



## รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบบริหารแบบคำขอภายในองค์กร โดยใช้เฟรมเวิร์ก ONEWEB 4.0  
WORKFORCE SYSTEM USING FRAMEWORK ONEWEB 4.0

นายภาณุพงศ์ จำรัสพงษ์

ภาควิชา วิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560



## รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบบริหารแบบคำขอภายในองค์กร โดยใช้เฟรมเวิร์ก ONEWEB 4.0  
WORKFORCE SYSTEM USING FRAMEWORK ONEWEB 4.0

นายภาณุพงศ์ จำรัสพงษ์

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา ระบบบริหารแบบคำขอภายในองค์กร โดยใช้เฟรมเวิร์ก ONEWEB 4.0

ชื่อ-สกุล นักศึกษา	นายภาณุพงศ์ จำรัสพงษ์
คณะ	วิศวกรรมศาสตร์
ภาควิชา	วิศวกรรมคอมพิวเตอร์
สาขา	วิศวกรรมสารสนเทศ
ชื่อ-สกุล อาจารย์นิเทศ	ผศ.ดร.วันวิสา ชัชวงษ์
ชื่อ-สกุล ผู้นิเทศงาน	นายไพโรจน์ ชันแก้ว
ชื่อสถานประกอบการ	บริษัท อีฟวาแลนท์ จำกัด

### บทคัดย่อ

เนื่องจากการทำงานในองค์กร มีความจำเป็นต้องมีระบบการจัดการทรัพยากรมนุษย์และอุปกรณ์อิเล็กทรอนิกส์ที่มีประสิทธิภาพ เพื่อการจัดการที่เป็นระบบภายในองค์กร โครงการนี้เป็นการพัฒนาโปรแกรมสำหรับนำมาใช้ในการบริหารแบบคำขอภายในองค์กร เพื่อเพิ่มประสิทธิภาพและอำนวยความสะดวกการจัดการภายในองค์กร ส่วนที่พัฒนาประกอบด้วย การคำนวณชั่วโมงการทำงาน การตรวจสอบความสมบูรณ์ของข้อมูล การแบ่งผู้ใช้งานระบบ การดึงข้อมูลจากฐานข้อมูลมาแสดง และการสร้างอีเมล ซึ่งโครงการที่พัฒนานี้ จะทำให้การบริหารองค์กรมีประสิทธิภาพดียิ่งขึ้น และเป็นการทดสอบเฟรมเวิร์ก ONEWEB 4.0 ขององค์กรไปภายในตัว เพื่อหาข้อผิดพลาดและแก้ไขให้สามารถนำไปใช้ในองค์กรได้อย่างมีประสิทธิภาพ

**Co-operative Title:** WORKFORCE SYSTEM USING FRAMEWORK ONEWEB 4.0  
**Student Intern Name:** Mr. Panupong Jamraspong  
**Faculty:** Engineering  
**Department:** Computer Engineering  
**Major:** Information Engineering  
**Advisor Name:** Asst. Prof. Vanvisa Chutchavong  
**Mentor Name:** Mr. Praisan Kahnkaew  
**Company:** Avalant Co., Ltd

## ABSTRACT

It is essential for organization to have the great effective resource management of human and electronic equipment for the purpose of arrange the system in the organization. This project is the developing program for the internal application form in administration which increase efficiency and easiness in the organization. The part of development includes to calculate working hours, to verify of completed data, to divide network users, to extracting data from the database and generate E-mail which this project has better effectiveness. Moreover, ONEWEB 4.0, the framework testing, of internal organization, finding and solving the fault are able to adapt in the organization effectively.

## กิตติกรรมประกาศ

รายงานวิชาสหกิจศึกษาฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์จาก ผศ.ดร.วันวิสา ชัชวงษ์ อาจารย์ที่ปรึกษาที่คอยชี้แนะแนวทางการปฏิบัติสหกิจศึกษา นอกจากนี้ตลอดระยะเวลา การปฏิบัติ สหกิจศึกษายังได้รับความช่วยเหลือจากหลายๆ ฝ่าย ดังนี้

ขอขอบคุณ นายไพโรจน์ ชันแก้ว พนักงานที่ปรึกษารวมทั้งบุคลากรบริษัท อีฟวาแลนท์ จำกัด และ ท่านอื่นที่ไม่ได้กล่าวนามที่กรุณาเสียสละเวลาอันมีค่าให้คำปรึกษาและคำแนะนำในการทำงาน ผู้จัดทำซาบซึ้ง ในความกรุณาอย่างยิ่ง

สุดท้ายนี้ขอขอบคุณครอบครัวของผู้จัดทำเป็นอย่างสูงที่เป็นกำลังใจและแรงผลักดันในทุกๆ ด้านเสมอ มา

ภาณุพงศ์ จำรัสพงษ์

## สารบัญ

### หน้า

บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญรูป.....	ฉ
สารบัญตาราง.....	ณ
บทที่ 1 บทนำ.....	1
1.1 รายละเอียดสถานประกอบการ.....	1
1.2 ความเป็นมาและความสำคัญ.....	2
1.3 วัตถุประสงค์โครงการ.....	3
1.4 ขอบเขตของโครงการ.....	3
1.5 แผนเวลาโครงการ.....	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	5
2.1 ภาษาจาวา.....	5
2.2 jQuery.....	12
2.3 SQL.....	13
2.4 PostgreSQL.....	14
2.5 จาวากับการเขียนโปรแกรมฐานข้อมูลด้วย JDBC.....	15
2.6 หลักการของระบบฐานข้อมูล.....	18
2.7 สัญลักษณ์ Flowchart และความหมาย.....	21
บทที่ 3 วิธีดำเนินงาน.....	24
3.1 เครื่องมือที่ใช้ในการพัฒนา.....	24
3.2 ขั้นตอนการเตรียมการ.....	27
3.3 การคำนวณชั่วโมงการทำงาน.....	32

## สารบัญ (ต่อ)

	หน้า
3.4 การดึงข้อมูลจากฐานข้อมูลมาแสดง.....	37
3.5 การตรวจสอบความสมบูรณ์ของข้อมูล.....	41
3.6 การแบ่งผู้ใช้งานระบบ.....	43
3.7 การสร้างอีเมล.....	46
บทที่ 4 ผลการดำเนินงาน.....	49
4.1 ผลลัพธ์หน้าการคำนวณชั่วโมงการทำงาน.....	49
4.2 ผลลัพธ์หน้าดึงข้อมูลจากฐานข้อมูลมาแสดง.....	55
4.3 ผลลัพธ์หน้าการตรวจสอบความสมบูรณ์ของข้อมูล.....	56
4.4 ผลลัพธ์หน้าการแบ่งผู้ใช้งานระบบ.....	58
4.5 ผลลัพธ์หน้าการสร้างอีเมล.....	60
บทที่ 5 สรุปผลการปฏิบัติงาน.....	61
5.1 ปัญหาที่พบ.....	61
5.2 แนวทางในการแก้ปัญหา.....	61
5.3 ข้อเสนอแนะ.....	61
เอกสารอ้างอิง.....	62

## สารบัญรูป

	หน้า
รูปที่ 1.1 สถานที่ตั้งบริษัท อีฟวาแลนท์ จำกัด .....	2
รูปที่ 2.1 สัญลักษณ์ของภาษา Java.....	5
รูปที่ 2.2 แสดงให้เห็นถึงรายละเอียดของรถที่นำมาสร้างเป็นคลาส.....	8
รูปที่ 2.3 สัญลักษณ์ของ jQuery.....	12
รูปที่ 2.4 สัญลักษณ์ของ PostgreSQL.....	14
รูปที่ 2.5 แสดงการติดต่อกับฐานข้อมูล.....	15
รูปที่ 2.6 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....	19
รูปที่ 2.7 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....	20
รูปที่ 3.1 แสดงหน้าจอ Framework ONEWEB 4.0.....	24
รูปที่ 3.2 แสดงสัญลักษณ์โปรแกรม Eclipse.....	25
รูปที่ 3.3 แสดงสัญลักษณ์โปรแกรม DbVisualizer.....	26
รูปที่ 3.4 แสดงตัวอย่างหน้าต่างต่างของ App Designer.....	27
รูปที่ 3.5 แสดงตัวอย่างหน้าต่างต่างของ Front End.....	28
รูปที่ 3.6 แสดงตัวอย่างการใช้ Action กับไฟล์ที่ผูก.....	28
รูปที่ 3.7 แสดงตัวอย่างการผูกไฟล์.....	29
รูปที่ 3.8 แสดงแผนผังการปรับแต่งด้วยไฟล์ Java.....	29
รูปที่ 3.9 แสดงหน้าต่างของ Front End หน้าคำนวณชั่วโมงการทำงาน.....	32
รูปที่ 3.10 แสดงหน้าต่างกรอกค่าคำนวณชั่วโมงการทำงาน (ก่อนปรับแต่ง).....	32
รูปที่ 3.11 แสดงผังงานของการคำนวณชั่วโมงการทำงานหน้าที่ 1.....	33
รูปที่ 3.12 แสดงผังงานของการคำนวณชั่วโมงการทำงานหน้าที่ 2.....	34
รูปที่ 3.13 แสดงการเรียกใช้ฟังก์ชัน loadUpdateManyManual().....	37
รูปที่ 3.14 แสดงหน้าต่างของแถบด้านล่างในหน้าหลักไม่มี employee name.....	37
รูปที่ 3.15 แสดงโค้ดการเรียกใช้งานระดับโมดูล.....	38
รูปที่ 3.16 แสดงข้อมูลทั้งหมดที่กรอกในหน้า Add.....	38
รูปที่ 3.17 แสดงแผนภาพการทำงานของกรอดึงข้อมูลจากฐานข้อมูล.....	39

## สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.18	แสดงโครงสร้าง Hashmap ที่ใช้สำหรับเก็บข้อมูล.....	40
รูปที่ 3.19	แสดงโค้ดเรียกใช้งาน Hashmap.....	40
รูปที่ 3.20	แสดงหน้าเว็บแอปพลิเคชันที่ต้องการตรวจสอบความสมบูรณ์ของข้อมูล.....	41
รูปที่ 3.21	แสดงโค้ดการเรียกใช้งานระดับเอนทิตี.....	42
รูปที่ 3.22	แสดง Hashmap ก่อนใส่ข้อมูลของ Status และ Asset No.....	42
รูปที่ 3.23	แสดง Hashmap หลังใส่ข้อมูลของ Status และ Asset No.....	42
รูปที่ 3.24	แสดงหน้าเว็บแอปพลิเคชันที่ต้องการแบ่งผู้ใช้งานระบบ.....	43
รูปที่ 3.25	แสดงโค้ดที่เรียกดูข้อมูลในโปรแกรม Dbvisualizer .....	44
รูปที่ 3.26	แสดงข้อมูลผู้ใช้งานที่มีในระบบทั้งหมด.....	44
รูปที่ 3.27	แสดงโค้ดแบ่งผู้ใช้งานระบบ.....	45
รูปที่ 3.28	แสดงโค้ดคำสั่ง SQL ก่อนและหลังเพิ่มเงื่อนไข.....	45
รูปที่ 3.29	แสดงแผนภาพการทำงานของการสร้างอีเมล.....	46
รูปที่ 3.30	แสดงคำสั่งให้เชื่อมต่อกับฐานข้อมูลโดยใช้ JDBC.....	47
รูปที่ 3.31	แสดงเทมเพลตของข้อความในฐานข้อมูล.....	47
รูปที่ 3.32	แสดงโค้ดที่ผู้งานส่งข้อมูลเพื่อไปสร้างอีเมล.....	48
รูปที่ 4.1	แสดงหน้าคำนวณชั่วโมงการทำงาน (ก่อนกรอกข้อมูล).....	49
รูปที่ 4.2	แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Start Join date.....	49
รูปที่ 4.3	แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Latest Join date.....	50
รูปที่ 4.4	แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Latest time.....	51
รูปที่ 4.5	แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Roll out date.....	51
รูปที่ 4.6	แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Roll time.....	52
รูปที่ 4.7	แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Latest time.....	52
รูปที่ 4.8	แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Latest Join date.....	53
รูปที่ 4.9	แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Start time.....	53
รูปที่ 4.10	แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Start Join date.....	54

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.11 แสดงหน้าคำนวณชั่วโมงการทำงานเมื่อกรอกข้อมูลครบ.....	54
รูปที่ 4.12 แสดงหน้าต่างของแถบด้านล่างในหน้าหลักก่อน Add ข้อมูล.....	55
รูปที่ 4.13 แสดงหน้าต่างของหน้า Add พนักงาน.....	55
รูปที่ 4.14 แสดงหน้าเว็บหลังจากที่ Add พนักงาน.....	56
รูปที่ 4.15 แสดงหน้าเว็บที่ต้องการตรวจสอบความสมบูรณ์ของข้อมูล.....	57
รูปที่ 4.16 แสดงหน้าเว็บที่มีข้อความแจ้งเตือนเมื่อใส่ข้อมูลไม่ครบ.....	57
รูปที่ 4.17 แสดงหน้าต่างแก้ไขข้อมูล.....	57
รูปที่ 4.18 แสดงหน้าเว็บหลักหลังจากแก้ไขข้อมูล.....	58
รูปที่ 4.19 แสดงหน้าเว็บแอปพลิเคชันที่ต้องการแบ่งผู้ใช้งานระบบ.....	58
รูปที่ 4.20 แสดงภาพการล็อกอินเข้าสู่ระบบด้วย admin.....	59
รูปที่ 4.21 แสดงภาพการล็อกอินเข้าสู่ระบบด้วย req1.....	59
รูปที่ 4.22 แสดงข้อมูลที่ผู้ใช้งานต้องการแทนค่าในอีเมล.....	60
รูปที่ 4.23 แสดงอีเมลที่แทนค่าข้อมูลเรียบร้อยแล้ว.....	60
รูปที่ 4.24 แสดงอีเมลบน Internet Explorer.....	60

## สารบัญตาราง

	หน้า
ตารางที่ 1.1 ระยะเวลาการดำเนินงาน.....	4
ตารางที่ 2.1 คำที่สงวนไว้ในภาษาจาวา.....	6
ตารางที่ 2.2 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง.....	19
ตารางที่ 2.3 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม.....	20
ตารางที่ 2.4 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....	21

# บทที่ 1

## บทนำ

### 1.1 รายละเอียดสถานประกอบการ

#### 1.1.1 ประวัติความเป็นมาของหน่วยงาน

บริษัท อีฟวาแลนท์ จำกัด ก่อตั้งขึ้นเมื่อปี พ.ศ.2545 เริ่มต้นจากให้บริการทางด้านซอฟต์แวร์แก๊งค์กร และเป็น IBM Partner ในปี 2553 บริษัทได้รับการรับรองมาตรฐานการพัฒนาซอฟต์แวร์ CMMI Level 5 เป็นแห่งแรกของประเทศไทย อีกทั้งในปีดังกล่าวยังได้รับรางวัลจากการประกวดซอฟต์แวร์ระดับประเทศ TICTA Award และรางวัล เจ้าฟ้าไอทีเป็นจำนวน 2 รางวัล ในการประกวด 2 ประเภท ได้แก่ Tools & Infrastructure และ Financial Industry Applications ในขณะเดียวกันก็ได้พัฒนา OneWeb ระบบแพลตฟอร์มสำหรับการพัฒนาซอฟต์แวร์แบบ Configuration-based ที่ช่วยให้ทุกองค์กรสามารถพัฒนาซอฟต์แวร์ ที่มีคุณภาพได้ตามแนวคิดของการทำ Agile โดยไม่ต้องลงทุนกับทีมพัฒนาซอฟต์แวร์ภายในองค์กร จำนวนมากอีกต่อไป [1,2]

#### 1.1.2 วิสัยทัศน์และพันธกิจ

##### วิสัยทัศน์ (Vision)

บริษัท อีฟวาแลนท์ จำกัด ต้องการเป็นบริษัทชั้นนำในการให้คำปรึกษาทางด้านโซลูชันระบบสารสนเทศ (Information Technology) ในระดับภูมิภาคเอเชีย

##### พันธกิจ (Mission)

เพื่อช่วยเหลือลูกค้าของเราออกแบบโซลูชันทั้งทางด้านธุรกิจและด้านระบบสารสนเทศ เพื่อให้เกิดผลตอบแทนจากการลงทุนผ่านผลกำไรและผลการดำเนินงานที่ดีขึ้น

#### 1.1.3 โครงสร้างบริษัท

กรรมการผู้จัดการ นายอัศวพล บุญวรเศรษฐ์

แผนกมีทั้งหมด 6 แผนก ได้แก่

1. Products หน้าที่สร้างผลิตภัณฑ์เพื่อตอบสนองความต้องการของลูกค้า
2. Solutions หน้าที่หาวิธีการที่สามารถตอบโจทย์อะไรให้แก่ลูกค้าได้บ้าง
3. Professional Service ทำหน้าที่พัฒนาซอฟต์แวร์แก่ลูกค้าตามที่ลูกค้าต้องการ

4. Sales & Mktg ทำหน้าที่ขายและนำเสนอผลิตภัณฑ์ของบริษัทแก่ลูกค้า
5. Back Office ทำหน้าที่จัดการบริหารทรัพยากรมนุษย์
6. Auditor ทำหน้าที่ตรวจสอบคุณภาพซอฟต์แวร์และทำบัญชี

#### 1.1.4 ที่ตั้งของหน่วยงาน

เลขที่ 20 ชั้น 15 อาคาร บุผจิต ถนนสาทรเหนือ แขวงสีลม เขตบางรัก กรุงเทพมหานคร 10500 [4]



รูปที่ 1.1 สถานที่ตั้งบริษัท อฟวาแลนท์ จำกัด

#### 1.1.5 ตำแหน่งและหน้าที่งานที่นักศึกษาได้รับมอบหมาย

ตำแหน่ง Java Developer

หน้าที่งานที่ได้รับมอบหมาย Developer

#### 1.1.6 ระยะเวลาปฏิบัติงาน

ปฏิบัติงานสหกิจศึกษาเป็นระยะเวลา 17 สัปดาห์

นับตั้งแต่วันที่ 7 สิงหาคม 2560 – 30 พฤศจิกายน 2560

## 1.2 ความเป็นมาและความสำคัญ

การบริหารจัดการแบบคำขอภายในองค์กร มีความสำคัญต่อองค์กรเป็นอย่างมาก เนื่องจากมีความจำเป็นที่ต้องมีระบบการจัดการทรัพยากรมนุษย์และอุปกรณ์อิเล็กทรอนิกส์ อันส่งผลต่อการทำงานโดยรวมขององค์กรและการสร้างผลผลิตให้แก่องค์กร จึงจัดทำโครงการนี้เพื่อเป็นการพัฒนาระบบบริหารแบบคำขอภายในองค์กร โดยใช้เฟรมเวิร์ก ONEWEB 4.0 ให้มีความสะดวกและมีประสิทธิภาพดียิ่งขึ้น

### 1.3 วัตถุประสงค์โครงการ

1. เพื่อเพิ่มประสิทธิภาพให้ระบบที่สร้างด้วยเฟรมเวิร์ก ONEWEB 4.0 มีความสมบูรณ์มากขึ้น
2. เพื่อทดสอบเฟรมเวิร์ก ONEWEB 4.0 ของบริษัท หาจุดบกพร่องและแก้ไขให้มีประสิทธิภาพยิ่งขึ้น

### 1.4 ขอบเขตของโครงการ

โครงการพัฒนาระบบบริหารแบบคำขอภายในองค์กร โดยใช้เฟรมเวิร์ก ONEWEB 4.0 มีส่วนประกอบดังนี้

#### 1.4.1 การคำนวณชั่วโมงการทำงาน

จะทำการคำนวณชั่วโมงในแบบคำขอที่ต้องการพนักงาน ในการทำแต่ละโครงการ แล้วบันทึกเข้าระบบเพื่อให้ฝ่ายจัดการทรัพยากรมนุษย์ดำเนินการหาพนักงานที่มีความพร้อมที่จะปฏิบัติการในช่วงเวลาดังกล่าวตามที่ระบุไว้

#### 1.4.2 การดึงข้อมูลจากฐานข้อมูลมาแสดง

จะทำการส่งข้อมูลที่เป็นรหัสพนักงานแล้วไปค้นหาชื่อของพนักงานในฐานข้อมูลแล้วส่งกลับมาจากนั้นไปแสดงแทนในฟิลด์ที่ได้กำหนดไว้ให้

#### 1.4.3 การตรวจสอบความสมบูรณ์ของข้อมูล

จะทำการตรวจสอบข้อมูลในฟิลด์ที่จำเป็นต้องกรอกข้อมูล ก่อนที่จะบันทึกข้อมูลลงฐานข้อมูล หากว่ากรอกข้อมูลไม่ครบจะมีข้อความแสดงเตือนผู้ใช้งาน

#### 1.4.4 การแบ่งผู้ใช้งานระบบ

จะทำการดึงข้อมูลที่เกี่ยวข้องกับผู้ใช้ระบบมาจากฐานข้อมูล เพื่อให้สามารถแก้ไขหรือดูข้อมูลเฉพาะส่วนของตัวเองเท่านั้น ไม่สามารถเห็นของผู้ใช้งานคนอื่นได้

#### 1.4.5 การสร้างอีเมล

จะทำการส่งค่าที่ต้องการจะแทนเข้าไป แล้วทำการแทนที่ค่านั้นๆในข้อความ เพื่อให้ได้อีเมลที่มีรายละเอียดตรงตามที่ต้องการจะใช้งาน

