



TEAM 6

CODING **STANDARD**

TEMPORARY TAG
PERMISSION SYTEM

TEAM 6

SOFTWARE ENGINEERING
BURAPHA UNIVERSITY

CREATED BY TEAM 6 | 2021

การควบคุมการเปลี่ยนแปลงเอกสาร

ตารางที่ 0-1 การควบคุมการเปลี่ยนแปลงเอกสาร

เวอร์ชันปัจจุบัน		1.0.0		
วันที่แก้ไขเวอร์ชันล่าสุด		17 กรกฎาคม 2564		
ผู้จัดทำเอกสารนี้		ทีมพัฒนาซอฟต์แวร์ทีม 6		
เวอร์ชัน	วันที่	รายการที่แก้ไข	ผู้แก้ไข	หมายเหตุ
1.0.0	27 สิงหาคม 2563	สร้างเทมเพลต	นายปิยะศักดิ์	Team 0/63
2.0.0	17 กรกฎาคม 2564	นำมาใช้	ฐิติมา โพพิลา	

สารบัญ

หน้า

การควบคุมการเปลี่ยนแปลงเอกสาร.....	ข
สารบัญตาราง	ง
สารบัญรูปภาพ.....	จ
บทนำ	ฉ
ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (CODING STANDARDS).....	7
1. การตั้งชื่อไฟล์และคลาส	7
2. การตั้งชื่อฟังก์ชัน	8
3. การตั้งชื่อตัวแปร.....	10
4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล.....	11
5. การเขียนคอมเมนต์.....	12
ส่วนที่ 2 มาตรฐานส่วนติดต่อผู้ใช้งาน (UI STANDARDS).....	17
1. การแสดงสีปุ่ม	17
2. การแสดงตำแหน่งปุ่ม	18
3. การแสดงข้อความยืนยัน	18
4. การแสดงอื่น ๆ	19

สารบัญตาราง

ตารางที่

หน้า

0-1 การควบคุมการเปลี่ยนแปลงเอกสาร.....ก

สารบัญรูปภาพ

รูปภาพที่

หน้า

รูปภาพที่ 1 แสดงรายการปุ่ม และสีปุ่ม (Button Color).....	17
รูปภาพที่ 2 แสดงการจัดวางตำแหน่งปุ่ม (Button Position).....	17
รูปภาพที่ 3 แสดงการจัดวางตำแหน่งปุ่ม (Button Position).....	18
รูปภาพที่ 4 แสดงการแสดงกล่องข้อความยืนยัน (Confirm Box).....	18

บทนำ

ในการสร้างซอฟต์แวร์ จะทำอย่างไรให้ผู้ที่ทำการพัฒนาซอฟต์แวร์เข้าใจในส่วนของ Source Code ของระบบได้ตรงกัน หากเกิดปัญหาในภายหลัง หรือระบบถูกนำไปพัฒนาต่อ จะทำอย่างไรให้ผู้ที่ทำการแก้ไขครั้งถัดไป สามารถแก้ไขและเข้าใจซอฟต์แวร์นั้นได้ ทางทีมพัฒนาซอฟต์แวร์ได้เห็นถึงปัญหาเหล่านี้จึงจัดทำคู่มือมาตรฐานการพัฒนาซอฟต์แวร์ขึ้นมาเพื่อให้ผู้ที่ต้องการแก้ไขซอฟต์แวร์ต่อไปเข้าใจในส่วนของ Source Code ได้มากขึ้น อีกทั้งยังเกิดข้อผิดพลาดในการพัฒนาซอฟต์แวร์น้อยที่สุด และเพื่อให้เป็นแนวทางปฏิบัติเดียวกันของสมาชิกในทีมพัฒนา

ส่วนที่ 1 มาตรฐานการเขียนโปรแกรม (Coding Standards)

มาตรฐานการเขียนโปรแกรมนี้เป็นมาตรฐานที่กำหนดขึ้นเพื่อการพัฒนากระบวนการจัดการการวางของ (Temporary Tag Permission System : TTP) โดยใช้การพัฒนาสถาปัตยกรรมแบบ MVC (Model-View-Controller) โดยส่วนที่มีการกำหนดมาตรฐานประกอบด้วยไฟล์ในโฟลเดอร์ controllers, models, views, config และ helpers ดังนั้นจึงมีการกำหนดมาตรฐานการเขียนโปรแกรมแบ่งตามหัวเรื่อง และ MVC รวมถึง config, helpers และมาตรฐานเกี่ยวกับฐานข้อมูล โดยอ้างอิงจากคู่มือมาตรฐานการพัฒนาซอฟต์แวร์ (SDS) ดังนี้

