



## รายงานสหกิจศึกษาฉบับสมบูรณ์

เว็บไซต์เก็บข้อมูลสินค้าคงคลัง (แบ็กเอนด์)  
Web-based Inventory System (Backend)

นางสาวฐิติรัตน์ ปฐมพรสุริยะ

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิชาวิศวกรรมสารสนเทศ  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560



## รายงานสหกิจศึกษาฉบับสมบูรณ์

เว็บไซต์เก็บข้อมูลสินค้าคงคลัง (แบ็กเอนด์)

Web-based Inventory System (Backend)

นางสาวฐิติรัตน์ ปฐมพรสุริยะ

ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา	เว็บไซต์เก็บข้อมูลสินค้าคงคลัง (แบ็กเอนด์)
ชื่อ-สกุล นักศึกษา	ฐิติรัตน์ ปฐมพรสุริยะ
คณะ	วิศวกรรมศาสตรบัณฑิต สาขาวิชา วิศวกรรมสารสนเทศ
ชื่อ-สกุล อาจารย์นิเทศ	ผศ.มยุรี เลิศเวชกุล
ชื่อ-สกุล ผู้นิเทศงาน	พิบูลย์ พรประสิทธิ์
สถานประกอบการ	บริษัทไอบีเอ็ม โซลูชั่นส์ ดิลิเวอรี จำกัด

### บทคัดย่อ

ในการปฏิบัติงานในทีมเน็ตเวิร์ค บริษัท ไอบีเอ็ม โซลูชั่นส์ ดิลิเวอรี จำกัด จะมีการนำเอาอุปกรณ์เครือข่ายออกไปเพื่อใช้ในการติดตั้งให้แก่ลูกค้า หรือนำอุปกรณ์ไปเพื่อประกอบการทำงานด้วยเสมอ ซึ่งระเบียบในการทำงานได้กำหนดไว้ให้พนักงานจะต้องลงบันทึกการนำอุปกรณ์ออกและการนำเข้าอุปกรณ์ทุกครั้ง แต่อย่างไรก็ตามมักเกิดปัญหาที่เกิดจากการที่มีผู้นำอุปกรณ์ออกไปโดยไม่ลงบันทึกข้อมูล หรือเกิดจากความล่าช้าในการรวมไฟล์ข้อมูลซึ่งจัดเก็บในรูปแบบของไฟล์เอ็กเซล (Microsoft Excel) ทำให้ข้อมูลที่มีอยู่ในระบบไม่เป็นข้อมูลที่ล่าสุด เมื่อเช็คสินค้าคงคลังจะพบว่าจำนวนอุปกรณ์ที่มีอยู่ในระบบไม่ตรงกับจำนวนสินค้าที่มีอยู่จริง จึงมีแนวคิดในการจัดทำระบบข้อมูลสินค้าคงคลังขึ้นมาเพื่อใช้ประโยชน์ในการติดตามอุปกรณ์ สามารถรู้สถานที่เก็บอุปกรณ์ และจำนวนอุปกรณ์ได้แม่นยำมากยิ่งขึ้น ระบบข้อมูลสินค้าคงคลังที่พัฒนาขึ้นมีส่วนช่วยให้การทำงานในทีมเน็ตเวิร์คเป็นระบบและมีประสิทธิภาพมากยิ่งขึ้น

Cooperative Title	Inventory log Website (Backend)
Student intern name	Thitirat Pathompornsuriya
Faculty	Engineering Department Information Technology
Advisor name	Asst.Mayuree Lertwetchakul
Mentor name	Piboon Ponprasit
Company	IBM Solutions Delivery Co., Ltd.

## ABSTRACT

Generally, task of the network operation team, IBM Solutions Delivery Co., Ltd. is to install networking devices at customer's site. The operation rules defined that storing and taking-out the network device activities must be recorded correctively, and up-to-date. But the problem always occurs when network operation team member does not record some device taking activity or because of delaying of file merging procedure since the inventory data could be kept in many separated Excel files. This makes the network device data in the inventory system incorrect. Web-based inventory system is developed as to solve this problem. The web-based inventory system could be used to record the inventory activities from anywhere, anytime, user able to query the devices data to show a device's location, the number of available devices more easily. By utilizing the web-based inventory system, the correct information of the inventory system makes the network operation team working is pretty more systematically and more efficiently.

## กิตติกรรมประกาศ

โครงการเว็บไซต์เก็บข้อมูลสินค้าคงคลังนี้ สำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลืออย่างสูงจาก ผศ.มยุรี เลิศเวชกุลอาจารย์นิเทศ และ คุณพิบูลย์ พรประสิทธิ์ ผู้นิเทศงาน ที่ได้กรุณาให้คำปรึกษา แนะนำ และตรวจสอบแก้ไขข้อบกพร่องในทุกขั้นตอนของการจัดทำโครงการ ดิฉันขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณบิดา มารดา เพื่อนนักศึกษา ตลอดจนผู้ที่เกี่ยวข้องทุกท่านที่ไม่ได้กล่าวนามไว้ ณ ที่นี้ที่ให้อำนาจใจ และมีส่วนช่วยเหลือให้โครงการฉบับนี้สำเร็จลุล่วงได้ด้วยดี

ท้ายที่สุด ดิฉันหวังเป็นอย่างยิ่งว่าโครงการฉบับนี้จะเป็นประโยชน์กับผู้สนใจไม่มากนักน้อย

นางสาวฐิติรัตน์ ปฐมพรสุริยะ

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ .....	III
สารบัญ .....	IV
สารบัญตาราง .....	VI
สารบัญภาพ .....	VII
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญ .....	1
1.2 วัตถุประสงค์ของโครงการ .....	1
1.3 ขอบเขตของโครงการ .....	2
1.4 วิธีการดำเนินการ .....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	3
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง .....	4
2.1 ทฤษฎีภาษาที่ใช้ในการสร้างเว็บไซต์ .....	4
2.1.1 SQL .....	4
2.1.2 PHP .....	10
2.2 ทฤษฎีระบบฐานข้อมูล (database) .....	18
2.2.1 ความหมายของระบบฐานข้อมูล .....	18
2.2.2 โครงสร้างของฐานข้อมูล .....	19
2.2.3 ชนิดของข้อมูลของเขตข้อมูลในตารางข้อมูล .....	20
2.2.4 องค์ประกอบของระบบฐานข้อมูล .....	21
2.2.5 ระบบจัดการฐานข้อมูล (DBMS) .....	22
2.2.6 ผังงาน (Flowchart) .....	27
2.2.7 Use case diagram .....	30
2.2.8 ER Diagram .....	35
2.2.9 คีย์ (key) .....	40

## สารบัญ(ต่อ)

	หน้า
บทที่3 วิธีการดำเนินงาน.....	41
3.1 ขั้นตอนและวิธีการดำเนินงาน.....	41
3.1.1 ลงโปรแกรมที่ใช้ในการทำงาน.....	41
3.1.2 ออกแบบ Flowchart.....	42
3.1.3 ออกแบบ Use case diagram.....	43
3.1.4 ออกแบบระบบฐานข้อมูล.....	43
3.2 สร้างระบบฐานข้อมูล.....	44
3.2.1 ทำระบบ Login.....	44
3.2.2 เชื่อมระบบฐานข้อมูลกับหน้าเว็บ.....	45
บทที่ 4 ผลการดำเนินงาน.....	46
4.1 การเข้าสู่ระบบ.....	46
4.1.1 Sign in.....	46
4.1.2 Register.....	46
4.2 โครงสร้างของเว็บไซต์.....	47
4.2.1 Admin.....	47
4.2.2 Implementer.....	48
4.2.3 ScoA&D.....	49
4.2.4 Network monitor.....	50
4.3 การเปลี่ยนรหัส.....	51
บทที่ 5 สรุปผลดำเนินการ และข้อเสนอแนะ.....	51
5.1 การดำเนินการจัดทำโครงการ.....	51
5.1.1 วัตถุประสงค์ของโครงการ.....	51
5.1.2 วัสดุ อุปกรณ์ เครื่องมือหรือโปรแกรมที่ใช้ในการพัฒนา.....	51
5.2 สรุปผลการดำเนินโครงการ.....	51
5.3 ข้อเสนอแนะและปัญหาอุปสรรค.....	52
เอกสารอ้างอิง.....	53

## สารบัญตาราง

ภาพที่	หน้า
2.1 แสดงคำสั่ง select.....	6
2.2 แสดงตัวแปรในภาษา PHP.....	14
2.3 แสดงตัวดำเนินการทางคณิตศาสตร์.....	14
2.4 แสดงตัวดำเนินการทางด้านเพิ่มลดค่า.....	14
2.5 แสดงตัวดำเนินการทางด้านตรรกศาสตร์.....	15
2.6 แสดงตัวดำเนินการด้านการเปรียบเทียบ.....	15
2.7 แสดงตัวเลขชนิดต่างๆ และขนาดข้อมูล.....	21
2.8 แสดงสัญลักษณ์ Flowchart.....	30
2.9 เปรียบเทียบสัญลักษณ์ที่ใช้ใน ER diagram.....	36

## สารบัญภาพ

ภาพที่	หน้า
2.1 การประมวลผลไปให้เครื่อง Client.....	12
2.2 โครงสร้างข้อมูล.....	20
2.3 องค์ประกอบระบบฐานข้อมูล.....	22
2.4 DBMS architecture.....	24
2.5 องค์ประกอบของระบบจัดการฐานข้อมูล.....	25
2.6 Software DBMS.....	26
2.7 โครงสร้างแบบลำดับ.....	27
2.8 การทำงานของโครงสร้างแบบมีทางเลือกif...then...else.....	28
2.9 การทำงานของโครงสร้างแบบมีทางเลือกcase.....	28
2.10 การทำงานของการทำซ้ำแบบ do while.....	29
2.11 การทำงานของการทำซ้ำแบบ do until.....	29
2.12 สัญลักษณ์ Use case.....	31
2.13 สัญลักษณ์ Actor.....	31
2.14 สัญลักษณ์ System Boundary.....	31
2.15 สัญลักษณ์ Connection.....	32
2.16 สัญลักษณ์ Extend Relationship.....	32
2.17 Use Case Diagram ที่มีความสัมพันธ์แบบ Extend Relationship.....	33
2.18 สัญลักษณ์ Include Relationship.....	33
2.19 Use Case Diagram ที่มี Included Relationship.....	34
2.20 ความสัมพันธ์ระหว่าง Use Case แบบ Generalization / Specialization.....	35
2.21 ความสัมพันธ์ระหว่าง Use Case แบบ Generalization / Specialization.....	35
2.22 แสดงความสัมพันธ์แบบ Generalization / Specialization ระหว่าง Actor.....	35
2.23 ตัวอย่าง Regular Entity และสัญลักษณ์ Regular Entity.....	37
2.24 ตัวอย่าง Weak Entity และสัญลักษณ์ Weak Entity.....	37
2.25 ตัวอย่าง Composite Entity และสัญลักษณ์ Composite Entity.....	37

## สารบัญภาพ(ต่อ)

ภาพที่	หน้า
2.26 ตัวอย่างชนิดของ Attribute .....	38
2.27 แสดงความสัมพันธ์แบบ 1:1 .....	39
2.28 แสดงความสัมพันธ์แบบ 1:M.....	39
2.29 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม.....	39
3.1 โปรแกรม XAMPP.....	41
3.2 โปรแกรม phpMyAdmin .....	41
3.3 โปรแกรม Sublime text 2.....	42
3.4 Flowchart แสดงการทำงานของเว็บไซต์ .....	42
3.5 Use case แสดงการทำงานของเว็บไซต์ .....	43
3.6 ER diagram แสดงการทำงานของเว็บไซต์ .....	43
3.7 โครงสร้างตาราง user.....	44
3.8 แสดงค่าของตาราง user.....	44
3.9 โค้ดการเชื่อม database กับ หน้าเว็บไซต์.....	45
4.1 sign in page.....	46
4.2 register page.....	46
4.3 แสดง movement.....	47
4.4 แสดงการค้นหาอุปกรณ์.....	47
4.5 แสดงการ import excel file ในหน้า serial.....	48
4.6 แสดงการเพิ่มสมาชิกใหม่.....	48
4.7 แสดงการแก้ไขข้อมูลของ implementer.....	49
4.8 แสดงการตรวจสอบข้อมูลของ ScoA&D.....	49
4.9 แสดง Network monitor ไม่สามารถแก้ไขข้อมูลได้ .....	50
4.10 แสดงปุ่ม change password .....	50
4.11 แสดงการ change password .....	50

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญ

ในยุคปัจจุบันที่โลกขับเคลื่อนด้วยข้อมูลดิจิทัลผ่านทางเครือข่ายเน็ตเวิร์ค และมีแนวโน้มที่ข้อมูลเหล่านั้นจะกระจายสู่สังคมอย่างแพร่หลายมากยิ่งขึ้นเมื่อมีผู้ใช้งานอินเทอร์เน็ตเพิ่มขึ้นทุกวัน และเนื่องจากแนวโน้มของระบบอินเทอร์เน็ตซึ่งเติบโตอย่างต่อเนื่องนี้ ประชากรจำนวนมากรับฟังข่าวสารจากระบบอินเทอร์เน็ตกันมากขึ้น ไม่ว่าจะเป็นองค์กรไม่แสวงหาผลประโยชน์ องค์กรภาครัฐ หน่วยงานต่าง ๆ นิติบุคคล หรือกระทั่งบุคคลธรรมดา หลากอาชีพหลายสาขาจึงให้ความสำคัญกับการสร้างสื่อทางอินเทอร์เน็ต การสร้างเว็บไซต์หรือสนับสนุนให้มีการรับทำเว็บไซต์เป็นของตนเองเพื่อประชาสัมพันธ์ ประสานงาน เก็บรวบรวมข้อมูลหรือกิจกรรมต่าง ๆ โดยเฉพาะในวงการธุรกิจ ปัจจุบันเว็บไซต์เปรียบเหมือนสิ่งหนึ่งที่ไม่อาจขาดได้ไปเสียแล้ว ไม่ว่าจะบริษัทใหญ่หรือเล็กต่างก็ทำเว็บไซต์เป็นของตนเองทั้งนั้น เพราะเว็บไซต์คือช่องทางสำคัญที่จะทำให้บริษัทมีโอกาสได้นำเสนอสินค้าของตนให้ผู้คนจำนวนมากได้เห็น ชิมชั๊ป และรับทราบถึงสินค้าหรือบริการของบริษัทตนเอง

เนื่องจากในอดีตได้มีการเก็บรวบรวมข้อมูลลงในโปรแกรม Microsoft Excel ซึ่งมีข้อมูลจำนวนมาก ทำให้ใช้เวลานานในการค้นหาข้อมูล และข้อมูลที่ได้อาจเป็นข้อมูลที่ไม่ถูกต้อง เนื่องจากข้อมูลที่ได้นั้นยังไม่ได้มีการอัปเดต เพราะเมื่อมีการเคลื่อนย้ายอุปกรณ์ในคลังสินค้าแล้วไม่ได้มีการแก้ไขข้อมูลลงในไฟล์รวม ทำให้ข้อมูลที่ได้เกิดความผิดพลาดได้

ดังนั้นการทำเว็บไซต์เก็บสินค้าคงคลังซึ่งเป็นเว็บไซต์ที่เก็บรวบรวมตำแหน่งและข้อมูลของอุปกรณ์ทุกชนิดที่อยู่ภายในองค์กร เพื่อเป็นประโยชน์ต่อการค้นหา หรือเรียกกลับคืนเพื่อซ่อมบำรุง สามารถเข้าถึงข้อมูลได้ตลอดเวลา เพิ่มความสะดวกสบายในการทำงาน ช่วยลดเวลาในการทำงาน และยังช่วยเพิ่มประสิทธิภาพในการทำงานได้อีกด้วย

#### 1.2 วัตถุประสงค์ของโครงการ

เว็บไซต์เก็บข้อมูลสินค้าคงคลังนี้จัดทำขึ้นเพื่อความสะดวกสบายในการตรวจสอบจำนวนสินค้าและตำแหน่งของสินค้าให้ถูกต้องแม่นยำมากขึ้น และเพื่อเพิ่มประสิทธิภาพในการทำงาน โดยที่ทุกคนในองค์กรสามารถเข้าถึงข้อมูลได้ตลอดเวลา ทำให้มีความรวดเร็วในการค้นหาและตรวจสอบอุปกรณ์

นอกจากนี้ยังช่วยลดปัญหาข้อมูลไม่อัปเดต ซึ่งในอดีตที่เก็บข้อมูลลงในโปรแกรม Microsoft Excel นั้นเกิดปัญหาคือเมื่อมีการเคลื่อนย้ายหรือเพิ่มอุปกรณ์ลงในคลังสินค้าแล้ว ไม่ได้แก้ไขข้อมูลลงในไฟล์รวม ทำให้เกิดความผิดพลาดของข้อมูลได้ จึงเกิดเป็นระบบเว็บไซต์ซึ่งง่ายต่อการแก้ไขข้อมูล โดยการกดแก้ไขเพียงครั้งเดียวข้อมูลก็จะอัปเดตไปในทุกๆ ฐานข้อมูล

