้ใช้ (UI Standards) 21](#_Toc87282158)

[1. การแสดงสีปุ่ม (Button Color) 21](#_Toc87282159)

[2. การจัดวางตำแหน่งปุ่ม (Button Position) 22](#_Toc87282160)

[3. การแสดงกล่องข้อความยืนยัน (Confirm Box) 23](#_Toc87282161)

[4. การแสดงผลอื่นๆ 23](#_Toc87282162)

# ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (Coding Standards)

มาตรฐานการเขียนโปรแกรมนี้เป็นมาตรฐานที่กำหนดขึ้นในบริบทของการพัฒนาระบบโดยใช้ CodeIgniter Framework ซึ่งเป็นสถาปัตยกรรมแบบMVC(Model-View-Controller) โดยส่วนที่มีการกำหนดมาตรฐาน ประกอบด้วยไฟล์ในโฟลเดอร์ controllers, models, views, config และ helpers ดังนั้นจึงมีการกำหนด มาตรฐานการเขียนโปรแกรมแบ่งตามหัวเรื่อง และ MVC รวมถึง config, helpers และมาตรฐานเกี่ยวกับ ฐานข้อมูล ดังนี้

## 1. การตั้งชื่อไฟล์และคลาส

เกี่ยวกับชื่อไฟล์Controller, Model, View รวมทั้ง ชื่อคลาสของ Controller, Model

### 1.1. การตั้งชื่อไฟล์และคลาสของ Controller

หลักการตั้งชื่อไฟล์และคลาส

1. ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (\_) ได้เช่น project.php
2. การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับชื่อไฟล์และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่ เช่น Project
3. ควรตั้งชื่อตามโมดูล หรืองานของระบบนั้นๆ เช่น การจัดการหลักสูตร ควรตั้งชื่อว่า course\_management
4. คอนโทรลเลอร์หลักของระบบ ควรตั้งชื่อด้วย ชื่อระบบ\_controller เช่น emeeting\_controller
5. ส่วนของข้อมูลพื้นฐานของระบบ ควรตั้งชื่อลงท้ายด้วย \_base เช่น ข้อมูลพื้นฐานของระบบ การจัดการประชุม ใช้ชื่อว่า meeting\_base
6. ส่วนของรายงานของระบบ ควรตั้งชื่อลงท้ายด้วย \_report เช่น project\_report
7. คอนโทรลเลอร์สำหรับการเชื่อมโยงข้อมูลกับระบบสารสนเทศอื่น ควรตั้งชื่อด้วย ชื่อระบบหรือ งาน\_service เช่น emeeting\_service
8. คอนโทรลเลอร์สำหรับการรับ - ส่งค่าในรูปแบบของ AJAX ควรตั้งชื่อด้วย ชื่อระบบหรือ งาน\_ajax เช่น emeeting\_ajax

ข้อยกเว้น

1. กรณีที่ต้องใช้ตัวเลขเป็นส่วนหนึ่งของชื่อไฟล์หรือคลาส ให้ตั้งอยู่ตำแหน่งท้ายสุด เช่น section1, section2 เป็นต้น
2. กรณีสร้างโฟลเดอร์ย่อย base, report, service หรือ ajax ไม่ต้องตั้งชื่อลงท้ายด้วย \_base, \_report, \_service หรือ \_ajax ตามลำดับ
3. กรณีCodeIgniter เวอร์ชัน 3 ตั้งชื่อไฟล์ขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่

ข้อห้าม

1. ห้ามตั้งชื่อขึ้นต้นด้วย c\_ เช่น c\_project, con\_project, controller\_project
2. ห้ามตั้งชื่อขั้นต้นด้วย ชื่อระบบ\_ ยกเว้นคอนโทรลเลอร์หลักของระบบเท่านั้น

### 1.2. การตั้งชื่อไฟล์และคลาสของ Controller

หลักการตั้งชื่อไฟล์และคลาส

1. โมเดลต้องประกอบด้วย 2 ไฟล์คือ da และ m
2. ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (\_) ได้เช่น da\_ppm\_project.php, m\_ppm\_project.php
3. การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับชื่อไฟล์และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่ เช่น Da\_ppm\_project, M\_ppm\_project
4. ควรตั้งชื่อตามชื่อตารางในฐานข้อมูลเท่านั้น
5. โมเดลหลักของระบบ ควรตั้งชื่อด้วย ชื่อระบบ\_model เช่น emt\_model

ข้อยกเว้น

1. กรณีCodeIgniter เวอร์ชัน 3 ตั้งชื่อไฟล์ขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่

### 1.3. การตั้งชื่อไฟล์ view

หลักการตั้งชื่อไฟล์

1. ชื่อไฟล์ต้องขึ้นต้นด้วย v\_ เช่น v\_project.php

2. ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (\_) ได้เช่น v\_project\_detail.php

3. ชื่อ View ต้องสอดคล้องกับชื่อฟังก์ชัน หรือคลาส เช่น ฟังก์ชันชื่อ projecttype\_input ควร ตั้งชื่อไฟล์ว่า v\_projecttype\_input.php