1. การตั้งชื่อไฟล์และคลาส

1. การตั้งชื่อไฟล์และคลาสของ Controller

หลักการตั้งชื่อไฟล์และคลาส มีดังนี้

- 1) การตั้งชื่อโครงสร้างไคเรคทอรีของ Controller มีความเหมาะสม และสื่อความหมาย
- 2) ตั้งชื่อไฟล์ด้วยตัวอักษรอังกฤษตัวแรกเป็นตัวพิมพ์ใหญ่จากนั้นตัวพิมพ์เล็กทั้งหมด

และตัวอักษรติดกันทั้งหมด

- 3) ชื่อไฟล์ หรือคลาสจะต้องมีค่ามากกว่า 1 คำ ให้คั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้

เช่น Final_project.php

- 4) การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับชื่อไฟล์ และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่
- 5) คอนโทรลเลอร์หลักของระบบควรจะต้องตั้งชื่อด้วย ชื่อระบบ_controller

เช่น evaluation_controller

- 6) คอนโทรลเลอร์สำหรับการเชื่อมโยงข้อมูลกับระบบสารสนเทศอื่นควรตั้งชื่อด้วย

ชื่อระบบ หรืองาน_service เช่น evaluation_service

2. การตั้งชื่อไฟล์และคลาสของ Model

หลักการตั้งชื่อไฟล์และคลาส มีดังนี้

- 1) โมเดลต้องประกอบด้วย 2 ไฟล์คือ da และ m เท่านั้น
- 2) ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้

เช่น da_pos_title.php, m_pos_title.php

- 3) การตั้งชื่อคลาสต้องเป็นชื่อเดียวกันกับชื่อไฟล์ และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่

เช่น Da_pos_title, M_pos_title

- 4) ควรตั้งชื่อตามชื่อตารางในฐานข้อมูลเท่านั้น
- 5) โมเดลหลักของระบบควรตั้งชื่อด้วย ชื่อระบบ_model เช่น ev_model

3. การตั้งชื่อไฟล์ View

หลักการตั้งชื่อไฟล์ มีดังนี้

- 1) ชื่อไฟล์ต้องขึ้นต้นด้วย v_ ชื่อหน้า view นั้นๆ เช่น v_pos_title.php
- 2) ตั้งชื่อไฟล์ด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นคำด้วยเครื่องหมายขีดล่าง (_) ได้

เช่น v_pos_title.php

- 3) ชื่อ View ต้องสอดคล้องกับชื่อฟังก์ชัน หรือคลาส เช่น ฟังก์ชันชื่อ pos_input ควรตั้งชื่อไฟล์ว่า v_pos_index.php

- 4) ชื่อไฟล์ต้องลงท้ายด้วยการระบุการทำงาน เช่น v_position_insert.php เป็นการระบุว่าเพิ่มข้อมูล

2. การตั้งชื่อฟังก์ชัน

1.การตั้งชื่อฟังก์ชัน Controller

หลักการตั้งชื่อฟังก์ชัน ซึ่งจะต้องตั้งชื่อฟังก์ชันด้วยตัวอักษรพิมพ์เล็กเท่านั้น และคั่นด้วยเครื่องหมายขีดล่าง (_) ได้ เช่น position_input()

2.1.1 หมวดของฟังก์ชัน

- 1) ฟังก์ชันสำหรับการบันทึก หรือแก้ไขฐานข้อมูล
 - สำหรับบันทึกอย่างเดียว ตั้งชื่อลงท้ายด้วย _insert เช่น position_insert()
 - สำหรับแก้ไขอย่างเดียว ตั้งชื่อลงท้ายด้วย _edit เช่น position_edit()
- 2) ฟังก์ชันสำหรับการลบข้อมูลในฐานข้อมูล
 - ตั้งชื่อลงท้ายด้วย _delete เช่น position_delete()
- 3) ฟังก์ชันสำหรับการนำเข้าและอ่านข้อมูลจากไฟล์โดยเฉพาะ
 - ตั้งชื่อลงท้ายด้วย _import เช่น position_import()
- 4) ฟังก์ชันสำหรับส่งออกข้อมูลในรูปแบบต่างๆ
 - ส่งออกข้อมูลรูปแบบไฟล์ Word ตั้งชื่อลงท้ายด้วย _word เช่น

position_word()

- ส่งออกข้อมูลรูปแบบไฟล์ Excel ตั้งชื่อลงท้ายด้วย _excel เช่น
- position_excel()

- ส่งออกข้อมูลรูปแบบไฟล์ PDF ตั้งชื่อลงท้ายด้วย _pdf เช่น
- position_pdf()