สามารถเข้าถึงข้อมูลได้ง่าย เนื่องด้วยการเชื่อมต่อทางอินเทอร์เน็ตเป็นสิ่งที่สามารถเชื่อมทุกคนจาก ทุกมุมโลกให้สามารถติดต่อสื่อสารกันได้ผ่านเครือข่ายเน็ตเวิร์ก การมีเว็บไซต์เพื่อเก็บสินค้าคงคลังจึงเป็นการเพิ่มความสะดวกรบายให้กับผู้ใช้ เพราะไม่จำเป็นต้องเข้าไปในคลังสินค้าเองให้เสียเวลา

### 1.3 ขอบเขตของโครงการ

ขอบเขตการทำงานในการขนย้ายหรือเพิ่มอุปกรณ์ในคลังสินค้าจะมีลักษณะ คือสร้างเว็บไซต์สำหรับเก็บข้อมูลอุปกรณ์ทั้งหมดภายในคลังสินค้า แบ่งระบบงานต่างๆ ดังนี้

#### 1.3.1 ขอบเขตด้านเนื้อหา

เกี่ยวกับข้อมูลของอุปกรณ์ทุกชนิดที่อยู่ในคลังสินค้า เช่น serial number, model, brand, location name, ip address, change number เป็นต้น

#### 1.3.2 โปรแกรมที่ใช้ในการดำเนินงาน

- โปรแกรม XAMPP
- โปรแกรม Phpmyadmin
- โปรแกรม Sublime
- โปรแกรม Web Browser

### 1.4 วิธีการดำเนินการ

- ศึกษาหาความรู้เกี่ยวกับข้อมูลพื้นฐานทั่วไปในการสร้างเว็บไซต์
- ศึกษาหาข้อมูลเกี่ยวกับการสร้างระบบฐานข้อมูล
- ออกแบบระบบฐานข้อมูล
- ติดตั้งโปรแกรมในการสร้างเว็บไซต์ (xampp, phpmyadmin, sublime)
- ศึกษาและเรียนรู้โปรแกรมและภาษาที่ใช้ (SQL, PHP, HTML, Javascript)
- สร้างระบบฐานข้อมูลตามที่ได้ออกแบบไว้
- เชื่อมระบบฐานข้อมูลกับหน้าเว็บ
- ทำระบบ login

- ทดสอบการใช้งานของเว็บไซต์
- แก้ไขและปรับปรุงให้ใช้งานได้ดีขึ้น

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- เพิ่มความสะดวกสบายในการค้นหาข้อมูล
- ช่วยลดเวลาในการทำงาน ไม่ต้องเสียเวลาเข้าไปตรวจเช็คข้อมูลในคลังสินค้าจริง
- ข้อมูลที่ได้มีความถูกต้องแม่นยำ ทำให้ช่วยเพิ่มประสิทธิภาพในการทำงานมากขึ้น

## บทที่ 2

### แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีภาษาที่ใช้ในการสร้างเว็บไซต์

##### 2.1.1 SQL

SQL ย่อมาจาก Structured Query Language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) และเป็นระบบเปิด (open system) หมายถึงเราสามารถใส่คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่านระบบฐานข้อมูลที่แตกต่างกันจะได้ผลลัพธ์ที่เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูลชนิดใดก็ได้โดยไม่ติดขัดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง และเป็นภาษาที่ได้รับความนิยมเพราะเข้าใจง่ายไม่ซับซ้อน ภาษา SQL สามารถแบ่งออกเป็น 2 ส่วน

- 1) ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language: DDL) เป็นภาษาที่มีไว้สำหรับจัดการฐานข้อมูลโดยเฉพาะได้แก่ สร้างฐานข้อมูล, แก้ไข หรือลบฐานข้อมูล โดยในภาษา DDL นั้นประกอบไปด้วยคำสั่ง ดังนี้
  - CREATE คือคำสั่งสำหรับการสร้างนิยาม หรือสร้างฐานข้อมูล สามารถใช้รูปแบบคำสั่งได้ดังนี้

```
CREATE TABLE <ชื่อตาราง>
(
  <ชื่อ Field><ชนิดข้อมูล (ขนาดข้อมูล)>,
  <ชื่อ Field><ชนิดข้อมูล (ขนาดข้อมูล)>
)
```

เช่น

```
CREATE TABLE STUDENT
(ID INT(5) Not NULL,
NAME CHAR(10) ,
AGE integer)
```

- ALTER คือคำสั่งในการแก้ไขโครงสร้างของตารางข้อมูล สามารถใช้รูปแบบคำสั่งได้ดังนี้

```
ALTER TABLE <ชื่อตารางที่ตั้งขึ้น>
<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมน์ ประเภทข้อมูล>);
```

เช่น

```
ALTER TABLE STUDENT
ADD (LASTNAME Char(10));
```

- DROP คือคำสั่งสำหรับลบตารางข้อมูลสามารถใช้รูปแบบคำสั่งได้ดังนี้

```
DROP TABLE <ชื่อตารางข้อมูล>
```

เช่น

```
DROP TABLE STUDENT
```

### ดัชนี (INDEX)

INDEX มีความสำคัญมากต่อระบบฐานข้อมูลเชิงสัมพันธ์เนื่องจากระบบฐานข้อมูลแบบรีเลชันส์ (RDBMS) จะใช้ดัชนีในการค้นหาข้อมูลที่ต้องการได้อย่างรวดเร็ว โดยปกติแล้ว ถ้าไม่มีการประกาศดัชนีจะต้องค้นหาข้อมูลในตารางแบบเรียงตั้งแต่แถวที่หนึ่งจนถึงแถวสุดท้าย ทำให้การสร้างดัชนีช่วยเพิ่มความเร็วในการดึงข้อมูล และยังสามารถนำไปใช้ในการควบคุมคอลัมน์ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลที่ไม่ซ้ำกัน (Unique)

การสร้างดัชนีใช้คำสั่งดังนี้

```
CREATE [UNIQUE] INDEX <ชื่อตารางที่ตั้งขึ้น>
ON (<ชื่อตารางที่สร้างดัชนี> (<ชื่อคอลัมน์_1> [, <ชื่อคอลัมน์_2>] ...);
```

การลบดัชนีใช้คำสั่งดังนี้

```
DROP INDEX <ชื่อดัชนี>
```

2) ภาษาที่ใช้สำหรับการจัดการข้อมูล (Data Manipulation Language: DML) คือภาษาสำหรับจัดการข้อมูลที่จัดเก็บอยู่ในตารางข้อมูล ซึ่งในกลุ่มภาษา DML นั้นจะครอบคลุมการจัดการข้อมูลทั้งหมด เช่น การเพิ่ม, แก้ไข, ค้นหา และลบข้อมูล โดยคำสั่งต่าง ๆ มีดังนี้

- SELECT คือคำสั่งสำหรับค้นหาข้อมูลที่จัดเก็บอยู่ในตารางข้อมูล โดยมีรูปแบบคำสั่งทั่วไปคือ

```
SELECT <ชื่อคอลัมน์>
FROM <ชื่อตาราง>
WHERE <ชื่อคอลัมน์>
```

ตารางที่ 2.1 แสดงคำสั่ง select

คำสั่ง	หน้าที่	จำเป็นต้องมี(ใช่/ไม่)
SELECT	แสดงข้อมูลที่สนใจ	ใช่
FROM	ตารางที่ต้องการหาข้อมูล	ใช่
WHERE	กรองแถวข้อมูลที่ต้องการจากข้อมูลในตารางที่ระบุไว้ในคีย์เวิร์ด FROM	ไม่
ORDER BY	ระบุวิธีการเรียงลำดับผลลัพธ์	ไม่
GROUP BY	ใช้จัดกลุ่มแถวข้อมูลที่ได้ถูกรองมาแล้วจากคีย์เวิร์ด WHERE	ไม่
HAVING	ระบุเงื่อนไข	ไม่

ตัวอย่างเช่น

ตาราง Store\_Information

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

คำสั่ง : SELECT store\_name, SUM(Sales) FROM Store\_Information GROUP BY  
store\_name  
ผลลัพธ์ที่ได้

store_name	SUM(Sales)
Los Angeles	\$1800
San Diego	\$250
Boston	\$700

- INSERT คือคำสั่งสำหรับการเพิ่มข้อมูลลงไปในตารางข้อมูล โดยมีรูปแบบคำสั่ง  
ดังนี้

แบบที่ 1 เป็นการเพิ่มข้อมูลโดยไม่กำหนดคอลัมน์คือ เมื่อเพิ่มข้อมูลต้อง  
เพิ่มให้ครบทุกคอลัมน์ ดัง Syntax ด้านล่าง

```
INSERT INTO <ชื่อตาราง>
VALUES (ค่าคอลัมน์ที่1, ค่าคอลัมน์2, ค่าคอลัมน์3,...)
```

แบบที่ 2 เป็นการเพิ่มข้อมูลแบบกำหนดชื่อคอลัมน์โดย เวลาเพิ่มต้องบอก  
ชื่อคอลัมน์ที่ต้องการเพิ่ม ไม่จำเป็นต้องเพิ่มทุกคอลัมน์ ดัง Syntax ด้านล่าง

```
INSERT INTO <ชื่อตาราง> (คอลัมน์ 1, คอลัมน์ 2, คอลัมน์ 3,...)
VALUES (ค่าคอลัมน์ที่1, ค่าคอลัมน์2, ค่าคอลัมน์3,...)
```

ตัวอย่างเช่น

คำสั่ง : INSERT INTO country (CountryCode, CountryName) VALUES ('CH', 'Chaina')

ผลลัพธ์ที่ได้

CountryCode	CountryName
TH	Thailand
EN	England
US	United stated
CH	Chaina

- UPDATE คือคำสั่งสำหรับการปรับปรุง หรือแก้ไขข้อมูลในตารางข้อมูล โดยมีรูปแบบคำสั่งดังนี้

```
UPDATE <ชื่อตาราง >
SET <ค่าที่ต้องการ>
WHERE <เงื่อนไข>
```

ตัวอย่างเช่น

คำสั่ง : UPDATE country SET CountryCode = 'JP',CountryName='Japan' WHERE  
CountryCode = 'CH'

ผลลัพธ์ที่ได้

CountryCode	CountryName
TH	Thailand
EN	England
US	United stated
JP	Japan

- DELETE คือคำสั่งในการลบข้อมูลในตารางข้อมูล โดยมีรูปแบบคำสั่งดังนี้

```
DELETE FROM<ชื่อตาราง>
WHERE เงื่อนไข
```

ตัวอย่างเช่น

คำสั่ง : DELETE FROM country WHERE CountryCode = 'JP'

ผลลัพธ์ที่ได้

CountryCode	CountryName
TH	Thailand
EN	England
US	United stated

3) ภาษาควบคุมข้อมูล (Data Control Language : DCL) เป็นชุดคำสั่งที่ใช้สำหรับควบคุมสิทธิ์ของผู้ใช้ในการใช้ข้อมูล รวมทั้งส่วนที่ใช้ควบคุมการใช้งานฐานข้อมูลจากผู้ใช้หลายๆ คนพร้อมกัน มี 2 คำสั่ง ได้แก่

- GRANT เป็นคำสั่งเพื่อกำหนดสิทธิของการเข้าถึงข้อมูลในฐานข้อมูลว่าผู้ใช้แต่ละคน มีสิทธิกระทำการใด ๆ กับข้อมูลเช่น เพิ่ม, ลบ, แก้ไขข้อมูลในตารางใดบ้างหรือกำหนดให้สามารถดูข้อมูลได้เพียงอย่างเดียวเท่านั้นโดยมีรูปแบบคำสั่งดังนี้

```
GRANT <privilege>
ON <db_name.object>
TO <user_name>
[IDENTIFIED BY ' password ' ]
[WITH GRANT OPTION]
```

- GRANT คือ คำสั่งที่ให้สิทธิในการใช้งานข้อมูล
- privileges คือ สิทธิที่ต้องการอนุญาตให้ใช้งานซึ่งแบ่งออกเป็นหลายระดับ
- db\_name.object คือชื่อตารางหรือชื่อวิว แต่ถ้าหากต้องการกำหนดสิทธินั้นๆ ในทุกตารางที่มีในฐานข้อมูลที่กำหนด ให้ใช้ “db\_name.\*” หรือถ้าต้องการให้เข้าถึงข้อมูลในทุกๆฐานข้อมูลและทุกตารางให้ใช้ “ \*.\* ”
- user\_name คือ ชื่อผู้ใช้ที่ต้องการให้สิทธินั้น

ตัวอย่างเช่น ต้องการกำหนดสิทธิในการเข้าถึงข้อมูลในฐานข้อมูล Registration ให้กับอาจารย์วิทย์มานันท์สวงศ์โดยอาจารย์วิทย์สามารถเรียกดูข้อมูล และปรับปรุงข้อมูลในตารางต่างๆ ได้ใช้คำสั่งดังนี้

```
GRANT select, update
ON registration.*
TO Vittaya@localhost
IDENTIFIED BY “vit1234”;
```

- REVOKE คือคำสั่งที่มีไว้สำหรับการยกเลิกสิทธิในการเข้าถึงข้อมูลของบุคคลต่างๆ โดยมีรูปแบบคำสั่งดังนี้

```
REVOKE <SELECT, INSERT, UPDATE, DELETE>
ON <table name>
FROM <user name>;
```

ตัวอย่างเช่น

```
REVOKE INSERT, DELETE ON CUSTOMERSTAB FROM Wichai,Nattapol;
```

ผลจากคำสั่งนี้ Wichai และ Nattapol จะไม่สามารถเพิ่มเติมข้อมูลหรือลบข้อมูลในตารางลูกค้า (CUSTOMERSTAB) ได้

### 2.1.2 PHP

PHP เกิดมาในปี 1994 โดย Rasmus Lerdorf โปรแกรมเมอร์ชาวสหรัฐอเมริกาได้คิดค้นสร้างเครื่องมือที่ใช้ในการพัฒนาเว็บส่วนตัวของเขา โดยใช้ข้อดีของภาษาซี และ Perl เรียกว่า Personal Home Page และได้สร้างส่วนติดต่อฐานข้อมูลชื่อว่า Form Interpreter (FI) รวมทั้งสองส่วน เรียกว่า PHP/FI ซึ่งเป็นจุดเริ่มต้นของ PHP

PHP เป็นภาษาสคริปต์ที่ประมวลผลที่ฝั่งเซิร์ฟเวอร์ แล้วส่งผลลัพธ์ไปแสดงผลที่ฝั่งไคลเอ็นต์(Client) ผ่านบราวเซอร์เช่นเดียวกับ CGI และ ASP ต่อมาผู้ใช้งานเพิ่มมากขึ้นจึงมีการร้องขอให้มีการพัฒนาประสิทธิภาพของ PHP/FI ให้สูงขึ้น Rasmus Lerdorf ก็ได้ผู้ที่มาช่วยพัฒนาอีก 2 คนคือ Zeev Surask และ Andi Gutmans ชาวอิสราเอล ซึ่งปรับปรุงโค้ดของ Lerdorf ใหม่โดยใช้ C++ ต่อมาก็มีเพิ่มมาอีก 3 คน คือ Stig Bakken รับผิดชอบความสามารถในการติดต่อ Oracle, Shane Caraveo รับผิดชอบดูแล PHP บน Windows9x/NT, และ Jim Winastead รับผิดชอบการตรวจ ความบกพร่องต่างๆ และได้เปลี่ยนชื่อเป็น Professional Home Page

PHP เป็นภาษาจำพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่าสคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริปต์เช่น JavaScript, Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมาเพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้น PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting

language เป็นเครื่องมือที่สำคัญชนิดหนึ่ง ที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น PHP ได้รับการพัฒนาขึ้นมาเพื่อแทนที่ SSI รูปแบบเดิมๆ โดยให้มีความสามารถ และมีส่วนเชื่อมต่อกับเครื่องมือชนิดอื่นมากขึ้น เช่น ติดต่อกับคลังข้อมูล (database) เป็นต้น

PHP ได้รับการเผยแพร่เป็นครั้งแรกในปี ค.ศ. 1994 จากนั้นก็มีการพัฒนาต่อมาตามลำดับเป็นเวอร์ชัน 1 ในปี 1995 เวอร์ชัน 2 (ตอนนั้นใช้ชื่อว่า PHP/FI) ในช่วงระหว่าง 1995 ถึง 1997 และเวอร์ชัน 3 ช่วง 1997 ถึง 1999 จนถึงเวอร์ชัน 4 ในปัจจุบัน

PHP เป็นผลงานที่เติบโตมาจากกลุ่มของนักพัฒนาในเชิงเปิดเผยรหัสต้นฉบับหรือ Open Source ดังนั้น PHP จึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลายโดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับ Apache Webserver ระบบปฏิบัติการอย่างเช่น Linux หรือ FreeBSD เป็นต้น ในปัจจุบัน PHP สามารถใช้ร่วมกับ Web Server หลายๆตัวบนระบบปฏิบัติการอย่างเช่น Windows 95/98/NT เป็นต้น