ข้อยกเว้น

1. กรณี 1 ฟังก์ชันเรียกมากกว่า 1 view ให้ขยายชื่อไฟล์จากชื่อเดิมได้ เช่นฟังก์ชันชื่อ project\_input() เรียกไฟล์ v\_project\_input\_section1.php และ v\_project\_input\_section2.php 2. กรณี 1 view ถูกเรียกจากหลายฟังก์ชัน ให้ตั้งชื่อตามชื่อฟังก์ชันแรกที่เรียกใช้
2. กรณี 1 view ถูกเรียกจากหลายฟังก์ชัน ให้ตั้งชื่อตามชื่อฟังก์ชันแรกที่เรียกใช้

## 2. การตั้งชื่อฟังก์ชัน

หลักการตั้งชื่อฟังก์ชัน

1. ตั้งชื่อฟังก์ชันด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (\_) ได้เช่น project\_input()

หมวดของฟังก์ชัน

1. ฟังก์ชันสำหรับแสดงหน้าจอการบันทึกหรือแก้ไขข้อมูล
   1. หน้าจอบันทึกข้อมูลอย่างเดียว หรือทั้งบันทึกและแก้ไข ตั้งชื่อลงท้ายด้วย \_input หรือ ตั้งชื่อเป็น input เช่น projecttype\_input(), input()
   2. หน้าจอการแก้ไขข้อมูลอย่างเดียว ตั้งชื่อลงท้ายด้วย \_edit หรือตั้งชื่อเป็น edit เช่น projecttype\_edit(), edit()
2. ฟังก์ชันสำหรับการบันทึกหรือแก้ไขลงฐานข้อมูล
   1. สำหรับบันทึกอย่างเดียว ตั้งชื่อลงท้ายด้วย \_insert หรือตั้งชื่อเป็น insert เช่น projecttype\_insert(), insert()
   2. สำหรับแก้ไขอย่างเดียว ตั้งชื่อลงท้ายด้วย \_update หรือตั้งชื่อเป็น update เช่น projecttype\_update(), update()
   3. สำหรับบันทึกและแก้ไข ตั้งชื่อลงท้ายด้วย \_save หรือตั้งชื่อเป็น save เช่น projecttype\_save(), save()
3. ฟังก์ชันสำหรับการลบข้อมูลในฐานข้อมูล ตั้งชื่อลงท้ายด้วย \_delete หรือตั้งชื่อเป็น delete เช่น projecttype\_delete(), delete()
4. ฟังก์ชันสำหรับการแสดงผล
   1. หน้าหลักสำหรับแสดงข้อมูล ตั้งชื่อลงท้ายด้วย \_show หรือตั้งชื่อเป็น show เช่น projectlist\_show(), show()
   2. หน้าสำหรับแสดงข้อมูลแบบลงรายละเอียด ตั้งชื่อลงท้ายด้วย \_detail หรือตั้งชื่อเป็น detail เช่น projectlist\_detail(), detail()
5. ฟังก์ชันสำหรับการนำเข้าและอ่านข้อมูลจากไฟล์โดยเฉพาะ (ไฟล์Excel) ตั้งชื่อลงท้ายด้วย \_import หรือตั้งชื่อเป็น import เช่น person\_import(), import()
6. ฟังก์ชันสำหรับส่งออกข้อมูลในรูปแบบต่างๆ
   1. ส่งออกข้อมูลรูปแบบไฟล์Excel ตั้งชื่อลงท้ายด้วย \_excel หรือตั้งชื่อเป็น excel เช่น actionplan\_excel(), excel()
   2. ส่งออกข้อมูลรูปแบบไฟล์Word ตั้งชื่อลงท้ายด้วย \_word หรือตั้งชื่อเป็น word เช่น actionplan\_word(), word()
   3. ส่งออกข้อมูลรูปแบบไฟล์PDF ตั้งชื่อลงท้ายด้วย \_pdf หรือตั้งชื่อเป็น pdf เช่น actionplan\_pdf(), pdf()
   4. ส่งออกข้อมูลรูปแบบตัวอย่างก่อนพิมพ์ตั้งชื่อลงท้ายด้วย \_print หรือตั้งชื่อเป็น print เช่น actionplan\_print(), print()
   5. ส่งออกข้อมูลหลายรูปแบบในฟังก์ชันเดียว ตั้งชื่อลงท้ายด้วย \_export หรือตั้งชื่อเป็น export เช่น actionplan\_export(), export()
7. ฟังก์ชันสำหรับแสดง Popup ตั้งชื่อลงท้ายด้วย \_popup หรือตั้งชื่อเป็น popup เช่น projecttype\_insert\_popup(), projecttype\_save\_popup(), popup()
8. ฟังก์ชันสำหรับการเชื่อมโยงข้อมูลกับระบบสารสนเทศที่เกี่ยวข้อง
   1. การรับข้อมูล ตั้งชื่อขึ้นต้นด้วย get\_service\_ หรือตั้งชื่อเป็น get\_service เช่น get\_service\_person, get\_service()
   2. การส่งข้อมูล ตั้งชื่อขึ้นต้นด้วย post\_service\_ หรือตั้งชื่อเป็น post\_service เช่น post\_service\_person, post\_service()
9. ฟังก์ชันสำหรับรับ - ส่งค่าในรูปแบบ AJAX ตั้งชื่อลงท้ายด้วย \_ajax หรือตั้งชื่อเป็น ajax เช่น projecttype\_ajax, ajax()

### 2.2 การตั้งชื่อฟังก์ชันของ Model

ฟังก์ชันในไฟล์ da

1. ประกอบด้วย ฟังก์ชันหลัก 4 ฟังก์ชัน เท่านั้น ได้แก่ insert(), update(), delete(), get\_by\_key()