## 1.5 แผนเวลาโครงการ

ตารางที่ 1.1 ระยะเวลาการดำเนินงาน

กิจกรรมดำเนินงาน	ระยะเวลา			
	สิงหาคม	กันยายน	ตุลาคม	พฤศจิกายน
1. เลือกหัวข้อปัญหา				
2. ทำแบบฝึกและอบรม ONEWEB 4.0				
3. ดำเนินการ Customize บน ONEWEB 4.0 ตามระยะเวลาที่กำหนด				
4. ตรวจสอบและแก้ไขส่วนที่ Customize ทั้งหมด				
5. จัดทำรูปเล่มสหกิจศึกษา พร้อมทั้งจัดทำรูปแบบการนำเสนอ				

## 1.6 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ทำให้เฟรมเวิร์ก ONEWEB 4.0 ของบริษัทมีประสิทธิภาพมากขึ้น
- 1.5.2 ช่วยอำนวยความสะดวก แก่ผู้ใช้งานในระบบบริหารแบบคำขอภายในองค์กร
- 1.5.3 สามารถนำความรู้จากการทำโครงการ ไปประยุกต์ใช้กับโครงการอื่นๆต่อไปได้

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 ภาษาจาวา



รูปที่ 2.1 สัญลักษณ์ของภาษา Java

ภาษาจาวา (Java programming language) เป็นภาษาโปรแกรมเชิงวัตถุ (อังกฤษ: Object Oriented Programming) พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่ ซัน ไมโครซิสเต็มส์ ภาษาจาวาถูกพัฒนาขึ้นในปี พ.ศ. 2534 (ค.ศ. 1991) โดยเป็นส่วนหนึ่งของ โครงการกรีน (the Green Project) และสำเร็จออกสู่สาธารณะในปี พ.ศ. 2538 (ค.ศ. 1995) ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส (C++) โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แต่ว่ามีปัญหาทางลิขสิทธิ์ จึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทนและแม้ว่าจะมีชื่อคล้ายกัน แต่ภาษาจาวาไม่มีความเกี่ยวข้องใดๆ กับภาษาจาวาสคริปต์ (JavaScript) ปัจจุบันมาตรฐานของภาษาจาวาดูแลโดย Java Community Process ซึ่งเป็นกระบวนการอย่างเป็นทางการที่อนุญาตให้ผู้สนใจเข้าร่วมกำหนดความสามารถในจาวาแพลตฟอร์มได้

จุดมุ่งหมายหลัก 4 ประการ ในการพัฒนาจาวา คือ

1. ใช้ภาษาโปรแกรมเชิงวัตถุ
2. ไม่ขึ้นกับแพลตฟอร์ม (สถาปัตยกรรม และ ระบบปฏิบัติการ)
3. เหมาะกับการใช้ในระบบเครือข่าย พร้อมมีไลบรารีสนับสนุน
4. เรียกใช้งานจากระยะไกลได้อย่างปลอดภัย [3]

### 2.1.1 Keywords in Java

Keyword คือคำที่สงวนไว้ในภาษาจาวา หมายความว่าไม่สามารถนำคำเหล่านี้ไปประกาศเป็นชื่อตัวแปร เมธอด หรือว่าคลาสได้ เพราะว่า Keyword ถูกใช้โดยคอมไพเลอร์เพื่อให้ทำงานได้สมบูรณ์ ตารางข้างล่างนี้เป็นลิสต์ของ Keyword ในภาษา Java [4]

ตารางที่ 2.1 คำที่สงวนไว้ในภาษาจาวา

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	While

### 2.1.2 การเขียนโปรแกรมเชิงวัตถุ

การเขียนโปรแกรมเชิงวัตถุ (Object-oriented programming: OOP) เป็นรูปแบบการเขียนโปรแกรมที่ใช้แนวคิดของออบเจ็ค ซึ่งในออบเจ็คนั้นจะประกอบไปด้วยสมาชิก 2 แบบ คือ ตัวแปรซึ่งเป็นข้อมูลหรือกำหนดคุณลักษณะของออบเจ็ค และเมธอด เป็นฟังก์ชันการทำงานที่ออบเจ็คสามารถทำได้ ในการสร้างออบเจ็คนั้นจะอยู่ภายใต้การกำหนดของคลาส โดยคลาสคือการกำหนดว่าออบเจ็คจะมีสมาชิกอะไรบ้าง และกำหนดการทำงานของเมธอด ดังนั้นออบเจ็คจะสร้างจากคลาส เรียกออบเจ็คว่า instances ของคลาส

ภาษาต่างๆ เป็นจำนวนมากสนับสนุนการเขียนโปรแกรมเชิงวัตถุ เช่น Java C++ C# Python PHP Ruby และภาษาอื่นๆ ซึ่ง Java ถือว่าเป็นภาษาของออบเจ็คอย่างเต็มรูปแบบ การเขียนโปรแกรมเชิงวัตถุ

นี่เป็นการพัฒนารูปแบบของการเขียนโปรแกรมไปอีกขั้นจากการเขียนโปรแกรมแบบเดิมที่เคยเขียนคือ Procedural programming

### คุณสมบัติของ OOP

การเขียนโปรแกรมเชิงวัตถุ มีข้อดีและข้อได้เปรียบจากการเขียนโปรแกรมแบบเดิมมากมาย ซึ่งสิ่งที่สำคัญที่สุดของมันคือการนำโค้ดกลับมาใช้ใหม่ และคุณสมบัติอื่นๆ ดังนี้

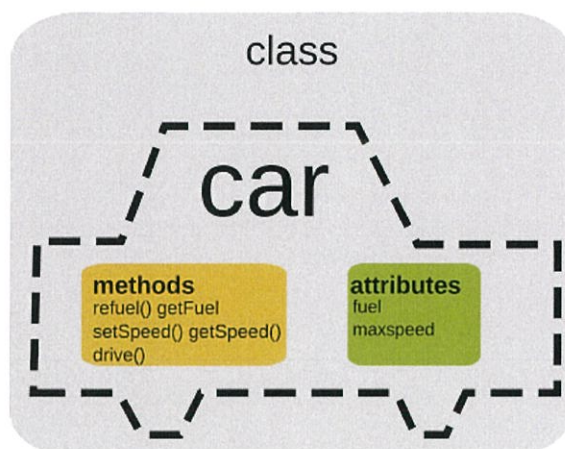
- Code reuse: การนำโค้ดกลับมาใช้ใหม่ ได้อย่างมีประสิทธิภาพ
- Maintenance: ง่ายต่อการบำรุงรักษาและปรับปรุง แต่ละคลาสจะมีขอบเขตการทำงานและเมธอดของตัวเอง
- Classes: คลาสที่สร้างขึ้นนั้น สามารถนำไปสร้างเป็นออบเจกต์ได้ไม่จำกัด และแต่ละออบเจกต์มีสมาชิกเป็นของตัวเอง
- Inheritance: คุณสมบัติการสืบทอด คลาสสามารถสืบทอดจากคลาสอื่นได้ ซึ่งเป็นการ reuse code และพัฒนาต่อเติมความสามารถจากคลาสเดิมที่มีอยู่
- Polymorphism: คุณสมบัติของออบเจกต์ที่หลากหลายรูปแบบ ซึ่งสามารถจัดการได้จาก Base คลาสของตัวเอง
- Encapsulation: เป็นคุณสมบัติการห่อหุ้ม เป็นแนวคิดแบบนามธรรมที่ไม่จำเป็นต้องรู้ว่าเมธอดหรือในคลาสนั้นทำงานยังไง แต่สามารถทำงานได้สำเร็จและให้ผลลัพธ์ได้
- Abstraction: คือการอธิบายการทำงานของคลาสและเมธอดในแบบนามธรรม เพื่อนำไปใช้สำหรับออบเจกต์ที่มีการทำงานแตกต่างกัน เช่น บอกว่าสัตว์สามารถเคลื่อนที่ การเคลื่อนที่คือการบอกอย่างเป็นนามธรรม เพราะว่าสัตว์แต่ละตัวมีวิธีการเคลื่อนที่ที่แตกต่างกัน เช่น สุนัขจะเดิน นกจะบิน ส่วนเต่าจะคลาน เป็นต้น

### คลาสและออบเจกต์

ส่วนประกอบของการเขียนโปรแกรมเชิงวัตถุที่สำคัญคือ คลาสและออบเจกต์ ในตอนแรกเราต้องสร้างหรือประกาศคลาสขึ้นมา ซึ่งประกอบไปด้วยตัวแปร และเมธอดในการทำงาน หลังจากนั้นจะนำไปสร้างออบเจกต์ การสร้างคลาสก็เหมือนการออกแบบรูปแบบของวัตถุอย่างหนึ่งที่เราจะสร้างขึ้นมา เช่น คุณจะสร้างรถขึ้นมาเราก็ต้องเขียนแบบ เราเรียกสิ่งนี้ว่าคลาส และเรานำไปสร้างรถ ซึ่งก็คือ ออบเจกต์

ในตัวอย่างเป็นรูปที่แสดงให้เห็นถึงรายละเอียดของรถที่นำมาสร้างเป็นคลาส จะเห็นได้ว่าจะมี attribute เมธอดเพื่ออธิบายเกี่ยวกับรถ ซึ่งรถก็จะมีคุณสมบัติของมัน เช่น จำนวนเชื้อเพลิง ความเร็วสูงสุด และอื่นๆ

ส่วนของการทำงาน จะเป็นการเติมน้ำมัน การเพิ่มและลดความเร็ว การขับเคลื่อนรถออกไป และสตาร์ทหรือหยุดเครื่องยนต์ เป็นต้น ซึ่งสิ่งเหล่านี้จะถูกกำหนดไว้ในคลาส



รูปที่ 2.2 แสดงให้เห็นถึงรายละเอียดของรถที่นำมาสร้างเป็นคลาส

### 2.1.3 Collection

คือ ออบเจ็กต์ที่สามารถเก็บข้อมูลของออบเจ็กต์อื่นในตัวเองได้แบบไดนามิกส์ และเป็นเครื่องมืออำนวยความสะดวกที่จะช่วยให้โปรแกรมเมอร์เขียนโปรแกรมได้รวดเร็วและมีประสิทธิภาพกว่า ในภาษาจาวานั้นมี Collection มากมายที่สามารถใช้ในการเขียนโปรแกรมซึ่ง Collection ถูก Implement มาจาก Interfaces Collection ซึ่งเป็นไลบรารีของภาษาที่อยู่ใน Package java.util และใน Collection ส่วนมากนั้นสืบทอดมาจาก Abstract class และมีการ Implement บาง Interfaces รวมกัน

Collection นั้นเป็นการเก็บข้อมูลแบบไดนามิกส์ นั้นหมายความว่าสามารถเพิ่มข้อมูลเข้าไปใน Collection ได้ไม่จำกัด โดยที่ไม่ต้องกำหนดขนาดสูงสุดในการเก็บข้อมูลล่วงหน้า Collection มีเมธอดที่ทำงานด้วยอัลกอริทึมที่มีประสิทธิภาพเป็นจำนวนมากให้สามารถใช้งานได้ทันที โดยที่ไม่ต้องเขียนขึ้นอีก ยกตัวอย่างเช่น การเรียงข้อมูลภายใน Collection เป็นต้น Collection นั้นเป็น Generic type ซึ่งมีรูปแบบเป็น  $G<T>$  จะแนะนำ Collection ในภาษาจาวาเบื้องต้น ดังนี้

- การใช้ Array list ในภาษาจาวา
- การใช้ Map ในภาษาจาวา

#### การใช้ Array list ในภาษาจาวา

Array list คือ Collection ที่ใช้เก็บข้อมูลเหมือนอาร์เรย์ปกติ แต่สิ่งที่แตกต่างกันคือ Array list สามารถย่อและขยายของตัวเองได้อัตโนมัติตามข้อมูลที่มีอยู่ภายในนั่นเอง ทำให้มันยืดหยุ่นในการทำงานเพราะ

ไม่ต้องกำหนดขนาดสูงสุดล่วงหน้าเหมือนกับอาเรย์ปกติ ซึ่งถูกสืบทอดมาจาก Abstract list คลาส และมีการ Implement Collection Interfaces สำหรับเมธอดการทำงานของตัวมันเอง จะใช้ Iterator ในการวนอ่านข้อมูลจากอาเรย์ และการใช้งานเมธอดต่างๆ ของ Array list ในการทำงานจะใช้ Array list ในภาษาจาวานั้นต้องทำการ import java.util.ArrayList ไลบรารี

```
ArrayList<String> names = new ArrayList<String>();
```

เป็นคำสั่งในการประกาศอาเรย์ลิสต์จากคลาส ArrayList และในภายในเครื่องหมาย <String> นั้นหมายถึง Array list จะใช้สำหรับเก็บของออบเจ็ค String หมายความว่าสามารถเก็บออบเจ็คอื่นก็ได้ โดยการกำหนดคลาสของออบเจ็คในเครื่องหมาย <>

- เมธอด add() ใช้เพิ่มข้อมูลลงไป Array list ข้อมูลที่เพิ่มลงไปนั้นต้องเป็น String เพราะว่าได้กำหนด Array list สำหรับเก็บข้อมูล String ในตอนประกาศ
- เมธอด size() สำหรับนับจำนวนสมาชิก
- เมธอด get() ใช้ใส่ข้อมูลใน Index เป็นพารามิเตอร์ ซึ่ง Index ของ Array list นั้นจะเหมือนกับอาเรย์ปกติคือข้อมูลที่เพิ่มเข้าอันแรกจะมี Index เริ่มจาก 0 และเพิ่มไปที่ละ 1
- เมธอด indexOf() ใช้รับค่าของ Index ของข้อมูลที่ต้องการในอาเรย์
- เมธอด set() ในคำสั่งเหล่านี้จึงเป็นการเปลี่ยนข้อมูลในอาเรย์
- เมธอด remove() เป็นการนำข้อมูลออกจากอาเรย์ ผลลัพธ์ของการลบก็คือ Index ภายในอาเรย์ หลังจากตำแหน่งที่ถูกลบจะเลื่อนลงมาหนึ่ง สามารถลบใส่พารามิเตอร์เป็น Index
- เมธอด clear() ใช้สำหรับลบข้อมูลทั้งหมดออกจากอาเรย์
- เมธอด isEmpty() เพื่อตรวจสอบว่า Collection นั้นว่างหรือมีข้อมูลหรือไม่

#### การใช้ Maps ในภาษาจาวา

Maps เป็น Collection ที่มีรูปแบบในการเก็บข้อมูลเป็นแบบ Key Value และเป็นคลาสที่ Implement มาจาก java.util.Map Interface ในภาษาจาวานั้นมี Map หลายประเภท แต่ที่ใช้บ่อยๆ คือ HashMap และ TreeMap โดยที่ HashMap จะเก็บข้อมูลโดยไม่มีการเรียงของ Key และ Value ส่วน TreeMap จะมีการเรียงข้อมูลภายใน ทำให้การเข้าถึงข้อมูลภายใน Map จะเร็วกว่า ในทางกลับกัน HashMap การเพิ่มข้อมูลจะเร็วกว่า

การใช้งาน Map ในการเก็บข้อมูล โดยมีชื่อ Key และค่า Value จะสร้างออบเจ็กต์ด้วยคลาส HashMap และใช้ Iterator สำหรับวนอ่านค่า Key และ Value ภายใน Map เพื่อแสดงข้อมูลภายใน Map และใช้เมธอดที่สำคัญ

```
HashMap<String, String> country = new HashMap<>();
```

เป็นคำสั่งในการประกาศ Map โดยใช้คลาส HashMap สำหรับเก็บข้อมูลของประเทศ ทั้ง Key และ Value ของ Map นี้เป็นประเภทข้อมูลแบบ String

- เมธอด put() ใช้เพิ่มข้อมูลลงไปใน Hashmap
- เมธอด get() เป็นเมธอดสำหรับรับค่าของออบเจ็กต์ภายใน Map โดยมี Key ของออบเจ็กต์เป็นพารามิเตอร์
- การวนอ่านค่า Key Value ภายใน Map จะใช้ HashMap.Entry Iterator ที่ใช้สำหรับทำงานกับ Map ประเภทต่าง และแสดงผล Key และ Value ออกทางหน้าจอ
- เมธอด remove() เป็นเมธอดในการนำออบเจ็กต์ของ Key ที่ระบุออกจาก Map
- เมธอด containsKey() ใช้เพื่อตรวจสอบว่ามี Key ดังกล่าวอยู่ใน Map หรือไม่
- เมธอด containsValue() ใช้เพื่อตรวจสอบว่ามี Value อยู่หรือไม่
- เมธอด size() ใช้สำหรับหาขนาด
- เมธอด clear() ใช้สำหรับลบข้อมูลทั้งหมดออกจาก Map

#### 2.1.4 Exception

คือกรณีที่โปรแกรมพยายามจะทำงานบางอย่าง แต่เกิดข้อผิดพลาดขึ้น แล้วโปรแกรมไม่สามารถจัดการข้อผิดพลาดนั้นได้ ซึ่งทำให้เกิด exception ขึ้น และส่งผลทำให้โปรแกรมหยุดทำงาน

Exception เกิดขึ้นในขณะที่โปรแกรมทำงาน ยกตัวอย่างเช่น โปรแกรมกำลังจะเปิดไฟล์ขึ้นมา แต่ไฟล์ที่ต้องการไม่มีอยู่ เป็นต้น รูปแบบการใช้ exception ในภาษาจาวาเป็นดังนี้

```
try {
    // try to do something
} catch (Exception1 ex1) {
    // handle for exception 1
}
...
} catch (ExceptionN exN) {
```

```

        // handle for exception N
    } finally {
        // Always proceed this block whether
        // an exception is thrown or not
    }

```

ในบล็อกคำสั่งของ try เป็นส่วนของโปรแกรมที่อาจจะทำให้เกิด exception ขึ้น และในแต่ละ catch บล็อกเป็นการจัดการกับ exception แต่ละแบบ และบล็อก finally โปรแกรมจะเข้ามาทำงานเสมอไม่ว่าจะเกิด exception ในบล็อกของคำสั่ง try หรือไม่ก็ตาม

ในภาษาจาวานั้นมีไลบรารีมาตรฐานของ exception ที่สามารถให้เราจัดการกับข้อผิดพลาดประเภทต่างๆ ได้ ซึ่ง extend มาจากคลาส Exception สำหรับในทฤษฎีนี้ จะยกตัวอย่างการใช้ในบางส่วนเท่านั้น

- เมธอด reader.nextInt() นั้นใช้สำหรับอ่านค่าตัวเลข ในการใส่ข้อมูลเข้ามาเป็นไปได้ที่ผู้ใช้จะใส่อย่างอื่นที่ไม่ใช่ตัวเลข เช่น ข้อความ ซึ่งจะทำให้เกิด InputMismatchException ขึ้น

การใช้คำสั่ง try catch เพื่อจัดการ exception

เพื่อจัดการและเตรียมรับมือกับ exception ที่อาจจะเกิดขึ้น เราใช้คำสั่ง try เป็นบล็อกสำหรับสิ่งที่พยายามจะทำ และคำสั่ง catch ในการตรวจจับ exception ที่จะเกิดขึ้นและจัดการกับสิ่งนั้น

การใช้คำสั่ง Finally

ในบล็อกของคำสั่ง finally นั้นจะประมวลผลเสมอ ไม่ว่าจะเกิด exception หรือไม่ จะใช้สำหรับการทำงานที่สำคัญ เช่น การปิดไฟล์ หรือปิดการเชื่อมต่อของ stream

- เมธอด e.printStackTrace(); ที่ใช้สำหรับ IDE ในการบอกรายละเอียดเกี่ยวกับข้อผิดพลาดให้กับเรา เช่น บรรทัดที่ทำให้เกิดความผิดพลาดและคลาสที่รัน ใช้สำหรับการ debug โปรแกรม

#### 2.1.5 DAO (DATA ACCESS OBJECT)

DAO หรือ data access object เป็นแนวคิดของทาง Microsoft เพื่อให้ทางนักพัฒนาระบบ (Programmer) เข้าใช้งาน Database ได้โดยง่าย กล่าวคือ การใช้ DAO คือการแยกส่วนการทำงานเป็น 2 ส่วน โดยส่วนแรกจะทำการ Query ข้อมูลตรงไปพื้นฐานข้อมูล และอีกส่วนคือการเรียกใช้งาน Function ของส่วนแรกที่ทำการ Query ไปยังฐานข้อมูล

สามารถใช้ DAO เพื่อแยก Data Layer ออกจาก Business Layer ให้ชัดเจน จะทำให้การพัฒนาาระบบง่ายขึ้นและสะดวกต่อการใช้งาน หรือจะพูดได้อีกอย่างหนึ่งว่า DAO จะคอยเป็นตัวกลางในการเชื่อมข้อมูลระหว่างฐานข้อมูลกับหน้าจอที่ใช้ในการแสดงผลข้อมูล เช่น หากมีฐานข้อมูลที่เก็บข้อมูลสินค้าเอาไว้แล้วให้ก็ทำการดึงข้อมูลมาแสดงผลที่หน้าจอ ทำได้ 2 แบบ

แบบที่ 1 คือเรียกมาแสดงผลตรงๆจากฐานข้อมูล (โดยการดึงข้อมูลสินค้ามาแสดงผลแบบปกติเขียนโปรแกรมเพื่อติดต่อฐานข้อมูลโดยตรง)

แบบที่ 2 ทำผ่านตัว DAO โดยทำการสร้าง Function ที่ทำการดึงข้อมูลสินค้าเอาไว้ Function (โดย Function นี้ก็จะทำการดึงข้อมูลสินค้าแบบปกติ) แล้วหน้าแสดงผลหน้าจอ ก็ทำการเรียกผ่าน Function ที่ได้สร้างขึ้นไว้ก่อนหน้านี้นี้ เท่านั้นจะได้ข้อมูลสินค้ามาแสดงที่หน้าจอ (ก็เปรียบได้ว่า DAO ทำตัวเองเป็น Interface ให้ระหว่างหน้าจอที่แสดงผลกับฐานข้อมูล) [5]