- ส่งออกข้อมูลรูปแบบตัวอย่างก่อนพิมพ์ตั้งชื่อลงท้ายด้วย `_print` เช่น `position_print()`
- ส่งออกข้อมูลหลายรูปแบบในฟังก์ชันเดียว ตั้งชื่อลงท้ายด้วย `_export` เช่น `position_export()`
- 5) ฟังก์ชันสำหรับการเชื่อมโยงข้อมูลกับระบบสารสนเทศที่เกี่ยวข้อง
- การรับข้อมูล ตั้งชื่อขึ้นต้นด้วย `get_service_` ชื่อข้อมูล เช่น `ge_service_position()`
- การส่งข้อมูล ตั้งชื่อขึ้นต้นด้วย `post_service_` ชื่อข้อมูล เช่น `post_service_position`

2. การตั้งชื่อฟังก์ชัน Model

ฟังก์ชันในไฟล์ `Da` ประกอบด้วย ฟังก์ชันหลัก 4 ฟังก์ชัน เท่านั้น ได้แก่ `insert()`, `update()`, `delete()` และ `get_by_key()`

2.2.1 ฟังก์ชันในไฟล์ `M`

- 1) ฟังก์ชันอื่น ๆ นอกเหนือจากฟังก์ชันในไฟล์ `da` เช่น การคิวรีข้อมูลต่าง ๆ การอัปเดตบางฟิลด์ การลบโดยไม่อ้างอิงหลัก
- 2) ตั้งชื่อฟังก์ชันด้วยตัวอักษรพิมพ์เล็กเท่านั้น และค้นคำด้วยเครื่องหมายขีดล่าง (`_`) ได้เช่น `get_all()`
- 3) โครงสร้างของชื่อฟังก์ชัน `action_data_by_condition(for_something)`
 - `action` คือ การกระทำ ตัวอย่างเช่น `get`, `search`, `count`, `update`
 - `data` คือ ข้อมูลที่ต้องการ ตัวอย่างเช่น `position`, `form`, `level`
 - `by_condition` คือ เงื่อนไขการค้นหา เช่น `by_posid`, `by_posname`
 - `for_something` คือ ถูกเรียกใช้เพื่อฟังก์ชัน โมดูล หรือเงื่อนไขโดยเฉพาะ (ถ้าสำคัญ) เช่น `for_ajax`

2.2.2 หมวดของฟังก์ชัน

- 1) ฟังก์ชันสำหรับคิวรีดึงข้อมูล
 - สำหรับดึงข้อมูลทั่วไป ไม่มีการค้นหา หรือค้นหาแบบมีเงื่อนไขไม่ซับซ้อน ได้แก่ ดึงข้อมูลทั้งหมด (`get_all`) ข้อมูลที่ขึ้นต่อกัน เช่น ตำแหน่งขึ้นอยู่กับเวลาที่เลือก เป็นต้น ให้มีการขึ้นต้นด้วย `get_` ชื่อข้อมูล เช่น `get_position()`
- 2) ฟังก์ชันสำหรับคิวรีโดยเรียกใช้ `SQL function`

- ให้ตั้งชื่อฟังก์ชันขึ้นต้นด้วย SQL function เช่น count_person(), sum_salary(), max_salary() และ avg_salary()
- 3) ฟังก์ชันสำหรับคิวรีเพื่อตรวจสอบข้อมูล
 - สำหรับ return ค่า เป็น binary เช่น 0,1 TRUE, FALSE Y,N ให้ตั้งชื่อขึ้นต้นด้วย check เช่น check_active_position()
- 4) ฟังก์ชันสำหรับอัปเดตบางฟิลด์
 - กรณีอัปเดต 1 - 2 ฟิลด์ให้ตั้งชื่อว่า update_ชื่อฟิลด์ที่ต้องการอัปเดต เช่น update_firstname(), update_lastname()
 - กรณีอัปเดตมากกว่า 2 ฟิลด์ให้ตั้งชื่อว่า update_ชื่อการทำงานนั้น ๆ เช่น update_position_level() คือการอัปเดตฟิลด์ของ position ว่าอยู่ level ไหน
- 5) ฟังก์ชันสำหรับลบ โดยไม่อ้างอิง PK
 - ให้ตั้งชื่อว่า delete_by_ชื่อฟิลด์ เช่น delete_by_posid(), delete_by_level_and_pos_id()

3. การตั้งชื่อตัวแปร

1. ตัวแปรแทน Object ของ Model

การตั้งชื่อตัวแปรต้องขึ้นต้นด้วย m ต่อด้วยชื่อย่อของตาราง เช่น m_position ใช้ชื่อตัวแปรว่า mpos

2. ตัวแปร Array

การตั้งชื่อตัวแปรที่บ่งบอกว่าเป็นชุดของ Array ให้ขึ้นต้นด้วย arr ตัวอย่างเช่น \$arr_ps, \$arr_dp