ฟังก์ชันในไฟล์ m

1. ฟังก์ชันอื่นๆ นอกเหนือจากฟังก์ชันในไฟล์da เช่น การคิวรี่ข้อมูลต่างๆ การอัปเดตบางฟิลด์ การลบโดยไม่อ้างคีย์หลัก เป็นต้น
2. ตั้งชื่อฟังก์ชันด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (\_) ได้เช่น get\_all()
3. 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\_pjid, by\_pjname
   4. for\_something คือ ถูกเรียกใช้เพื่อฟังก์ชัน โมดูล หรือเงื่อนไขโดยเฉพาะ (ถ้าสำคัญ) เช่น for\_mission, for\_ajax

หมวดของฟังก์ชัน

1. ฟังก์ชันสำหรับคิวรี่ดึงข้อมูล
   1. สำหรับดึงข้อมูลทั่วไป ไม่มีการค้นหา หรือค้นหาแบบมีเงื่อนไขไม่ซับซ้อน ได้แก่ดึง ข้อมูลทั้งหมด (get\_all) ข้อมูลที่ขึ้นต่อกัน เช่น สาขาขึ้นอยู่กับคณะที่เลือก เป็นต้น ให้ ขึ้นต้นด้วย get\_ เช่น get\_project()
   2. สำหรับดึงข้อมูลที่มีการค้นหาแบบมีเงื่อนไขที่ซับซ้อน ให้ขึ้นต้นด้วย search\_ เช่น search\_cousestr\_by\_csname\_and\_dpid
2. ฟังก์ชันสำหรับคิวรี่โดยเรียกใช้SQL function
   1. ให้ตั้งชื่อขึ้นต้นด้วย SQL function เช่น count\_person(), sum\_salary(), max\_salary(), avg\_salary()
3. ฟังก์ชันสำหรับคิวรี่เพื่อตรวจสอบข้อมูล
   1. สำหรับ return ค่า เป็น binary เช่น 0,1 TRUE, FALSE Y,N ให้ตั้งชื่อขึ้นต้นด้วย check เช่น check\_active\_person()
4. ฟังก์ชันสำหรับอัปเดตบางฟิลด์
   1. กรณีอัปเดต 1 - 2 ฟิลด์ให้ตั้งชื่อว่า update\_ชื่อฟิลด์ที่ต้องการอัปเดต เช่น update\_firstname(), update\_prefix\_firstname()
   2. กรณีอัปเดตมากกว่า 2 ฟิลด์ให้ตั้งชื่อว่า update\_ชื่อการทำงานนั้นๆ เช่น update\_person\_retire() คือการอัปเดตฟิลด์สถานะของบุคลากรและวันที่ออก
5. ฟังก์ชันสำหรับลบ โดยไม่อ้างอิง PK
   1. ให้ตั้งชื่อว่า delete\_by\_ชื่อฟิลด์ เช่น delete\_by\_dept\_id(), delete\_by\_dept\_id\_and\_pos\_id()

## 3. การตั้งชื่อตัวแปร

### 3.1 ตัวแปรแทน Object ของ Model

ขึ้นต้นด้วย m ต่อด้วยชื่อย่อของตาราง เช่น m\_hr\_person ใช้ชื่อตัวแปรว่า mps

### 3.2. ตัวแปรที่รับค่ามาจากฐานข้อมูล

หลักการตั้งชื่อตัวแปร

1. ให้ตั้งชื่อตัวแปรเป็นตัวอักษรพิมพ์เล็กทั้งหมด
2. กรณีรับค่าหลาย record ให้ใช้ขึ้นต้นด้วย rs\_ชื่อย่อหรือชื่อเต็มของข้อมูล เช่น $rs\_ps, $rs\_person
3. กรณีรับค่า record เดียว ให้ใช้ขึ้นต้นด้วย qu\_ชื่อย่อหรือชื่อเต็มของข้อมูล เช่น $qu\_ps, $qu\_person
4. กรณีรับค่าฟิลด์เดียว หรือจากฟังก์ชันของ SQL ให้ตั้งชื่อให้สื่อความหมาย เช่น รับค่าจาก ฟังก์ชัน sum\_salary() ให้ตั้งชื่อว่า $sum\_salary
5. กรณีรับค่าเป็น Array ใช้สำหรับ drop down list ต้องขึ้นต้นด้วย opt\_ เช่น $opt\_province

### 3.3 ตัวแปรสำหรับรับค่าจาก Fetch Array และ Object

ความแตกต่างระหว่าง array และ object และต้องตั้งชื่อตัวแปรรับค่าคนละแบบ

หลักการตั้งชื่อตัวแปร

1. ให้ตั้งชื่อตัวแปรด้วยตัวอักษรพิมพ์เล็กทั้งหมด
2. กรณีรับค่าจากการ Fetch array § ค่า Key ให้ตั้งชื่อว่า key\_ข้อมูลนั้นๆ เช่น $key\_ps § ค่า Value ให้ตั้งชื่อว่า val\_ข้อมูลนั้นๆ เช่น $val\_ps
3. กรณีรับค่าจากการ Fetch object ให้ตั้งชื่อว่า row\_ข้อมูลนั้นๆ เช่น $row\_ps