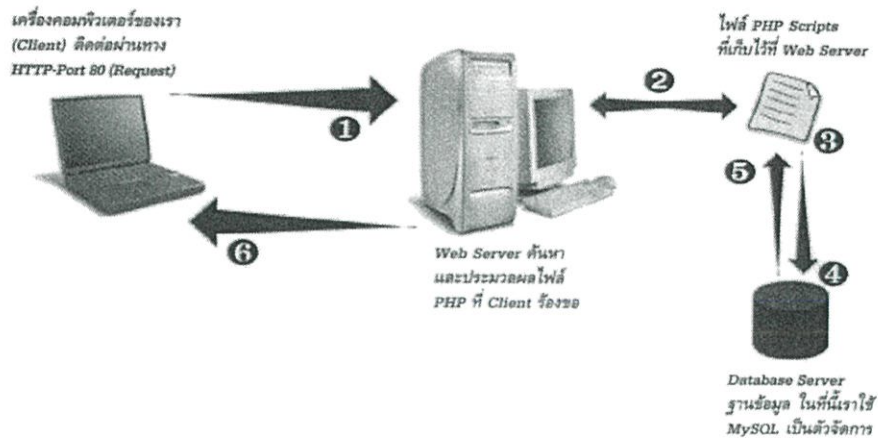
ภาษาสคริปต์ (Scripting Language) โสมเพจจำเป็นต้องหาอะไรแปลกใหม่ มาลงในเว็บไซต์เสมอ หากมีเพียงแค่ HTML เว็บไซต์จะไม่มี การตอบโต้อะไรกลับมา ภาษาสคริปต์สำหรับเว็บไซต์นั้นเขียนได้หลายภาษาและมีรูปแบบการเขียนอยู่ 2 แบบคือ

1) Client-Side Scripting เป็นการเขียนโปรแกรมภาษาสคริปต์ ให้ทำงานบน Web Browser โดยเขียนโปรแกรมแทรกหรือแฝง (Embed) เข้าไปเป็นส่วนหนึ่งของเอกสาร HTML โปรแกรมภาษาสคริปต์ประเภทนี้ได้แก่ JavaScript, VBScript

2) Server-Side Scripting เป็นการเขียนโปรแกรมภาษาสคริปต์ ให้ทำงานบน Web Server โดย Web Browser จะเป็นเพียงแค่ตัวที่แสดงผลการทำงานเท่านั้น โปรแกรมที่ทำงานบน Web Server เหล่านี้เราเรียกว่า CGI Script ซึ่งสามารถเขียนได้หลายภาษา เช่น Perl, C, Pascal, VB เป็นต้น และโปรแกรมภาษาสคริปต์ประเภทนี้ได้แก่ JSP, ASP, PHP

### หลักการการทำงานของ PHP

ฝั่ง Client จะทำการร้องขอหรือเรียกใช้งานไฟล์ PHP ที่เก็บในเครื่อง Server ส่วนฝั่ง Server จะทำการประมวลผลไฟล์ PHP ตามที่ Client ทำการร้องขอและทำการประมวลผลไฟล์ PHP จากนั้นติดต่อส่งข้อมูลกลับมายัง Client ตามภาพที่ 2.1



ภาพที่ 2.1 การประมวลผลไปให้เครื่อง Client

### โครงสร้างพื้นฐานของภาษาPHP

การเขียนโค้ดโปรแกรมจำเป็นต้องมีสัญลักษณ์ที่บ่งบอกถึงขอบเขตของPHP เพื่อที่จะแยกโค้ด PHP ออกจากโค้ด HTML ได้อย่างชัดเจน ซึ่งสัญลักษณ์ต่าง ๆ ที่เราสามารถนำมาใช้แยกโค้ด PHP ได้ดังนี้

#### แบบที่ 1

```
<?
echo ("PHP Hello World\n");
?>
```

#### แบบที่ 2

```
<?php
echo ("PHP Hello World\n");
?>
```

#### แบบที่ 3

```
<script language="php">
echo ("PHP Hello World\n");
</script>
```

## แบบที่ 4

```
<% echo ("PHP Hello World\n");>
<%= $ตัวแปร; %>
```

แบบที่ 4 นี้จะใช้ได้กับ PHP เวอร์ชัน 3.0.4 ขึ้นไป และต้องมีการแก้ไขไฟล์ PHP.ini ด้วย ซึ่งจะอยู่ใน ส่วนของไฟล์ php.ini ดังนี้

```
.....
; Language Options ;
.....
asp_tag = On ; allow ASP=style <% %> tags
```

ถ้าต้องการใช้งานในแบบที่ 4 ให้แก้ไขไฟล์ php.ini ซึ่งจะอยู่ในส่วนของ Language Options โดยเปลี่ยนที่บรรทัดของ asp tags เปลี่ยนจาก off ให้เป็น on ก็สามารถที่จะใช้งานได้

## การกำหนดตัวแปรและชนิดของข้อมูล

### กฎการตั้งชื่อตัวแปร

- เริ่มต้นด้วย \$
- ตามด้วย A-Z หรือ a-z หรือ 0-9 เช่น \$myvalue; \$MyValue;
- ไม่ควรเว้นวรรค แต่ใช้ \_ แทน เช่น \$my\_value; \$My\_Value;
- Case Sensitive ตัวพิมพ์ใหญ่/เล็กถือเป็นคนละตัว
- ไม่ตั้งชื่อซ้ำคำสั่ง เช่น \$\_POST; \$\_SESSION; \$\_GET; \$\_SERVER; ฯลฯ

### ตัวอย่างเช่น

```
$a = 1234; //ตัวอย่างที่1
```

```
$b_boy = -123; //ตัวอย่างที่2
```

ประโยชน์ของตัวแปรคือการเก็บค่าข้อมูลชั่วคราวเพื่อใช้ในการประมวลผล ซึ่งค่าข้อมูลที่ตัวแปรสามารถเก็บได้จะมีทั้งตัวอักษร ตัวเลขดังตารางต่อไปนี้

ตารางที่ 2.2 แสดงตัวแปรในภาษา PHP

ค่าข้อมูล	คำอธิบาย
Integers	เก็บข้อมูลตัวเลขจำนวนเต็ม เช่น 452, -45
Float	เก็บข้อมูลตัวเลขที่มีทศนิยม เช่น 405.23
Strings	เก็บข้อมูลตัวอักษร ข้อความ เช่น "Hello", "World"
Arrays	เก็บข้อมูลเป็นชุด กลุ่มสมาชิก
Objects	เก็บข้อมูลในลักษณะของการเรียกใช้ เป็น class object หรือ function
Type juggling	เก็บข้อมูลในลักษณะที่ขึ้นอยู่กับ Operator

ตารางที่ 2.3 แสดงตัวดำเนินการทางคณิตศาสตร์

เครื่องหมาย	ความหมาย	ตัวอย่าง
+	Addition แทนการบวก	$\$x+\$y$
-	Subtraction แทนการลบ	$\$x-\$y$
*	Multiplication แทนการคูณ	$\$x*\$y$
/	Division แทนการหาร	$\$x/\$y$
%	Modules แทนการหารเอาเศษ	$\$x\%\$y$

ตารางที่ 2.4 แสดงตัวดำเนินการทางด้านเพิ่มลดค่า

เครื่องหมาย	ความหมาย	ชื่อตัวดำเนินการ
$++\$a$	เพิ่มค่าที่ละ 1 ก่อนแล้วให้ค่าตัวแปร	Pre-increment
$\$a++$	ให้ค่าตัวแปรก่อนแล้วค่อยเพิ่มที่ละ 1	Post-increment
$--\$a$	ลดค่าที่ละ 1 ก่อนแล้วให้ค่าตัวแปร	Pre-decrement
$\$a--$	ให้ค่าตัวแปรก่อนแล้วค่อยลดที่ละ 1	Post-decrement

ตารางที่ 2.5 แสดงตัวดำเนินการทางด้านตรรกศาสตร์

เครื่องหมาย	ความหมาย
!a	เป็นค่าจริงเมื่อ a เป็นเท็จ
a && b	เป็นค่าจริงเมื่อค่าของ a และ b เป็นจริงหรือเท็จเหมือนกัน
a    b	เป็นค่าจริงเมื่อค่าของ a และ b ไม่เป็นเท็จทั้งคู่
a and b	เหมือนกับ && แต่มีความสำคัญน้อยกว่า
a or b	เหมือนกับ    แต่มีความสำคัญน้อยกว่า

ตารางที่ 2.6 แสดงตัวดำเนินการด้านการเปรียบเทียบ

เครื่องหมาย	ความหมาย
a == b	ค่าของ a เท่ากับค่าของ b
a != b	ค่าของ a ไม่เท่ากับ b
a < b	ค่าของ a น้อยกว่า b
a > b	ค่าของ a มากกว่า b
a <= b	ค่าของ a น้อยกว่าหรือเท่ากับ b
a >= b	ค่าของ a มากกว่าหรือเท่ากับ b

## คำสั่งอื่นๆ

- `str_replace()` ใช้ในการแทนที่ คำ อักษร หรือข้อมูล ในประโยคมีรูปแบบคำสั่งดังนี้

```
str_replace(find,replace,string,count)
```

- find คือ ข้อมูลที่ทำการค้นหา
- replace คือ ข้อมูลที่จะนำไปแทนที่
- string คือ ประโยคหรือข้อความที่จะทำการแทนที่คำ
- count คือ ตำแหน่งหรือจำนวนของข้อมูลที่จะแทนที่ (จะใส่หรือไม่ใส่ก็ได้)

ตัวอย่างเช่น

```
<?php
```

```
$string = "Hello world";
```

```
echo "result : ".str_replace("world","john",$string);
```

```
?>
```

ผลลัพธ์ที่ได้: Hello john

- `explode()` ใช้สำหรับแบ่งแยกคำออกจากประโยคให้อยู่ในรูปแบบ เป็นคำๆ โดยใช้สัญลักษณ์ หรืออักษรเป็นตัวกำหนดจุดที่จะแยกคำออกมาในรูปแบบคำสั่งดังนี้

```
explode(separator,string,limit)
```

- separator คือ สัญลักษณ์ หรืออักษรเป็นตัวกำหนดจุดที่จะแยกคำ
- string คือ ข้อความที่จะแยกออก
- limit คือ กำหนดจำนวน index array ที่เก็บค่าคำที่แยกออกมา กำหนดหรือไม่ก็ได้

ตัวอย่างเช่น

```
<?php
```

```
$str = "สวัสดี เพื่อนๆ ทุกคน";
```

```
$result = explode(" ", $str);
```

```
print_r ($result);
```

```
echo "<br/>";
```

```
$result = explode(" ", $str, 1);
```

```
print_r ($result);
```

```
?>
```

ผลลัพธ์ที่ได้

```
Array ([0] => สวัสดี [1] => เพื่อนๆ [2] => ทุกคน)
```

```
Array ([0] => สวัสดี [1] => เพื่อนๆ ทุกคน)
```

- `substr()` เป็นคำสั่งตัดข้อความที่ต้องการออกมาจากประโยคมีรูปแบบคำสั่งดังนี้

```
substr (string,start,length)
```

- string = ประโยคเต็มๆที่ต้องการจะตัดออก
- start = ตำแหน่งตัวอักษรที่จะเริ่มตัด
- length = ความยาวของคำที่ต้องการตัดออกมา

ตัวอย่างเช่น

```
<?php
    $date = date("dmY");
    echo "วันที่ต้องการจะตัดออก = ".$date;
    echo "<br/>วัน = ".substr($date, 0, 2);
    echo "<br/>เดือน = ".substr($date, 2, 2);
    echo "<br/>ปี = ".substr($date, 4, 4);
?>
```

ผลลัพธ์ที่ได้

วันที่ต้องการจะตัดออก = 05082016

วัน = 05

เดือน = 08

ปี = 2016

## Comments

Comments คือข้อความที่เขียนไว้ในโปรแกรมเพื่ออธิบายโค้ด โดยใส่ไว้ในโปรแกรมที่ต้องการอธิบาย ซึ่งจะไม่มีผลกับการทำงานของโปรแกรมในระหว่างที่โปรแกรมทำงาน

การใส่ Comments โดยทั่วไปจะมี 2 แบบ

- // ใช้สำหรับ comment1 บรรทัด

เช่น

```
//echo $strAddress; โปรแกรมจะไม่สนใจและประมวลผลในส่วนนี้
```

- /\* PHP \*/ ใช้ Comment มากกว่า 1 บรรทัด

เช่น

```
/* echo $strAddress; โปรแกรมจะไม่สนใจและประมวลผลในส่วนนี้
echo $strEmail; */
```

นอกจากนี้ในภาษา PHP จะใช้ "." (dot) ในการเชื่อมต่อค่าให้เป็นประโยค ก่อนที่จะทำการแสดงผลออกมา เพื่อสื่อสารกับผู้ใช้งานโปรแกรม

ตัวอย่างเช่น

```
<?php
    $Name = "ด.ญ.วาสนา รักเรียน";
    $Class = "1/2";
    $Number = 1;
    echo $Name." อยู่ห้อง ป.".$Class." เลขที่ " .$Number;
?>
```

ผลลัพธ์ที่ได้

ด.ญ.วาสนา รักเรียน อยู่ห้อง ป.1/2 เลขที่ 1

## 2.2 ทฤษฎีระบบฐานข้อมูล (database)

### 2.2.1 ความหมายของระบบฐานข้อมูล

การจัดการข้อมูลด้วยระบบการจัดการฐานข้อมูลการจัดเก็บข้อมูลที่มีปริมาณมากๆ ในระบบคอมพิวเตอร์โดยการเก็บไว้ในแฟ้มข้อมูลแต่ละแฟ้ม อาจเกิดปัญหาข้อมูลซ้ำซ้อนกัน เมื่อมีการแก้ไขข้อมูลที่มีอยู่หลายแฟ้มข้อมูล จนทำให้ข้อมูลมีความขัดแย้งกันเอง จึงได้มีการเปลี่ยนการจัดการจัดเก็บข้อมูลให้อยู่ในรูปของฐานข้อมูลแทน เพื่อความสะดวกในการบันทึกข้อมูล แก้ไขข้อมูล และค้นหาข้อมูล

ระบบการจัดการฐานข้อมูล (Data Base Management System: DBMS) หมายถึงซอฟต์แวร์ที่สร้างขึ้นเพื่อรวบรวมข้อมูลให้เป็นระบบเพื่อจะได้นำไปเก็บรักษา เรียกใช้หรือนำมาปรับปรุงให้ทันสมัยได้ง่าย ทั้งนี้จำเป็นต้องคำนึงถึงการรักษาความปลอดภัยของข้อมูลเป็นเรื่องสำคัญด้วยระบบการจัดการฐานข้อมูล คือ โปรแกรมที่ใช้เป็นเครื่องมือในการจัดการฐานข้อมูลซึ่งประกอบด้วยหน้าที่ต่างๆ ในการจัดการกับข้อมูลรวมทั้งภาษาที่ใช้ทำงานกับข้อมูลโดยมักจะใช้ภาษา SQL ในการติดต่อระหว่างกับผู้ใช้เพื่อให้สามารถกำหนดการสร้างการเรียกดูการบำรุงรักษาฐานข้อมูลรวมทั้งการจัดการควบคุมการเข้าถึงฐานข้อมูล ซึ่งถือเป็นการป้องกันความปลอดภัยในฐานข้อมูลเพื่อป้องกันไม่ให้ผู้ที่ไม่มีความสิทธิ์การใช้งานเข้ามาละเมิดข้อมูลในฐานข้อมูลที่เป็นศูนย์กลางได้นอกจากนี้ DBMS ยังมีหน้าที่ในการรักษาความมั่นคงและความปลอดภัยของข้อมูลการสำรองข้อมูลและการเรียกคืนข้อมูลในกรณีที่ข้อมูลเกิดความเสียหาย

สรุปได้ว่า ระบบการจัดการข้อมูลคือ โปรแกรมที่ทำหน้าที่ในการกำหนดลักษณะข้อมูลที่จะเก็บไว้ในฐานข้อมูลอำนวยความสะดวกในการบันทึกข้อมูลลงในฐานข้อมูล กำหนดผู้ที่ได้รับอนุญาตให้ใช้ฐานข้อมูลได้พร้อมกับการกำหนดด้วยว่าให้ใช้แบบใดเช่น ให้อ่านข้อมูลได้อย่างเดียวหรือให้แก้ไขข้อมูลได้ด้วย นอกจากนี้ยังอำนวยความสะดวกในการค้นหาข้อมูล และการแก้ไขปรับปรุงข้อมูล ทำให้ผู้ใช้สามารถเข้าถึงข้อมูลได้ง่าย สะดวกและมีประสิทธิภาพเสมือนเป็นตัวกลางระหว่างผู้ใช้กับฐานข้อมูลให้สามารถติดต่อกันได้