### ข้อดีและข้อเสีย

ข้อดีของการแยก data access code ออกจาก business code ก็คือ ผู้ที่เขียน business code ไม่จำเป็นต้องรู้เกี่ยวกับฐานข้อมูลหรือการเขียน sql statement การแยก data access layer ออกมาก็ทำให้ไม่ต้องเขียน code ซ้ำๆ หรือ copy + paste และทำให้ความซับซ้อนของ code ลดลงอย่างมาก ส่วนข้อเสีย คือ medium-scale และ large-scale application มีโอกาสมากที่จะมีการ query หรือ update ข้อมูลเดียวกันมากกว่า 1 function

## 2.2 jQuery



รูปที่ 2.3 สัญลักษณ์ของ jQuery

jQuery (เจควีรี่) เปิดตัวครั้งแรกในงานบาร์แคมป์นิวยอร์ก โดย จอห์น เรซิก (John Resig) เมื่อวันที่ 14 มกราคม พ.ศ. 2549 ตัวโค้ดของเจควีรี่มีลิขสิทธิ์และสัญญาอนุญาตแบบโอเพนซอร์ซ โดยใช้สัญญาอนุญาตของ GFDL และ MIT License

jQuery คือ JavaScript Library ซึ่งถูกออกแบบมาเพื่อให้การเขียน JavaScript นั้นมีความสะดวกและง่ายขึ้น เพราะว่าการนำ JavaScript เอาไปประยุกต์กับงานจำพวกเว็บ (Client-side JavaScript) นั้นเป็นสิ่งที่ยุ่งยาก ไม่ว่าจะเป็นเรื่องความไม่เข้ากันของ Web Browser แต่ละค่าย, DOM หรือ API เป็นต้น

ดังนั้น jQuery จึงรวมเอา Object และ Function ต่างๆ ที่จำเป็นมารวบรวมไว้ในรูปแบบของ Library พอเป็นเช่นนี้แล้ว ไม่ว่าโค้ดที่เขียนจะใช้ JavaScript จำนวนหลายบรรทัด ก็สามารถทำให้สั้นลงได้ อาจทำให้เหลือสั้นเพียงแค่บรรทัดเดียวเท่านั้น

jQuery ประกอบด้วยฟีเจอร์ต่างๆ ดังนี้

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

## 2.3 SQL

SQL ย่อมาจาก Structured query language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และคำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์และเป็นภาษาหนึ่ง ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

1. Select query ใช้สำหรับดึงข้อมูลที่ต้องการ
2. Update query ใช้สำหรับแก้ไขข้อมูล
3. Insert query ใช้สำหรับการเพิ่มข้อมูล
4. Delete query ใช้สำหรับลบข้อมูลออกไป

ปัจจุบันมีซอฟต์แวร์ระบบจัดการฐานข้อมูล (DBMS) ที่สนับสนุนการใช้คำสั่ง SQL เช่น Oracle DB2 MS-SQL และ MS-Access นอกจากนี้ภาษา SQL ถูกนำมาใช้เขียนร่วมกับโปรแกรมภาษาต่างๆ เช่น ภาษา C/C++ VisualBasic และ Java

ประโยชน์ของภาษา SQL

1. สร้างฐานข้อมูลและ ตาราง
2. สนับสนุนการจัดการฐานข้อมูล ซึ่งประกอบด้วย การเพิ่ม การปรับปรุง และการลบข้อมูล
3. สนับสนุนการเรียกใช้หรือ ค้นหาข้อมูล

ประเภทของคำสั่งภาษา SQL

1. ภาษานิยามข้อมูล (Data Definition Language: DDL) เป็นคำสั่งที่ใช้ในการสร้างฐานข้อมูล กำหนดโครงสร้างข้อมูลว่ามี Attribute ไต ชนิดของข้อมูล รวมทั้งการเปลี่ยนแปลงตาราง และการสร้างดัชนี คำสั่ง : CREATE DROP และ ALTER
2. ภาษาจัดการข้อมูล (Data Manipulation Language: DML) เป็นคำสั่งที่ใช้ในการเรียกใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง คำสั่ง : SELECT INSERT UPDATE และ DELETE
3. ภาษาควบคุมข้อมูล (Data Control Language: DCL) เป็นคำสั่งที่ใช้ในการกำหนดสิทธิการอนุญาต หรือ ยกเลิก การเข้าถึงฐานข้อมูล เพื่อป้องกันความปลอดภัยของฐานข้อมูล คำสั่ง : GRANT และ REVOKE

## 2.4 PostgreSQL



รูปที่ 2.4 สัญลักษณ์ของ PostgreSQL

PostgreSQL (โพสท์เกรสควิแอล) คือ PostgreSQL เรียกได้ว่าเป็นระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (object-relational) แบบ ORDBMS โดยสามารถใช้รูปแบบคำสั่งของภาษา SQL ได้เกือบทั้งหมด นอกจากนี้ยังเป็นระบบฐานข้อมูลที่ทันสมัยที่สุดของ Open Source ที่สามารถนำไปใช้งานได้โดยไม่มี

ค่าใช้จ่ายใด ได้มีการพัฒนามาจาก POSTGRES 4.2 โดยมหาวิทยาลัยแคลิฟอร์เนีย (Berkeley Computer Science department, University of California.)

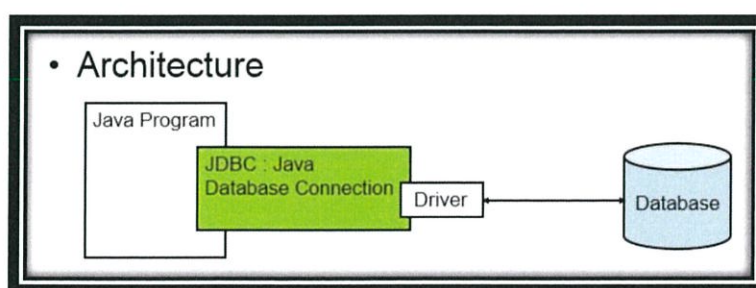
PostgreSQL สามารถทำงานบนระบบปฏิบัติการได้ทั้ง Linux UNIX (AIX BSD HP-UX SGI Irix Mac OS X Solaris Tru64) และ Windows

ดังนั้น PostgreSQL (โพสท์เกรสคิวแอล) ระบบจัดการฐานข้อมูลเป็นอีกหนึ่งทางเลือกสำหรับหลายๆ องค์กร เพื่อช่วยในการจัดการฐานข้อมูลต่างๆให้เป็นไปตามแผนการดำเนินการที่วางไว้ โปรแกรม PostgreSQL (โพสท์เกรสคิวแอล) เป็นที่นิยมอย่างมากเพราะสามารถใช้ได้โดยไม่ต้องเสียค่าใช้จ่าย และยังมีการอัปเดตให้ทันสมัยอยู่เสมอ

## 2.5 จาวากับการเขียนโปรแกรมฐานข้อมูลด้วย JDBC

ในการเขียนโปรแกรมให้สามารถทำงานร่วมกันกับข้อมูลที่มีลักษณะเป็นตารางที่มีความสัมพันธ์กัน หรือที่รู้จักกันดีว่าฐานข้อมูลนั้นภาษาจาวาได้มี API (Application Programming Interface) หรือชุดคำสั่งมาตรฐานไว้สำหรับทำงานเฉพาะ ซึ่งเรียกว่า JDBC (Java Database Connectivity)

JDBC นั้นทำหน้าที่เชื่อมโยงโปรแกรมจาวากับฐานข้อมูลเข้าด้วยกัน อีกทั้งยังสามารถจัดการกับข้อมูลต่างๆ ที่อยู่ในฐานข้อมูลนั้นด้วย โดยการใช้งานร่วมกับคำสั่งในภาษา SQL พื้นฐานทั่วไปซึ่ง SQL นั้นเป็นมาตรฐานในการจัดการกับฐานข้อมูลในฐานข้อมูลแทบทุกยี่ห้อที่เราใช้งานกันตั้งแต่ระบบฐานข้อมูลส่วนบุคคล เช่น Access Foxpro และ Paradox ไปจนถึงระบบฐานข้อมูลแบบไคลเอนท์เซิร์ฟเวอร์ เช่น Oracle SQLServer และ Informix เป็นต้น



รูปที่ 2.5 แสดงการติดต่อกับฐานข้อมูล

ดังนั้นนอกจากจะต้องเรียนรู้เกี่ยวกับ JDBC แล้วการทำความเข้าใจเป็นอย่างดีกับภาษา SQL จะทำให้สามารถเขียนโปรแกรมที่ใช้งานกับฐานข้อมูลได้อย่างมีประสิทธิภาพด้วย

### 2.5.1 SQL และฐานข้อมูล

จากที่กล่าวมา SQL เป็นภาษาพื้นฐานที่ถูกใช้ในการสร้าง เรียกค้น ตรวจสอบและจัดการกับฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ซึ่งมองข้อมูลในฐานข้อมูลประกอบไปด้วยตารางต่างๆที่สัมพันธ์กันอยู่ ซึ่งควรจะรู้ถึงคำศัพท์ที่ใช้งานบ่อยๆ เมื่อพูดถึงฐานข้อมูลและภาษา SQL ดังนี้

- 1) Column หรือข้อมูลเดี่ยวๆที่มีชื่อ ชนิด และค่ากำหนดไว้
- 2) Row (ในเชิงของแนวคิด) ใช้รวบรวมกลุ่มของ Column ที่แตกต่างกันเอาไว้
- 3) Table เป็นที่รวบรวมกลุ่มของ Row เข้าด้วยกัน
- 4) Database เก็บรวบรวม Table ไว้มากมายโดยแต่ละ Table จะมีความสัมพันธ์กัน

### 2.5.2 จาวากับ JDBC

การเขียนโปรแกรมจาวาที่มีการเชื่อมต่อกับฐานข้อมูลนั้น เมธอดหรือคลาสต่างๆที่ใช้จะถูกกำหนดไว้ใน `java.sql.*` และ `javax.sql.*` โดยแบ่งชนิดของการติดต่อฐานข้อมูลได้เป็น 2 แบบด้วยกัน

#### 2.5.2.1 การติดต่อฐานข้อมูลแบบ JDBC โดยตรง

วิธีนี้เป็นการติดต่อฐานข้อมูลกับโปรแกรมจาวาผ่านไดรเวอร์ของ JDBC ที่เขียนขึ้นสำหรับการติดต่อฐานข้อมูลเฉพาะอย่างโดยตรง เช่น การติดต่อกับฐานข้อมูล Oracle และ MySQL ก็จะใช้ไดรเวอร์ที่เขียนขึ้นมาต่างหาก โดยสิ่งที่สำคัญในการใช้พวกไดรเวอร์เหล่านี้คือ มักจะมีราคาสูงหรืออีกนัยหนึ่งคือ จะติดมาด้วยกันกับฐานข้อมูลทั้งหลาย หรือแม้แต่จะเป็นฐานข้อมูลที่เปิดให้ใช้ได้ฟรี ก็อาจจำเป็นที่จะต้องลงทุนซื้อไดรเวอร์ JDBC นี้มาด้วยราคาแพง ดังนั้นสำหรับนักพัฒนาทั่วไปที่ไม่ได้เกี่ยวข้องกับธุรกิจที่ต้องใช้ฐานข้อมูลใหญ่ๆ ประสิทธิภาพสูงมากๆ จึงนิยมใช้กันเพียงการติดต่อแบบ Bridge ที่จะกล่าวถึงในหัวข้อถัดไป ตัวอย่างการใช้ JDBC Driver ผ่าน JDBC URL ดังนี้

```
jdbc:couldscape:rmi:databaseName
jdbc:weblogic:oracle:databaseName
jdbc:atinav:localhost:5000:C:\dir\dbl.mdb
```

#### 2.5.2.2 การติดต่อฐานข้อมูลแบบ JDBC-ODBC Bridge

การใช้ Bridge ในการเชื่อมโยงนี้สามารถนำไปใช้ได้กับฐานข้อมูลทุกๆชนิด โดยทำการสร้าง Data Source เอาไว้ก่อน แล้วทำการสร้างการติดต่อไปยัง JDBC-ODBC URL ดังนี้

```
jdbc:odbc:DataSourceName
```

### 2.5.3 การเชื่อมต่อโปรแกรมกับฐานข้อมูล

ในการเชื่อมต่อฐานข้อมูลใด ๆ กับโปรแกรมจาวาประกอบไปด้วย 2 ขั้นตอนที่สำคัญดังนี้

#### 1) Load JDBC Driver มีรูปแบบการใช้ทั่วไป ดังนี้

```
Class.forName(DriverClassName);
```

ซึ่งในที่นี้ เราทำการติดต่อฐานข้อมูลแบบ JDBC-ODBC Bridge ดังนั้น ไดรเวอร์ที่ใช้จึงมีเพียงอันเดียวเท่านั้นไม่ว่าจะเป็นฐานข้อมูลอะไรก็ตามคือ “sun.jdbc.odbc.JdbcOdbcDriver” โดยการโหลดไดรเวอร์นี้จะทำภายใต้การตรวจสอบ Exception (ป้องกันการผิดพลาดจากการโหลด) ดังนี้

```
try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
} catch(java.lang.ClassNotFoundException e) {
    System.out.print("ClassNotFoundException:");
    System.out.println(e.getMessage());
}
```

“ClassNotFoundException” นั้นจะทำการตรวจสอบชื่อของไดรเวอร์ที่ระบุว่ามีอยู่จริง และสามารถนำกลับมาใช้ได้จริงในโปรแกรมหรือไม่

#### 2) การติดต่อไปยัง Data Source ด้วย “DriverManager” มีรูปแบบการใช้ดังนี้

```
Connection con = DriverManager.getConnection(URL, USERNAME, PASSWORD);
```

URL ที่ใช้สำหรับ JDBC-ODBC Bridge นั้นก็จะใช้เหมือนกันในทุกๆฐานข้อมูลเช่นกันกับไดรเวอร์คือ “jdbc:odbc:DataSourceName” โดยถ้าไม่ได้ระบุ username/password ให้ปล่อยว่างไว้ได้

```
String url = "jdbc:odbc:AAA_DB";
Connection con = DriverManager.getConnection(url, "", "");
```

### 2.5.4 การเพิ่มข้อมูลในตาราง

การเพิ่มข้อมูลหรือแก้ไขข้อมูลในตารางนั้นสามารถทำได้ด้วยคำสั่ง SQL ปกติ โดยจาวาจะมีเมธอดที่ทำการประมวลผลคำสั่ง SQL ประเภท INSERT และ UPDATE คือ Statement.executeUpdate() โดย object Statement ก็สามารสร้างได้จากเมธอด Connection.createStatement()

### 2.5.5 การดึงข้อมูลจากตารางออกมาใช้

การเรียกข้อมูลมาใช้นั้นสามารถทำได้โดยใช้คำสั่ง SQL SELECT ซึ่งเมธอดที่ใช้ในการประมวลผลก็คือ Statement.executeQuery() ผลลัพธ์ที่ได้เป็นค่าของ ResultSet

```
ResultSet rs = stmt.executeQuery("SELECT * FROM AAA");
```

ค่าของ ResultSet "rs" จะประกอบด้วยข้อมูลที่เกี่ยวข้องกันกับกลุ่มข้อมูลที่ได้ทั้งหมด ไม่ว่าจะ เป็นลำดับของผลลัพธ์ ค่าของผลลัพธ์ ค่าถัดไป หรือก่อนหน้านั้น และอื่นๆ โดยนำค่าเหล่านั้นมาใช้ได้จากเมธอดที่สำคัญต่างๆ เช่น next() getString() getInt() และ getFloat() เป็นต้น

ในการอ่านค่าที่ SELECT ขึ้นมานั้นเมธอดที่ใช้จะขึ้นต้นด้วย getXXXX() และตามด้วยชนิดของข้อมูลที่ต้องการจะอ่าน เช่น คอลัมน์ "Name" มีชนิดของข้อมูลเป็น VARCHAR หรือข้อความนั่นเอง ก็จะใช้เมธอด getString() ในการอ่านค่าและใส่พารามิเตอร์เป็นชื่อของคอลัมน์เพื่อบอกให้รู้ว่าเป็นการอ่านค่าของคอลัมน์ใด ก็คือ getString("Name") ซึ่งเช่นกันกับชนิดอื่นๆ

## 2.6 หลักการของระบบฐานข้อมูล

### 2.6.1 ส่วนประกอบของข้อมูล

ตารางจะประกอบด้วย 2 ส่วนหลัก คือ คอลัมน์ (Column) สำหรับระบุประเภทของข้อมูลและแถว (Row) ซึ่งประกอบด้วยข้อมูลของแต่ละคอลัมน์ในแถวนั้นๆ

#### 2.6.1.1 คอลัมน์ (Column)

คอลัมน์หรือฟิลด์ จะถูกกำหนดเป็นชื่อที่ไม่ซ้ำกันสำหรับในตารางหนึ่งๆ โดยมีการระบุประเภทของข้อมูลที่จะถูกนำมาเก็บยังฟิลด์นี้ รวมถึงการกำหนดขนาดความยาวของข้อมูลที่สามารถรองรับในแต่ละฟิลด์เหล่านี้ได้ หรือความสามารถในการยอมรับข้อมูลที่ไม่ทราบค่า (Unknow)

### 2.6.1.2 แถว (Row)

แถวหรือเรคคอร์ด (Record) แทนลักษณะของสิ่งหนึ่งสิ่งใดที่อยู่ภายในบรรทัดเดียวกัน

### 2.6.1.3 คีย์ (Key)

เค้าร่างของตารางหรือรีเลชัน (Table/Relation Schema) ประกอบด้วยรายละเอียดข้อมูลของคอลัมน์ต่างๆ ซึ่งคอลัมน์ใดคอลัมน์หนึ่งในตารางมีโอกาสที่จะมีคุณสมบัติเป็นคีย์ โดยคีย์นั้นๆ สามารถใช้ในการแสดงหรือบ่งบอกถึงค่าของข้อมูลที่เหลือในแต่ละแถว หรือใช้ในการเชื่อมโยงข้อมูลกับข้อมูลอีกตารางหนึ่ง ประเภทของคีย์แบ่งออกเป็น 2 ประเภทหลัก ดังนี้

#### 1) คีย์หลัก (Primary Key)

เป็นคอลัมน์ที่มีคุณสมบัติของข้อมูลที่เก็บค่าไม่ซ้ำกัน คุณสมบัตินี้จะสามารถระบุว่าข้อมูลนั้นเป็นข้อมูลของแถวใด เช่น รหัสของพนักงาน เลขที่ 0001 สามารถระบุว่าเป็นรหัสของพนักงานชื่อ AAAA

#### 2) คีย์นอก (Foreign Key)

เป็นคอลัมน์ในตารางหนึ่งที่สามารถใช้ในการเชื่อมโยงข้อมูลกับอีกตารางหนึ่งที่มีคอลัมน์เดียวกันนี้ปรากฏอยู่

## 2.6.2 ประเภทความสัมพันธ์ของข้อมูล

เนื่องจากความสัมพันธ์ระหว่างข้อมูลต่างๆ ที่จัดเก็บอยู่ในฐานข้อมูลจัดเป็นส่วนที่สำคัญ มีผลต่อโครงสร้างของฐานข้อมูล แล้วเป็นส่วนที่ใช้แบ่งฐานข้อมูลออกเป็นฐานฐานข้อมูลประเภทต่างๆ สำหรับความสัมพันธ์ของข้อมูล สามารถแบ่งได้เป็น 3 ลักษณะดังนี้

### 2.6.2.1 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One-to-One)

ความสัมพันธ์แบบหนึ่งต่อหนึ่ง คือ ข้อมูลที่เก็บอยู่ในเอ็นทิตีหนึ่งมีความสัมพันธ์กับข้อมูลที่เก็บอยู่ในเอ็นทิตีอื่นเพียงหนึ่งระเบียนเท่านั้น และสมาชิกของในเอ็นทิตีอื่นมีความสัมพันธ์กับเอ็นทิตีเดิมเพียงหนึ่งระเบียนด้วย สามารถเขียนเป็นแผนผังได้ดังภาพ



รูปที่ 2.6 แสดงความสัมพันธ์แบบหนึ่งต่อหนึ่ง

## ตารางที่ 2.2 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

รหัสพนักงาน	ชื่อพนักงาน
0001	พอใจ ที่มีอยู่
0002	ใจดี สบายใจ
0003	มั่งมี ศรีสุข

### 2.6.2.2 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-to-Many)

ความสัมพันธ์แบบหนึ่งต่อกลุ่ม คือ สมาชิกจำนวนหนึ่งระเบียบที่เก็บอยู่ในเอ็นทิตีหนึ่งมีความสัมพันธ์กับสมาชิกที่เก็บอยู่ในเอ็นทิตีอื่นจำนวนหลายๆ ระเบียบ อาจจะตั้งแต่หนึ่งระเบียบขึ้นไป เพราะฉะนั้นความสัมพันธ์ที่เขียนเป็น หนึ่งต่อกลุ่ม สัญลักษณ์ที่ใช้คือ 1:M สามารถเขียนเป็นแผนผังได้ดังนี้



รูปที่ 2.7 แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่ม

## ตารางที่ 2.3 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

แผนก	ชื่อพนักงาน
การเงินและบัญชี	พอใจ ที่มีอยู่
การเงินและบัญชี	ยุวชน คนดี
ฝ่ายบุคคล	ใจดี สบายใจ
ฝ่ายบุคคล	มั่งมี ศรีสุข