### 3.4 ตัวแปร Array

ชื่อตัวแปรที่บ่งบอกว่าเป็นชุดของ Array ให้ขึ้นต้นด้วย arr ตัวอย่างเช่น $arr\_ps, $arr\_dp

### 3.5 ตัวแปร JSON

กรณีที่ต้องการส่งข้อมูลผ่านตัวแปรกลับมาในรูปแบบ JSON ให้ขึ้นต้นด้วย json ตัวอย่างเช่น json\_data, json\_message

### 3.6 ตัวแปรนับรอบของลูป

กรณีต้องการตั้งตัวแปรเพื่อใช้นับรอบของลูป สามารถใช้ตัวแปรในรูปแบบ Single ได้แต่ต้องสื่อ ความหมาย เช่น

1. ใช้ตัวแปร $i เพื่อนับบรรทัดของลูป
2. ใช้ตัวแปร $i, $j, $k หรือ $m, $n หรือ $x, $y, $z ร่วมกัน กรณีมีลูปมากกว่า 1 ลูปได้ตามความ เหมาะสม

## 4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล

### 4.1. การตั้งชื่อฐานข้อมูล

ข้อบังคับ

1. ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
2. ต้องคั่นด้วยเครื่องหมายขีดล่าง (\_)

หลักการตั้งชื่อฐานข้อมูล

1. ขึ้นต้นด้วยชื่อย่อของไซต์เช่น pi, sci, buu
2. ตามด้วยชื่อระบบ หรือ ชื่อย่อของระบบ เช่น hr, emeeting, spms
3. ลงท้ายด้วยคำว่า db ตัวอย่างเช่น pi\_hrdb, sci\_emeetingdb, buu\_spmsdb

### 4.2 การตั้งชื่อตาราง

ข้อบังคับ

1. ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
2. ต้องคั่นด้วยเครื่องหมายขีดล่าง (\_)

หลักการตั้งชื่อตาราง

1. ขึ้นต้นด้วยชื่อย่อของระบบในฐานข้อมูล ความยาวไม่เกิน 4 ตัวอักษร เช่น hr, emt, spms
2. ตามด้วยชื่อโมดูลการทำงาน หรือบ่งบอกว่าใช้เก็บข้อมูลนั้นๆ ตัวอย่างเช่น hr\_person, hr\_province, emt\_agenda

### 4.3 การตั้งชื่อฟิลด์

ข้อบังคับ

1. ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
2. ต้องคั่นด้วยเครื่องหมายขีดล่าง (\_)

หลักการตั้งชื่อฟิลด์

1. ต้องขึ้นต้นด้วยชื่อย่อของตาราง ความยาวไม่เกิน 4 ตัวอักษร (ไม่รวมชื่อฐานข้อมูล) เช่น ตาราง hr\_person ชื่อย่อเป็น ps หรือตาราง hr\_admin ชื่อย่อเป็น am เป็นต้น
2. หลังชื่อย่อของตาราง ให้ระบุชื่อฟิลด์นั้นๆ โดยมีชื่อฟิลด์ที่ต้องบังคับใช้ในรูปแบบเดียวกัน ดังนี้
   1. ชื่อฟิลด์ที่เป็นคีย์หลัก ต้องลงท้ายด้วย id เช่น ps\_id
   2. ชื่อฟิลด์ที่เป็นความหมายหรือข้อมูลหลักของตาราง ต้องลงท้ายด้วย name เช่น ps\_name, ps\_first\_name, ps\_last\_name
   3. ชื่อฟิลด์ที่บ่งบอกถึงลำดับของข้อมูล ต้องลงท้ายด้วย seq เช่น am\_seq, dp\_seq
   4. ชื่อฟิลด์ที่บ่งบอกถึงลำดับชั้น ต้องลงท้ายด้วย level เช่น dp\_level
   5. ชื่อฟิลด์FK จากตารางอื่น ให้ใช้ชื่อเดิมมาต่อท้าย เช่น ps\_pf\_id, ps\_dp\_id
   6. ชื่อฟิลด์FK ตารางตารางเดียวกันและบ่งบอกความสัมพันธ์แม่ลูก ต้องลงท้ายด้วย parent\_id เช่น dp\_parent\_id

ข้อยกเว้น

1. ชื่อย่อของตารางมีความยาวมากกว่า 4 ตัวอักษรได้กรณีที่ไม่สามารถลดคำให้น้อยลงได้ (ถ้า จำเป็น)

### 4.4. การเขียน Comment ของตารางและฟิลด์

ทุกตารางและทุกฟิลด์ต้องมีการคอมเม้นท์หรือนิยามความหมายกำกับไว้ให้ครบถ้วน ไม่มีข้อยกเว้น