### 2.2.2 โครงสร้างของฐานข้อมูล

1) บิต (Bit) เป็นหน่วยข้อมูลที่มีขนาดเล็กที่สุด ซึ่งเป็นข้อมูลที่เครื่องคอมพิวเตอร์สามารถเข้าใจและนำไปใช้งานได้ ได้แก่ เลข 0 และ เลข 1

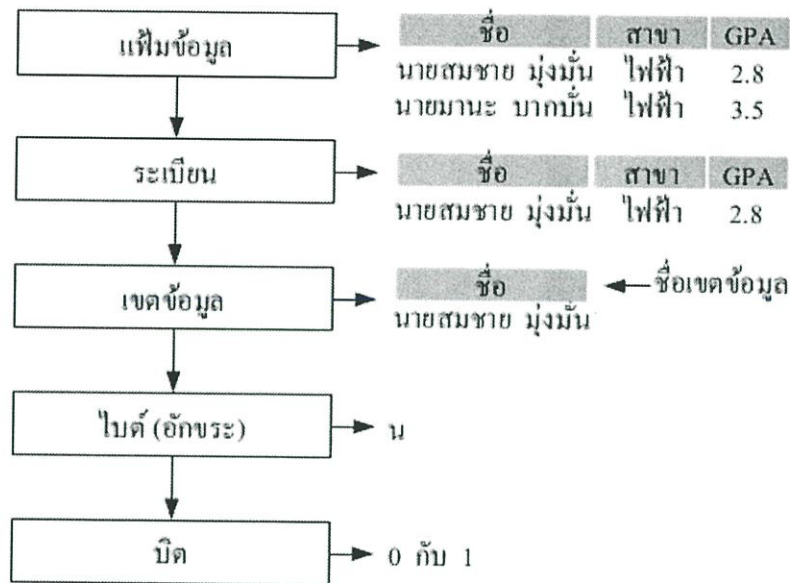
2) ไบต์ (Byte) หรือ อักขระ (Character) ได้แก่ ตัวเลข ตัวอักษร หรือ สัญลักษณ์พิเศษ 1 ตัว เช่น 0,1...9,A, B,...Z ซึ่ง 1 ไบต์ จะเท่ากับ 8 บิต หรือ ตัวอักขระ 1 ตัว

3) ฟิลด์ (Field) คือ อักขระ ตั้งแต่ 1 ตัวขึ้นไป รวมกันเป็น ฟิลด์ เช่น เลขประจำตัว ชื่อสกุล เป็นต้น

4) เรคคอร์ด (Record) คือ การนำเอาฟิลด์หลายฟิลด์และมีความสัมพันธ์มารวมกลุ่มกัน เช่น นักเรียนแต่ละคนจะมีข้อมูลที่เกี่ยวข้องกับ ชื่อ สกุล อายุ เพศ เกรดเฉลี่ย โดยข้อมูลในลักษณะนี้คือ 1 เรคคอร์ด

5) แฟ้มข้อมูล หรือ ไฟล์ (Files) คือ เรคคอร์ดหลายๆ เรคคอร์ดรวมกัน และเป็นเรื่องเดียวกันเช่น แฟ้มข้อมูลนักเรียนห้อง ม.1/1 จำนวน 50 คน ทุกคนจะมีข้อมูลเกี่ยวกับ ชื่อ สกุล เพศ อายุ เกรดเฉลี่ย ซึ่งข้อมูลทั้งหมดนี้ของนักเรียนจำนวน 50 คนนี้ เรียกว่า แฟ้มข้อมูล

6) ฐานข้อมูล (Database) คือ การเก็บรวบรวมไฟล์หรือแฟ้มข้อมูลหลายๆ ไฟล์ที่เกี่ยวข้องมารวมกัน



ภาพที่ 2.2 โครงสร้างข้อมูล

### 2.2.3 ชนิดของข้อมูลของเขตข้อมูลในตารางข้อมูล

ชนิดของข้อมูล (Data Type) แบ่งเป็นและมีความหมายดังนี้

1) Text เป็นข้อมูลชนิดตัวอักษร ขนาดความกว้าง 255 ตัว ซึ่งโปรแกรมจะกำหนดค่าตั้งต้น (Default) ไว้ 50 ตัว ใช้กำหนดให้ฟิลด์สำหรับเก็บข้อมูลเป็นอักขระ ได้แก่ ตัวอักษร ตัวเลข ช่องว่าง เครื่องหมายวรรคตอนหรือสัญลักษณ์อื่นๆ

2) Memo เป็นข้อมูลแบบข้อความใช้บันทึกรายละเอียดที่ไม่อาจกำหนดได้อย่างแน่นอน สามารถบันทึกข้อมูลได้ถึง 64,000 ตัว

3) Number ใช้กำหนดให้ฟิลด์ที่เก็บข้อมูลเป็นตัวเลขที่คำนวณได้ เช่น ราคาสินค้า หรือ จำนวนสิ่งของ โดยฟิลด์ที่กำหนดด้วยชนิดของข้อมูลชนิดนี้จะรับเฉพาะตัวเลขหรือจุดทศนิยมเท่านั้น ตัวเลขแบ่งออกเป็นตัวเลขต่างๆชนิด ซึ่งมีขนาดข้อมูลแตกต่างกัน ได้แก่ Byte Integer Long integer Single และ Double

การกำหนดขนาดฟิลด์ (Field Size) ให้ข้อมูลชนิด Number

ตารางที่ 2.7 แสดงตัวเลขชนิดต่างๆ และขนาดข้อมูล

ขนาดเขตข้อมูล	ค่าข้อมูลที่บันทึกได้	ตำแหน่งทศนิยม	ขนาดข้อมูล
Byte	0-255	-	1 Byte
Integer	-32,768 ถึง 32,767	-	2 Byte
Long Integer	-22,147,483,648 ถึง +2,147,483,648	-	4 Byte
Single	$-3.4 \times 10^{38}$ ถึง $+3.4 \times 10^{38}$	7 ตำแหน่ง	4 Byte
Double	$-1.797 \times 10^{308}$ ถึง $+1.797 \times 10^{308}$	15 ตำแหน่ง	8 Byte

ข้อมูลที่ Database จะแสดงออกมาจากฟิลด์ต่างๆ จะถูกกำหนดด้วยรูปแบบที่แน่นอน ซึ่งขึ้นอยู่กับชนิดของข้อมูลที่กำหนดให้ฟิลด์นั้นๆ เช่น ข้อมูลชนิด Number จะถูกจัดให้ชิดขอบฟิลด์ทางขวาเสมอ ในขณะที่ข้อมูลชนิด Text จะถูกจัดให้ชิดขอบซ้ายของฟิลด์เสมอ

4) Data / Time ใช้กำหนดให้ฟิลด์ที่ต้องการเก็บข้อมูลประเภทวันที่ เวลา

5) Currency ตัวเลขทางการเงิน ใช้กำหนดกับข้อมูลเกี่ยวกับสกุลเงินตราของประเทศต่างๆ

6) AutoNumber ตัวเลขที่ใช้ในการนับระยะเป็น เป็นข้อมูลตัวเลขเรียงลำดับที่โปรแกรมกำหนดให้เองและจะเป็นตัวเลขที่ไม่ซ้ำกัน ซึ่งถ้าผู้ใช้เลือกทำเป็น Primary Key จะมีรูปกุญแจนำหน้าชื่อฟิลด์

7) Yes/No เก็บค่าข้อมูลทางตรรกศาสตร์มี 2 ค่า ใช้กำหนดให้ฟิลด์ที่ต้องการเก็บ ข้อมูล โดยใช้ตัวอักษรตัวเดียวที่แสดงค่าความจริงเป็นใช่ (Yes, True) หรือไม่ใช่ (No, False) ซึ่งฟิลด์ที่กำหนดด้วยชนิดของข้อมูลนี้จะรับค่า Y หรือ y (Yes), N หรือ n (No), T หรือ t (True) และ F หรือ f (false) โดยความยาวฟิลด์นี้กำหนดไว้เพียง 1 ตัวอักษรเท่านั้น

8) OLE Object ใช้กำหนดให้กับฟิลด์ที่เก็บข้อมูลเป็นรูปภาพ

9) Hyperlink ชนิดข้อมูลสำหรับเก็บที่อยู่ของไฟล์ หรือเว็บไซต์ที่อยู่ในอินเทอร์เน็ต

#### 2.2.4 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยส่วนใหญ่แล้ว เป็นระบบที่มีการนำเอาคอมพิวเตอร์มาช่วยในกระบวนการจัดเก็บข้อมูล ค้นหาข้อมูล ประมวลผลข้อมูล เพื่อให้ได้สารสนเทศที่ต้องการแล้วนำไปใช้ในการปฏิบัติงานและบริหารงานของผู้บริหาร โดยอาศัยโปรแกรมเข้ามาช่วยจัดการข้อมูล ระบบฐานข้อมูลมีองค์ประกอบ 5 ประเภท คือ

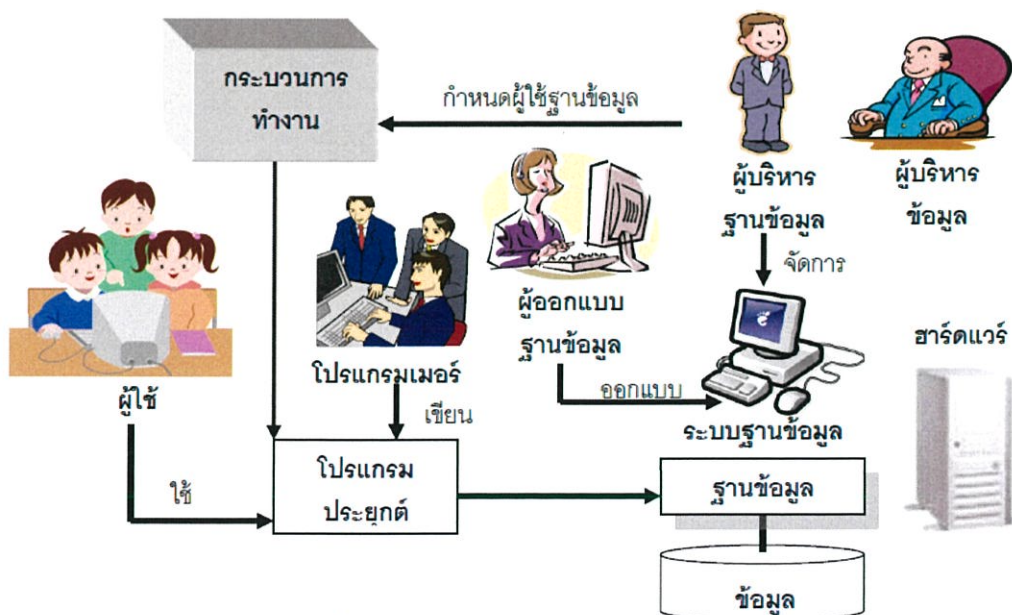
1) ฮาร์ดแวร์ (Hardware) คือ เครื่องคอมพิวเตอร์และจำนวนพื้นที่สำหรับเก็บข้อมูล ซึ่งมีหลายระดับการใช้งานให้เลือกใช้ขึ้นอยู่กับลักษณะงานขององค์กรและผลิตภัณฑ์ ระบบจัดการฐานข้อมูลต้องมีประสิทธิภาพในการจัดเก็บข้อมูลสูงทั้งด้านความเร็วและความจุข้อมูล ต้องมีการปรับปรุงอยู่เสมอ เพื่อให้การใช้งานมีประสิทธิภาพมากขึ้น

2) โปรแกรม (Program หรือ Software) ซึ่งมีหน้าที่ควบคุมดูแลการสร้างฐานข้อมูล การเรียกใช้ข้อมูล และ การจัดทำรายงาน เรียกว่า โปรแกรมระบบจัดการฐานข้อมูล (Database Management System: DBMS)

3) ข้อมูล (Data) ข้อมูลในฐานข้อมูลเป็นส่วนที่ถูกนำมาใช้งาน ถูกเก็บอยู่ในอุปกรณ์อิเล็กทรอนิกส์ โดยในมุมมองของผู้ใช้งานนั้นข้อมูลจะถูกเก็บไว้ในตารางต่างๆ ของฐานข้อมูล

4) บุคลากร (People) คือ ผู้ใช้งาน (User) พนักงานปฏิบัติการ (Operator) นักวิเคราะห์และออกแบบระบบ (System Analyst) ผู้เขียนโปรแกรมประยุกต์ใช้งาน (Programmer) และ ผู้บริหารฐานข้อมูล (Database Administrator: DBA)

5) ขั้นตอนการปฏิบัติงาน (Procedure) เป็นขั้นตอนและวิธีการต่าง ๆ ในการปฏิบัติงาน เพื่อการทำงานที่ถูกต้องและเป็นไปตามขั้นตอนที่ได้กำหนดไว้ จึงควรทำเอกสารที่ระบุขั้นตอนการทำงานของหน้าที่ต่าง ๆ ในระบบฐานข้อมูล ทั้งขั้นตอนปกติ และขั้นตอนในสภาวะที่ระบบเกิดปัญหา (Failure)



ภาพที่ 2.3 องค์ประกอบระบบฐานข้อมูล

### 2.2.5 ระบบจัดการฐานข้อมูล (DBMS)

Database Management System (DBMS) คือ ซอฟต์แวร์หรือโปรแกรมหรือชุดคำสั่งที่ทำหน้าที่บริหารและดูแลจัดการเกี่ยวกับระบบฐานข้อมูล โดยช่วยอำนวยความสะดวกให้แก่ผู้ใช้งานในหลายๆด้าน เช่น ช่วยการสร้างฐานข้อมูล การปรับปรุงแก้ไข การเข้าถึงข้อมูล การเรียกใช้ข้อมูล บริหารความถูกต้องของข้อมูล ควบคุมการแสดงผล และการจัดการเกี่ยวกับระบบแฟ้มข้อมูลภายในฐานข้อมูล ตลอดจนควบคุมดูแลความซ้ำซ้อนและความสัมพันธ์ของข้อมูลแต่ละชนิดในระบบฐานข้อมูล

DBMS ถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาด้าน Data Independence ที่ไม่มีในระบบแฟ้มข้อมูล ทำให้มีความเป็นอิสระจากทั้งส่วนของฮาร์ดแวร์ และข้อมูลภายในฐานข้อมูล กล่าวคือโปรแกรม DBMS นี้จะมีความทำงานที่ไม่ขึ้นอยู่กับรูปแบบ (Platform) ของตัวฮาร์ดแวร์ที่นำมาใช้กับระบบฐานข้อมูล รวมทั้งมีรูปแบบในการอ้างถึงข้อมูลที่ไม่ขึ้นอยู่กับโครงสร้างทางกายภาพของข้อมูลด้วยการใช้ Query Language ในการติดต่อกับข้อมูลในฐานข้อมูลส่งผลให้ผู้ใช้สามารถเรียกใช้ข้อมูลจากฐานข้อมูลได้โดยไม่จำเป็นต้องทราบถึงประเภทหรือขนาดของข้อมูลนั้น และยังสามารถกำหนดลำดับที่ของฟิลต์ในการแสดงผลได้โดยไม่ต้องคำนึงถึงลำดับที่จริงของฟิลต์นั้น

#### หน้าที่ของระบบจัดการฐานข้อมูล

- รับคำสั่งจากภายนอก แล้วแปลงคำสั่ง (Compile) ให้อยู่ในรูปแบบที่ฐานข้อมูลเข้าใจ
- นำคำสั่งที่ผ่านการ Compile แล้ว ไปสั่งให้ฐานข้อมูลทำงานตามที่ต้องการ ไม่ว่าจะเป็นการเรียกข้อมูล การปรับแต่งแก้ไข ลบข้อมูล หรือสร้างข้อมูลเพิ่มเติม
- รักษาความสัมพันธ์ (Relationship) ของข้อมูลภายในฐานข้อมูลให้มีความถูกต้องอยู่เสมอ
- เก็บรายละเอียดทุกชนิดที่เกี่ยวข้องกับข้อมูลภายในฐานข้อมูลไว้ในพจนานุกรมข้อมูล เพื่อให้โปรแกรมเมอร์หรือผู้บริหารระบบสามารถพัฒนาโปรแกรมสำหรับเชื่อมต่อหรือเรียกข้อมูลภายในไปใช้
- ควบคุมให้ฐานข้อมูลทำงานได้อย่างถูกต้องและมีประสิทธิภาพ
- รักษาความปลอดภัยและป้องกันความเสียหายที่จะเกิดขึ้นกับข้อมูลภายในฐานข้อมูล โดยจะตรวจสอบคำสั่งภายนอกที่จะเข้ามาจัดการฐานข้อมูล และปฏิเสธคำสั่งที่อาจสร้างความเสียหายกับระบบฐานข้อมูล
- ควบคุมการใช้งานข้อมูลแบบหลายคนพร้อมกัน เพื่อจัดลำดับการ Update ข้อมูล

- ทำหน้าที่สำรองข้อมูล (Backup) และกู้ข้อมูลเมื่อเกิดความเสียหายขึ้นกับฐานข้อมูล (Recovery)