### 2.6.2.3 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many-to-Many)

สมาชิกจำนวนตั้งแต่หนึ่งระเบียบขึ้นไปในเอ็นทิตีหนึ่งมีความสัมพันธ์กับสมาชิกในเอ็นทิตีที่สองจำนวนหลายๆ ระเบียบ ในขณะที่เดียวกันหากมองย้อนกลับ สมาชิกในเอ็นทิตีที่สองกลับไปที่มีความสัมพันธ์กับสมาชิกในเอ็นทิตีที่หนึ่ง จำนวนหลายๆ ระเบียบเช่นกัน เพราะฉะนั้นความสัมพันธ์ที่เขียนเป็น กลุ่ม ต่อกลุ่ม สัญลักษณ์ที่ใช้คือ M:N สามารถเขียนเป็นแผนผังได้ดังนี้



รูปที่ 2.8 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม

ตารางที่ 2.4 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

รายวิชา	นักศึกษา
วิทยาศาสตร์	พอใจ ที่มีอยู่
วิทยาศาสตร์	ใจดี สบายใจ
ภาษาอังกฤษ	พอใจ ที่มีอยู่
ภาษาอังกฤษ	ใจดี สบายใจ
ภาษาอังกฤษ	ยุวชน คนดี

## 2.7 สัญลักษณ์ Flowchart และความหมาย

ผังงาน (Flowchart) คือ รูปภาพ (Image) หรือสัญลักษณ์ (Symbol) ที่ใช้เขียนแทนขั้นตอน คำอธิบาย ข้อความ หรือคำพูด ที่ใช้ในอัลกอริทึม (Algorithm) เพราะการนำเสนอขั้นตอนของงานให้เข้าใจตรงกัน ระหว่างผู้เกี่ยวข้อง ด้วยคำพูด หรือข้อความทำได้ยากกว่า [7]

ผังงานแบ่งได้ 2 ประเภท

### 1. ผังงานระบบ (System Flowchart)

คือ ผังงานที่แสดงขั้นตอนการทำงานในระบบอย่างกว้างๆ แต่ไม่เจาะลงในระบบงานย่อย



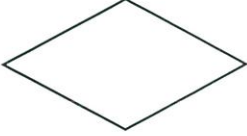
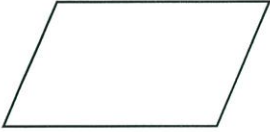


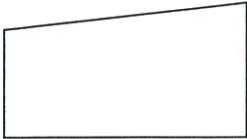
### 2. ผังงานโปรแกรม (Program Flowchart)

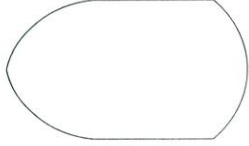

คือ ผังงานที่แสดงถึงขั้นตอนในการทำงานของโปรแกรม ตั้งแต่รับข้อมูล คำนวณ จนถึง

แสดงผลลัพธ์

ประโยชน์ของ Flowchart

1. ทำให้เข้าใจ และแยกแยะปัญหาได้ง่าย (Problem Define)
2. แสดงลำดับการทำงาน (Step Flowing)
3. หาข้อผิดพลาดได้ง่าย (Easy to Debug)
4. ทำความเข้าใจโปรแกรมได้ง่าย (Easy to Read)
5. ไม่ขึ้นกับภาษาใดภาษาหนึ่ง (Flexible Language)

สัญลักษณ์	ความหมาย
 Terminator	การเริ่มต้น หรือ การสิ้นสุด
 Process	การกำหนดค่า หรือ การประมวลผลทั่วไป
 Decision	การตัดสินใจ การเปรียบเทียบ จะมีผลใน 2 ทิศทาง คือ กรณีผลตรวจสอบเงื่อนไขเป็นเท็จ และเป็นจริง
 Data	รับ หรือ แสดงข้อมูล โดยไม่ระบุชนิดอุปกรณ์
 Predefined Process	โปรแกรมย่อย หรือ โมดูล เริ่มทำงานหลักจากจบคำสั่งในโปรแกรมย่อยแล้ว จะกลับมาทำคำสั่งต่อไป
 Connector	จุดเชื่อมต่อในหน้าเดียวกัน
 Manual Input	การรับข้อมูลเข้าทางแป้นพิมพ์

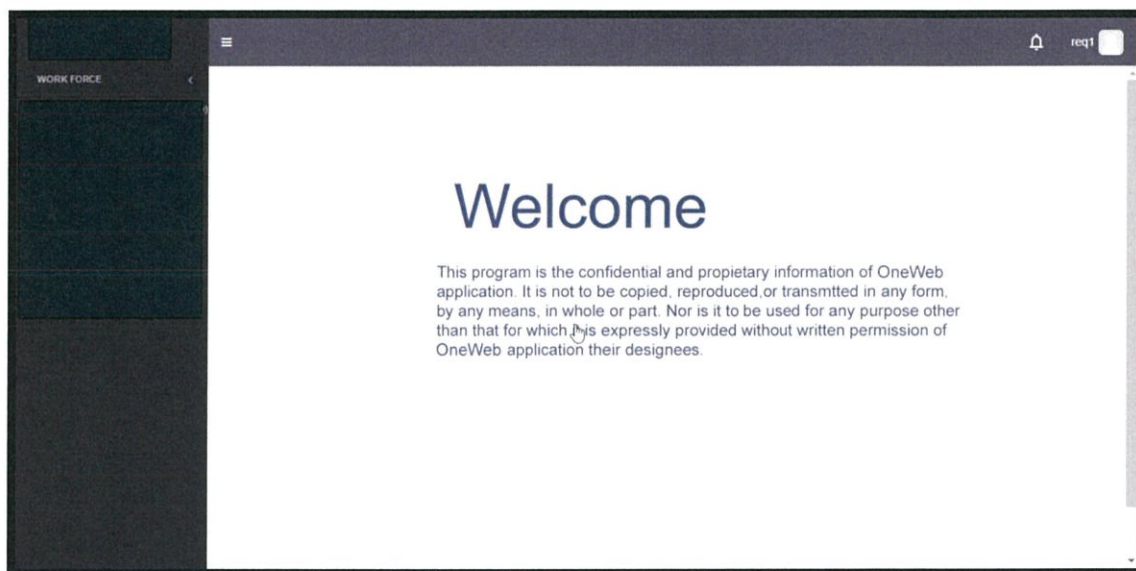
 <p>Display</p>	จอภาพแสดงผล
 <p>Off-page Connector</p>	จุดเชื่อมต่อกับหน้า

## บทที่ 3

### วิธีดำเนินงาน

#### 3.1 เครื่องมือที่ใช้ในการพัฒนา

##### 3.1.1 Framework ONEWEB 4.0



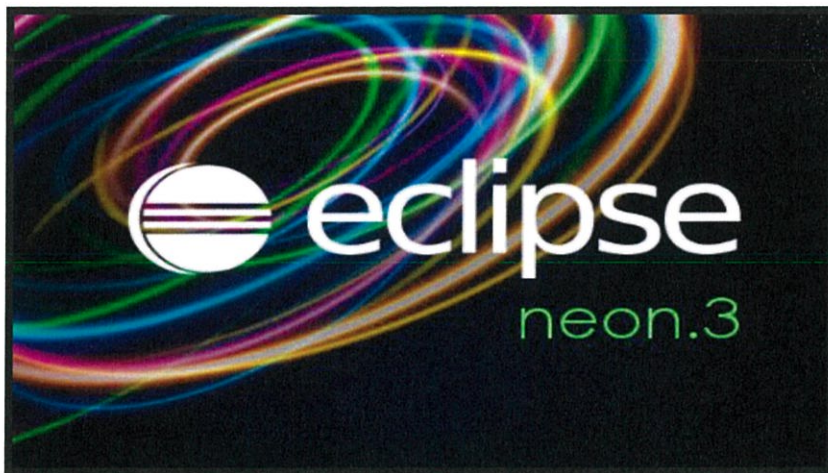
รูปที่ 3.1 แสดงหน้าจอ Framework ONEWEB 4.0

เป็น ‘All-In-One Digital Platform’ ที่สามารถปฏิรูปความคิดของคนในองค์กรให้ออกมาเป็น แอปพลิเคชันและสามารถตอบโจทย์ธุรกิจ การใช้งานของคนในองค์กรได้อย่างครอบคลุม มั่นคง และยั่งยืนในยุค Digital Transformation ทำให้ ONEWEB 4.0 รองรับการเติบโตขององค์กรทุกขนาด ทุกอุตสาหกรรม ให้สามารถออกแบบและพัฒนาแอปพลิเคชันธุรกิจ (Business Application) ขององค์กร เพื่อใช้ในการแข่งขันทำธุรกิจอย่างรวดเร็วและมีประสิทธิภาพ อีกทั้งยังสามารถประหยัดค่าใช้จ่ายในการบำรุงรักษาและการปรับปรุงระบบในระยะยาว ซึ่งแอปพลิเคชันที่พัฒนาขึ้นมาได้นั้นจะมีลักษณะเป็นเว็บแอปพลิเคชันระบบเปิด (Open Standard) และในส่วนของเว็บเทคโนโลยี อาทิ Java HTML5 และ CSS3 อยู่ในฉากหลัง เป็น Open Platform มาตรฐานการทำงานที่จำเป็นสำหรับองค์กรรุ่นใหม่ ทำให้สามารถเชื่อมต่อกับแพลตฟอร์มอื่นๆ ได้สะดวก รองรับความต้องการของนักพัฒนา และความต้องการของผู้ใช้งาน นอกจากนี้ยังรองรับการใช้งานคลาวด์ทั้งไพรเวตคลาวด์และพับลิค คลาวด์อีกด้วย

ONEWEB พัฒนาขึ้นภายใต้แนวคิดของความยืดหยุ่น เพื่อให้ส่วนต่างๆ ของแอปพลิเคชันสามารถปรับเปลี่ยนได้ตามความต้องการของลูกค้า จึงเป็นเครื่องมือที่มีความยืดหยุ่นอย่างมาก และมีความสามารถในการปรับลดหรือเพิ่มตามต้องการ ให้สามารถออกแบบแอปพลิเคชันที่รองรับการใช้งานของ User ตั้งแต่ระดับไม่กี่ User จนถึงการใช้งานในระดับพันหรือหมื่น User โดยแบ่งหน้าที่การพัฒนาแอปพลิเคชันออกเป็น 3 ส่วน ได้แก่ ส่วนของ App Designer จะดูแลเรื่องฐานข้อมูลและการปรับแต่ง User interface (UI) ที่ช่วยลดระยะเวลาการปรับแต่งแอปพลิเคชันลงได้เป็นอย่างมาก และในส่วนของกระบวนการออกแบบ (Process Designer) คือ เครื่องมือในการออกแบบกระบวนการของธุรกิจ ซึ่งจะทำให้การบริหารระบบการทำงานภายในและภายนอกสอดคล้องและเป็นแบบแผนที่ชัดเจน ส่วนสุดท้าย คือ App Space ซึ่งเป็นโมดูลในการบริหารจัดการแอปพลิเคชันที่พัฒนาขึ้นมาในองค์กรอย่างบูรณาการ

ทั้ง 3 ส่วนเป็นกระบวนการพัฒนาแอปพลิเคชันที่ตรงตามแนวคิดของ Agile Methodology ที่จะช่วยลดความยุ่งยากและเวลาในการจัดทำเอกสารของโครงการในการพัฒนาแอปพลิเคชันทิ้งไป โดยเน้นผลลัพธ์ที่สามารถพัฒนาแอปพลิเคชันให้ตรงตามความต้องการของธุรกิจได้อย่างรวดเร็ว และยังเป็นการแก้ปัญหาในกระบวนการพัฒนาแอปพลิเคชันแบบเดิมๆ ที่ใช้เวลานาน แต่ผลลัพธ์ที่ออกมาไม่สามารถตอบสนองความต้องการของธุรกิจ หรือใช้เวลาพัฒนานาน ทำให้ไม่ทันต่อการเปลี่ยนแปลงโดยเฉพาะในยุคดิจิทัล

### 3.1.2 โปรแกรม Eclipse



รูปที่ 3.2 แสดงสัญลักษณ์โปรแกรม Eclipse

Eclipse คือโปรแกรมที่ใช้สำหรับพัฒนาภาษาจาวาเป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็นซอฟต์แวร์ Open Source ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแค่พัฒนา plugin สำหรับงานนั้นขึ้นมา และนำ Plug-in นั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มีมาพร้อมกับ Eclipse เมื่อเรา download มาครั้งแรกก็คือ องค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียนและ Debug โปรแกรมภาษาจาวา

ข้อดีของโปรแกรม Eclipse คือ ติดตั้งง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชัน รองรับภาษาต่างประเทศอีกหลายภาษา มี plugin ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์หลายชนิด เช่น HTML Java C JSP EJB XML และ GIF ที่สำคัญเป็นฟรีแวร์ที่สามารถใช้งานได้กับระบบปฏิบัติการ Windows Linux และ Mac OS

### 3.1.3 โปรแกรม DbVisualizer



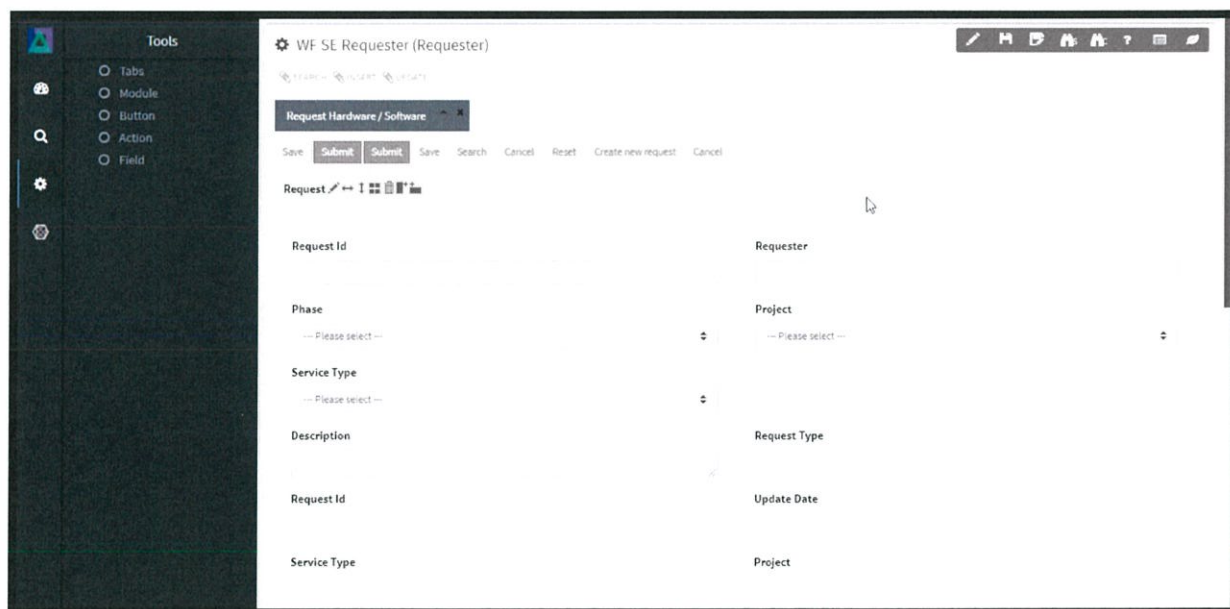
รูปที่ 3.3 แสดงสัญลักษณ์โปรแกรม DbVisualizer

เป็นเครื่องมือฐานข้อมูลสำหรับนักพัฒนาและผู้บริหารฐานข้อมูลช่วยให้มีทั้งการพัฒนาและการบำรุงรักษาฐานข้อมูล นอกจากนี้ยังมีการสนับสนุนเสรีจอตโนมัตินตัวแก้ไข SQL ภาพการสร้างหรือแก้ไขตาราง การสนับสนุนการดำเนินการของหลายคำสั่งสคริปต์ SQL แผนภูมิของข้อมูลการจัดการข้อมูล CLOB ส่งออก Schema ฐานข้อมูลนำเข้าข้อมูลจากไฟล์ CSV สร้างแบบสอบถาม นาวิเกเตอร์คีข้อมูลต่างประเทศอธิบายการสนับสนุนแผนและรวมถึงการบริหารจัดการของมุมมอง ทริกเกอร์ ตารางสรุปคำพ้องความหมาย ฟังก์ชัน ขั้นตอนตารางระบบเรียกคอลัมน์คีหลักลิติ จำกัด อ้างอิง DDL โปรแกรม โมดูล เพิ่มข้อมูล และพื้นที่ตาราง

### 3.2 ขั้นตอนการเตรียมการ

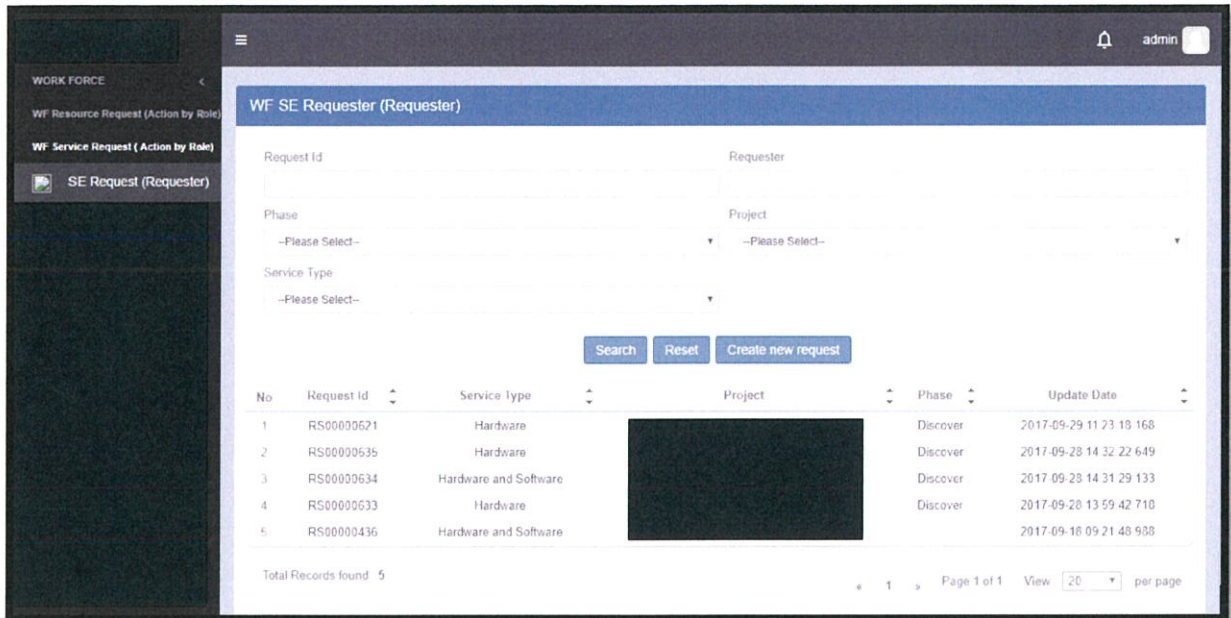
1. ติดตั้งโปรแกรมเครื่องมือที่ใช้ในการพัฒนา
2. ศึกษาเครื่องมือที่ใช้ในการพัฒนาเว็บแอปพลิเคชัน
3. เข้าฝึกอบรมการพัฒนา ONEWEB 4.0 เพื่อศึกษาวิธีการพัฒนาเว็บแอปพลิเคชันอย่างถูกต้อง
4. เมื่อได้รับมอบหมายงานจากพนักงานที่ปรึกษาแล้ว เพื่อให้สามารถพัฒนางานได้ตรงตามความต้องการของผู้ใช้ต้องศึกษาความต้องการที่จะพัฒนาระบบจากพนักงานที่ปรึกษาให้ชัดเจน

สำหรับในส่วนของหน้าต่างเว็บแอปพลิเคชัน ภายในทีมจะมีพนักงานที่รับผิดชอบในการสร้างหน้าต่างเว็บแอปพลิเคชันให้ตรงตามความต้องการในการนำไปใช้งาน และส่วนนักศึกษาจะต้องเขียนโปรแกรมเพื่อปรับแต่งเพิ่ม เพื่อให้การทำงานของเว็บแอปพลิเคชันสมบูรณ์ยิ่งขึ้น โดยมีตัวอย่างหน้าต่างของ App Designer หน้าต่างของ Front End และตัวอย่างการผูกไฟล์กับหน้าที่ปรับแต่ง ดังรูป ตามลำดับ



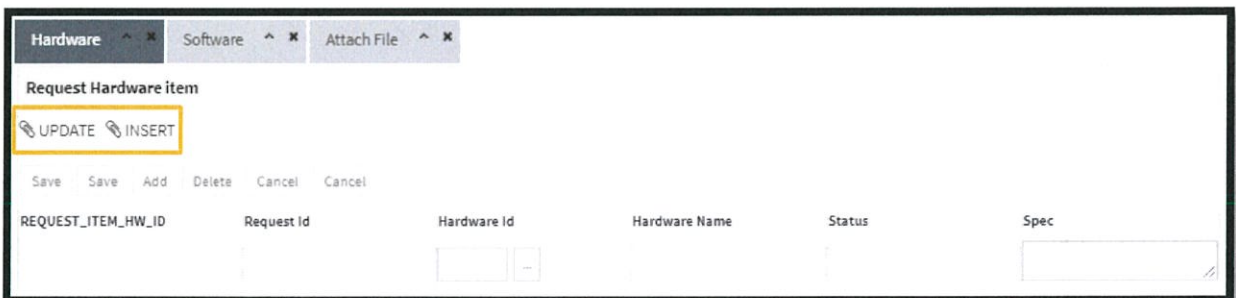
รูปที่ 3.4 แสดงตัวอย่างหน้าต่างของ App Designer