3. ตัวแปรนับรอบของลูป

กรณีต้องการตั้งตัวแปร เพื่อใช้นับรอบของลูป สามารถใช้ตัวแปรในรูปแบบ Single ได้ แต่จะต้องสื่อความหมาย เช่น

- 1) ใช้ตัวแปร \$i เพื่อนับบรรทัดของลูป
- 2) ใช้ตัวแปร \$i, \$j และ \$k หรือ \$x, \$y และ \$z ร่วมกัน กรณีมีลูป มากกว่า 1 ลูป

ได้ตามความเหมาะสม

4. การจัดทำมาตรฐานเกี่ยวกับฐานข้อมูล

1. การตั้งชื่อฐานข้อมูล

ข้อบังคับ

- 1) ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
- 2) คั่นคำด้วยเครื่องหมายขีดล่าง (_)

5.3.1 หลักการตั้งชื่อฐานข้อมูล

- 1) ตั้งชื่อขึ้นต้นด้วยชื่อย่อของระบบ เช่น evs
- 2) ตามด้วยชื่อข้อมูล หรือชื่อย่อข้อมูล เช่น evs_person, ev_position

4.2 การตั้งชื่อตาราง

ข้อบังคับ

- 1) ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
- 2) คั่นด้วยเครื่องหมายขีดล่าง (_)

5.3.1 หลักการตั้งชื่อตาราง

- 1) ขึ้นต้นด้วยชื่อย่อของระบบในฐานข้อมูล ความยาวไม่เกิน 4 ตัวอักษร เช่น ev
- 2) ตามด้วยชื่อโมดูลการทำงาน หรือต้องบ่งบอกว่าใช้เก็บข้อมูลนั้น ๆ

ตัวอย่าง เช่น ev_position และ ev_level

4.3 การตั้งชื่อฟิลด์

ข้อบังคับ

- 1) ต้องเป็นตัวอักษรพิมพ์เล็กทั้งหมด
- 2) คั่นด้วยเครื่องหมายขีดล่าง (_)

4.3.1 หลักการตั้งชื่อฐานข้อมูล

- 1) ต้องขึ้นต้นด้วยชื่อย่อของตาราง ความยาวไม่เกิน 4 ตัวอักษร (ไม่รวมชื่อฐานข้อมูล) เช่น ตาราง ev_pos ชื่อย่อเป็น pos หรือตาราง ev_admin ชื่อย่อเป็น adm
- 2) หลังชื่อย่อของตาราง ให้ระบุชื่อฟิลด์นั้น ๆ โดยมีชื่อฟิลด์ที่ต้องบังคับใช้

ในรูปแบบเดียวกัน ดังนี้

- ชื่อฟิลด์ที่เป็นคีย์หลัก ต้องลงท้ายด้วย id เช่น pos_id
- ชื่อฟิลด์ที่เป็นความหมายหรือข้อมูลหลักของตาราง ต้องลงท้ายด้วย

detail เช่น pos_detail และ ps_level

- ชื่อฟิลด์ที่บ่งบอกถึงลำดับชั้น ต้องลงท้ายด้วย level เช่น pos_level
- ชื่อฟิลด์ FK จากตารางอื่น ให้ใช้ชื่อเดิมมาต่อท้าย เช่น pos_ps_id

และ pos_lv_id

ข้อยกเว้น

ชื่อย่อของตารางต้องมีความยาวมากกว่า 4 ตัวอักษรได้ กรณีที่ไม่สามารถลดคำให้น้อยลงได้ (ถ้าจำเป็น)

4.4 การเขียนคอมเมนต์ (Comment) ของตารางและฟิลด์

หลักการเขียนคอมเมนต์ มีดังนี้

- 1) ตาราง ให้นิยามความหมายว่า ตารางเก็บข้อมูลอะไร หรือใช้สำหรับทำอะไร
เช่น ตาราง ev_position คือ ตารางเก็บข้อมูลตำแหน่ง
- 2) ฟิลด์ให้นิยามความหมายว่าใช้เก็บข้อมูลอะไร เช่น
 - pos_id คือ รหัสตำแหน่ง
 - pos_name คือ ชื่อตำแหน่ง
- 3) การระบุตัวอย่างของข้อมูล หากฟิลด์นั้นมีตัวอย่างของข้อมูลชัดเจน ให้ใส่ต่อท้ายในเครื่องหมายวงเล็บด้วย เช่น
 - pos_status คือ สถานะของระเบียบวาระ (Y=ใช้งาน, N=ไม่ใช้งาน)
- 4) ฟิลด์ที่อ้างอิงจากฟิลด์อื่น (FK) ให้อธิบายความหมายเดียวกันกับตารางตั้งต้น (PK) และต่อท้าย ด้วยว่ามาจากตารางไหนและ/หรือฐานข้อมูลไหน (ระบุชื่อฐานข้อมูลด้วย หากอยู่คนละฐานข้อมูล) เช่น
 - pos_person_id คือ รหัสบุคลากร (ตาราง ev_person)
 - pos_form_id คือ รหัสฟอร์ม (ตาราง ev_form)