- กำหนดสิทธิ์ของผู้ใช้ทุกคนที่เข้ามาใช้ฐานข้อมูล

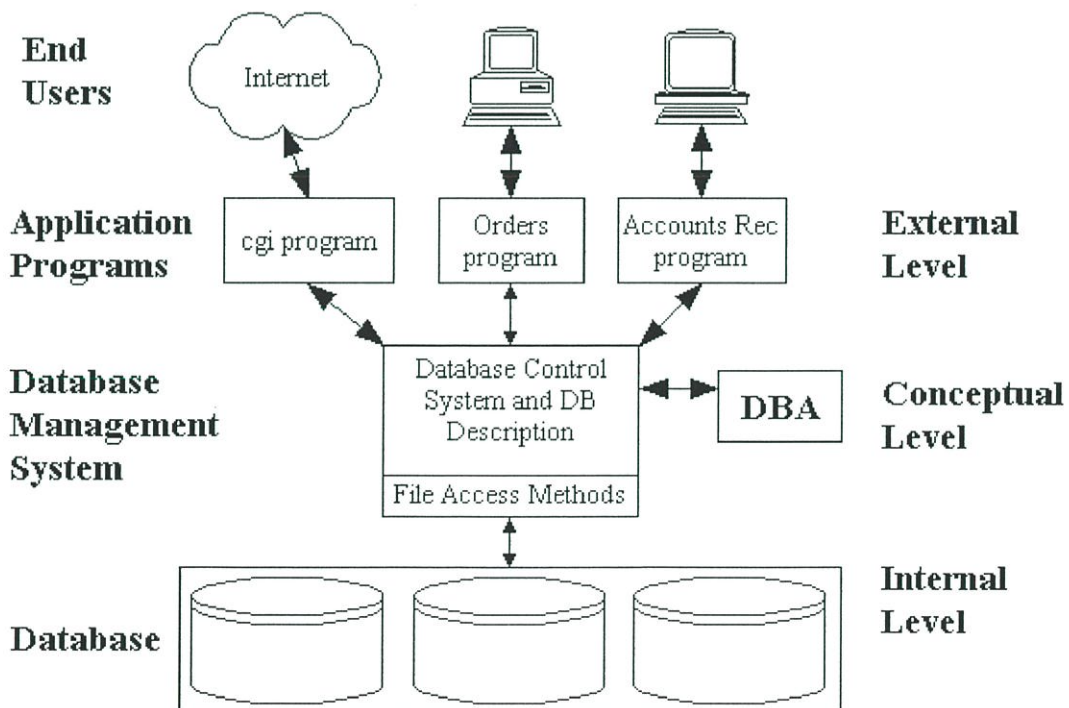
- สร้าง Input/Edit/Report/Query forms เพื่อการเรียกใช้และการเข้าถึงระบบฐานข้อมูล

### โครงสร้างการทำงานของระบบจัดการฐานข้อมูล

- ระดับที่ 1 ระดับภายใน (Internal Level) จะแสดงรายละเอียดของการจัดเก็บในตัวกลาง และวิธีการเรียกใช้ ฐานข้อมูล

- ระดับที่ 2 ระดับแนวคิด (Conceptual Level) แสดงโครงสร้างรวมของฐานข้อมูลให้กับผู้ใช้ทุกประเภท โดยโครงสร้างแนวคิดจะแสดงข้อมูลเกี่ยวกับข้อมูล (Metadata) เท่านั้น เช่น ตัวข้อมูล ชนิดของข้อมูล และข้อกำหนดข้อมูล

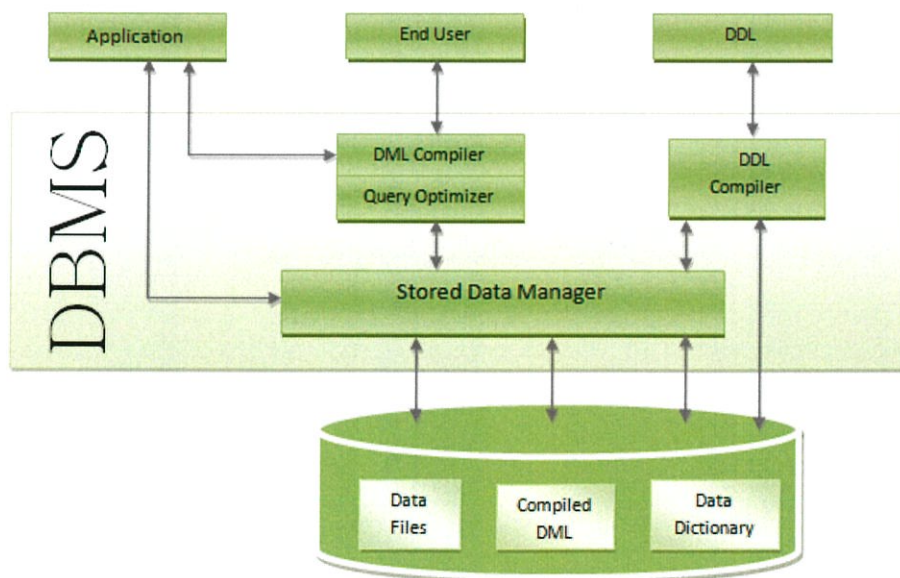
- ระดับที่ 3 ระดับภายนอก (External Level) ประกอบด้วยโครงสร้างภายนอก หรือมุมมองภาพของผู้ใช้ (user views) หลายๆตัว ด้วยกัน แต่ละตัวจะใช้แสดงมุมมองภาพของผู้ใช้เฉพาะกลุ่มเท่านั้น



ภาพที่ 2.4 DBMS architecture

### องค์ประกอบของระบบจัดการฐานข้อมูล

- 1) ส่วนจัดการฐานข้อมูล (Database manager) ประสานระหว่างข้อมูลที่จัดเก็บในฐานข้อมูล โปรแกรมประยุกต์ และส่วนคำร้องขอ โดยกำหนดการกระทำต่างๆ ผ่านส่วนจัดการแฟ้มข้อมูล (File manager) เพื่อไปกระทำและจัดการกับข้อมูลที่เก็บอยู่ในฐานข้อมูลระดับกายภาพ
- 2) ส่วนประมวลผลสอบถาม (Query processor) แปลงประโยคคำสั่งภาษาสอบถามให้อยู่ในรูปแบบของคำสั่งที่ส่วนจัดการฐานข้อมูลเข้าใจ
- 3) ส่วนแปลงภาษาจัดการข้อมูล (data manipulation language precompiled) แปลงประโยคคำสั่งภาษา DML ให้อยู่ในรูปแบบที่ส่วนรหัส object ของโปรแกรมประยุกต์สามารถเข้าใจได้เพื่อส่งต่อรหัสคำสั่งนั้นไปยังส่วนจัดการฐานข้อมูล
- 4) ส่วนแปลภาษานิยามข้อมูล (data definition language precompiled) แปลงประโยคคำสั่งภาษา DLL ให้อยู่ในรูปแบบของ metadata ซึ่งเก็บรายละเอียดของโครงสร้างต่างๆ ของข้อมูลโดยเก็บไว้ใน data dictionary
- 5) ส่วนรหัส object ของโปรแกรมประยุกต์ (application program object code) ทำหน้าที่แปลงคำสั่งต่างๆ ของโปรแกรมประยุกต์ และแปลงคำสั่งภาษา DML ที่ส่งต่อมาจากส่วนแปลภาษาจัดการข้อมูลให้อยู่ในรูป object code ซึ่งจะส่งต่อให้ส่วนจัดการฐานข้อมูลเพื่อจัดการกับข้อมูลในฐานข้อมูลต่อไป



ภาพที่ 2.5 องค์ประกอบของระบบจัดการฐานข้อมูล

ตัวอย่างซอฟต์แวร์ DBMS ที่ได้รับความนิยม

- Oracle
- SAP DB
- IBM DB2
- Sybase SQL
- Microsoft SQL Server
- PostgreSQL
- Pervasive SQL
- Microsoft Access
- SQLite
- Progress
- MySQL
- Firebird



ภาพที่ 2.6 Software DBMS

## 2.2.6 ผังงาน (Flowchart)

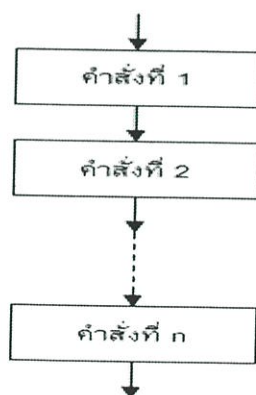
Flow chart คือ แผนภาพที่ใช้สำหรับลำดับขั้นตอนและวิธีการทำงานแบบเป็นลำดับ โดยการทำงานจะเริ่มจากจุดเริ่มต้น (Start) ไปยัง จุดสิ้นสุด (Stop) ของโปรแกรม ซึ่งเราสามารถให้ Flowchart เป็นเครื่องมือในการตรวจสอบอัลกอริทึมด้วยวิธีการไล่โปรแกรมได้ (Tracing) เพื่อหาข้อผิดพลาดของโปรแกรม (Bug) ของโปรแกรมได้อีกด้วยโดยจะใช้ รูปภาพ (Image) หรือ สัญลักษณ์ (Symbol) ที่ใช้เขียนแทนขั้นตอน คำอธิบาย ข้อความ หรือคำพูดที่ใช้ในกิจกรรมปฏิบัติงาน เพื่อให้ง่ายต่อการเข้าใจ

ผังงานแบ่งได้ 2 ประเภท

- 1) ผังงานระบบ (System Flowchart) คือ ผังงานที่แสดงขั้นตอนการทำงานในระบบอย่างกว้างๆ แต่ไม่เจาะลงในระบบงานย่อย
- 2) ผังงานโปรแกรม (Program Flowchart) คือ ผังงานที่แสดงถึงขั้นตอนในการทำงานของโปรแกรม ตั้งแต่รับข้อมูล คำนวณ จนถึงแสดงผลลัพธ์

### โครงสร้างของผังงาน

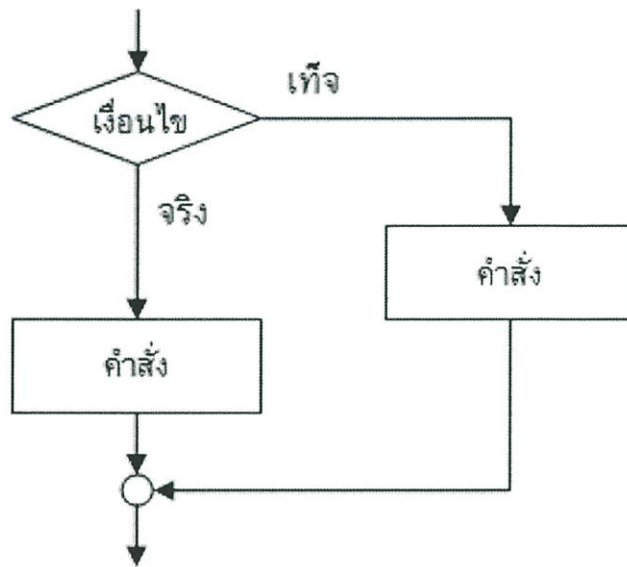
- 1) โครงสร้างแบบลำดับ (Sequential structure) คือ โครงสร้างแสดงขั้นตอนการทำงานที่เป็นไปตามลำดับก่อนหลัง และแต่ละขั้นตอนจะถูกประมวลผลเพียงครั้งเดียวเท่านั้น สามารถแสดงการทำงานของโครงสร้างนี้ ดังรูป



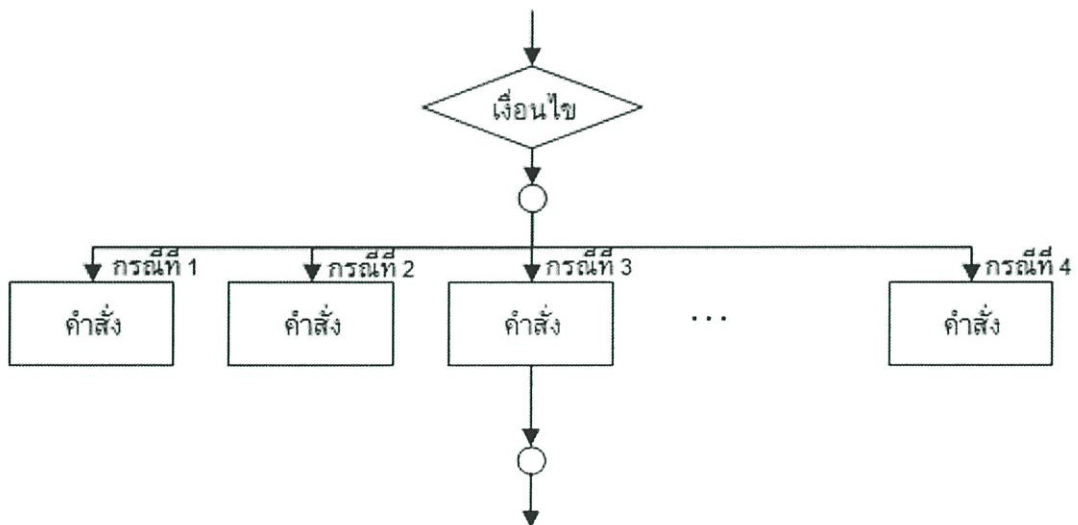
ภาพที่ 2.7 โครงสร้างแบบลำดับ

- 2) โครงสร้างแบบมีทางเลือก (Selection structure) คือ โครงสร้างที่มีเงื่อนไข ขั้นตอนการทำงานบางขั้นตอนต้องมีการตัดสินใจเพื่อเลือกวิธีการประมวลผลขั้นต่อไป และจะมีบางขั้นตอนที่ไม่ได้รับการประมวลผล การตัดสินใจอาจมีทางเลือก 2 ทางหรือมากกว่าก็ได้ โครงสร้างที่มีทางเลือก

เพียง 2 ทาง เราเรียกชื่อว่าโครงสร้างแบบ if...then...else และโครงสร้างที่มีทางเลือกมากกว่า 2 ทาง เรียกชื่อว่า โครงสร้างแบบ case ซึ่งสามารถแสดงการทำงานของโครงสร้างนี้โดยใช้ผังงานได้ดังรูป



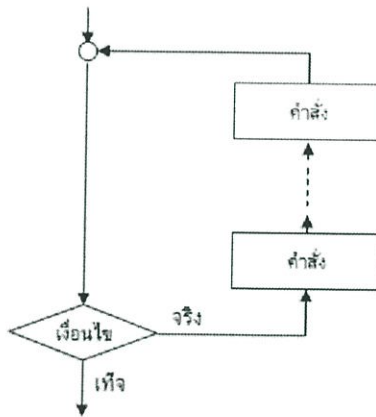
ภาพที่ 2.8 การทำงานของโครงสร้างแบบมีทางเลือกif...then...else



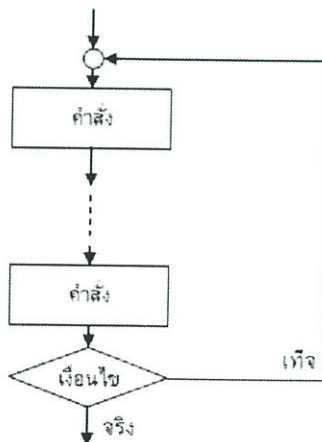
ภาพที่ 2.9 การทำงานของโครงสร้างแบบมีทางเลือกcase

3) โครงสร้างแบบทำซ้ำ (Repetition structure) คือ โครงสร้างที่ขั้นตอนการทำงานบางขั้นตอนได้รับการประมวลผลมากกว่า 1 ครั้ง ทั้งนี้ขึ้นอยู่กับเงื่อนไขบางประการ โครงสร้างแบบทำซ้ำนี้ต้องมีการตัดสินใจในการทำงานซ้ำ และลักษณะการทำงานของโครงสร้างแบบนี้มี 2 ลักษณะ ได้แก่

- การทำซ้ำแบบ do while เป็นแบบที่มีการตรวจสอบเงื่อนไขในการทำซ้ำทุกครั้งก่อนดำเนินการกิจกรรมใดๆ ถ้าเงื่อนไขเป็นจริงจะทำงานซ้ำไปเรื่อยๆ และหยุดเมื่อเงื่อนไขเป็นเท็จ
- การทำซ้ำแบบ do until เป็นแบบที่ทำกิจกรรมซ้ำเรื่อยๆ จนกระทั่งเงื่อนไขที่กำหนดเป็นจริงแล้วจึงหยุดการทำงาน โดยแต่ละครั้งที่เสร็จสิ้นการดำเนินการแต่ละรอบจะต้องมีการตรวจสอบเงื่อนไข



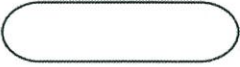
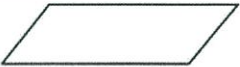



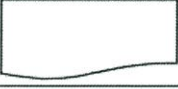
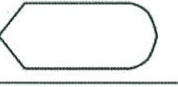