จากรูปที่ 3.4 หน้าต่างของ App Designer ด้านซ้ายคือเครื่องมือต่างๆของ App Designer ส่วนด้านขวาคือพื้นที่สำหรับ Drag & Drop หน้าเว็บแอปพลิเคชัน เมื่อทำการจัดเรียงฟิลด์ต่างๆเรียบร้อยแล้ว App Designer ก็จะสร้างหน้าเว็บแอปพลิเคชันขึ้นมาโดยเว็บที่ถูกสร้างขึ้นมานั้นจะสามารถบันทึก ค้นหาหรือลบข้อมูลได้ และเมื่อนักพัฒนาโปรแกรมเขียนไฟล์ Java หรือ JavaScript ก็จะนำไฟล์นั้นมาผูกกับหน้า App Designer ที่ต้องการปรับแต่งเพิ่มเติม



รูปที่ 3.5 แสดงตัวอย่างหน้าต่างต่างของ Front End

จากรูปที่ 3.5 หน้าต่างของ Front End ด้านซ้ายคือแถบเมนูต่างๆที่เกิดจากการสร้างหน้าของ App Designer หลายๆหน้า แล้วเอามาจัดไว้เป็นหมวดหมู่เพื่อให้งานได้ตรงตามความต้องการ ส่วนด้านขวาคือพื้นที่สำหรับให้งานตามที่ได้ออกแบบใน App Designer



รูปที่ 3.6 แสดงตัวอย่างการใช้ Action กับไฟล์ที่ผูก

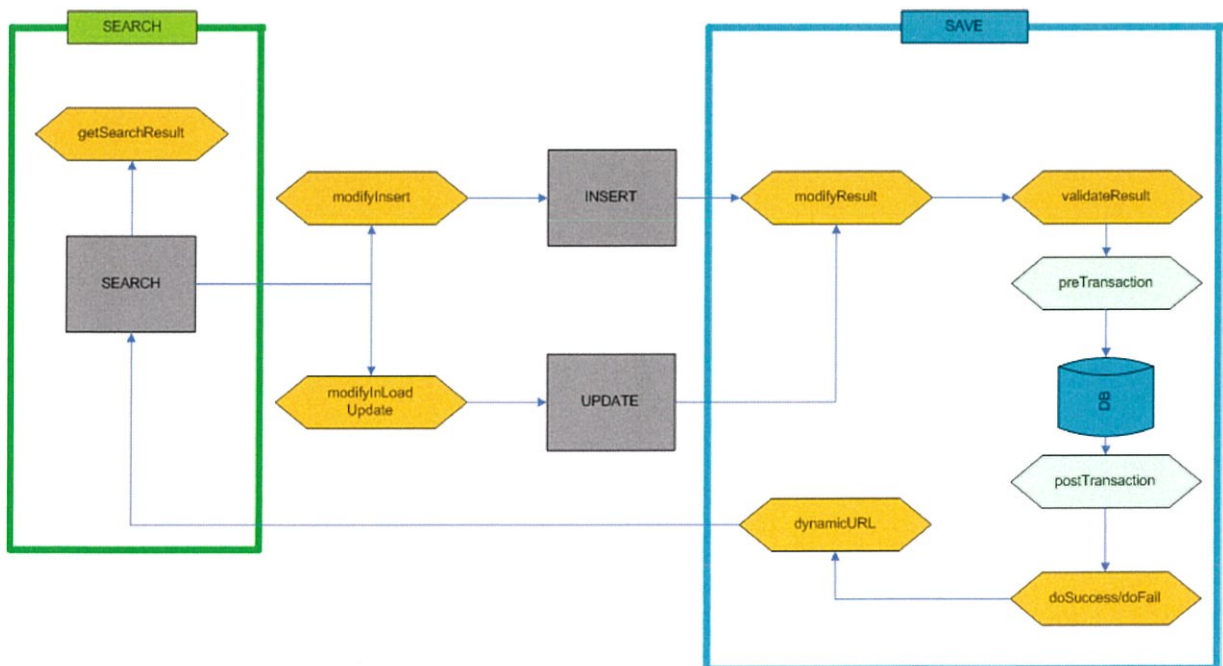
จากรูปที่ 3.6 การผูกไฟล์จะมี Action ให้เลือกอยู่ 3 แบบ ได้แก่ Search Update และ Insert โดยต้องวิเคราะห์หน้าเว็บแอปพลิเคชันที่ทำการปรับแต่งว่าตรงกับ Action ในรูปแบบใด จะสามารถทำงานได้ถูกต้องตามที่ต้องการ



รูปที่ 3.7 แสดงตัวอย่างการผูกไฟล์

จากรูปที่ 3.7 เมื่อเลือกที่ Action เข้าไปจะเจอกับหน้าต่างกำหนดค่า Action Field โดยช่องหมายเลข 1 จะใช้สำหรับผูกไฟล์ที่เป็น JavaScript ส่วนช่องหมายเลข 2 จะใช้สำหรับผูกไฟล์ที่เป็น Java ให้ใส่ Path ของไฟล์ที่เขียนโปรแกรม

### 3.2.1 สิ่งสำคัญสำหรับการปรับแต่ง ONEWEB 4.0



รูปที่ 3.8 แสดงแผนผังการปรับแต่งด้วยไฟล์ Java

## 1. การปรับแต่งเพิ่มเติมด้วยไฟล์ Java

### 1.1 การใช้ Extends และ Implements class

การใช้ คำสั่ง Extends คือการเอาคลาสที่มีอยู่เดิมมาประยุกต์ เพิ่มเมธอดหรือตัวแปรเข้าไปได้ ส่วนการใช้ "implement" จะเป็นการ implement class ประเภท interface ซึ่งในคลาสนั้น ทุกๆเมธอดจะเป็น abstract method (สำหรับเมธอดที่เป็น abstract เมื่อ extends หรือ implement มาจะต้อง Overriding เมธอดนั้นเสมอๆ) ดังนั้นจะต้อง Overriding ทุกเมธอด

### 1.2 วิธีรับเซสชันเอนทิตี

- เซสชันเอนทิตีเก็บโดยใช้ชื่อคลาส "EntityFormHandler"
- เซสชันเอนทิตีที่อยู่ในชื่อเซสชันจะใช้ Entity id + "\_session" เช่น

"EN\_3331123753954\_session"

### 1.3 วิธีรับเซสชันโมดูล

- เซสชันโมดูลเก็บโดยใช้ชื่อคลาส "MasterFormHandler"
- เซสชันโมดูลเก็บไว้ในชื่อเซสชันจะใช้ module id + "\_session" เช่น

"MD66128805\_session"

### 1.4 วิธีป้อนข้อมูลทั้งหมดให้กับโมดูล

- สำหรับโมดูลแบบ one to one จะใช้เมธอด

MasterFormHandler.getStoreHashMap() โดยค่าทั้งหมดที่ส่งกลับมาในโมดูลจะเป็นแบบ Hashmap

- สำหรับโมดูลแบบ one to many จะใช้เมธอด

MasterFormHandler.getStoreActionList() โดยค่าทั้งหมดที่ส่งกลับมาในโมดูลจะเป็นแบบเวกเตอร์

### 1.5 วิธีการรับข้อมูลจากฟิลด์เฉพาะที่อยู่ภายในโมดูล

- การดึงข้อมูลจากโมดูลแบบ Hashmap (จากชื่อก่อนหน้า) จะใช้ Key ที่เป็นไอดี

ของฟิลด์ จากหน้าที่เป็นส่วนของหลังบ้าน (หน้า App Designer)

### 1.6 วิธีใส่ข้อความแจ้งเตือนความผิดพลาดที่หน้าจอ

- จะใส่ข้อความสำหรับแจ้งเตือนโดยใช้คำสั่ง

EntityFormHandler.getFormErrors()

### 1.7 วิธีการปรับเปลี่ยนค่าของฟิลด์บนหน้าจอป้อนข้อมูลแบบ one to many

- สร้างคลาส java แล้วใช้เมธอด "modifyInsertMany"

- หลังจากแก้ไขข้อมูลแล้วให้เรียกใช้ "super.modifyInsertMany (moduleID, hStoreHashMap, processMode, row);"

1.8 วิธีแก้ไขค่าฟิลด์เพิ่มหรือแก้ไข จะได้ผลลัพธ์ในตารางหลังจากบันทึกหน้าป้อปอัพ

- สร้างคลาส java แล้วใช้เมธอด “modifyResultMany”

- หลังจากแก้ไขข้อมูลแล้วให้เรียกใช้ “super.modifyResultMany(moduleID, hStoreHashMap, processMode, row);”

## 2 การปรับแต่งเพิ่มเติมด้วย Java Script

2.1 วิธีสรุปข้อมูลหรือแก้ไขข้อมูลหลังจากบันทึกป้อปอัพตามปกติ

- สร้างไฟล์ Java Script โดยรูปแบบฟังก์ชัน Override จะใช้ module\_id + "manualUS ()"

2.2 วิธีสรุปข้อมูลหลังจากลบบันทึก

- สร้างไฟล์ Java Script โดยรูปแบบฟังก์ชัน Override จะใช้ module\_id + "CreateRowForManual()"

2.3 วิธีการปรับเปลี่ยนค่าของฟิลด์ในหน้าจ้อป้อปอัพของโมดูลแบบ one to many หลังจากคลิก Add

- สร้างไฟล์ Java Script โดยรูปแบบฟังก์ชัน Override จะใช้ popupActionFlowManual (moduleID)

- ต้องตรวจสอบว่า moduleID ที่ใช้ทำงานเฉพาะโมดูลที่กำหนดเท่านั้นหรือไม่

2.4 วิธีการปรับเปลี่ยนค่าของฟิลด์ในหน้าจ้อป้อปอัพของโมดูลแบบ one to many หลังจากคลิกที่ Edit

- สร้างไฟล์ Java Script โดยรูปแบบฟังก์ชัน Override จะใช้ loadUpdateManyManual (moduleID)

- ต้องตรวจสอบว่า moduleID ที่ใช้ทำงานเฉพาะโมดูลที่กำหนดเท่านั้นหรือไม่

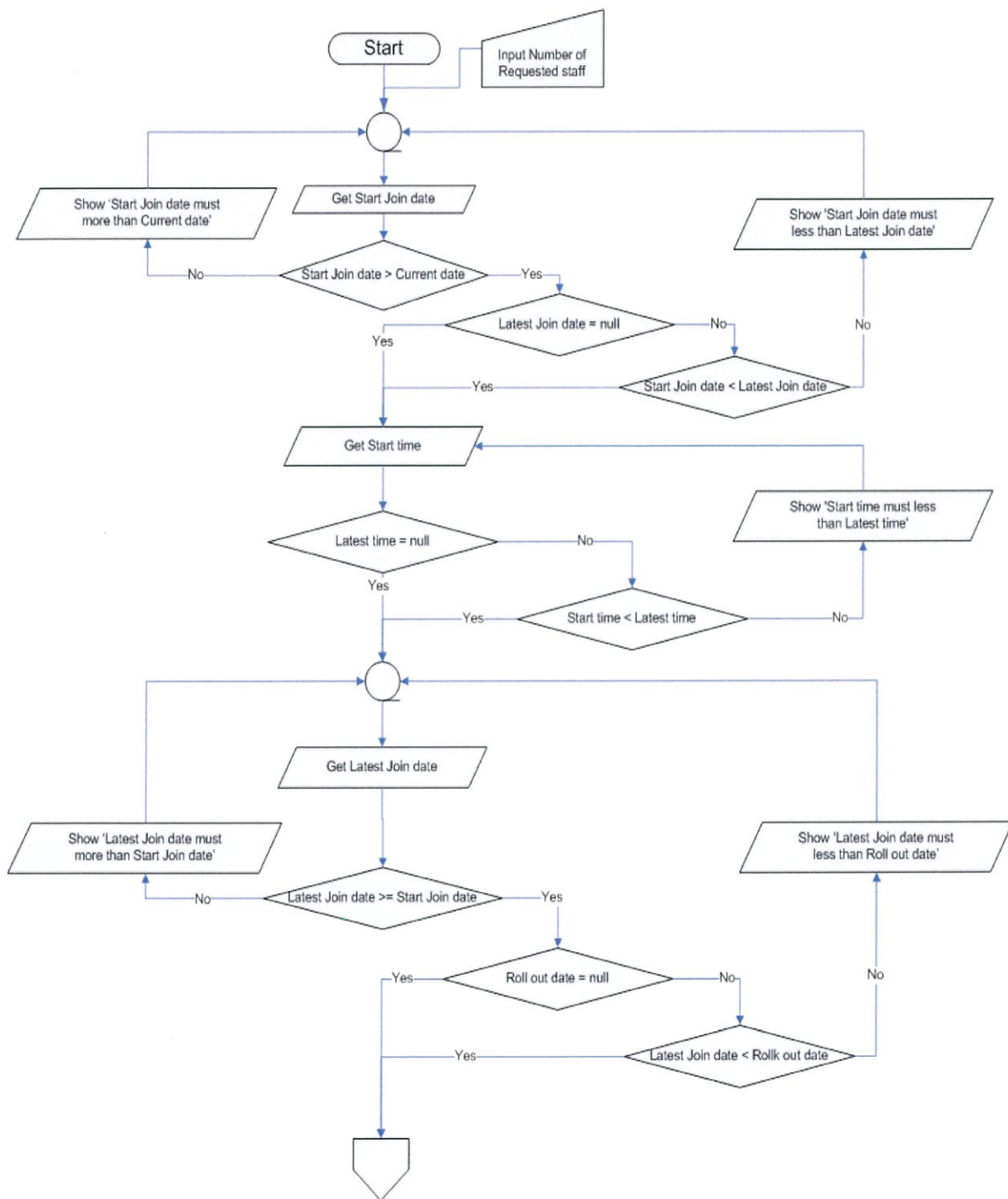
### 3.3 การคำนวณชั่วโมงการทำงาน

รูปที่ 3.9 แสดงหน้าต่างของ Front End หน้าคำนวณชั่วโมงการทำงาน

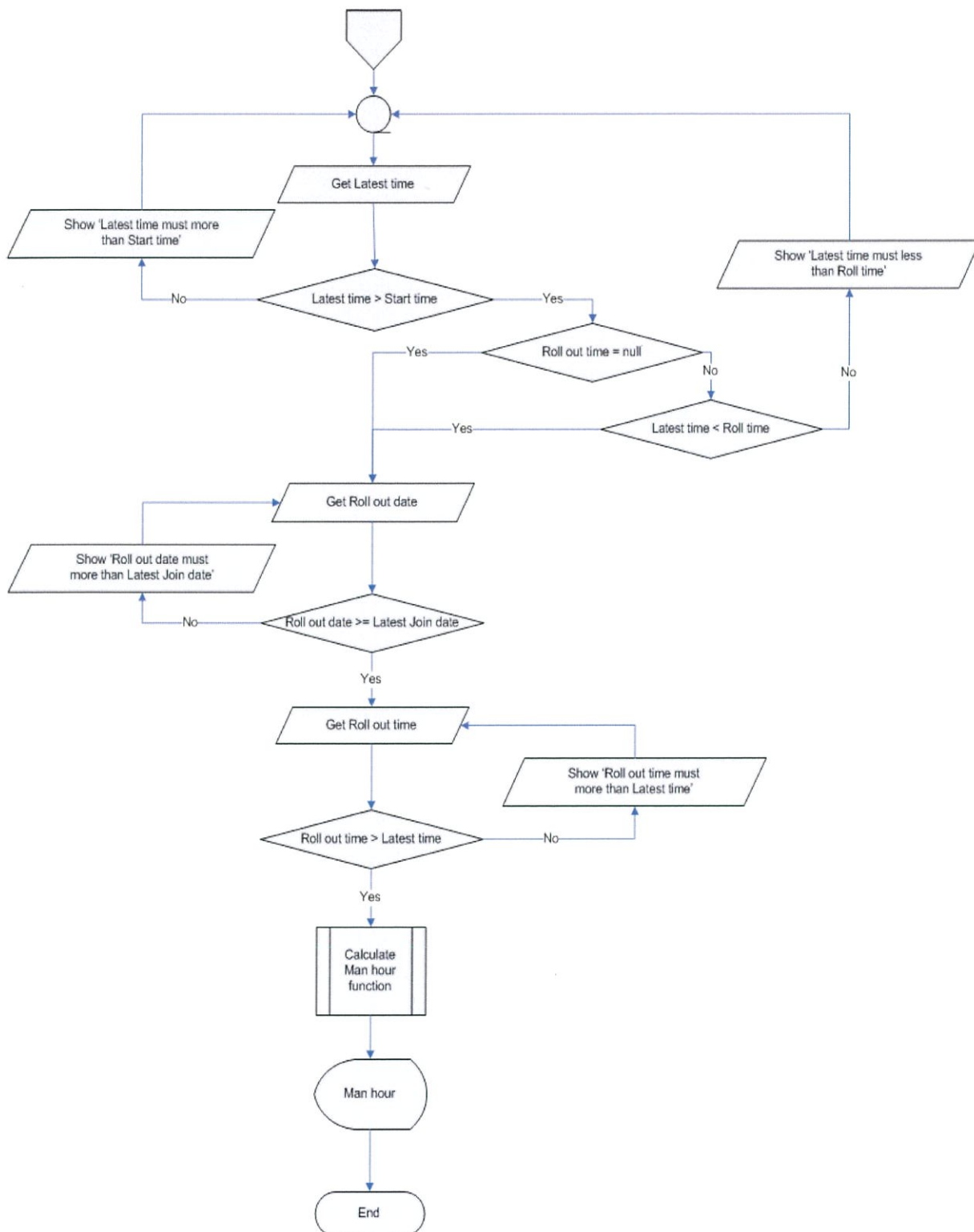
สำหรับหน้าคำนวณชั่วโมงการทำงานจะใช้ภาษา JavaScript ในการปรับแต่ง เพราะมีการคำนวณค่าภายในหน้าเดิม ไม่มีการรีเฟรชหน้าต่างก่อนที่จะแสดงผลลัพธ์

รูปที่ 3.10 แสดงหน้าต่างกรอกค่าคำนวณชั่วโมงการทำงาน (ก่อนปรับแต่ง)

ผังงานของการคำนวณชั่วโมงการทำงาน



รูปที่ 3.11 แสดงผังงานของการคำนวณชั่วโมงการทำงานหน้าที่ 1



รูปที่ 3.12 แสดงผังงานของการคำนวณชั่วโมงการทำงานหน้าที่ 2

## อธิบายผังงาน

1. เริ่มต้นโปรแกรม
2. รับข้อมูลจากคีย์บอร์ด ซึ่งเป็นจำนวนของพนักงานที่ต้องการ
3. รับข้อมูลวันที่เริ่มทำงาน (Start Join date)
4. ตรวจสอบเงื่อนไขวันที่เริ่มทำงาน (Start Join date) ต้องมากกว่าวันที่ปัจจุบัน (Current date) ถ้าผลที่ได้คือ Yes จะไปตรวจสอบเงื่อนไขต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Start Join date must more than Current date” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลวันที่เริ่มงานอีกครั้ง
5. ตรวจสอบเงื่อนไขวันที่เริ่มงานล่าสุด (Latest Join date) มีข้อมูลหรือไม่ (ใช้เพื่อตรวจสอบกรณีที่ยกรอกข้อมูลรอบแรกเสร็จ แล้วต้องการแก้ไขอีกครั้ง) ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะไปตรวจสอบเงื่อนไขต่อไป
  - 5.1 จากการณืที่ตรวจสอบเงื่อนไข ผลที่ได้คือ No จะตรวจสอบเงื่อนไขวันที่เริ่มทำงานต้องน้อยกว่าวันที่เริ่มงานล่าสุดหรือไม่ ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Start Join date must less than Latest Join date” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลวันที่เริ่มงานอีกครั้ง
6. รับข้อมูลเวลาของวันที่เริ่มทำงาน (Start time)
7. ตรวจสอบเงื่อนไขเวลาของวันที่เริ่มงานล่าสุด (Latest time) มีข้อมูลหรือไม่ (ใช้เพื่อตรวจสอบกรณีที่ยกรอกข้อมูลรอบแรกเสร็จ แล้วต้องการแก้ไขอีกครั้ง) ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะไปตรวจสอบเงื่อนไขต่อไป
  - 7.1 จากการณืที่ตรวจสอบเงื่อนไข ผลที่ได้คือ No จะตรวจสอบเงื่อนไขเวลาของวันที่เริ่มทำงานต้องน้อยกว่าเวลาของวันที่เริ่มทำงานล่าสุดหรือไม่ ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Start time must less than Latest time” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลเวลาของวันที่เริ่มทำงานอีกครั้ง
8. รับข้อมูลวันที่เริ่มงานล่าสุด (Latest Join date)
9. ตรวจสอบเงื่อนไขวันที่เริ่มทำงานล่าสุด (Latest Join date) ต้องมากกว่าหรือเท่ากับวันที่เริ่มทำงาน (Start Join date) ถ้าผลที่ได้คือ Yes จะไปตรวจสอบเงื่อนไขต่อไป แต่ถ้าผลที่ได้คือ No จะแสดง

ข้อความ “Latest Join date must more than Start Join date” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลวันที่เริ่มทำงานล่าสุดอีกครั้ง

10. ตรวจสอบเงื่อนไขวันที่สิ้นสุด (Roll out date) มีข้อมูลหรือไม่ (ใช้เพื่อตรวจสอบกรณีที่กรอกข้อมูลรอบแรกเสร็จ แล้วต้องการแก้ไขอีกครั้ง) ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะไปตรวจสอบเงื่อนไขต่อไป

10.1 จากการมีที่ตรวจสอบเงื่อนไข ผลที่ได้คือ No จะตรวจสอบเงื่อนไขวันที่เริ่มทำงานล่าสุดต้องน้อยกว่าวันที่สิ้นสุดหรือไม่ ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Latest Join date must less than Roll out date” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลวันที่เริ่มทำงานล่าสุดอีกครั้ง

11. รับข้อมูลเวลาของวันที่เริ่มทำงานล่าสุด (Latest time)