5. การเขียนคอมเมนต์

5.1 คอมเมนต์คลาสของ Controller และ Model

ข้อบังคับ

- 1) ให้เขียนคอมเมนต์คลาสกำกับในไฟล์คลาสทุกไฟล์ไม่มีข้อยกเว้น
- 2) เขียนคอมเมนต์คลาสไว้บรรทัดแรกของไฟล์
- 3) เขียนคอมเมนต์คลาสด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ ยกเว้นเป็นชื่อตัวแปรหรือข้อความเฉพาะ

5.1.1 หลักการเขียนคอมเมนต์

- 1) บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /*
- 2) บรรทัดที่ 2 ระบุชื่อไฟล์ หรือชื่อคลาส เช่น Base_position
- 3) บรรทัดที่ 3 ระบุการทำงานของคลาสแบบคร่าว ๆ เช่น Base Data of Position Management
- 4) บรรทัดที่ 4 ระบุชื่อผู้สร้างไฟล์ หรือคลาส หลังหัวข้อ @author เช่น @author Piyasak
- 5) บรรทัดที่ 5 ระบุวันที่สร้างไฟล์ หรือคลาส ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหัวข้อ @Create Date เช่น @Create Date 2563-08-29
- 6) บรรทัดที่ 6 ถ้ามีการแก้ไข หรือมีการโค้ดเพิ่มเติม ให้ระบุวันที่แก้ไขไฟล์ หรือคลาส ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหัวข้อ @Update Date เช่น @Create Date 2563-08-29
- 7) บรรทัดที่ 6 ใช้เครื่องหมายปิดคอมเมนต์ คือ */

หมายเหตุ :

- 1) แต่ละบรรทัดให้ใส่เครื่องหมาย * เว้นวรรค 1 ครั้งนำหน้าเสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 6)
 - 2) ข้อความคอมเมนต์หลังหัวข้อ @author ให้เคาะ tab 1 ครั้ง
- ตัวอย่างการคอมเมนต์คลาส
- ```
/*
 * Member
 * Member Model of Member
 * @author : Somsak
 * @Create Date : 2563-8-25
 */
```

## 5.2 คอมเมนต์ฟังก์ชันใน Controller, Model และ Helper

### ข้อบังคับ

- 1) เขียนคอมเมนต์กำกับทุกฟังก์ชัน ไม่มีข้อยกเว้น
- 2) เขียนคอมเมนต์ฟังก์ชันไว้ด้านบน ก่อนประกาศฟังก์ชันนั้น ๆ
- 3) เขียนคอมเมนต์ฟังก์ชันด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยตัวอักษรพิมพ์ใหญ่ ยกเว้นเป็นชื่อตัวแปรหรือข้อความเฉพาะ

## 5.2.1 หลักการเขียนคอมเมนต์ส่วนของคลาส

- 1) บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /\*
- 2) บรรทัดที่ 2 ระบุฟังก์ชัน เช่น position\_insert
- 3) บรรทัดที่ 3 ระบุการทำงานของฟังก์ชันแบบคร่าว ๆ เช่น Insert position in database after form add
- 4) บรรทัดที่ 4 ระบุข้อมูลเข้า (Input) หลังหวัข้อ @input กรณีไม่มีให้ใส่เครื่องหมายขีด (-) เช่น @input position\_name, position\_level
- 5) บรรทัดที่ 5 ระบุข้อมูลที่ส่งกลับคืน (Output) หลังหวัข้อ @output กรณีไม่มีให้ใส่เครื่องหมายขีด (-) เช่น @output The last insert id (pos\_id)
- 6) บรรทัดที่ 6 ระบุชื่อผู้สร้างฟังก์ชัน หลังหวัข้อ @author เช่น @author Tanadon
- 7) บรรทัดที่ 7 ระบุวันที่สร้างฟังก์ชัน ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหวัข้อ @Create Date เช่น @Create Date 2563-08-29
- 8) บรรทัดที่ 8 ถ้ามีการแก้ไข หรือมีการโค้ดเพิ่มเติม ระบุวันที่แก้ไขฟังก์ชัน ในรูปแบบปี พ.ศ. - เดือน - วัน หลังหวัข้อ @Update Date เช่น @Update Date 2563-08-29
- 9) บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเมนต์ คือ \*/