ภาพที่ 2.10 การทำงานของการทำซ้ำแบบ do while



ภาพที่ 2.11 การทำงานของการทำซ้ำแบบ do until

## สัญลักษณ์ของ Flowchart

ตารางที่ 2.8 แสดงสัญลักษณ์ Flowchart

สัญลักษณ์รูปภาพ	ความหมาย
	จุดเริ่มต้น (start) หรือจุดสิ้นสุด (stop)
	รับข้อมูล (input) หรือแสดงผลข้อมูล (output)
	รับข้อมูลนำเข้าจากคีย์บอร์ด (Input from keyboard)
	การคำนวณ (Process)
	การตัดสินใจ (Decision) หรือ การเปรียบเทียบ (compare)
	แสดงผลข้อมูลออกทางเครื่องพิมพ์ (printer)
	การแสดงผลออกทางจอภาพ (display)
	การทำงานย่อย (subprogram)
	จุดเชื่อมต่อในหน้าเดียวกัน (connector)
	จุดเชื่อมต่อไปหน้าอื่น (Off page connector)
	ทิศทาง (Flow Line)

### 2.2.7 Use case diagram

สัญลักษณ์ที่สำคัญของ Use Case Diagram มีดังต่อไปนี้

- Use Case คือ หน้าหน้าที่ระบบต้องกระทำ ใช้สัญลักษณ์รูปวงรี พร้อมทั้งเขียนชื่อ Use Case ซึ่งต้องใช้คำกริยาหรือกริยาวลีก็ได้



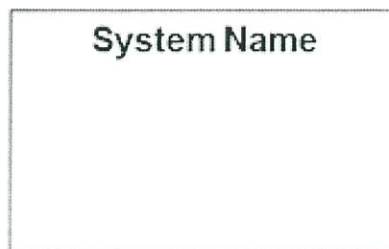
ภาพที่ 2.12 สัญลักษณ์ Use case

- Actor คือ ผู้เกี่ยวข้องกับระบบ ซึ่งรวมทั้ง Primary Actor และ Stakeholder Actor ที่เป็นมนุษย์ ในที่นี่จะใช้สัญลักษณ์รูปคน (Stick Man Icon) เหมือนกัน พร้อมทั้งเขียนชื่อ Actor ไว้ด้านล่างของสัญลักษณ์ด้วย แต่หากเป็น Actor ที่ไม่ใช่มนุษย์ เช่น ระบบงานอื่นที่อยู่นอกเหนือระบบที่เราสนใจ จะใช้รูปสี่เหลี่ยมแล้วเขียนคำว่า “<<actor>>” ไว้ด้านบนแทน



ภาพที่ 2.13 สัญลักษณ์ Actor

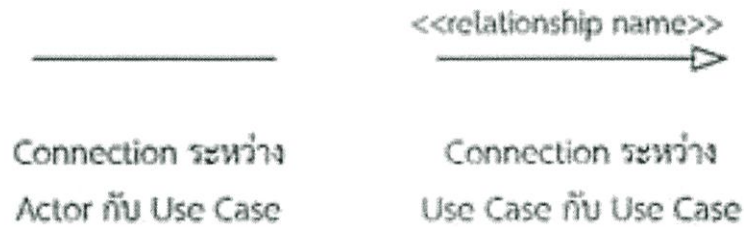
- System Boundary เส้นแบ่งขอบเขตระหว่างระบบกับผู้กระทำต่อระบบ (Use Case กับ Actor) ใช้รูปสี่เหลี่ยมเป็นสัญลักษณ์ พร้อมทั้งเขียนชื่อระบบไว้ด้านใน ซึ่งสำคัญมากทุกการเขียนจะต้องไม่ลืมเขียน System Boundary



ภาพที่ 2.14 สัญลักษณ์ System Boundary

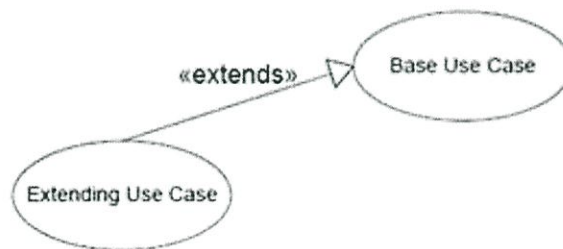
- Connection คือ เส้นที่ลากเชื่อมต่อระหว่าง Actor กับ Use Case ที่มีปฏิสัมพันธ์กัน ใช้เส้นตรงไม่มีหัวลูกศรเป็นสัญลักษณ์ของ Connection ส่วน Connection ที่ใช้เชื่อมต่อระหว่าง Use

Case กับ Use Case กรณีที่ Use Case นั้นมีความสัมพันธ์ซึ่งกันและกัน จะใช้สัญลักษณ์เส้นตรงมีหัวลูกศร พร้อมทั้งเขียนชื่อความสัมพันธ์ไว้ตรงกลางเส้นด้วย โดยเขียนไว้ภายในเครื่องหมาย<<...>>

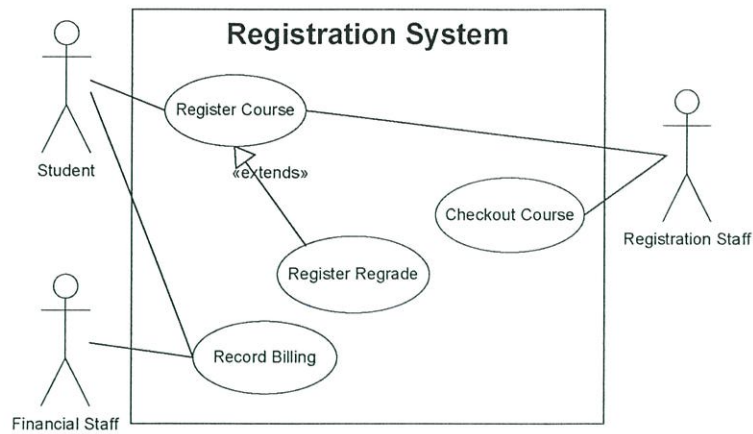


ภาพที่ 2.15 สัญลักษณ์ Connection

- Extend Relationship เป็นความสัมพันธ์แบบขยายหรือเพิ่ม เกิดขึ้นในกรณีที่บาง Use Case ดำเนินกิจกรรมของตนเองไปตามปกติ แต่อาจจะมีเงื่อนไขหรือสิ่งกระตุ้นบางอย่างที่ส่งผลให้กิจกรรมตามปกติของ Use Case นั้นถูกรบกวนจนเบี่ยงเบนไป ซึ่งเราสามารถแสดงเงื่อนไขหรือสิ่งกระตุ้นเหล่านั้นได้ในรูปของ “Use Case” และเรียกความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “Extend Relationship” โดยเรียก Use Case ที่ถูกรบกวนหรือ Use Case ที่ดำเนินงานตามปกติว่า “Base Use Case” และเรียก Use Case ที่ทำหน้าที่รบกวนหรือกระตุ้น Base Use Case ว่า “Extending Use Case” ซึ่งการเขียนสัญลักษณ์ Extend Relationship จะเขียนใน Connection เช่น <<extend>>



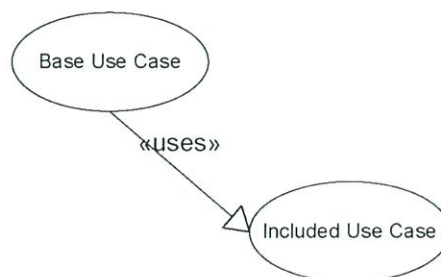
ภาพที่ 2.16 สัญลักษณ์ Extend Relationship



ภาพที่ 2.17 Use Case Diagram ที่มีความสัมพันธ์แบบ Extend Relationship

จากภาพที่ 2.17 สังเกตที่ Use Case “Register Course” ซึ่งเป็น Base Use Case คือ ทำหน้าที่ รับลงทะเบียนตามปกติ แต่เมื่อมีเงื่อนไขหรือมีเหตุการณ์พิเศษเกิดขึ้น คือ “นักศึกษาบางคน อาจมีการลงทะเบียนเรียนซ้ำเพื่อปรับเกรดด้วย (Regrade)” จึงได้เพิ่ม Extending Use Case เพื่อมารองรับหน้าที่พิเศษดังกล่าว นั่นคือ “Register Regrade”

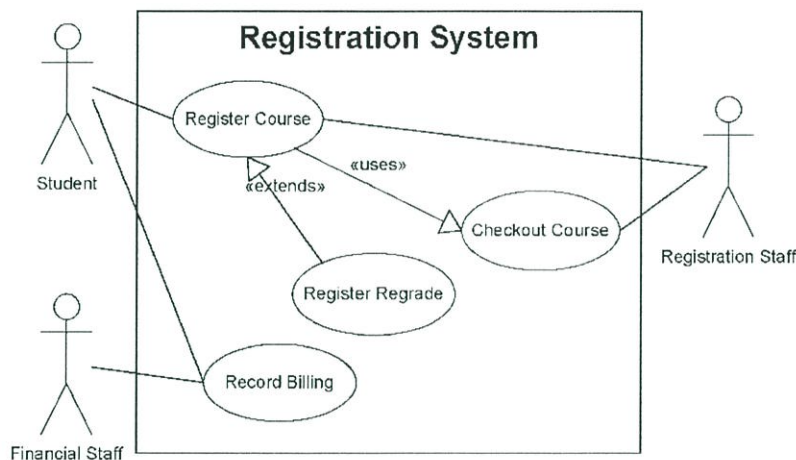
- Include Relationship ความสัมพันธ์อีกรูปแบบหนึ่งของ Use Case Diagram ก็คือ ความสัมพันธ์แบบเรียกใช้เกิดขึ้นในกรณีที่ Use Case หนึ่งไปเรียกหรือดึงกิจกรรมของอีก Use Case หนึ่งมาใช้เพื่อให้กิจกรรมนั้นเกิดขึ้นจริงใน Use Case ของตนเอง หรือกล่าวให้ง่ายกว่านั้นคือกิจกรรมใน Use Case หนึ่ง อาจจะถูกผนวกเข้าไปรวมกับกิจกรรมของอีก Use Case หนึ่ง เราเรียกความสัมพันธ์ระหว่าง Use Case ในลักษณะนี้ว่า “Include Relationship” โดย Use Case ที่ทำหน้าที่ดึงกิจกรรมมาจาก Use Case อื่นๆ เรียกว่า “Base Use Case” ในขณะที่ Use Case ที่ถูกเรียก หรือถูกดึงกิจกรรมมาใช้ เรียกว่า “Included Use Case” สามารถเขียนเส้น Connection ได้ในทิศทางตรงกันข้ามกับ Extend Relationship โดยเริ่มต้นลากเส้นตรงจาก Base Use Case หันลูกศรชี้ไปที่ Included Use Case แล้วเขียนชื่อว่า «include» ไว้ตรงกลาง



ภาพที่ 2.18 สัญลักษณ์ Include Relationship

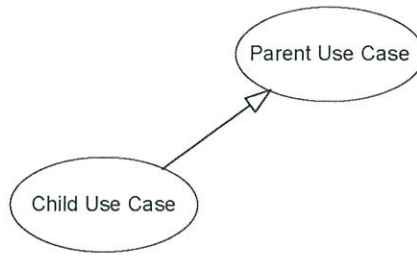
ความสัมพันธ์ระหว่าง Use Case แบบ Include เป็นการสนับสนุนหลักการนำกลับมาใช้ใหม่ ของ Use Case (Use Case Reusability) กล่าวคือ Use Case หนึ่งสามารถถูก Include ได้โดย Base Use Case หลายๆตัว และสามารถถูก Include ได้มากกว่าหนึ่งครั้งด้วย

ดังนั้นจากภาพที่ 2.17 เมื่อพิจารณาแล้ว Use Case “ตรวจสอบรายวิชา (Checkout Course)” สามารถถูกเรียกใช้จาก Use Case “ลงทะเบียนเรียน (Register Course)” ได้ ดังนั้น Use Case “Checkout Course” มีความสัมพันธ์กับ Use Case “Register Course” แบบ Include แสดง Use Case Diagram ได้ดังภาพที่ 2.19



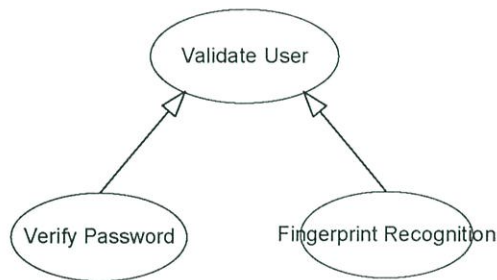
ภาพที่ 2.19 Use Case Diagram ที่มี Included Relationship

- Generalization/Specialization Relationship ที่เกิดขึ้นระหว่าง Use Case มีคุณสมบัติแตกต่างจาก Generalization/Specialization ที่เกิดขึ้นระหว่างคลาส คือ ความสัมพันธ์ลักษณะดังกล่าวที่เกิดขึ้นใน Use Case นี้จะไม่มีการถ่ายทอดคุณลักษณะ (ไม่มีการ Inherit) แต่จะใช้เพื่อแสดงความสัมพันธ์แบบจำแนกแยกแยะประเภทของ Use Case เท่านั้น อย่างไรก็ตาม Use Case ที่เป็น Use Case หลักในการจำแนกประเภทจะเรียกว่า “Parent Use Case” ส่วน Use Case ที่ถูกจำแนกแยกแยะออกมาจะเรียกว่า “Child Use Case” ส่วนสัญลักษณ์เชื่อมความสัมพันธ์ให้ใช้เส้นตรงลูกศรไปร่งลากจาก Child Use Case ให้ลูกศรชี้ไปที่ Parent Use Case



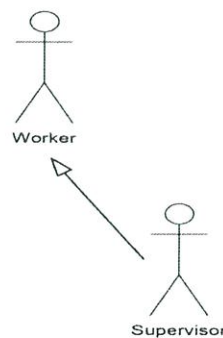
ภาพที่ 2.20 ความสัมพันธ์ระหว่าง Use Case แบบ Generalization / Specialization

ใช้ Generalization / Specialization ในกรณีที่ต้องการแสดงความสัมพันธ์ในเชิงการจำแนกแยกแยะประเภทของ Use Case เช่น การตรวจสอบความถูกต้องของผู้ใช้งานระบบ (Validate user) สามารถกระทำได้หลายวิธี ได้แก่ การตรวจสอบจากรหัสผ่าน (Verify Password) และการตรวจจากลายนิ้วมือ (Fingerprint Recognition) เป็นต้น



ภาพที่ 2.21 ความสัมพันธ์ระหว่าง Use Case แบบ Generalization / Specialization

นอกจากนี้ยังสามารถใช้ความสัมพันธ์ลักษณะนี้กับ Actor ได้ด้วย ตัวอย่างเช่น เราสามารถใช้ UML อธิบายข้อเท็จจริง “คนคุมงาน (Worker Supervisor) จัดเป็นคนงานประเภทพิเศษที่มีหน้าที่พิเศษกว่าคนงาน (Worker) ทั่วไป”



ภาพที่ 2.22 แสดงความสัมพันธ์แบบ Generalization / Specialization ระหว่าง Actor

## 2.2.8 ER Diagram

Entity Relationship Diagram (ER Diagram) คือ แบบจำลองที่ใช้อธิบายโครงสร้างของฐานข้อมูล ซึ่งเขียนออกมาในลักษณะของรูปภาพ

ตารางที่ 2.9 เปรียบเทียบสัญลักษณ์ที่ใช้ใน ER Diagram

### A Comparison of ER Modeling Symbols

	Chen	Crow's Foot	Rein85	IDEF1X
Entity				
Relationship line				
Relationship				
Option symbol				
One (1) symbol	1			
Many (M) symbol	M			
Composite entity				
Weak entity				

โดยการอธิบายโครงสร้างและความสัมพันธ์ของข้อมูล (Relationship) ประกอบด้วย

1) Entity คือบุคคล สถานที่ วัตถุ หรือเหตุการณ์ที่ทำให้เกิดกลุ่มของ ข้อมูลที่ต้องการจัดเก็บ รวมทั้งสามารถบ่งชี้ถึงความเป็นเอกลักษณ์เฉพาะตัว (uniquely identifiable) นอกจากนี้ entity สามารถแบ่งออกได้เป็น 3 ประเภท

- Regular Entity or Strong Entity: เป็น entity ที่อยู่ได้โดยไม่ต้องอาศัย entity อื่นโดยมี Attribute หนึ่งแยกความแตกต่างของข้อมูล