หลักการเขียนคอมเม้นท์

1. ตาราง ให้นิยามความหมายว่า ตารางเก็บข้อมูลอะไร หรือใช้สำหรับทำอะไร เช่น ตาราง emt\_agenda คือ ตารางเก็บข้อมูลระเบียบวาระ
2. ฟิลด์ให้นิยามความหมายว่าใช้เก็บข้อมูลอะไร เช่น
   1. agd\_id คือ รหัสระเบียบวาระ
   2. agd\_name คือ ชื่อระเบียบวาระ
   3. agd\_seq คือ ลำดับที่ในการแสดงผล
3. การระบุตัวอย่างของข้อมูล หากฟิลด์นั้นมีตัวอย่างของข้อมูลชัดเจน ให้ใส่ต่อท้ายใน เครื่องหมายวงเล็บด้วย เช่น
   1. agd\_status คือ สถานะของระเบียบวาระ (Y=ใช้งาน, N=ไม่ใช้งาน)
4. ฟิลด์ที่อ้างอิงจากฟิลด์อื่น (FK) ให้อธิบายความหมายเดียวกันกับตารางตั้งต้น (PK) และต่อท้าย ด้วยว่ามาจากตารางไหนและ/หรือฐานข้อมูลไหน (ระบุชื่อฐานข้อมูลด้วย หากอยู่คนละ ฐานข้อมูล) เช่น
   1. agd\_person\_id คือ รหัสบุคลากร (ตาราง hr.Person)
   2. agd\_mt\_id คือ รหัสการประชุมย่อย (ตาราง emt\_meeting)

## 5. การตั้งชื่อตัวแปรของ Config

ข้อบังคับ

1. ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
2. ต้องคั่นด้วยเครื่องหมายขีดล่าง (\_)

หลักการตั้งชื่อฟิลด์

1. ขึ้นต้นด้วยชื่อย่อของระบบ (สอดคล้องกับชื่อไฟล์คอนฟิก) เช่น hr, emt, spms
2. แล้วตามด้วยชื่อข้อมูลนั้นๆ ตัวอย่างเช่น
   1. โฟลเดอร์ของระบบ ใช้คำว่า folder เช่น $config["hr\_folder"], $config["emt\_folder"]
   2. ที่อยู่ไฟล์ที่อัพโหลดของระบบ ใช้คำว่า upload\_path เช่น $config["hr\_upload\_path"]
   3. ที่ตั้งไดเรกทอรี่ของระบบ ใช้คำว่า root\_path เช่น $config["hr\_root\_path"]
   4. ชื่อฐานข้อมูลของระบบ ใช้คำว่า db\_name เช่น $config["hr\_db\_name"]
   5. ที่อยู่รูปภาพต่างๆ ของระบบ ใช้คำว่า image\_ชื่อข้อมูลนั้นๆ เช่น $config["hr\_image\_header"]
   6. ที่อยู่ไอคอนต่างๆ ของระบบ ใช้คำว่า icon\_ชื่อข้อมูลนั้นๆ เช่น $config["hr\_icon\_add"], $config["hr\_icon\_edit"], $config["hr\_icon\_delete"]

ข้อควรระวัง

1. ห้ามตั้งชื่อคอนฟิกซ้ำกับระบบอื่น หากต้องการเรียกคอนฟิกจากระบบอื่นสามารถเรียกใช้ได้ เลย (ถ้ามี) หรือหากต้องการตั้งชื่อคอนฟิกเกี่ยวกับระบบอื่นเอง ให้ตั้งชื่อคอนฟิกขึ้นต้นด้วยชื่อ ระบบของตัวเองก่อนเสมอ เพื่อป้องกันการเขียนทับคอนฟิกของระบบอื่น เช่น ระบบบุคลากร ต้องการมีคอนฟิกชื่อโฟลเดอร์ระบบ UMS ให้ตั้งชื่อว่า $config["hr\_ums\_folder"] เป็นต้น

## 6. การตั้งชื่อฟังก์ชันของ Helper

ข้อบังคับ

1. ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
2. ต้องคั่นด้วยเครื่องหมายขีดล่าง (\_)

หลักการตั้งชื่อฟิลด์

1. ขึ้นต้นด้วยชื่อย่อของระบบ (สอดคล้องกับชื่อไฟล์Helper) เช่น hr, emt, spms
2. แล้วตามด้วยชื่อฟังก์ชันการทำงานนั้นๆ ตัวอย่างเช่น ฟังก์ชันอ่านไฟล์ของระบบ ตัวอย่าง hr\_read\_file(), emt\_read\_file()

ข้อห้าม

1. ห้ามตั้งชื่อฟังก์ชันซ้ำกับ function\_helper ที่มีอยู่แล้ว
2. ห้ามเพิ่ม/ลบ/แก้ไขเกี่ยวกับฟังก์ชันในไฟล์function\_helper โดยพละการ ยกเว้นมีข้อสรุป จากหัวหน้าทีมทุกทีมแล้ว
3. ห้ามตั้งชื่อฟังก์ชันซ้ำกับระบบอื่น หากต้องการเรียกฟังก์ชันจากระบบอื่นสามารถเรียกใช้ได้เลย (ถ้ามี) หรือหากต้องการคัดลอกฟังก์ชันจากระบบอื่นมายังระบบตัวเอง ให้ตั้งชื่อฟังก์ชันขึ้นต้น ด้วยชื่อระบบของตัวเองก่อนเสมอ เพื่อป้องกันการเขียนทับ Helper ของระบบอื่น เช่น ระบบ บุคลากรต้องการมีฟังก์ชันดึงปีงบประมาณปัจจุบันเหมือนระบบกำกับงบฯ ให้ตั้งชื่อว่า hr\_get\_bgy() เป็นต้น
4. หลีกเลี่ยงการใช้ตัวเลขในการตั้งชื่อฟังก์ชัน สำหรับฟังก์ชันที่การทำงานคล้ายกัน แต่ต้องการตั้ง ชื่อให้แตกต่างกันโดยใช้ตัวเลข ควรตั้งชื่อฟังก์ชันให้สื่อถึงการทำงาน