### หมายเหตุ :

- 1) แต่ละบรรทัดให้ใส่เครื่องหมาย \* และเว้นวรรค 1 ครั้งนำหน้าเสมอ (ยกเว้นบรรทัดที่ 1 และบรรทัดที่ 8)
  - 2) ข้อความคอมเมนต์หลังหวัข้อ @input, @output, @author ให้เคาะ tab 1 ครั้ง ตัวอย่างการคอมเมนต์คลาส
- ```
/*
 * position_insert
 * Insert position in database
 * @input position_name, position_level
 * @output The last insert id (pos_id)
 * @author Piyasak
 * @Create Date 2563-09-01
 */
```

5.3 คอมเมนต์ส่วนของ View

ข้อบังคับ

- 1) ให้เขียนคอมเมนต์ส่วนของวิวกำกับทุกไฟล์ไม่มีข้อยกเว้น
- 2) เขียนคอมเมนต์ส่วนของวิวไว้บรรทัดแรกสุดของไฟล์
- 3) เขียนคอมเมนต์ส่วนของวิวด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่ ยกเว้นเป็นชื่อตัวแปรหรือข้อความเฉพาะ
- 4) ระบุคอมเมนต์ส่วนของวิวให้อยู่ในรูปแบบของ PHP ยกเว้นไฟล์วิวที่เป็น HTML ให้ใช้การคอมเมนต์แบบ HTML ได้

5.3.1 หลักการเขียนคอมเมนต์ส่วนของ View

- 1) บรรทัดที่ 1 ใช้เครื่องหมายเปิดคอมเมนต์ คือ /*
- 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 กรณีไม่มีให้ใส่เครื่องหมายขีด (-) เช่น @output Input form of position
- 6) บรรทัดที่ 6 ระบุชื่อผู้สร้างไฟล์ หลังหัวข้อ @author เช่น @author Somchai
- 7) บรรทัดที่ 7 ระบุวันที่สร้างไฟล์ ในรูปแบบ ปีพ.ศ. - เดือน - วัน หลังหัวข้อ @CreateDate เช่น @CreateDate 2563-08-30
- 8) บรรทัดที่ 8 ถ้ามีการแก้ไข หรือโค้ดเพิ่มเติม ให้ระบุวันที่แก้ไขไฟล์ ในรูปแบบปี พ.ศ. - เดือน - วัน หลังหัวข้อ @UpdateDate เช่น @UpdateDate 2563-08-30
- 9) บรรทัดที่ 8 ใช้เครื่องหมายปิดคอมเมนต์คือ */

หมายเหตุ :

- 1) แต่ละบรรทัด ให้ใส่เครื่องหมาย * เว้นวรรค 1 ครั้งนำหน้าเสมอ (ยกเว้นบรรทัดที่ 1 และ 8)
 - 2) ข้อความคอมเมนต์หลังหัวข้อ @input, @output, @author ให้เคาะ tab 1 ครั้ง
- ตัวอย่างการคอมเมนต์ส่วนของ View

```
/*
```

```
* v_position_input
```


- * Display input form of position for add
- * @input Array of name (arr_name)
- * @output Input form of position
- * @author Piyasak
- * @Create Date 2563-09-01
- */

5.4 คอมเมนต์บรรทัดเดียวหรือตัวแปรต่าง ๆ

หลักการเขียนคอมเมนต์ มีดังนี้

- 1) ให้คอมเมนต์ด้านบนก่อนประกาศตัวแปร หรือก่อนบรรทัดนั้น ๆ
- 2) ขึ้นต้นด้วยเครื่องหมายคอมเมนต์ คือ //
- 3) เว้นวรรค 1 ครั้ง ตามด้วยคอมเมนต์ที่ต้องการ โดยสามารถคอมเมนต์เป็นภาษาไทย

หรืออังกฤษได้ตามความเหมาะสม

ตัวอย่างการคอมเมนต์

// กำหนดค่าเริ่มต้นของ i

\$i=0;

5.5 คอมเมนต์ระบุขอบเขตส่วนการทำงานนั้น ๆ

ข้อบังคับ

- 1) เขียนคอมเมนต์ด้วยตัวอักษรภาษาอังกฤษเท่านั้น และขึ้นต้นด้วยอักษรตัวพิมพ์ใหญ่

- 2) ต้องระบุคอมเมนต์ไว้ทั้ง 2 ส่วน คือ ส่วนบนและส่วนท้ายของการทำงานนั้น ๆ

5.5.1 หลักการเขียนคอมเมนต์ส่วนบน

- 1) เปิด - ปิดคอมเมนต์ไว้ด้านบนก่อนเริ่มการทำงานนั้น ๆ โดยใช้เครื่องหมายคอมเมนต์แบบ PHP คือ /* และ */ ตามลำดับ หรือเครื่องหมายคอมเมนต์ใน HTML ก็ได้

- 2) ส่วนของข้อความให้เว้นวรรค 1 ครั้ง แล้วขึ้นต้นด้วย Start แล้วตามด้วยอธิบายส่วนของการทำงานนั้น ๆ

ตัวอย่างการคอมเมนต์

/* Start Form input of position */

...