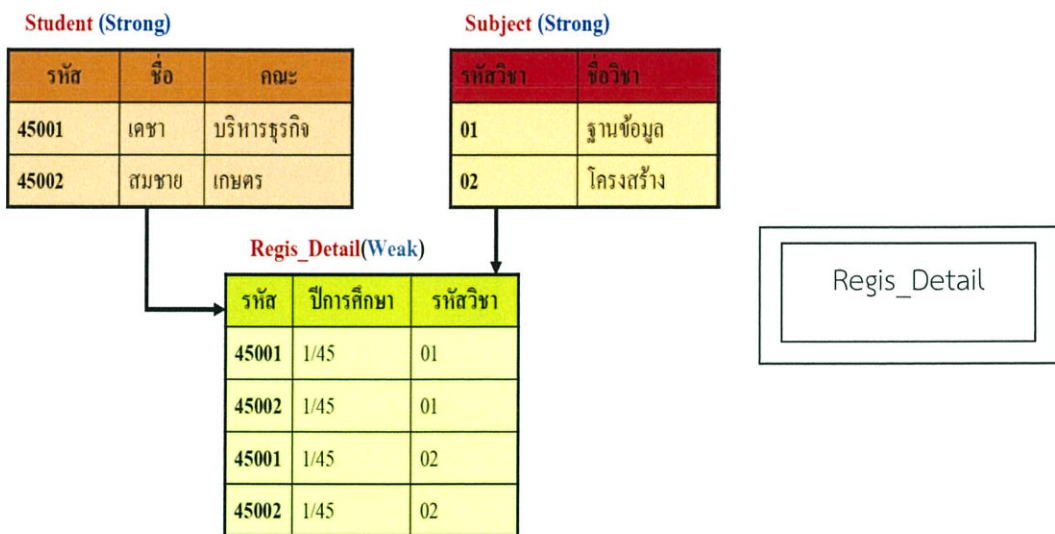
เอนทิตี STUDENT

SID	SName	Address	Grade
47035501	ประธาน สมพงษ์	111 หมู่ 5 สามเสน กรุงเทพฯ	3.5
47035502	อำนาจ รักดี	222 หมู่ 2 บางซื่อ กรุงเทพฯ	2.8
47035503	ประนอม รักอ่าน	333 หมู่ 1 หนองจอก กรุงเทพฯ	3.0

Student

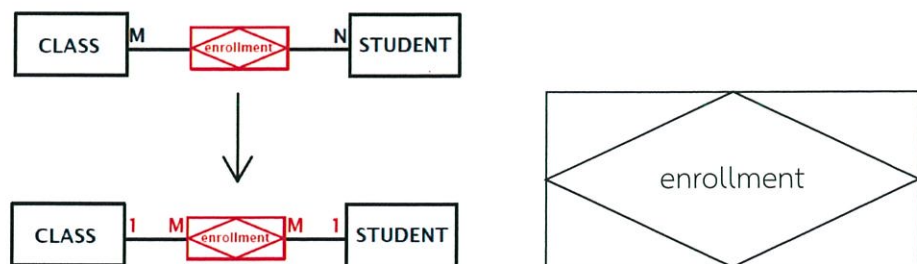
ภาพที่ 2.23 ตัวอย่าง Regular Entity และสัญลักษณ์ Regular Entity

- Weak Entity: Entity ที่ต้องอาศัย Attribute จาก Entity อื่นมาช่วยในแยกความแตกต่างของข้อมูล



ภาพที่ 2.24 ตัวอย่าง Weak Entity และสัญลักษณ์ Weak Entity

- Composite Entity: เป็น Entity ที่ถูกสร้างขึ้นเพื่อแปลงความสัมพันธ์แบบ M:M ให้เป็นแบบ 1:M ซึ่ง Primary Key ของ Composite Entity ก็คือ Key Attribute ของทั้ง 2 Entity หลัก



ภาพที่ 2.25 ตัวอย่าง Composite Entity และสัญลักษณ์ Composite Entity

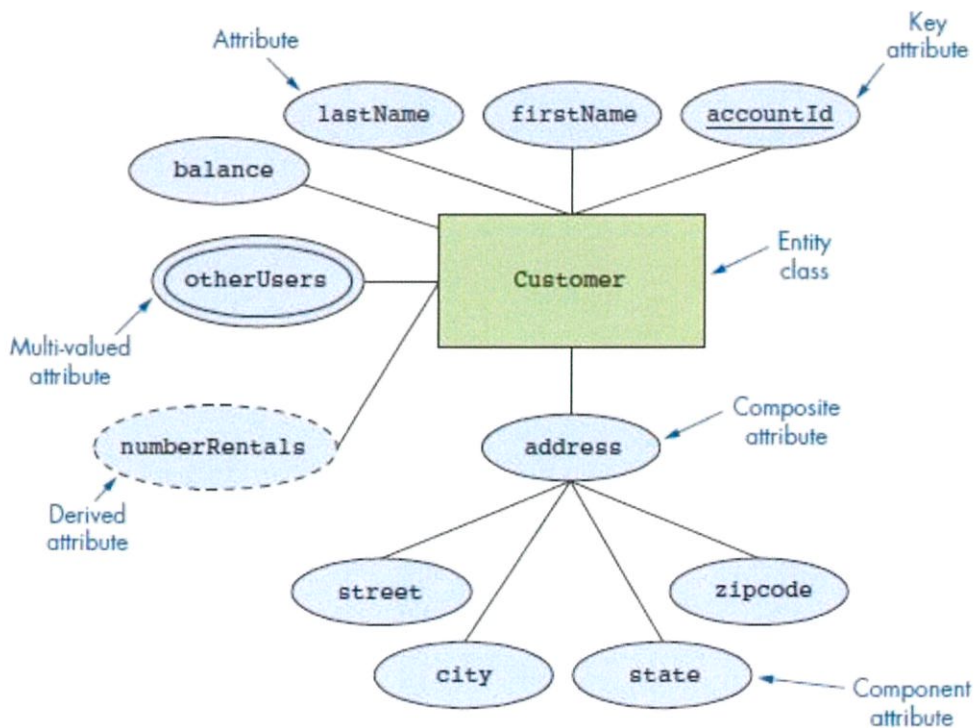
2) Attribute ใช้แสดงถึงคุณสมบัติของ entity เช่น ชื่อ นามสกุล เลขประจำตัว ที่อยู่ ฯลฯ แทนที่ด้วยชื่อของ attribute ในรูปวงรี

- Key attribute : attribute ที่ถูกกำหนดให้เป็นคีย์ของ entity โดยแทนที่ด้วย attribute ที่ถูกขีดเส้นใต้

- Multi-valued attribute : attribute ที่มีค่าบรรจุอยู่มากกว่าหนึ่งค่า โดยแทนที่ด้วยวงกลมรูปไข่ซ้อนกันสองวง

- Derived attribute : attribute ที่มีค่ามาจากการคำนวณของ attribute อื่น โดยแทนที่ด้วยวงกลมรูปไข่ที่เป็นเส้นประ

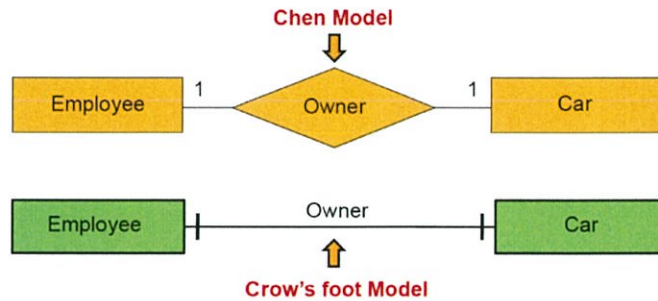
- Composite attribute : attribute ที่สามารถแยกออกเป็น attribute ย่อยๆ ได้หลาย attribute แทนที่โดยชื่อ attribute ในวงกลมรูปไข่ที่มีเส้นตรงลากไปเชื่อมโยงกับ attribute หลัก จากในภาพ attribute ที่ชื่อ address สามารถแยกออกเป็น attribute ย่อยที่ชื่อ street, city, state, zipcode ได้อีก



ภาพที่ 2.26 ตัวอย่างชนิดของ Attribute

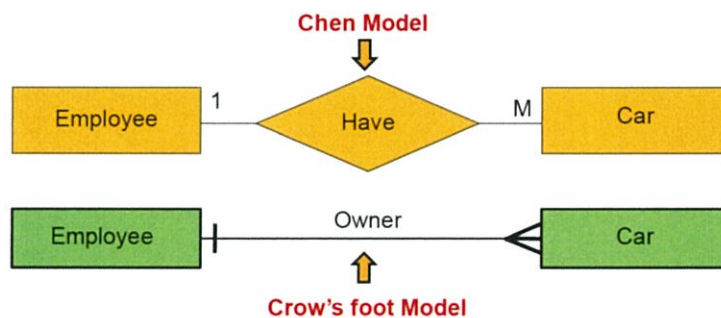
3) ความสัมพันธ์ (Relationship) คือ ความสัมพันธ์ระหว่างเอนทิตี

- One-to-One (1:1) เป็นความสัมพันธ์ของสมาชิกของใน Entity หนึ่งไปสัมพันธ์กับสมาชิกของใน Entity อีกอันหนึ่ง เพียงหนึ่งเดียวเท่านั้น



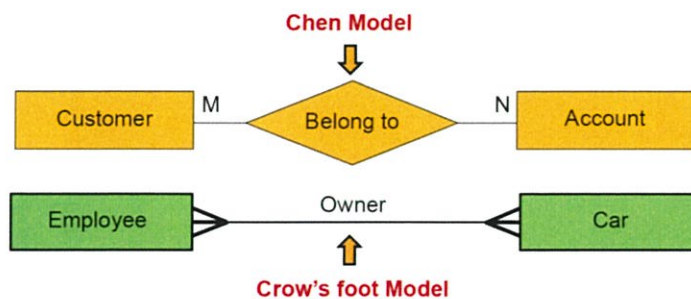
ภาพที่ 2.27 แสดงความสัมพันธ์แบบ 1:1

- One-to-Many (1:M) เป็นความสัมพันธ์ของสมาชิกของใน Entity หนึ่งไปสัมพันธ์กับสมาชิกของใน Entity อีกอันหนึ่ง มากกว่า 1 สมาชิก



ภาพที่ 2.28 แสดงความสัมพันธ์แบบ 1:M

- Many-to-Many (M:N) เป็นความสัมพันธ์ของสมาชิกมากกว่า 1 สมาชิก ของใน Entity หนึ่ง ไปสัมพันธ์กับสมาชิกของใน Entity อีกอันหนึ่ง มากกว่า 1 สมาชิก



ภาพที่ 2.29 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม

### 2.2.9 คีย์ (Key)

คุณสมบัติหนึ่งที่สำคัญของความสัมพันธ์ก็คือ ความเป็นเอกลักษณ์ (Uniqueness property) สิ่งที่ใช้กำหนดความเป็นเอกลักษณ์ของแถวในความสัมพันธ์ เรียกว่า คีย์ (key)

ฐานข้อมูลหลายๆ ฐานจะมีข้อมูลอยู่มากมาย ยิ่งฐานข้อมูลมีขนาดใหญ่ขึ้นก็จะมีข้อมูลจำนวนมากขึ้นเป็นเงาตามตัวข้อมูลเหล่านี้อาจมีค่าแตกต่างกัน คล้ายกัน หรือแม้กระทั่งเหมือนกัน ทำให้การแยกแยะโดยอาศัยเพียงตัวข้อมูลอย่างเดียวทำได้ยากลำบาก ดังนั้นจึงมีการกำหนดค่า Keys ประจำข้อมูลเพื่อทำให้การแยกแยะข้อมูลในฐานข้อมูลเป็นไปอย่างถูกต้อง

#### ประเภทของคีย์

1) คีย์หลัก (Primary key) คือ คีย์หลักที่ใช้ในการอ้างอิง Entity ในฐานข้อมูล การเลือกคีย์หลักสามารถเลือกได้จากเรคคอร์ดใดๆ ก็ได้ที่ไม่มีโอกาสซ้ำซ้อนกันบนฐานข้อมูลนั้นคีย์หลักเป็นข้อมูลสำคัญที่จะทำให้การเข้าถึงข้อมูลบนฐานข้อมูลเป็นไปได้อย่างรวดเร็ว ดังนั้นผู้ใช้จึงควรกำหนดคีย์หลักให้ชัดเจนตั้งแต่ขั้นตอนออกแบบฐานข้อมูล หากไม่มีข้อมูลใดเลยในฐานข้อมูลที่เหมาะสมที่จะเป็นคีย์หลักก็ควรที่จะกำหนด Record ใหม่สำหรับให้เป็นคีย์หลักโดยเฉพาะ

2) คีย์รอง (Secondary Key) คีย์สำรอง คือ คีย์เดี่ยวหรือคีย์ผสม (Single or Composite key) ซึ่งเมื่อใช้ในการค้นหาข้อมูลจากความสัมพันธ์จะได้มากกว่าหนึ่งเรคคอร์ด ต่างจากคีย์หลักที่ทำให้ข้อมูลในตารางไม่ซ้ำกัน ดังนั้นคีย์รองจึงไม่จำเป็นจะต้องเป็นเอกลักษณ์

3) คีย์นอก (Foreign key) คีย์นอก คือ คีย์เดี่ยวหรือคีย์ผสม ซึ่งปรากฏเป็นคีย์ทั่วไปของความสัมพันธหนึ่ง แต่ไปปรากฏเป็นอีกคีย์หลักในอีกความสัมพันธ์หนึ่ง คีย์นอกเป็นอีกคีย์หนึ่งที่มีความสำคัญมากในฐานข้อมูลเชิงสัมพันธ์ เนื่องจากเป็นตัวที่ใช้สร้างการเชื่อมต่อระหว่างความสัมพันธ์ การเปลี่ยนแปลงค่าของคีย์นอกจะต้องอาศัยความระมัดระวังเป็นอย่างมากเนื่องจากจะมีผลกระทบโดยตรงต่อข้อมูลในความสัมพันธ์อื่นที่มีการอ้างอิงถึงคีย์นอกตัวนี้ จึงมีกฎและเงื่อนไขที่บังคับใช้เพื่อให้ข้อมูลมีความถูกต้องอยู่เสมอ การกำหนดค่าให้กับคีย์นอกของความสัมพันธ์ที่อ้างอิงถึงจะต้องกำหนดค่าของคีย์ให้อยู่ในโดเมนเดียวกันกับความสัมพันธ์ที่คีย์นอกนั้นเป็นคีย์หลัก แต่คีย์นอกนั้นไม่จำเป็นจะต้องเป็นส่วนหนึ่งในคีย์หลักของความสัมพันธ์อื่น

4. ซุปเปอร์คีย์ (Superkey) คือกลุ่มของแอททริบิวต์ที่สามารถนำไปใช้ในการค้นหาข้อมูลที่เป็นเอกลักษณ์ได้

5. คีย์แข่งขัน (Candidate key) คีย์แข่งขัน คือซุปเปอร์คีย์และไม่มีกลุ่มย่อยของคีย์ใดในคีย์แข่งขันที่จะสามารถเป็นซุปเปอร์คีย์ได้อีก

## บทที่ 3

### วิธีการดำเนินงาน

#### 3.1 ขั้นตอนการดำเนินการ

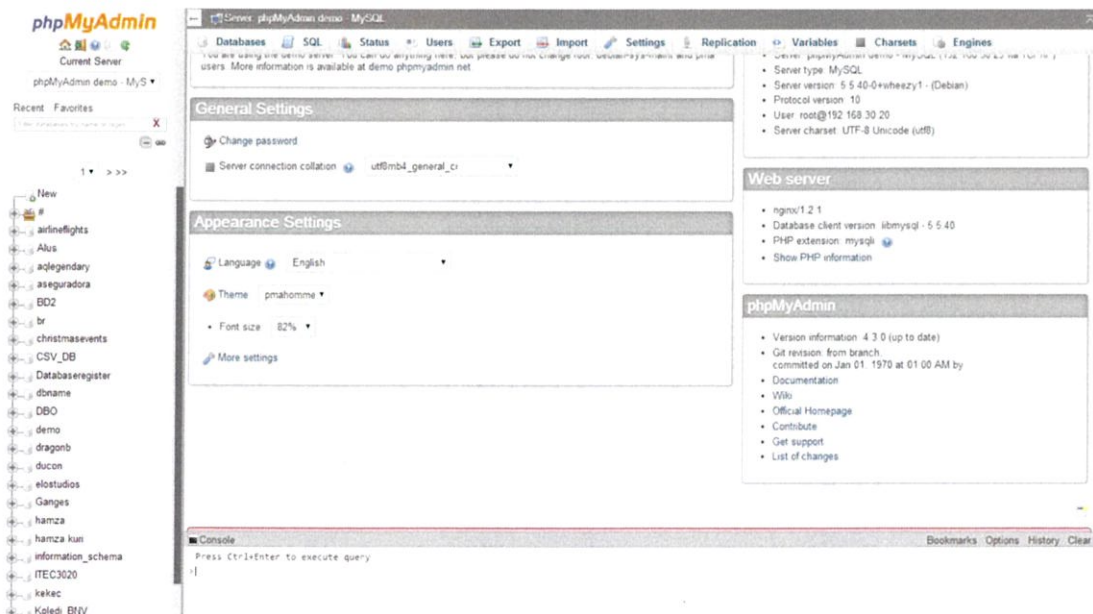
##### 3.1.1 ลงโปรแกรมที่ใช้ในการทำงาน

- XAMPP สามารถดาวน์โหลดโปรแกรม XAMPP ได้ที่เว็บไซต์ของ [apachefriends.org](http://apachefriends.org)