## 7. การเขียน Comments

### 7.1. Comment คลาสของ Controller และ Model

ในคลาสของ Controller และ Model ให้เขียนคอมเม้นท์รูปแบบเดียวกัน

ข้อบังคับ

1. ให้เขียนคอมเม้นท์คลาสกำกับในไฟล์คลาสทุกไฟล์ไม่มีข้อยกเว้น
2. เขียนคอมเม้นท์คลาสไว้บรรทัดแรกของไฟล์
3. เขียนคอมเม้นท์คลาสด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ ยกเว้นเป็นชื่อตัวแปรหรือข้อความเฉพาะ

หลักการเขียนคอมเม้นท์คลาส

1. บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเม้นท์คือ /\*
2. บรรทัดที่ 2 ระบุชื่อคลาส เช่น Baseposition
3. บรรทัดที่ 3 ระบุการทำงานของคลาสแบบคร่าวๆ เช่น Base Data of Position Management
4. บรรทัดที่ 4 ระบุชื่อผู้สร้างไฟล์คลาส หลังหัวข้อ @author เช่น @author Somchai
5. บรรทัดที่ 5 ระบุวันที่สร้างไฟล์คลาส ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหัวข้อ @Create Date เช่น @Create Date 2558-10-26
6. บรรทัดที่ 6 ใช้เครื่องหมายปิดคอมเม้นท์คือ \*/

หมายเหตุ :

1. แต่ละบรรทัด ให้ใส่เครื่อง \* เว้นวรรค 1 ครั้งนำหน้าเสมอ (ยกเว้นบรรทัดที่ 1 และ 6)
2. ข้อความคอมเม้นท์หลังหัวข้อ @author ให้เคาะ tab 1 ครั้ง

ตัวอย่างการคอมเม้นท์คลาส

/\*   
\* Baseposition   
\* Base Data of Position Management   
\* @author Somchai   
\* @Create Date 2558-10-26   
\*/

### 7.2. Comment ฟังก์ชันใน Controller, Model และ Helper

ในฟังก์ชันของ Controller, Model และ Helper ให้เขียนคอมเม้นท์รูปแบบเดียวกัน

ข้อบังคับ

1. ให้เขียนคอมเม้นท์ฟังก์ชันกำกับทุกฟังก์ชัน ไม่มีข้อยกเว้น
2. เขียนคอมเม้นท์ฟังก์ชันไว้ด้านบน ก่อนประกาศฟังก์ชันนั้นๆ
3. เขียนคอมเม้นท์ฟังก์ชันด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ ยกเว้นเป็นชื่อตัวแปรหรือข้อความเฉพาะ

หลักการเขียนคอมเม้นท์ฟังก์ชัน

1. บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเม้นท์คือ /\*
2. บรรทัดที่ 2 ระบุชื่อฟังก์ชัน เช่น position\_insert
3. บรรทัดที่ 3 ระบุการทำงานของฟังก์ชันแบบคร่าวๆ เช่น Insert position in database after form add
4. บรรทัดที่ 4 ระบุข้อมูลเข้า (Input) หลังหัวข้อ @input กรณีไม่มีให้ใส่เครื่องหมายขีด (-) เช่น @input postion\_name\_th, postion\_name\_en, position\_seq
5. บรรทัดที่ 5 ระบุข้อมูลที่ส่งกลับคืน (Output) หลังหัวข้อ @output กรณีไม่มีให้ใส่ เครื่องหมายขีด (-) เช่น @output The last insert id (pos\_id)
6. บรรทัดที่ 6 ระบุชื่อผู้สร้างฟังก์ชัน หลังหัวข้อ @author เช่น @author Somchai
7. บรรทัดที่ 7 ระบุวันที่สร้างฟังก์ชัน ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหัวข้อ @Create Date เช่น @Create Date 2558-10-26
8. บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเม้นท์คือ \*/

หมายเหตุ :

1. แต่ละบรรทัด ให้ใส่เครื่อง \* เว้นวรรค 1 ครั้งนำหน้าเสมอ (ยกเว้นบรรทัดที่ 1 และ 8)
2. ข้อความคอมเม้นท์หลังหัวข้อ @input, @output, @author ให้เคาะ tab 1 ครั้ง

ตัวอย่างการคอมเม้นท์ฟังก์ชัน

/\*   
\* position\_insert   
\* Insert position in database after form add   
\* @input postion\_name\_th, postion\_name\_en, position\_seq   
\* @output The last insert id (pos\_id) \* @author Somchai   
\* @Create Date 2558-10-26   
\*/

### 7.3. Comment ส่วนของ View

ข้อบังคับ

* 1. ให้เขียนคอมเม้นท์ส่วนของวิวกำกับทุกไฟล์ไม่มีข้อยกเว้น
  2. เขียนคอมเม้นท์ส่วนของวิวไว้บรรทัดแรกสุดของไฟล์
  3. เขียนคอมเม้นท์ส่วนของวิวด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ ยกเว้นเป็นชื่อตัวแปรหรือข้อความเฉพาะ
  4. ระบุคอมเม้นท์ส่วนของวิวให้อยู่ในรูปแบบของ PHP ยกเว้นไฟล์วิวเป็น HTML ให้ใช้การคอม เม้นท์แบบ HTML ได้