...

/* End Form input of position */

ส่วนที่ 2 มาตรฐานส่วนติดต่อผู้ใช้งาน (UI Standards)

มาตรฐานส่วนติดต่อผู้ใช้นี้ใช้ธีมเฟรมแบบ Bootstrap เป็นต้นแบบในการกำหนดมาตรฐานโดยรูปแบบของการแสดงผลจะมีลักษณะเหมือนกัน ดังนั้นจึงจัดทำมาตรฐานนี้ขึ้น เพื่อให้การแสดงผลส่วนติดต่อผู้ใช้ (User Interface) เป็นไปตามมาตรฐานเดียวกัน โดยอ้างอิงคู่มือมาตรฐานการพัฒนาซอฟต์แวร์ (SDS)

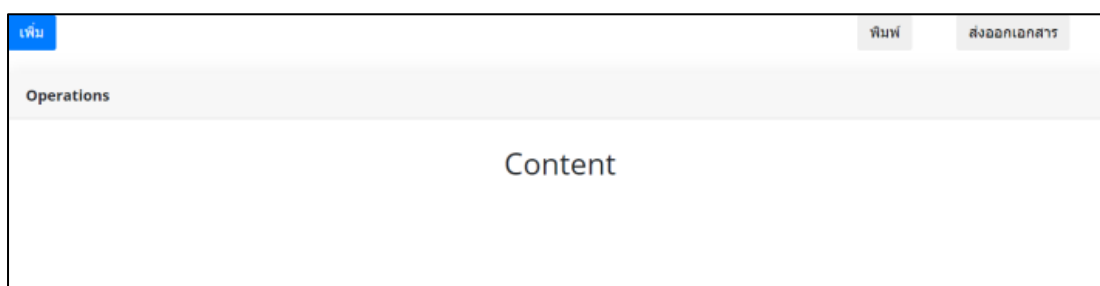
1. การแสดงสีปุ่ม (Button Color)

Operations			
	Normal	 	Exports
	Warning		Clear
 	Danger	  	Others

รูปภาพที่ 1 แสดงรายการปุ่ม และสีปุ่ม (Button Color)

- ปุ่มบันทึก แสดงเป็น สีเขียว
- ปุ่มแก้ไข แสดงเป็น สีส้ม
- ปุ่มลบ ปุ่มยกเลิก แสดงเป็น สีแดง
- ปุ่มพิมพ์ ปุ่มส่งออกเอกสาร แสดงเป็น สีขาว
- ปุ่มเคลียร์ และปุ่มย้อนกลับ แสดงเป็น สีเทา
- ปุ่มค้นหา ปุ่มเพิ่ม ปุ่มคัดลอก และอื่น ๆ แสดงเป็น สีน้ำเงิน และสีฟ้า

2. การจัดวางตำแหน่งปุ่ม (Button Position)