12. ตรวจสอบเงื่อนไขเวลาของวันที่เริ่มทำงานล่าสุดต้องมากกว่าเวลาของวันที่เริ่มงาน ถ้าผลที่ได้คือ Yes จะไปตรวจสอบเงื่อนไขต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Latest time must more than Start time” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลเวลาของวันที่เริ่มทำงานล่าสุดอีกครั้ง

13. ตรวจสอบเงื่อนไขเวลาของวันที่สิ้นสุด (Roll out time) มีข้อมูลหรือไม่ (ใช้เพื่อตรวจสอบกรณีที่กรอกข้อมูลรอบแรกเสร็จ แล้วต้องการแก้ไขอีกครั้ง) ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะไปตรวจสอบเงื่อนไขต่อไป

13.1 จากการมีที่ตรวจสอบเงื่อนไข ผลที่ได้คือ No จะตรวจสอบเงื่อนไขเวลาของวันที่เริ่มทำงานล่าสุดต้องน้อยกว่าเวลาของวันที่สิ้นสุดหรือไม่ ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Latest time must less than Roll time” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลเวลาของวันที่เริ่มทำงานล่าสุดอีกครั้ง

14. รับข้อมูลวันที่สิ้นสุด (Roll out date)

15. ตรวจสอบเงื่อนไขวันที่สิ้นสุดต้องมากกว่าหรือเท่ากับวันที่เริ่มทำงานล่าสุด ถ้าผลที่ได้คือ Yes จะไปรับข้อมูลอื่นต่อไป แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Roll out date must more than Latest Join date” ทางหน้าจอ แล้วย้อนกลับไปรับข้อมูลวันที่สิ้นสุดอีกครั้ง

16. รับข้อมูลเวลาของวันที่สิ้นสุด (Roll out time)

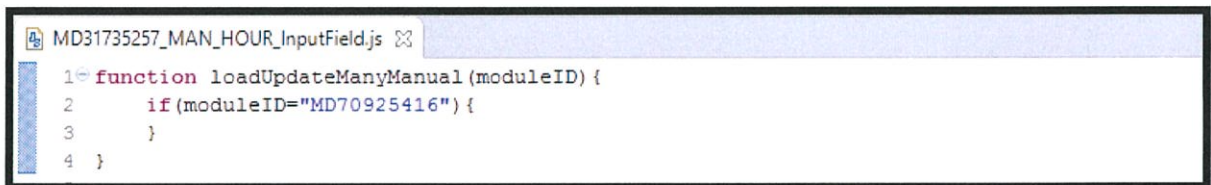
17. ตรวจสอบเงื่อนไขเวลาของวันที่สิ้นสุดต้องมากกว่าเวลาของวันที่เริ่มงานล่าสุด ถ้าผลที่ได้คือ Yes จะไปเรียกใช้ในส่วนของฟังก์ชันย่อย แต่ถ้าผลที่ได้คือ No จะแสดงข้อความ “Roll time must more than Latest time” ทางหน้าจอ แล้วย้อนกลับไปปรับข้อมูลเวลาของวันที่สิ้นสุดอีกครั้ง

18. ฟังก์ชันย่อยทำหน้าที่ในการคำนวณชั่วโมงการทำงานทั้งหมด โดยใช้อัลกอริทึมทางคณิตศาสตร์ เพื่อให้ได้ผลลัพธ์ตามที่ต้องการ

19. แสดงผลลัพธ์ชั่วโมงการทำงานออกทางหน้าจอ

20. สิ้นสุดโปรแกรม

เมื่อต้องการแก้ไขข้อมูลจะเรียกใช้งานฟังก์ชันที่ชื่อว่า loadUpdateManyManual(moduleID) แล้วเอาหมายเลข moduleID จากใน ONEWEB มาใส่ให้ถูกต้อง ใช้สำหรับเปลี่ยนค่าฟิลด์บนหน้าจอป๊อปอัพหลังจากที่กด Edit เพราะหน้าแก้ไขจะต้องทำการคำนวณเวลาใหม่เช่นเดียวกับการกรอกข้อมูลตอนแรก และเลือก Action สำหรับผูกไฟล์เป็น Update



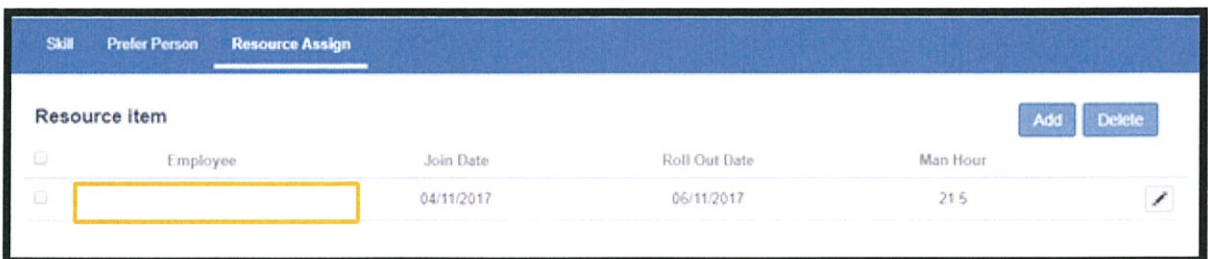
```

1 function loadUpdateManyManual (moduleID) {
2   if (moduleID='MD70925416') {
3   }
4 }

```

รูปที่ 3.13 แสดงการเรียกใช้ฟังก์ชัน loadUpdateManyManual()

### 3.4 การดึงข้อมูลจากฐานข้อมูลมาแสดง



Employee	Join Date	Roll Out Date	Man Hour
	04/11/2017	06/11/2017	215

รูปที่ 3.14 แสดงหน้าต่างของแถบด้านล่างในหน้าหลักไม่มี employee name

จากรูปที่ 3.14 แสดงหน้าต่างของแถบด้านล่างในหน้าหลัก ต้องพิจารณาก่อนว่าส่วนที่ต้องปรับแต่งเพิ่มเป็นระดับเอนทิตี (หน้าหลักทั้งหมด) หรือเป็นระดับโมดูล (บางส่วนของหน้าหลัก) ซึ่งแถบด้านล่างในหน้าหลัก

จะเป็นระดับโมดูล ต้องใช้ชื่อ module id + “\_session” หลังจากนั้นให้ไปดูแถบข้อมูลที่จะต้องใส่ว่าเป็นการใส่ข้อมูลแบบ one to one หรือ one to many ซึ่งหลังจากที่พิจารณาข้อมูลที่จะต้องใส่แล้ว เป็นการใส่ข้อมูลแบบ one to many และส่งข้อมูลกลับมาในรูปแบบของเวกเตอร์ ฉะนั้นจะต้องเลือกใช้งานใช้เมธอด MasterFormHandler.getStoreActionList() และเลือก Action สำหรับผูกไฟล์เป็น Insert ซึ่งการเขียนโค้ดแสดงได้ดังภาพต่อไปนี้

```
public class ManualEN_88691805359646ResourceAdmin extends ProcessHelper implements ProcessAction {
    Logger logger = Logger.getLogger(ManualEN_88691805359646ResourceAdmin.class);
    @Override
    public void modifyResultMany(String ModuleID, HashMap hStoreHashMap, String processMode, int row) {
        int updateRow = row;
        logger.debug("row :" + row);
        logger.debug("processMode :" + processMode);
        MasterFormHandler resourceAssigeformData = (MasterFormHandler)request.getSession().getAttribute(ModuleID+"_session");
        Vector resourceAssigeVect = resourceAssigeformData.getStoreActionList();
    }
}
```

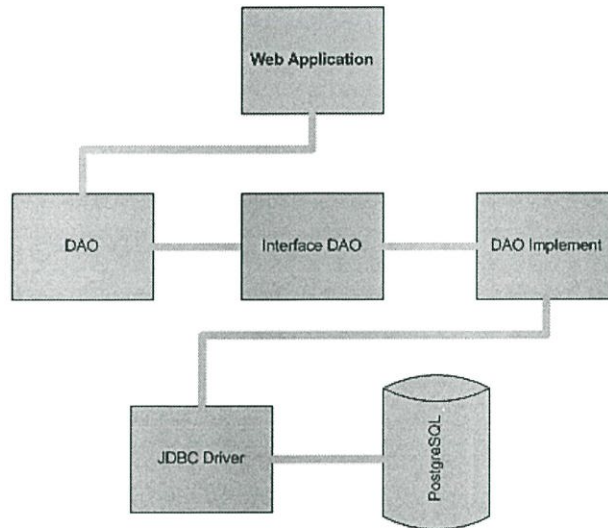
รูปที่ 3.15 แสดงโค้ดการเรียกใช้งานระดับโมดูล

Employee Id	Position
resourceadm1	resourceadm1
Join Date	Time
04/11/2017	08:00
Roll Out Date	Time
06/11/2017	05:30
Man Hour	21.5
Remark	
Test	

รูปที่ 3.16 แสดงข้อมูลทั้งหมดที่กรอกในหน้า Add

จากรูปที่ 3.16 ในช่องของรหัสของพนักงานไม่ได้ถูกดึงข้อมูลมาจากหน้า Add ที่ได้กรอกข้อมูลไว้ เนื่องจากข้อมูลที่สามารถเลือกได้เป็นรหัสของพนักงาน ไม่ใช่ชื่อของพนักงาน จึงไม่ต้องการให้นำรหัสพนักงานมาแสดงในแถบด้านล่างของหน้าหลัก

ฉะนั้นจะต้องทำการปรับแต่งเพิ่มเพื่อให้ชื่อของพนักงานมาแสดง โดยในหน้านี้สามารถเลือกใช้ภาษาจาวาในการปรับแต่งได้ เพราะข้อมูลจะถูกดึงไปแสดงหลังจากที่กดปุ่ม Save แล้วไปแสดงที่แถบด้านล่างของหน้าหลัก ซึ่งหมายความว่าจะมีการรีเฟรชหน้าหลักอีกครั้ง การดึงข้อมูลจากฐานข้อมูลมาแสดงมีหลักการดังรูป



รูปที่ 3.17 แสดงแผนภาพการทำงานของ การดึงข้อมูลจากฐานข้อมูล

จากรูปที่ 3.17 แผนภาพการทำงานของ การดึงข้อมูลจากฐานข้อมูลเริ่มต้นตั้งแต่เว็บแอปพลิเคชันใช้ DAO จะค่อยเป็นตัวกลางในการเชื่อมข้อมูลระหว่างฐานข้อมูลกับหน้าเว็บที่ใช้ในการแสดงผลข้อมูล ตัวไฟล์ DAO จะไปเรียกไฟล์ที่เก็บ Interface DAO ไว้ จากนั้นก็จะเรียกไฟล์ DAO Implement ภายในไฟล์จะมีการเขียนคำสั่งภาษา SQL ให้ไปดึงชื่อของพนักงานโดยที่ใช้รหัสพนักงานเป็นตัวค้นหา และมีคำสั่งให้เชื่อมต่อกับฐานข้อมูลโดยใช้ JDBC ซึ่งจะมีไดร์เวอร์สำหรับเชื่อมกับฐานข้อมูล PostgreSQL อยู่ ทำหน้าที่เชื่อมโยงโปรแกรมจาวากับฐานข้อมูลเข้าด้วยกัน อีกทั้งยังสามารถจัดการกับข้อมูลต่างๆ ที่อยู่ในฐานข้อมูลนั้นด้วย

เมื่อทราบหลักการการทำงานที่จะไปดึงข้อมูลมาจากฐานข้อมูลแล้ว อีกสิ่งที่สำคัญคือจะต้องทราบว่า Hashmap ที่ใช้ในการเก็บข้อมูลนั้นมีโครงสร้างเป็นแบบใด โดยสามารถเขียนคำสั่งภายในไฟล์เพื่อให้แสดงโครงสร้างออกมาดังรูป

```

1  {
2  INSERT = {
3  WF_RESOURCE_REQ_ITEM_ASSIGN = {
4  REQUEST_ITEM_ID = ,
5  EMPLOYEE_ID = resourceadm1,
6  ROLL_OUT_DATE = 2017 - 11 - 06,
7  START_JOIN_TIME_DESC = 08: 00,
8  ROLL_OUT_TIME_DESC = 05: 30,
9  START_JOIN_DATE = 2017 - 11 - 04,
10 ROLL_OUT_TIME = 05: 30,
11 REQUEST_ITEM_ASSIGN_ID = ,
12 EMP_POSITION = ,
13 START_JOIN_TIME = 08: 00,
14 REMARK = test,
15 MAN_HOUR = 21.5
16 }
17 }
18 }

```

รูปที่ 3.18 แสดงโครงสร้าง Hashmap ที่ใช้สำหรับเก็บข้อมูล

จากรูปที่ 3.18 แสดงโครงสร้าง Hashmap ที่ใช้สำหรับเก็บข้อมูล ในบทที่ 2 ได้กล่าวถึงการใช้งาน Hashmap ในการเก็บข้อมูลไปแล้ว โดยมีชื่อ Key และค่า Value อยู่ประกอประกกันเสมอ ดังนั้นเป้าหมายอยู่ที่ การจะเข้าไปดึงข้อมูลที่ Key คือ EMPLOYEE\_ID และ Value คือ resourceadm1 ซึ่งการที่จะเข้าไปถึงข้อมูลที่ ต้องการได้ จะต้องเข้าไปที่ละชั้นข้อมูล เริ่มต้นจากคีย์ตัวแรกคือ INSERT เมื่อเข้าไปแล้วให้เปรียบเสมือนว่า พบกับก้อน Hashmap ก้อนใหญ่หนึ่งก้อน ที่ยังไม่สามารถดึงข้อมูลที่ ต้องการได้ จึงต้องเข้าไปอีกชั้นที่มีคีย์คือ WF\_RESOURCE\_REQ\_ITEM\_ASSIGN หลังจากนั้นก็จะพบข้อมูลที่ ต้องการ ซึ่งการเขียนโค้ดแสดงได้ดังภาพ ต่อไปนี้

```

HashMap resourceAssigeHash = (HashMap) resourceAssigeVect.get(updateRow);
logger.debug("##### resourceAssigeformHash : " + resourceAssigeHash);
HashMap targetResourceAssigeHash = null;
if(resourceAssigeHash.containsKey("INSERT")){
    targetResourceAssigeHash = (HashMap) resourceAssigeHash.get("INSERT");
}else{
    targetResourceAssigeHash = (HashMap) resourceAssigeHash.get("UPDATE");
}
logger.debug("##### targetResourceAssigeHash : " + targetResourceAssigeHash);
HashMap ResourceAssigeHash = (HashMap) targetResourceAssigeHash.get("WF_RESOURCE_REQ_ITEM_ASSIGN");
logger.debug("##### ResourceAssigeHash : " + ResourceAssigeHash);
String employeeID = (String) ResourceAssigeHash.get("EMPLOYEE_ID");
logger.debug("##### employeeID : " + employeeID);

String name = "";
try{
    name = ManualDAOFactory.getManualWorkforceResourceDAO().getEmployeeNameById(employeeID);
}catch(Exception e){
    name = e.getMessage();
    e.printStackTrace();
}

ResourceAssigeHash.put("EMPLOYEE_NAME", name);

```

รูปที่ 3.19 แสดงโค้ดเรียกใช้งาน Hashmap

จากรูปที่ 3.19 แสดงตัวอย่างโค้ดเรียกใช้งาน Hashmap การเข้าไปภายในคีย์จะต้องใช้คำสั่ง get() แล้วตามด้วยชื่อ Key ที่ต้องการเข้า หลังจากที่เราเข้าไปเอาข้อมูลได้แล้วจะต้องนำไปใส่ไว้ในฟิลด์บนหน้าเว็บที่ต้องการจะให้เห็นชื่อออกมา โดยใช้คำสั่ง put() ตามด้วยชื่อฟิลด์และค่าที่ต้องการจะใส่ที่ถูกเก็บไว้ในตัวแปรที่ได้กำหนดไว้

### 3.5 การตรวจสอบความสมบูรณ์ของข้อมูล

The screenshot shows a web application interface for requesting hardware. At the top, there are 'Submit' and 'Back' buttons. The form contains several fields: Request Id (RS00000804), Requester (req1), Service Type\* (Hardware and Software), Project ([CFB01] The Crown Property Bureau), Phase (System Integration Test), Description (test1 send to pm1), Decision\* (Approve), and Status (In-Progress). Below the form, there are tabs for 'Hardware', 'Software', and 'Attach File'. Under the 'Hardware' tab, there is a table titled 'Request Hardware item' with columns: Hardware Name, Spec, Status, and Asset No. The 'Status' and 'Asset No' columns are highlighted with a yellow box.

รูปที่ 3.20 แสดงหน้าเว็บแอปพลิเคชันที่ต้องการตรวจสอบความสมบูรณ์ของข้อมูล

จากรูปที่ 3.20 แสดงหน้าเว็บแอปพลิเคชันที่ต้องการตรวจสอบความสมบูรณ์ของข้อมูล ในแถบด้านล่างของหน้าเว็บจะเห็นว่ายังมีข้อมูลในส่วนของ Status และ Asset No ที่ยังไม่ได้กรอกข้อมูล ซึ่งข้อมูลในส่วนนี้จำเป็นที่จะต้องใส่ ฉะนั้นต้องทำการปรับแต่งเพิ่มเติมเพื่อให้สามารถตรวจสอบว่าได้กรอกข้อมูลของ Status และ Asset No ก่อนที่จะกดปุ่ม Submit แล้วข้อมูลทั้งหมดจะถูกเก็บลงที่ฐานข้อมูล

สำหรับสิ่งที่ต้องพิจารณาก่อนที่จะทำการปรับแต่งเพิ่มเติมคือต้องวิเคราะห์ให้มากกว่าเป็นการตรวจสอบระดับเอนทิตี (หน้าหลักทั้งหมด) เพราะปุ่ม Submit ถูกใส่ไว้ในส่วนของหน้าหลัก ฉะนั้นจะต้องใช้ชื่อ entity id + “\_session” หลังจากนั้นต้องใส่ Vector Error ไว้สำหรับเพิ่มความที่ต้องการให้แสดงออกมาเพื่อให้ผู้ใช้งานทราบถึงส่วนที่ผิดพลาด และเลือก Action สำหรับผูกไฟล์เป็น Update ซึ่งการเขียนโค้ดแสดงได้ดังภาพต่อไปนี้

```

@Override
public boolean validateResult() {

    // TODO Auto-generated method stub
    // Get Request_ID from module first
    EntityFormHandler formEntity = (EntityFormHandler)request.getSession().getAttribute("EN_8927230061_v001_session");
    Vector vError = formEntity.getFormErrors();
}

```

รูปที่ 3.21 แสดงโค้ดการเรียกใช้งานระดับเอนทิตี

```

1 {
2   UPDATE = {
3     MODULE_FIELD_KEY = [REQUEST_ID, REQUEST_ITEM_HW_ID],
4     WF_SE_REQ_ITEM_HW = {
5       HARDWARE_ID = 1,
6       STATUS = null,
7       UPDATE_DATE = 2017 - 11 - 10,
8       ASSET_NO = null,
9       UPDATE_BY = admin,
10    REQUEST_ID = RS00000804,
11    REQUEST_ITEM_HW_ID = 192
12  }
13 }
14 }

```

รูปที่ 3.22 แสดง Hashmap ก่อนใส่ข้อมูลของ Status และ Asset No

จากรูปที่ 3.22 แสดง Hashmap ก่อนใส่ข้อมูล Status และ Asset No หลังจากที่ผูกกับหน้าเอนทิตีแล้ว ให้ตรวจสอบข้อมูลว่ามีข้อมูลอยู่หรือไม่ โดยปรีน log ออกมาดูโครงสร้าง Hashmap พบว่า Status เท่ากับ null และ Asset No เท่ากับ null ซึ่งกรณีนี้ จะไม่สามารถกด Submit แล้วย้อนกลับไปหน้าแรกได้

```

1 {
2   UPDATE = {
3     MODULE_FIELD_KEY = [REQUEST_ID, REQUEST_ITEM_HW_ID],
4     WF_SE_REQ_ITEM_HW = {
5       HARDWARE_NAME = Notebook,
6       HARDWARE_ID = 1,
7       STATUS = 01,
8       DIRTY_FLAG = DIRTY_FLAG,
9       ASSET_NO = 1232,
10    REQUEST_ID = RS00000804,
11    REQUEST_ITEM_HW_ID = 192,
12    SPEC = ram 4 Gig,
13    STATUS__DESC = Assigned
14  }
15 }
16 }