หลักการเขียนคอมเม้นท์ส่วนของ View

1. บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเม้นท์คือ /\*
2. บรรทัดที่ 2 ระบุชื่อไฟล์วิว เช่น v\_position\_input
3. บรรทัดที่ 3 ระบุการทำงานของวิวแบบคร่าวๆ เช่น Display input form of position for add
4. บรรทัดที่ 4 ระบุข้อมูลเข้า (Input) หลังหัวข้อ @input กรณีไม่มีให้ใส่เครื่องหมายขีด (-) เช่น @input Array of headings (arr\_head)
5. บรรทัดที่ 5 ระบุข้อมูลที่ส่งกลับคืน (Output) หลังหัวข้อ @output กรณีไม่มีให้ใส่ เครื่องหมายขีด (-) เช่น @output Input form of position
6. บรรทัดที่ 6 ระบุชื่อผู้สร้างฟังก์ชัน หลังหัวข้อ @author เช่น @author Somchai
7. บรรทัดที่ 7 ระบุวันที่สร้างฟังก์ชัน ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหัวข้อ @Create Date เช่น @Create Date 2558-10-26
8. บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเม้นท์คือ \*/

หมายเหตุ :

1. แต่ละบรรทัด ให้ใส่เครื่อง \* เว้นวรรค 1 ครั้งนำหน้าเสมอ (ยกเว้นบรรทัดที่ 1 และ 8)
2. ข้อความคอมเม้นท์หลังหัวข้อ @input, @output, @author ให้เคาะ tab 1 ครั้ง

ตัวอย่างการคอมเม้นท์ส่วนของ View

/\*   
\* v\_position\_input   
\* Display input form of position for add   
\* @input Array of headings (arr\_head)   
\* @output Input form of position   
\* @author Somchai   
\* @Create Date 2558-10-26   
\*/

### 7.4. Comment บรรทัดเดียว หรือตัวแปรต่างๆ

กรณีต้องการคอมเม้นท์เพื่อนิยามความหมายของตัวแปร หรือส่วนการทำงานบรรทัดนั้นๆ หรือคอม เม้นท์เพื่อระบุวันที่แก้ไข ผู้แก้ไข หรือหมายเหตุสำหรับกรณีที่มีการปรับแก้หรือเพิ่มเติมโปรแกรม (ไม่บังคับ)

หลักการเขียนคอมเม้นท์

1. ให้คอมเม้นท์ด้านบนก่อนประกาศตัวแปร หรือก่อนบรรทัดนั้นๆ
2. ขึ้นต้นด้วยเครื่องหมายคอมเม้นท์คือ //
3. เว้นวรรค 1 ครั้ง ตามด้วยคอมเม้นท์ที่ต้องการ โดยสามารถคอมเม้นท์เป็นภาษาไทยหรือ อังกฤษได้ตามความเหมาะสม

ตัวอย่างการคอมเม้นท์

// ตั้งค่าเริ่มต้นของ i   
$i = 0;

### 7.5. Comment ระบุขอบเขตส่วนการทำงานนั้นๆ

กรณีต้องการคอมเม้นท์ส่วนของการทำงานที่มีคำสั่งมากกว่า 1 บรรทัด เพื่อระบุขอบเขตการทำงาน นั้นๆ (ไม่บังคับ)

ข้อบังคับ

1. เขียนคอมเม้นท์ด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่
2. ต้องระบุคอมเม้นท์ไว้ทั้ง 2 ส่วน คือ ส่วนบนและส่วนท้ายของการทำงานนั้นๆ

หลักการคอมเม้นท์ส่วนบน

1. เปิด - ปิดคอมเม้นท์ไว้ด้านบนก่อนเริ่มการทำงานนั้นๆ โดยใช้เครื่องหมายคอมเม้นท์แบบ PHP คือ /\* และ \*/ ตามลำดับ หรือเครื่องหมายคอมเม้นท์ใน HTML ก็ได้
2. ส่วนของข้อความให้เว้นวรรค 1 ครั้ง แล้วขึ้นต้นด้วย Start แล้วตามด้วยอธิบายส่วนการทำงาน นั้นๆ

หลักการคอมเม้นท์ส่วนท้าย

1. เปิด - ปิดคอมเม้นท์ไว้ด้านล่างสุดหลังการทำงานนั้นๆ โดยใช้เครื่องหมายคอมเม้นท์แบบ PHP คือ /\* และ \*/ ตามลำดับ หรือเครื่องหมายคอมเม้นท์ใน HTML ก็ได้
2. ส่วนของข้อความให้เว้นวรรค 1 ครั้ง แล้วขึ้นต้นด้วย End แล้วตามด้วยอธิบายส่วนการทำงาน นั้นๆ

ตัวอย่างการคอมเม้นท์

/\* Start Form input of position \*/ .  
..   
...   
/\* End Form input of position \*/

# ส่วนที่2 มาตรฐานส่วนติดต่อผู้ใช้ (UI Standards)

มาตรฐานส่วนติดต่อผู้ใช้นี้ใช้เทมเพลตแบบ Bootstrap Framework เป็นต้นแบบในการกำหนดมาตรฐาน ซึ่ง เป็นรูปแบบของเทมเพลตที่ใช้พัฒนากันในหลายระบบ โดยรูปแบบของการแสดงผลจะมีลักษณะเหมือนกัน ดังนั้น จึงจัดทำมาตรฐานนี้ขึ้น เพื่อให้การแสดงผลส่วนติดต่อผู้ใช้(User Interface) เป็นไปตามมาตรฐานเดียวกัน