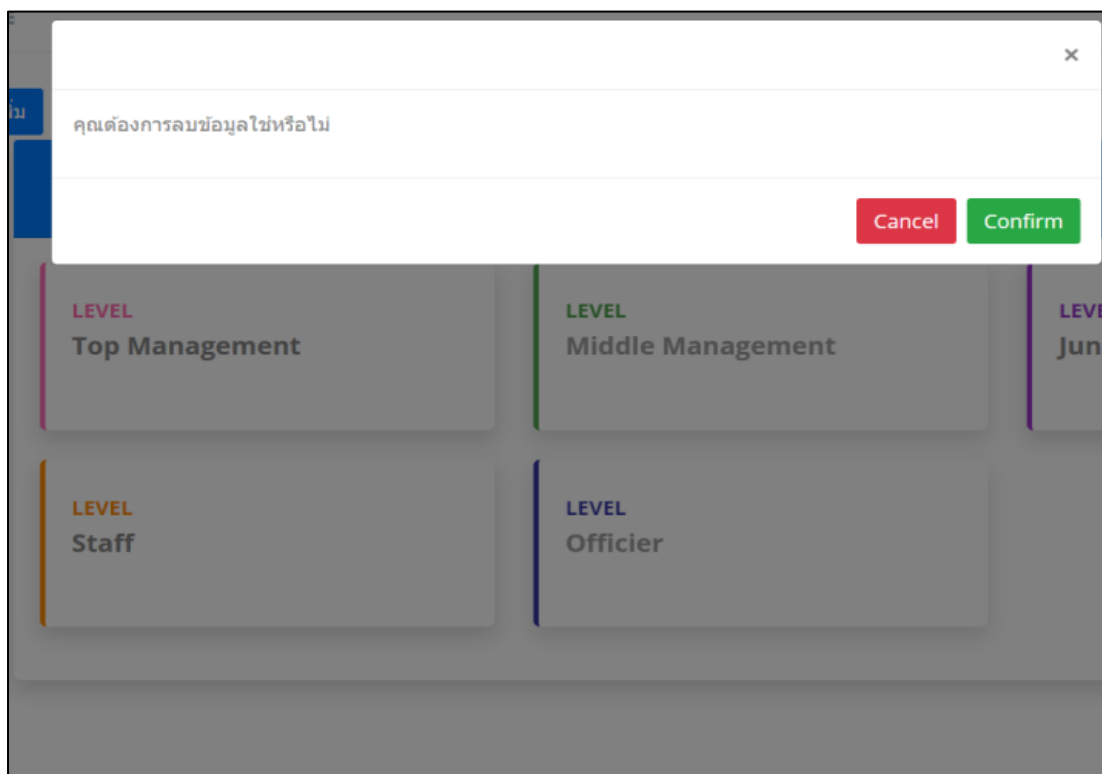
รูปภาพที่ 2 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)



รูปภาพที่ 3 แสดงการจัดวางตำแหน่งปุ่ม (Button Position)

- ปุ่มพิมพ์ และปุ่มส่งออกเอกสาร จัดวางตำแหน่ง ขวาบน
- ปุ่มเพิ่ม จัดวางตำแหน่ง ซ้ายบน
- ไอคอน หรือรูปภาพแทนการดำเนินการ จัดเรียงตามลำดับดังรูปภาพที่ 2
- ปุ่มพิมพ์ ปุ่มส่งออกเอกสาร จัดวางตำแหน่ง ขวาบน
- ปุ่มเคลียร์ จัดวางตำแหน่ง ซ้ายล่าง
- ปุ่มบันทึก ปุ่มยืนยัน ปุ่มแก้ไข ปุ่มลบ ปุ่มยกเลิก จัดวางตำแหน่ง ขวาล่าง

3. การแสดงข้อความยืนยัน (Confirm Box)



รูปภาพที่ 4 แสดงการแสดงกล่องข้อความยืนยัน (Confirm Box)

- ปุ่มตกลง แสดงเป็นสีเขียว จัดวางตำแหน่ง ขวาล่าง
- ปุ่มยกเลิก แสดงเป็นสีแดง จัดวางตำแหน่ง ขวาล่าง
- กรณีปุ่มตกลง สามารถแสดงเป็นปุ่มบันทึก ปุ่มยืนยัน ปุ่มแก้ไข ปุ่มลบ หรือปุ่ม

ดำเนินการอื่น ๆ ที่ต้องการใช้ให้สอดคล้องกับงาน โดยแสดงสีปุ่มตามดังรูปที่ 1

4. การแสดงผลอื่น ๆ

- การแสดงข้อความแจ้งเตือน Form Validation รูปแบบการแสดงผลแล้วแต่เทมเพลต แต่ควรเป็นรูปแบบเดียวกันทั้งระบบ
- การแสดง Tooltip ใช้สำหรับอธิบายรายละเอียดของปุ่ม (button) หรือลิงค์(link) ซึ่งจะมีการอธิบายรายละเอียดให้ชัดเจน กรณีต้องการคำอธิบายเพิ่มเติม และรูปแบบของการแสดงผลแล้วแต่เทมเพลต แต่ควรเป็นรูปแบบเดียวกันทั้งระบบ
- การแสดง Placeholder ใช้สำหรับอธิบายการกรอกข้อมูลในฟิลด์ โดยแสดงเป็นข้อความพื้นหลังในฟิลด์ ซึ่งควรมีการกำหนดว่าให้แสดงข้อความเป็นตัวอย่างข้อมูล ชื่อฟิลด์ข้อมูล หรืออื่น ๆ โดยขึ้นอยู่กับความ เหมาะสมของงาน
- การแสดง Icon หรือรูปภาพแทนการดำเนินการ เลือกใช้ตามรูปแบบ และความเหมาะสมของเทมเพลต แต่ควรเป็นรูปแบบเดียวกันทั้งระบบ
- การแสดง Datepicker และรูปแบบวันที่ (Date Format) ควรเป็นรูปแบบเดียวกันทั้งระบบ