```

รูปที่ 3.23 แสดง Hashmap หลังใส่ข้อมูลของ Status และ Asset No

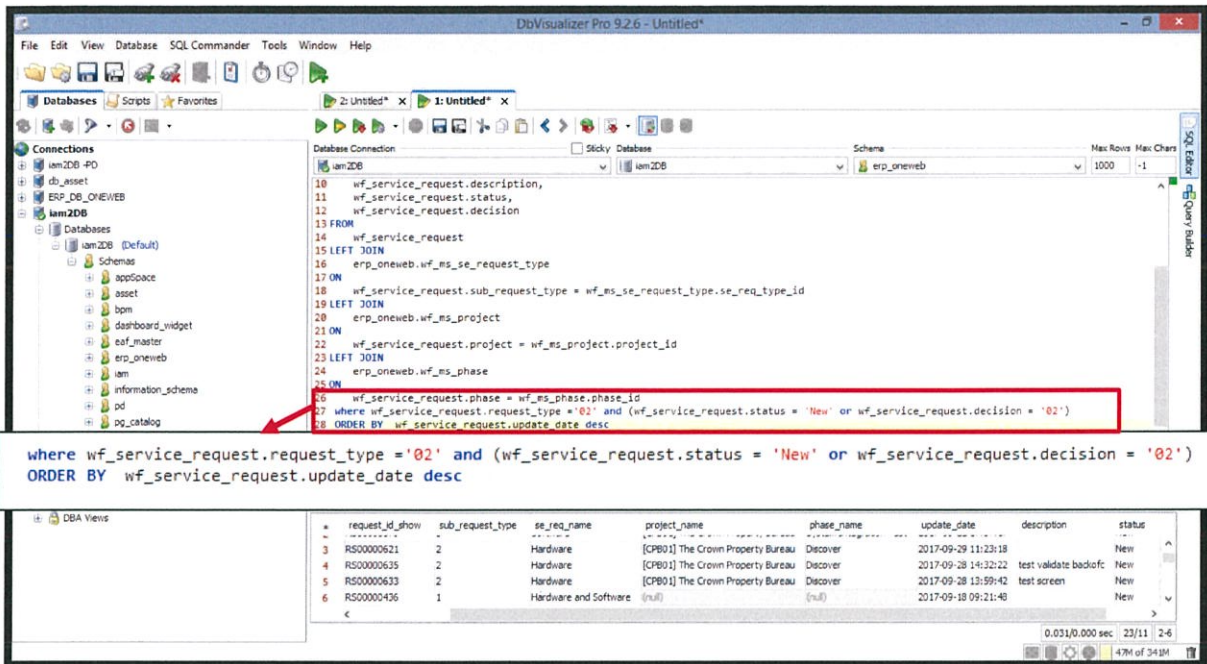
จากรูปที่ 3.23 แสดง Hashmap หลังใส่ข้อมูลของ Status และ Asset No ให้ตรวจสอบข้อมูลว่ามีอยู่หรือไม่ โดยปรีน log ออกมาดูโครงสร้าง Hashmap อีกครั้ง พบว่า Status เท่ากับ 01 ซึ่งหมายถึง Assigned และ Asset No เท่ากับ 1232 ซึ่งกรณีนี้จะสามารถกด Submit แล้วย้อนกลับไปหน้าแรกได้ เพราะข้อมูลมีอยู่ครบทุกฟิลด์ตามที่ต้องการและบันทึกลงในฐานข้อมูลต่อไป

### 3.6 การแบ่งผู้ใช้งานระบบ

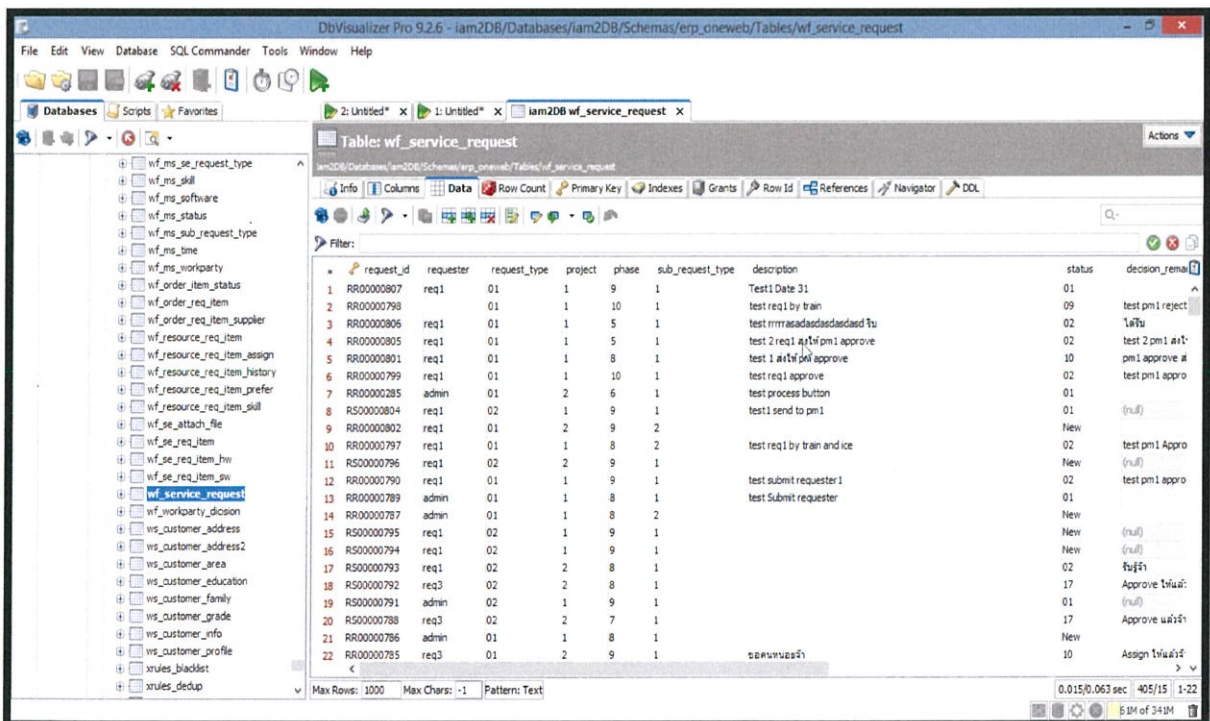
รูปที่ 3.24 แสดงหน้าเว็บแอปพลิเคชันที่ต้องการแบ่งผู้ใช้งานระบบ

จากรูปที่ 3.24 แสดงหน้าเว็บแอปพลิเคชันที่ต้องการแบ่งผู้ใช้งานระบบ วัตถุประสงค์คือต้องการให้ผู้ลงชื่อใช้งานระบบสามารถมองเห็นงานและแก้ไขงานเฉพาะของตนเองเท่านั้น เพราะถ้าไม่แบ่งผู้ใช้งานระบบไม่ว่าผู้ใช้งานจะลงชื่อเข้ามาด้วย Username อะไรตาม จะสามารถมองเห็นงานและแก้งานของทุกคนที่ใช้ระบบได้ หากเป็นเช่นนั้นระบบก็จะไม่มีความปลอดภัยด้วย ฉะนั้นจำเป็นต้องปรับแต่งเพิ่มเติมเพื่อให้ระบบมีความสมบูรณ์ยิ่งขึ้น

สำหรับการแบ่งผู้ใช้งานในระบบจะต้องพิจารณาว่าผู้ใช้งานแต่ละกลุ่มผู้ใช้งานมีบทบาทหน้าที่และสถานะเป็นอย่างไร เช่น ถ้าเป็นบทบาทของ Resource request คือมี request\_type เท่ากับ 01 และสถานะของผู้ใช้งานจะต้องเป็น New คือมีข้อมูล status เท่ากับ new หรือการถูกปฏิเสธงานกลับมาคือมี decision เท่ากับ 02 เป็นต้น จากนั้นให้เรียงข้อมูลโดยใช้คำสั่ง ORDER BY เพื่อให้ข้อมูลที่สร้างใหม่อยู่แถวบนสุดเรียงจากบนลงล่าง สามารถดูข้อมูลในฐานข้อมูลแล้วเขียนเงื่อนไขแสดงได้ดังรูปต่อไปนี้



รูปที่ 3.25 แสดงโค้ดที่เรียกดูข้อมูลในโปรแกรม Dbvisualizer



รูปที่ 3.26 แสดงข้อมูลผู้ใช้งานที่มีในระบบทั้งหมด

จากรูปที่ 3.26 แสดงข้อมูลผู้ใช้งานที่มีในระบบทั้งหมด จะพบว่าผู้ใช้งานหลากหลายบทบาทหน้าที่อยู่ในฐานข้อมูลปะปนกันอยู่ ไม่ได้แบ่งเป็นกลุ่มผู้ใช้งาน ดังนั้นการที่จะแบ่งผู้ใช้งานให้ไปแสดงในหน้าเว็บต้องเขียนคำสั่งเงื่อนไขดึงข้อมูลดูใน Dbvisualizer ถ้าได้ข้อมูลตรงตามที่กำหนดแล้ว ให้คัดลอกคำสั่ง SQL ที่ทดลองใน Dbvisualizer ไปใส่ในหน้าเอ็นทิตีที่ต้องการจะแบ่งกลุ่มผู้ใช้งานของ ONEWEB

การปรับแต่งเพิ่มเติมจะต้องพิจารณาว่าเป็นระดับเอ็นทิตีหรือระดับโมดูล ซึ่งการที่จะทำให้น้ำเว็บแอปพลิเคชันทั้งหน้าแบ่งผู้ใช้งานระบบจัดเป็นระดับเอ็นทิตี จึงต้องใช้ชื่อ entity id + “\_session” และเลือก Action สำหรับผูกไฟล์ที่ปุ่ม Search ซึ่งการเขียนโค้ดแสดงได้ดังภาพต่อไปนี้

```
public class ManualEN_3331123753954RequesterSearch extends ProcessHelper implements ProcessAction {
    Logger logger = Logger.getLogger(ManualEN_3331123753954RequesterSearch.class);

    @Override
    public HashMap getSearchResult() {
        MasterFormHandler moduleForm = (MasterFormHandler) request.getSession().getAttribute("MD29003177_session");
        String userName = (String) request.getSession().getAttribute("userName");
        StringBuilder appendSql = new StringBuilder(" and wf_service_request.requester = '");
        appendSql.append(userName).append("' ORDER BY ");

        String sql = moduleForm.getModuleM().getSqlWorkQueue();
        logger.info("##### old sql : " + sql);

        sql = sql.replace("ORDER BY", appendSql.toString());

        logger.info("##### new sql : " + sql);

        //moduleForm.getModuleM().setSqlWorkQueue(sql);

        return EAFManualUtil.loadTableData(request, sql, moduleForm.getPage(), moduleForm.getVolumePerPage());
    }
}
```

รูปที่ 3.27 แสดงโค้ดแบ่งผู้ใช้งานระบบ

จากรูปที่ 3.27 แสดงโค้ดแบ่งผู้ใช้งานระบบ และมีการเพิ่มโค้ดคำสั่ง SQL ในไฟล์ Java ที่จะนำไปผูกเพิ่มโดยใช้คำสั่ง append() ในการแทรกโค้ดคำสั่ง SQL

```
##### old sql :SELECT
where wf_service_request.request_type = '01' and (wf_service_request.status = 'New' or wf_service_request.decision = '02')
ORDER BY wf_service_request.update_date desc

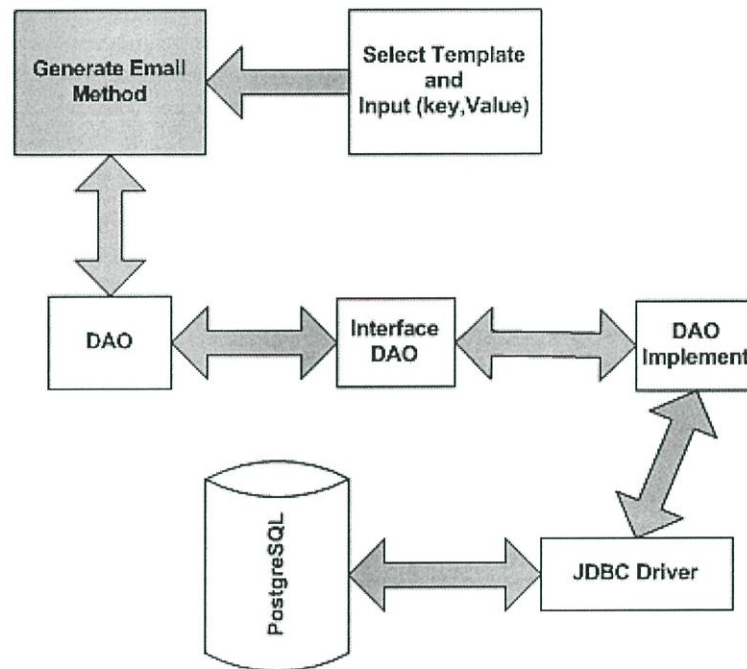
##### new sql :SELECT
where wf_service_request.request_type = '01' and (wf_service_request.status = 'New' or wf_service_request.decision = '02')
and wf_service_request.requester = 'admin' ORDER BY wf_service_request.update_date desc
```

รูปที่ 3.28 แสดงโค้ดคำสั่ง SQL ก่อนและหลังเพิ่มเงื่อนไข

จากรูปที่ 3.28 แสดงโค้ดคำสั่ง SQL ก่อนและหลังเพิ่มเงื่อนไขเมื่อนำไฟล์ Java ที่เขียนมาผูกแล้วจะได้เงื่อนไขในการแบ่งผู้ใช้งานระบบอย่างสมบูรณ์

### 3.7 การสร้างอีเมล

มีจุดประสงค์เพื่อที่สร้างอีเมลโดยที่มีรูปแบบของข้อความไว้แล้ว จากนั้นให้รับค่าเข้าไปแทนในข้อความที่อยู่ ซึ่งแนวคิดนี้สามารถนำไปต่อยอดพัฒนาใช้ในระบบต่อไป ใช้สำหรับส่งอีเมลแจ้งเตือนถึงผู้ที่เกี่ยวข้องเมื่อโปรเจกต์ดำเนินงานถึงจุดที่ต้องส่งงานให้กับพนักงานคนอื่นต่อไป โดยที่พนักงานไม่ต้องเขียนอีเมลส่งด้วยตนเอง ให้ระบบจัดการสร้างอีเมลขึ้นมา