## 1. การแสดงสีปุ่ม (Button Color)

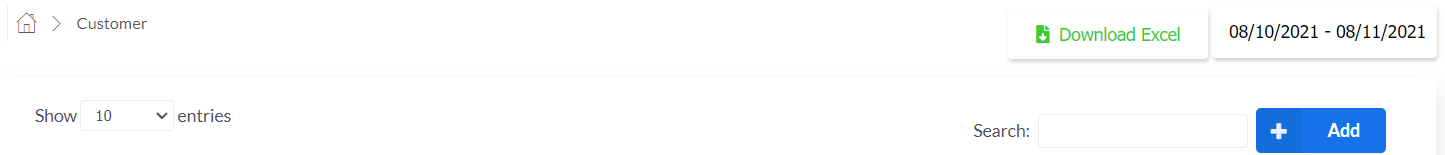
Graphical user interface, application, Teams

Description automatically generated

รูปภาพที่ 2-1 แสดงรายการปุ่ม และสีปุ่ม (Button Color)

* ปุ่ม**บันทึก** แสดงเป็น สีเขียว
* ปุ่ม**ยืนยัน, แก้ไข** แสดงเป็น สีส้ม
* ปุ่ม**ลบ** แสดงเป็น สีแดง
* ปุ่ม**พิมพ์, ส่งออกเอกสาร** แสดงเป็น สีขาว
* ปุ่ม**เคลียร์, ยกเลิก** แสดงเป็น สีเทา
* ปุ่ม**เพิ่ม** และ**อื่นๆ** แสดงเป็น สีน้ำเงิน, ฟ้า

## 2. การจัดวางตำแหน่งปุ่ม (Button Position)



Graphical user interface, application

Description automatically generated

รูปภาพที่ 2-2 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

* ปุ่ม**พิมพ์**, **ส่งออกเอกสาร** จัดวางตำแหน่งขวาบน
* ปุ่ม**เพิ่ม** จัดวางตำแหน่งขวาบน
* ไอคอน หรือรูปภาพแทนการดำเนินการ จัดเรียงตามลำดับดังรูปภาพที่ 2-2

Graphical user interface, application, Teams

Description automatically generated

รูปภาพที่ 2-3 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

* ปุ่ม**พิมพ์**, **ส่งออกเอกสาร** จัดวางตำแหน่งขวาบน
* ปุ่ม**เคลียร์**, **ยกเลิก** จัดวางตำแหน่งซ้ายล่าง
* ปุ่ม**ค้นหา**, **คัดลอก**, **บันทึก**, **ยืนยัน**, **แก้ไข**, **ลบ** จัดวางตำแหน่งขวาล่าง

## 3. การแสดงกล่องข้อความยืนยัน (Confirm Box)

Graphical user interface, text, application

Description automatically generated

* ปุ่ม**ตกลง** แสดงเป็นสีน้ำเงิน, ฟ้า จัดวางตำแหน่งขวาล่าง
* ปุ่ม**ยกเลิก** แสดงเป็นสีเทา จัดวางตำแหน่งซ้ายล่าง
* กรณีปุ่มตกลง สามารถแสดงเป็นปุ่มบันทึก, ยืนยัน, แก้ไข, ลบ หรือปุ่มดำเนินการอื่นๆ ที่ต้องการใช้ให้ สอดคล้องกับงาน โดยแสดงสีปุ่มตามดังรูปภาพที่ 2-1

## 4. การแสดงผลอื่นๆ

ประเด็นเกี่ยวกับมาตรฐานส่วนติดต่อผู้ใช้ที่ต้องกำหนดรูปแบบกันภายในทีมพัฒนา ให้เป็นมาตรฐานเดียวกัน ภายในระบบ หรือไซต์เดียวกัน มีดังนี้

* 1. **การแสดงข้อความแจ้งเตือน Form Validation** รูปแบบการแสดงผลแล้วแต่เทมเพลต แต่ควรเป็น รูปแบบเดียวกันทั้งระบบ หรือไซต์
  2. **การแสดง Tooltip** ใช้สำหรับอธิบายรายละเอียดของปุ่ม (button) หรือลิงค์(link) ซึ่งควรมีการอธิบาย รายละเอียดให้ชัดเจน กรณีต้องการคำอธิบายเพิ่มเติม และรูปแบบการแสดงผลแล้วแต่เทมเพลต แต่ควร เป็นรูปแบบเดียวกันทั้งระบบ หรือไซต์
  3. **การแสดง Placeholder** ใช้สำหรับอธิบายการกรอกข้อมูลในฟิลด์โดยแสดงเป็นข้อความพื้นหลังในฟิลด์ ซึ่งควรกำหนดว่าให้แสดงข้อความเป็นตัวอย่างข้อมูล ชื่อฟิลด์ข้อมูล หรืออื่นๆ โดยขึ้นอยู่กับความ เหมาะสมของงาน
  4. **การแสดง Icon หรือรูปภาพแทนการดำเนินการ** เลือกใช้ตามรูปแบบ และความเหมาะสมของเทมเพลต แต่ควรเป็นรูปแบบเดียวกันทั้งระบบ หรือไซต์
  5. **การแสดง Datepicker** และรูปแบบวันที่ (Date Format) ควรเป็นรูปแบบเดียวกันทั้งระบบ หรือไซต์