รูปที่ 3.29 แสดงแผนภาพการทำงานของการสร้างอีเมล

จากรูปที่ 3.29 แสดงแผนภาพการทำงานของการสร้างอีเมลเริ่มต้นตั้งแต่กรอกพารามิเตอร์เพื่อเลือกเทมเพลตที่ต้องการใช้งานและใส่ค่าข้อมูลที่เป็น Key และ Value ให้ครบ ซึ่งผู้ใช้งานจะต้องทราบจำนวนค่าที่ต้องการแทนลงไปข้อความ จากนั้นเมธอดจะเรียกใช้ DAO จะค่อยเป็นตัวกลางในการเชื่อมต่อข้อมูลระหว่างฐานข้อมูลกับเมธอดที่เรียกใช้งาน ตัวไฟล์ DAO จะไปเรียกไฟล์ที่เก็บ Interface DAO ไว้ จากนั้นก็จะเรียกไฟล์ DAO Implement ภายในไฟล์จะมีการเขียนคำสั่งภาษา SQL ให้เอาค่าพารามิเตอร์ที่ผู้ใช้งานกรอกไปค้นหาเทมเพลต และมีคำสั่งให้เชื่อมต่อกับฐานข้อมูลโดยใช้ JDBC ซึ่งจะมีไดรเวอร์สำหรับเชื่อมกับฐานข้อมูล PostgreSQL อยู่ ทำหน้าที่เชื่อมโยงโปรแกรมจาวากับฐานข้อมูลเข้าด้วยกัน อีกทั้งยังสามารถจัดการกับข้อมูลต่างๆ ที่อยู่ในฐานข้อมูลนั้นด้วย โดยจะค้นหาในฐานข้อมูลว่ามีเทมเพลตข้อความตรงกับที่ผู้ใช้งานกรอกค่าพารามิเตอร์เข้ามาหรือไม่ แล้วทำการค้นหา Key ที่ตรงกับผู้ใช้งานส่งเข้ามาและแทนค่า Value ลงในข้อความ จะได้ข้อความที่แทนค่าแล้วกลับมาที่เมธอดเพื่อนำไปใช้งานต่อไป



```

public class ResultGenerateMail {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        GenerateEmail emailtemp = new GenerateEmail();
        try {
            HashMap<String, String> result = new HashMap<String, String>();
            HashMap<String, String> subjectKey = new HashMap<String, String>();
            subjectKey.put("REQ_ID", "987654321");
            subjectKey.put("REQUEST_NAME", "Torpido");

            HashMap<String, String> contentKey = new HashMap<String, String>();
            contentKey.put("REQ_ID", "123456789");
            contentKey.put("RECEIVER_NAME", "RICH");
            contentKey.put("SYSTEM_NAME", "Test Software");

            result = emailtemp.getTemplate("01", "PM_APPROVE", "en",subjectKey,contentKey);
            System.out.println("Generate E-Mail >>>>>"+result);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

รูปที่ 3.32 แสดงโค้ดที่ผู้งานส่งข้อมูลเพื่อไปสร้างอีเมล

จากรูปที่ 3.32 แสดงโค้ดที่ผู้งานส่งข้อมูลเพื่อไปสร้างอีเมล จะเห็นว่าเทมเพลตที่ผู้ใช้งานต้องการใช้คือ T0002 ที่มาจากการรอกค่าพารามิเตอร์ request\_type เท่ากับ 01 template\_code เท่ากับ PM\_APPROVE และ lang เท่ากับ 1 จากนั้นเอาคีย์ REQ\_ID และ REQUEST\_NAME ไปค้นหาหาระหว่าง { } แล้วทำการแทนค่าลงไป ในชุดข้อมูลที่ 1 และเอาคีย์ REQ\_ID RECEIVER\_NAME และ SYSTEM\_NAME ไปค้นหาแล้วแทนค่าเช่นเดียวกันในชุดข้อมูลที่ 2 แล้วจะได้อีเมลที่สมบูรณ์แบบนำไปใช้งานต่อไป

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 ผลลัพธ์หน้าการคำนวณชั่วโมงการทำงาน

ให้ทำการกรอกข้อมูลเพื่อคำนวณชั่วโมงในแบบคำขอที่ต้องการพนักงานในการทำแต่ละโครงการ

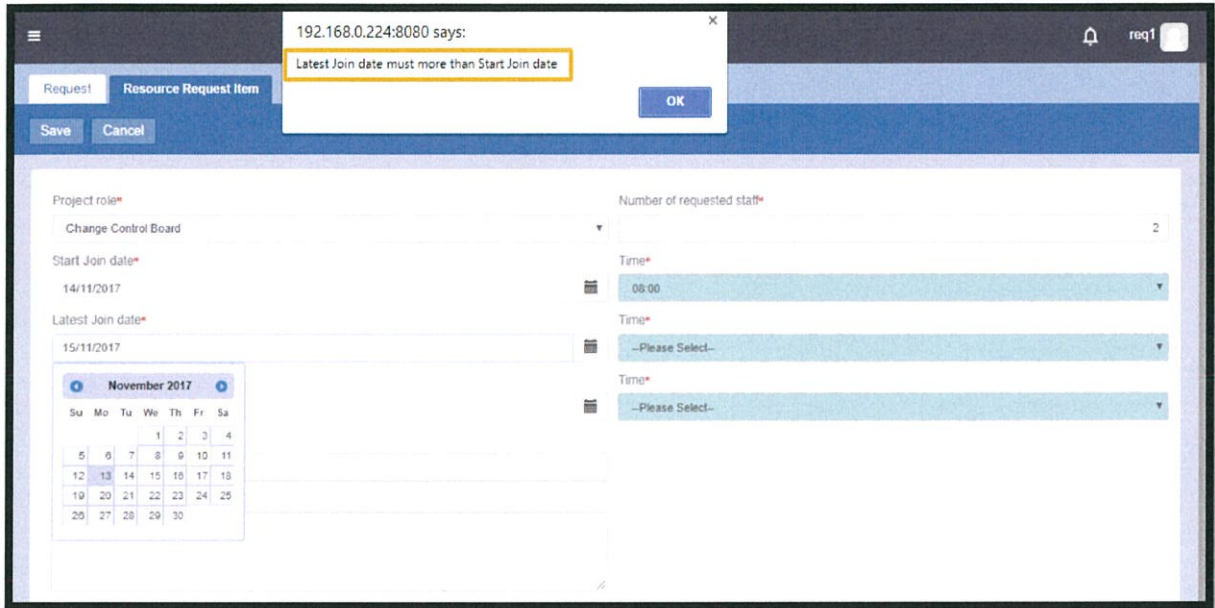
The screenshot shows a web application interface for 'Resource Request Item'. It features a header with 'Request' and 'Resource Request Item' tabs, and 'Save' and 'Cancel' buttons. The main form area contains several input fields: 'Project role\*' (dropdown menu with 'Change Control Board' selected), 'Start Join date\*', 'Latest Join date\*', 'Roll out date\*', 'Man hour\*', and 'Remark'. On the right side, there is a 'Number of requested staff\*' field with the value '2' and three 'Time\*' dropdown menus, each currently showing '--Please Select--'. The interface is clean and professional, with a blue header and white content area.

รูปที่ 4.1 แสดงหน้าคำนวณชั่วโมงการทำงาน (ก่อนกรอกข้อมูล)

This screenshot shows the same 'Resource Request Item' form as in Figure 4.1, but with a date picker open for the 'Start Join date\*' field. The date picker displays the month of November 2017, with dates from 1 to 30. A validation error message is shown in a yellow box: 'Start Join date must more than Current date'. The message is from the IP address 192.168.0.224:8080. The 'OK' button is visible next to the message. The form fields and layout are otherwise identical to the previous screenshot.

รูปที่ 4.2 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Start Join date

จากรูปที่ 4.2 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Start Join date โดยเลือกข้อมูลวันที่ของ Start Join date มีค่าน้อยกว่าวันที่ปัจจุบัน เช่น ปัจจุบันวันที่ 11/11/2017 แต่ไปเลือก Start Join date วันที่ 9/11/2017 เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Start Join date must more than Current date” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลวันที่เริ่มงานอีกครั้ง



รูปที่ 4.3 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Latest Join date

จากรูปที่ 4.3 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Latest Join date โดยเลือกข้อมูลวันที่ของ Latest Join date มีค่าน้อยกว่า Start Join date เช่น Start Join date เลือกวันที่ 14/11/2017 แต่ไปเลือก Latest Join date วันที่ 13/11/2017 เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Latest Join date must more than Start Join date” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลวันที่เริ่มทำงานซ้ำอีกครั้ง

The screenshot shows a web application interface for a 'Resource Request Item'. At the top, there is a notification box from IP 192.168.0.224:8080 with the message 'Latest time must more than Start time'. The form contains the following fields:

- Project role\***: Change Control Board
- Number of requested staff\***: 2
- Start Join date\***: 14/11/2017
- Time\***: 08:00
- Latest Join date\***: 14/11/2017
- Time\***: --Please Select--
- Roll out date\***: (empty)
- Time\***: --Please Select--
- Man hour\***: (empty)
- Remark**: (empty text area)

รูปที่ 4.4 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Latest time

จากรูปที่ 4.4 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Latest time โดยเลือกข้อมูลเวลาของ Latest time มีค่าน้อยกว่า Start time เช่น Start time เลือกเวลา 09.00 น. แต่ไปเลือก Latest time เวลา 08.30 น. เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Latest time must more than Start time” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลเวลาของวันที่เริ่มทำงานซ้ำอีกครั้ง

The screenshot shows the same web application interface. The notification box now displays the message 'Roll out date must more than Latest Join date'. The form fields are the same as in the previous image, but the 'Roll out date\*' field is now populated with a date from a calendar widget. The calendar shows the month of November 2017, with the date 14/11/2017 selected.

รูปที่ 4.5 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Roll out date

จากรูปที่ 4.5 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Roll out date โดยเลือกข้อมูลวันที่ของ Roll out date มีค่าน้อยกว่า Latest Join date เช่น Latest Join date เลือกวันที่ 14/11/2017 แต่ไปเลือก Roll out date วันที่ 13/11/2017 เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Roll out date must more than Latest Join date” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลวันที่สิ้นสุดอีกครั้ง

รูปที่ 4.6 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Roll time

จากรูปที่ 4.6 แสดงหน้าคำนวณชั่วโมงการทำงานกรอกข้อมูล Roll time โดยเลือกข้อมูลเวลาของ Roll time มีค่าน้อยกว่า Latest time เช่น Latest time เลือกเวลา 09.00 น. แต่ไปเลือก Roll time เวลา 08.30 น. เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Roll time must more than Latest time” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลเวลาของวันที่สิ้นสุดอีกครั้ง

รูปที่ 4.7 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Latest time

จากรูปที่ 4.7 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Latest time โดยเลือกข้อมูลเวลาของ Latest time ใหม่อีกครั้งแล้วมีค่ามากกว่า Roll time เช่น Roll time เลือกเวลา 10.00 น. แต่ไปเลือก Latest time เวลา 11.30 น. เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Latest time must less than Roll time” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลเวลาของวันที่เริ่มทำงานล่าสุดอีกครั้ง

รูปที่ 4.8 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Latest Join date

จากรูปที่ 4.8 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Latest Join date โดยเลือกข้อมูลวันที่ของ Latest Join date ใหม่อีกครั้งแล้วมีค่ามากกว่า Roll out date เช่น Roll out date เลือกวันที่ 14/11/2017 แต่ไปเลือก Latest Join date วันที่ 16/11/2017 เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Latest Join date must less than Roll out date” ทางหน้าจอ ต้องกลับไปกรอกข้อมูลวันที่เริ่มทำงานล่าสุดอีกครั้ง

รูปที่ 4.9 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Start time

จากรูปที่ 4.9 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Start time โดยเลือกข้อมูลเวลาของ Start time ใหม่อีกครั้งแล้วมีค่ามากกว่า Latest time เช่น Latest time เลือกเวลา 10.00 น. แต่ไปเลือก Start time เวลา 11.30 น. เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Start time must less than Latest time” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลเวลาของวันที่เริ่มทำงานอีกครั้ง

รูปที่ 4.10 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Start Join date

จากรูปที่ 4.10 แสดงหน้าคำนวณชั่วโมงการทำงานแก้ไขข้อมูล Start Join date โดยเลือกข้อมูลวันที่ของ Start Join date ใหม่อีกครั้งแล้วมีค่ามากกว่า Latest out date เช่น Latest out date เลือกวันที่ 14/11/2017 แต่ไปเลือก Start Join date วันที่ 17/11/2017 เป็นต้น ทำให้มีข้อความแจ้งเตือนว่า “Start Join date must less than Latest Join date” ทางหน้าจอ ต้องย้อนกลับไปกรอกข้อมูลวันที่เริ่มงานอีกครั้ง

รูปที่ 4.11 แสดงหน้าคำนวณชั่วโมงการทำงานเมื่อกรอกข้อมูลครบ

จากรูปที่ 4.11 แสดงหน้าคำนวณชั่วโมงการทำงานเมื่อกรอกข้อมูลครบ โดยในกรณีนี้ Latest Join date และ Roll out date เป็นวันเดียวกัน Latest time = 09.00 และ Roll time = 13.30 เนื่องจากผ่านตอนพักเที่ยงจะต้องลบออก 1 ชั่วโมง และต้องการจำนวนพนักงาน 2 คน จะคำนวณได้ดังนี้

$$(13.30 - 9 - 1) * 2 = 7 \text{ ชั่วโมง}$$

จะได้ว่าต้องการพนักงานมาโปรเจกต์นี้จำนวน 2 คน ใช้เวลาทำงานทั้งหมด 7 ชั่วโมง

#### 4.2 ผลลัพธ์หน้าดึงข้อมูลจากฐานข้อมูลมาแสดง

ต้องการส่งข้อมูลที่เป็นรหัสพนักงานเพื่อไปค้นหาชื่อของพนักงานในฐานข้อมูลแล้วส่งกลับมาจากนั้นให้แสดงแทนในฟิลด์ที่ได้กำหนดไว้ให้

Employee	Join Date	Roll Out Date	Man Hour
No record			

รูปที่ 4.12 แสดงหน้าต่างของแถบด้านล่างในหน้าหลักก่อน Add ข้อมูล

รูปที่ 4.13 แสดงหน้าต่างของหน้า Add พนักงาน

จากรูปที่ 4.13 แสดงหน้าต่างของหน้า Add พนักงานจะต้องเลือก Employee Id และระยะเวลาทำงานที่ต้องการจะกำหนดให้กับพนักงาน

Employee	Join Date	Roll Out Date	Man Hour
resourceadm1	04/11/2017	06/11/2017	215

รูปที่ 4.14 แสดงหน้าเว็บหลังจากที่ Add พนักงาน

จากรูปที่ 4.14 แสดงหน้าเว็บหลังจากที่ Add พนักงานแล้ว จะมีข้อมูลที่กรอกในหน้า Add มาแสดงที่แถบด้านล่างของหน้าหลัก พร้อมกับชื่อของพนักงานที่ทำการปรับแต่งเพิ่มเติมเรียบร้อยแล้ว

### 4.3 ผลลัพธ์หน้าการตรวจสอบความสมบูรณ์ของข้อมูล

ต้องการตรวจสอบข้อมูลในฟิลด์ที่จำเป็นต้องกรอกข้อมูล ก่อนที่จะบันทึกข้อมูลลงฐานข้อมูล หากว่ากรอกข้อมูลไม่ครบจะมีข้อความแสดงเตือนผู้ใช้งาน

Request Id	Requester
RS00000004	req1

Hardware Name	Spec	Status	Asset No
Notebook	ram 4 Gig		

รูปที่ 4.15 แสดงหน้าเว็บที่ต้องการตรวจสอบความสมบูรณ์ของข้อมูล

จากรูปที่ 4.15 แสดงหน้าเว็บที่ต้องการตรวจสอบความสมบูรณ์ของข้อมูล ซึ่งที่แถบด้านล่างของหน้าหลักที่ฟิลด์ของ Status และ Asset No ยังไม่มีข้อมูล เมื่อกดปุ่ม Submit จะมีข้อความแสดงเตือนดังรูป

รูปที่ 4.16 แสดงหน้าเว็บที่มีข้อความแจ้งเตือนเมื่อใส่ข้อมูลไม่ครบ

รูปที่ 4.17 แสดงหน้าต่างแก้ไขข้อมูล

จากรูปที่ 4.17 แสดงหน้าต่างแก้ไขข้อมูลที่ให้กรอกข้อมูล Asset No และ เลือก Status แล้วทำการบันทึกข้อมูลเพื่อแก้ไขให้ข้อมูลครบสมบูรณ์

The screenshot shows a web application interface for managing requests. At the top, there are 'Submit' and 'Back' buttons. The main form contains the following fields:

- Request Id: R50000004
- Requester: req1
- Service Type\*: Hardware and Software
- Project: [CPB01] The Crown Property Bureau
- Phase: System Integration Test
- Description: test1 send to pm1
- Decision\*: Approve
- Status: In-Progress
- Decision Remark\*: Test

Below the form, there are tabs for 'Hardware', 'Software', and 'Attach File'. Under the 'Hardware' tab, there is a table titled 'Request Hardware item' with the following data:

Hardware Name	Spec	Status	Asset No
Notebook	ram 4 Gig	Assigned	1232

รูปที่ 4.18 แสดงหน้าเว็บหลักหลังจากแก้ไขข้อมูล

จากรูปที่ 4.18 แสดงหน้าเว็บหลักหลังจากแก้ไขข้อมูล ที่แถบด้านล่างของหน้าหลักมีข้อมูลในฟิลด์ครบทุกฟิลด์ เมื่อทำการกดปุ่ม Submit ก็จะไม่มีความแจ้งเตือนแสดงขึ้นมา

#### 4.4 ผลลัพธ์หน้าการแบ่งผู้ใช้งานระบบ

ต้องการดึงข้อมูลที่เกี่ยวข้องกับผู้ใช้งานระบบมาจากฐานข้อมูล เพื่อให้สามารถแก้ไขหรือดูข้อมูลเฉพาะส่วนของตัวเองเท่านั้น ไม่สามารถเห็นของผู้ใช้งานคนอื่นได้

The screenshot shows a web application interface for 'WF Resource Request (Requester)'. The form contains the following fields:

- Request Id
- Requester
- Resource Type: -Please Select-
- Project1: -Please Select-
- Phase: -Please Select-
- Status: -Please Select-

Below the form, there are buttons for 'Search', 'Reset', and 'Create a New Request'. At the bottom, there is a table with the following columns: No, Request Id, Requester, Resource Type, Project, Phase, Create Date, and Status. Below the table, there is a search prompt: 'Please enter search criteria to search a record(s):'.

รูปที่ 4.19 แสดงหน้าเว็บแอปพลิเคชันที่ต้องการแบ่งผู้ใช้งานระบบ

เมื่อล็อกอินเข้าใช้งานในระบบแล้ว จะเห็นเฉพาะงานของตนเองเท่านั้น เพื่อความปลอดภัยของผู้ใช้งาน

The screenshot shows the 'WF Resource Request (Requester)' interface. At the top, there are search filters for Request Id, Requester, Resource Type, Project1, Phase, and Status. Below the filters are buttons for 'Search', 'Reset', and 'Create a New Request'. The main area displays a table of requests with columns: No, Request Id, Requester, Resource Type, Project, Phase, Create Date, and Status. The 'Requester' column for all rows is highlighted in yellow and contains the value 'admin'.

No	Request Id	Requester	Resource Type	Project	Phase	Create Date	Status
1	RR00000850	admin	Normal Resource	[CPB01] The Crown Property Bureau	System Integration Test	2017-11-07 08:42:43.738	New
2	RR00000808	admin	System Engineer	[CPB01] The Crown Property Bureau	System Integration Test	2017-10-31 10:43:18.594	New
3	RR00000787	admin	System Engineer	[CPB01] The Crown Property Bureau	Integrationtest	2017-10-16 14:29:06.123	New
4	RR00000786	admin	Normal Resource	[CPB01] The Crown Property Bureau	Integrationtest	2017-10-16 14:18:39.289	New
5	RR00000781	admin	Normal Resource	[CPB01] The Crown Property Bureau	Integrationtest	2017-10-11 17:31:48.507	New
6	RR00000780	admin	System Engineer	[INT01] Internal Project	Integrationtest	2017-10-11 17:27:46.657	New
7	RR00000628	admin	Normal Resource	[CPB01] The Crown Property Bureau	Discover	2017-09-28 12:16:25.082	New
8	RR00000773	admin	System Engineer	[INT01] Internal Project	User Acceptance Test	2017-10-11 13:52:48.262	New

รูปที่ 4.20 แสดงภาพการล็อกอินเข้าสู่ระบบด้วย admin

The screenshot shows the 'WF Resource Request (Requester)' interface. At the top, there are search filters for Request Id, Requester, Resource Type, Project1, Phase, and Status. Below the filters are buttons for 'Search', 'Reset', and 'Create a New Request'. The main area displays a table of requests with columns: No, Request Id, Requester, Resource Type, Project, Phase, Create Date, and Status. The 'Requester' column for all rows is highlighted in yellow and contains the value 'req1'.

No	Request Id	Requester	Resource Type	Project	Phase	Create Date	Status
1	RR00000802	req1	System Engineer	[INT01] Internal Project	System Integration Test	2017-10-30 16:06:21.487	New
2	RR00000785	req1	Normal Resource	[CPB01] The Crown Property Bureau	System Integration Test	2017-10-10 14:29:45.85	New
3	RR00000764	req1	Normal Resource	[INT01] Internal Project	System Integration Test	2017-10-10 14:28:47.654	New
4	RR00000761	req1	System Engineer	[CPB01] The Crown Property Bureau	Discover	2017-10-10 10:47:50.384	New
5	RR00000689	req1	Normal Resource	[CPB01] The Crown Property Bureau	Discover	2017-10-06 11:22:02.364	New
6	RR00000435	req1	Normal Resource	[CPB01] The Crown Property Bureau	System Integration Test	2017-09-15 19:27:41.007	New
7	RR00000439	req1	System Engineer	[CPB01] The Crown Property Bureau	Integrationtest	2017-09-18 09:56:57.648	New
8	RR00000594	req1	System Engineer	[CPB01] The Crown Property Bureau	Discover	2017-09-25 17:22:34.577	New
9	RR00000592	req1	System Engineer	[CPB01] The Crown Property Bureau	Discover	2017-09-25 17:20:30.38	New
10	RR00000516	req1	Normal Resource	[CPB01] The Crown Property Bureau	Discover	2017-09-21 15:21:54.689	New

รูปที่ 4.21 แสดงภาพการล็อกอินเข้าสู่ระบบด้วย req1

#### 4.5 ผลลัพธ์หน้าการสร้างอีเมล

ต้องการส่งค่าที่ต้องการจะแทนเข้าไป แล้วทำการแทนที่ค่านั้นๆในข้อความ เพื่อให้ได้อีเมลที่มีรายละเอียดตรงตามที่ต้องการจะใช้งาน

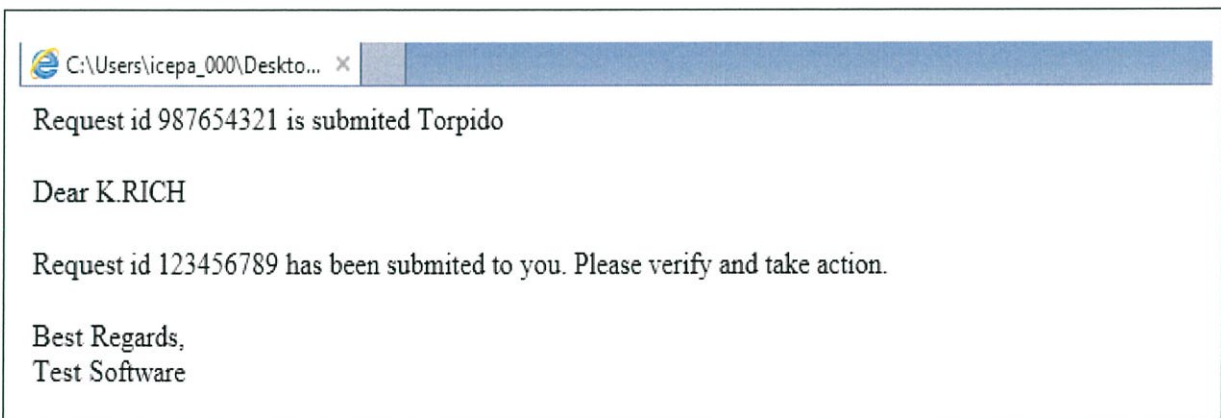
```
SubjectKey
("REQ_ID", "987654321")
("REQUEST_NAME", "Torpido")

ContentKey
("REQ_ID", "123456789")
("RECEIVER_NAME", "RICH")
("SYSTEM_NAME", "Test Software")
```

รูปที่ 4.22 แสดงข้อมูลที่ผู้ใช้งานต้องการแทนค่าในอีเมล

```
Generate E-Mail >>>>>
Request id 987654321 is submitted Torpido
<div>Dear K.RICH</div><div><br><div>Request id 123456789 has been submitted to you
Please verify and take action.</div><div><br>Best Regards,<br>Test Software</div>
```

รูปที่ 4.23 แสดงอีเมลที่แทนค่าข้อมูลเรียบร้อยแล้ว



รูปที่ 4.24 แสดงอีเมลบน Internet Explorer

จากรูปที่ 4.24 เมื่อนำไปใช้งานบนเว็บไซต์สามารถนำไปประยุกต์ใช้งานได้หลากหลาย หากเอาไปใช้ใน ระบบที่ต้องใช้งานเกี่ยวกับการส่งอีเมล จะช่วยเพิ่มความสำเร็จให้ระบบยิ่งขึ้น

## บทที่ 5

### สรุปผลการปฏิบัติงาน

โครงการพัฒนาระบบบริหารแบบคำขอภายในองค์กรโดยใช้เฟรมเวิร์ก ONEWEB 4.0 ทำให้สามารถเพิ่มพูนความรู้และทักษะในการพัฒนาระบบสารสนเทศ ทักษะในการเขียนโปรแกรม การแก้ไขปัญหา รวมถึงการเพิ่มประสบการณ์ในการทำงานที่ต้องเจอในอนาคต ซึ่งผู้จัดทำได้รับมอบหมายงานจากพนักงานที่ปรึกษา รวมทั้งสิ้น 5 งานดังนี้

1. การคำนวณชั่วโมงการทำงาน
2. การดึงข้อมูลจากฐานข้อมูลมาแสดง
3. การตรวจสอบความสมบูรณ์ของข้อมูล
4. การแบ่งผู้เข้าใช้งานระบบ
5. การสร้างอีเมล

#### 5.1 ปัญหาที่พบ

5.1.1 มีการสื่อสารกับพนักงานที่ปรึกษาหรือพี่ที่ทำงานร่วมกันไม่ชัดเจน จึงเกิดความเข้าใจที่ไม่ตรงกันทำให้ทำงานผิดพลาดส่งผลให้เกิดความล่าช้าในการทำงาน

5.1.2 ความรู้ในการปฏิบัติงานมีไม่เพียงพอ ต้องพยายามไฝ่หาความรู้เพื่อพัฒนาตนเองอยู่เสมอ

#### 5.2 แนวทางในการแก้ปัญหา

5.2.1 ทุกครั้งที่ได้รับมอบหมายงานจะต้องมีการทวนซ้ำให้พนักงานที่ปรึกษาฟังว่าผู้จัดทำเข้าใจถูกต้องหรือไม่ ถ้าไม่เข้าใจที่ตรงไหนก็จะสอบถามทันที

5.2.2 เมื่อผู้จัดทำเลิกงานหรือมีเวลาว่างจะหาข้อมูลเพิ่มเติมจากแหล่งข้อมูลต่างๆ เพื่อพัฒนาความรู้และทักษะในการทำงานของตนเองให้เพียงพอต่อการทำงาน

#### 5.3 ข้อเสนอแนะ

การทำงานใดๆ ควรเข้าใจในงานที่ต้องทำอย่างชัดเจนก่อนลงมือทำทุกครั้ง ถ้าลงมือทำทั้งๆที่ยังไม่เข้าใจงาน อาจเกิดความล่าช้าในงานและอาจทำให้เสียความน่าเชื่อถือได้ นอกจากนี้ควรหมั่นศึกษาหาความรู้อยู่เสมอเพื่อให้สามารถทำงานที่ได้รับมอบหมายได้ดี และเพิ่มโอกาสในการตอบรับเข้าทำงานในบริษัทที่ทำสหกิจอีกด้วย

## เอกสารอ้างอิง

- [1] ทีมงาน TechTalkThai. (2559, 25 เมษายน). Avalant OneWeb. ค้นเมื่อ 27 ตุลาคม 2560, จาก <https://www.techtalkthai.com/introduce-avalant-oneweb-configuration-based-enterprise-application-development/>.
- [2] MGR Online. (2559, 30 พฤศจิกายน). Avalant OneWeb. ค้นเมื่อ 27 ตุลาคม 2560, จาก <http://www.manager.co.th/game/viewnews.aspx?NewsID=9590000119645>
- [3] วิกิพีเดีย สารานุกรมเสรี. (2560, 29 มิถุนายน). ภาษาจาวา. ค้นเมื่อ 28 ตุลาคม 2560, จาก <https://th.wikipedia.org/wiki/ภาษาจาวา>
- [4] MarcusCode. (2560, 29 เมษายน). ภาษา Java. ค้นเมื่อ 28 ตุลาคม 2560, จาก <http://marcuscode.com/lang/java/object-oriented-programming>
- [5] boxsingle.com. (2556, 10 สิงหาคม). DAO (data access object) คือ. ค้นเมื่อ 28 ตุลาคม 2560, จาก <http://www.boxsingle.com/?page=Blog.ShowBlogDetail&blogID=4#>
- [6] MindPHP.COM. (2559, 8 พฤศจิกายน). jQuery คืออะไร. ค้นเมื่อ 29 ตุลาคม 2560, จาก <http://www.mindphp.com/คู่มือ/73-คืออะไร/3863-what-is-jquery.html>
- [7] P.Metee Blog (2558, มกราคม 16). สัญลักษณ์ Flowchart. ค้นเมื่อ 4 พฤศจิกายน 2560, จาก <http://meteeblog.blogspot.com/2015/01/flowchart.html>