ภาพที่ 3.1 โปรแกรม XAMPP

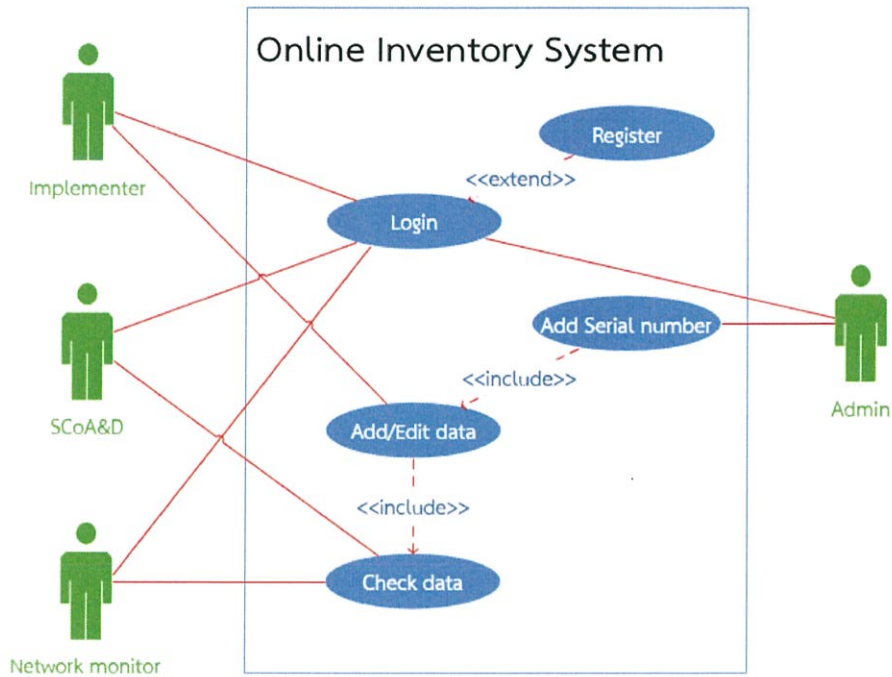
- phpMyAdmin สามารถดาวน์โหลดได้ที่เว็บไซต์ [phpmyadmin.net/downloads/](http://phpmyadmin.net/downloads/)



ภาพที่ 3.2 โปรแกรม phpMyAdmin

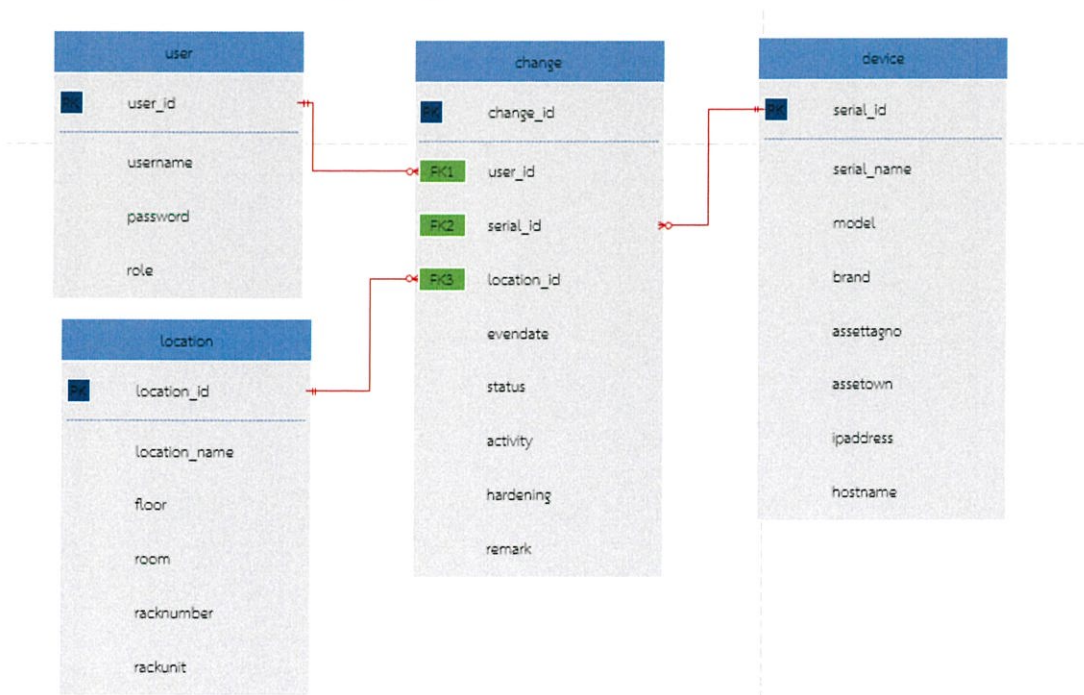


### 3.1.3 ออกแบบ Use case diagram



ภาพที่ 3.5 Use case แสดงการทำงานของเว็บไซต์

### 3.1.4 ออกแบบระบบฐานข้อมูล

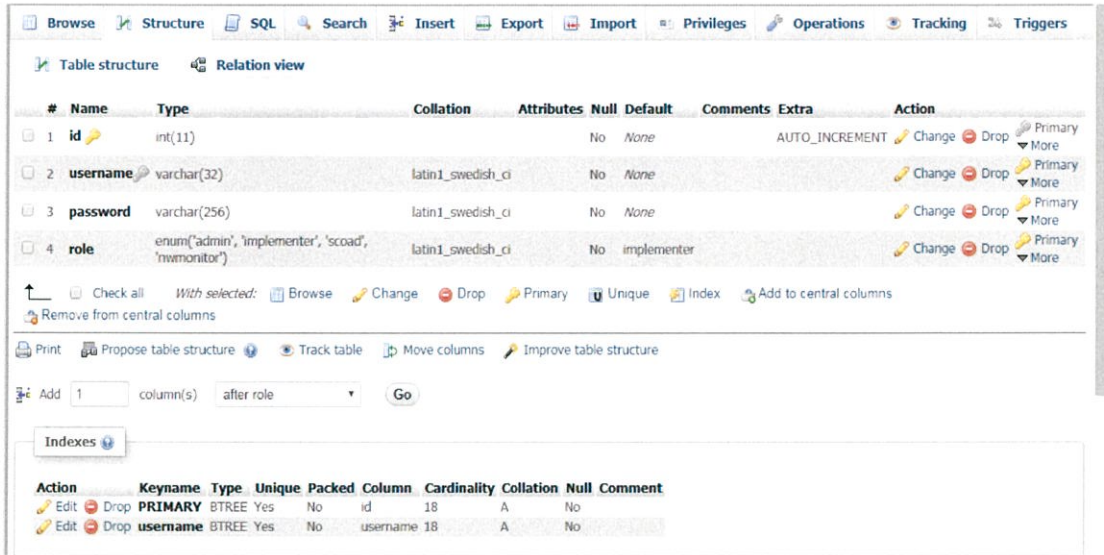


ภาพที่ 3.6 ER diagram แสดงการทำงานของเว็บไซต์

## 3.2 สร้างระบบฐานข้อมูล

### 3.2.1 ทำระบบ Login

- สร้าง table user สำหรับการเก็บข้อมูลของทุกคนที่เข้าสู่ระบบโดยมีการเก็บ id, username, password และrole



ภาพที่ 3.7 โครงสร้างตาราง user

- ใส่ค่าลงในตาราง user

+ Options			id	username	password	role
<input type="checkbox"/>	Edit Copy Delete	1	admin	admin	admin	admin
<input type="checkbox"/>	Edit Copy Delete	3	scoad	scoad	scoad	scoad
<input type="checkbox"/>	Edit Copy Delete	4	nwmonitor	nwmonitor	nwmonitor	nwmonitor
<input type="checkbox"/>	Edit Copy Delete	17	uninui	uninui	uninui	implementer
<input type="checkbox"/>	Edit Copy Delete	18	nunui	nunui	nunui	nwmonitor
<input type="checkbox"/>	Edit Copy Delete	27	tiger	tiger	tiger	implementer
<input type="checkbox"/>	Edit Copy Delete	29	we	we	we	implementer
<input type="checkbox"/>	Edit Copy Delete	32	hi	hi	hi	implementer

ภาพที่ 3.8 แสดงค่าของตาราง user

### 3.2.2 เชื่อมระบบฐานข้อมูลกับหน้าเว็บ

```

//update Items
if(isset($_POST['update_item'])){

    $edit_item_id = $_POST['edit_item_id'];
    $serial = $_POST['serial'];
    $model = $_POST['model'];
    $brand = $_POST['brand'];
    $assettagno = $_POST['assettagno'];
    $assettown = $_POST['assettown'];
    $sql = "UPDATE device SET
        serial='$serial',
        model='$model',
        brand='$brand',
        assettagno='$assettagno',
        assettown='$assettown'
        WHERE serial='$edit_item_id' ";
    if ($conn->query($sql) === TRUE) {
        echo "<script>>window.location.href='inventory.php'</script>";
    } else {
        echo "Error updating record: " . $conn->error;
    }
}

if(isset($_POST['delete'])){
    // sql to delete a record
    $delete_id = $_POST['delete_id'];
    $sql = "DELETE FROM $dtable WHERE serial='$delete_id' ";
    if ($conn->query($sql) === TRUE) {
        echo "<script>>window.location.href='inventory.php'</script>";
    } else {
        echo "Error deleting record: " . $conn->error;
    }
}
}

//Add Item
if(isset($_POST['add_item'])){
    $serial = $_POST['serial'];
    $model = $_POST['model'];
    $brand = $_POST['brand'];
    $assettagno = $_POST['assettagno'];
    $assettown = $_POST['assettown'];

    $sql = "INSERT INTO device (serial,model,brand,assettagno,assettown) VALUES
        ( '$serial', '$model', '$brand', '$assettagno', '$assettown' ) ";
    if ($conn->query($sql) === TRUE) {
        echo "<script>>window.location.href='inventory.php'</script>";
    } else {
        echo "Error updating record: " . $conn->error;
    }
}

//Addmove
if(isset($_POST['add_movement'])){
    $serial = $_POST['serial'];
    $model = $_POST['model'];
    $brand = $_POST['brand'];
    $assettagno = $_POST['assettagno'];
    $assettown = $_POST['assettown'];

    $sql_movem = "INSERT INTO $dtable_inven (serial) VALUES
        ( '$serial' ) ";
    if ($conn->query($sql_movem) === TRUE) {
        echo "<script>>window.location.href='admin.php'</script>";
    } else {
        echo "Error updating record: " . $conn->error;
    }
}
}

```

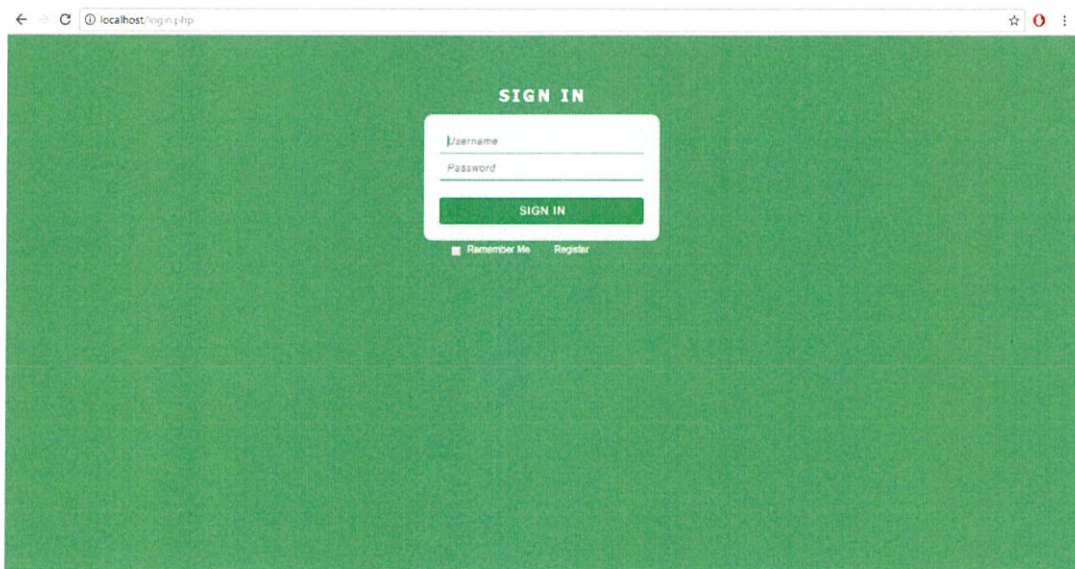
ภาพที่ 3.9 โค้ดการเชื่อม database กับหน้าเว็บไซต์

## บทที่ 4

### ผลการดำเนินงาน

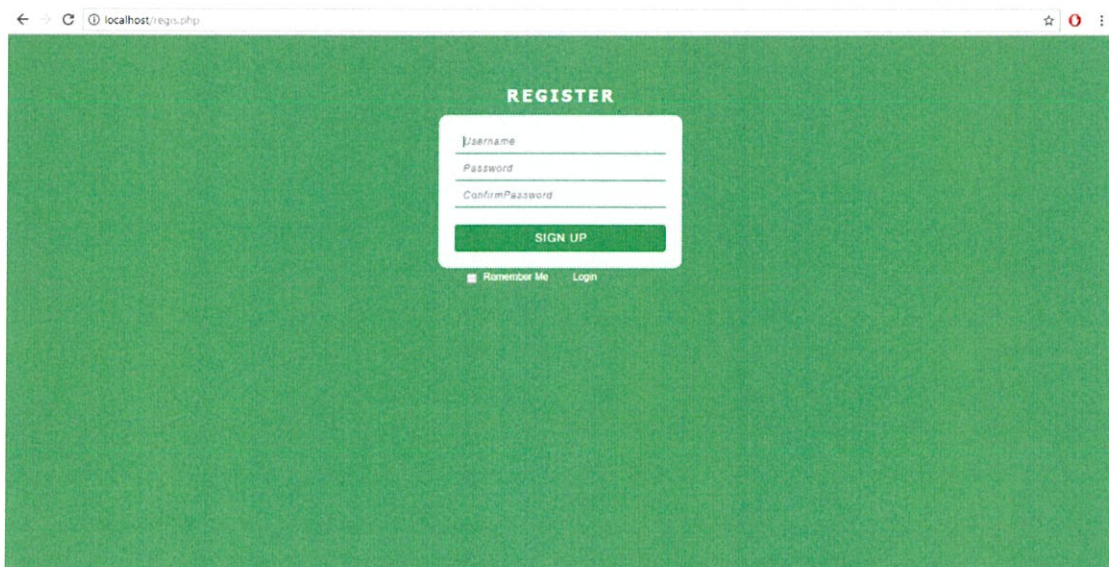
#### 4.1 การเข้าสู่ระบบ

##### 4.1.1 Sign in



ภาพที่ 4.1 sign in page

##### 4.1.2 Register



ภาพที่ 4.2 register page

## 4.2 โครงสร้างของเว็บไซต์

### 4.2.1 Admin

Admin มีหน้าที่เพิ่มอุปกรณ์ที่ต้องการจะย้ายไปที่หน้า movement เพื่อให้ implementer เข้ามาเพิ่มข้อมูล เช่น Change number, ip address, rack number เป็นต้น โดยที่ admin สามารถเพิ่ม/แก้ไข/ลบข้อมูลได้ทุกหน้า

- admin.php เป็นหน้าที่บอกภาพรวมของอุปกรณ์ที่ต้องการจะย้าย

ADMIN MOVEMENT SERIAL MEMBER

INVENTORY LOG

ADMIN (admin)

MOVEMENT

EXPORT

Show 10 entries

Search

ID	Serial	Model	Brand	Asset Tag No	Asset Owner	Lc
43	JMX1345Z0JY	ASA5505-BUN-K9	Cisco	CBAYX-001238	BAY	
44	1107200506000288	IB-1050-A-NS1GRID	Infoblox	CBAYS-000002	BAY	
45	0079052007000041	NS-ISG-2000	JUNIPER	N/A	BAY	ram
46	0079102009000136	NS-ISG-2000	JUNIPER	CBAYX-000088	BAY	
47	JMX1345Z0JY	ASA5505-BUN-K9	Cisco	CBAYX-001238	BAY	

ID Search Serial Search Model Search Brand Search Asset Tag No Search Asset Owner St

Showing 1 to 5 of 5 entries

Previous 1 Next

ภาพที่ 4.3 แสดง movement

- ssearch.php เป็นหน้าที่สามารถค้นหาอุปกรณ์ และสามารถเพิ่มไปที่ movement ได้

ADMIN MOVEMENT SERIAL MEMBER

INVENTORY LOG

ADMIN (admin)

SEARCH FOR ADD MOVEMENT

t Find

Show 10 entries

Serial	Model	Brand	Asset Tag No	Asset Owner	Action
FOX1420H0T8	7206VXRG2/2+VPNK9	Cisco	CBAYN-000021	BAY	+ -
JMX1345Z0UT	ASA5505-BUN-K9	Cisco	CBAYX-001501	BAY	+ -

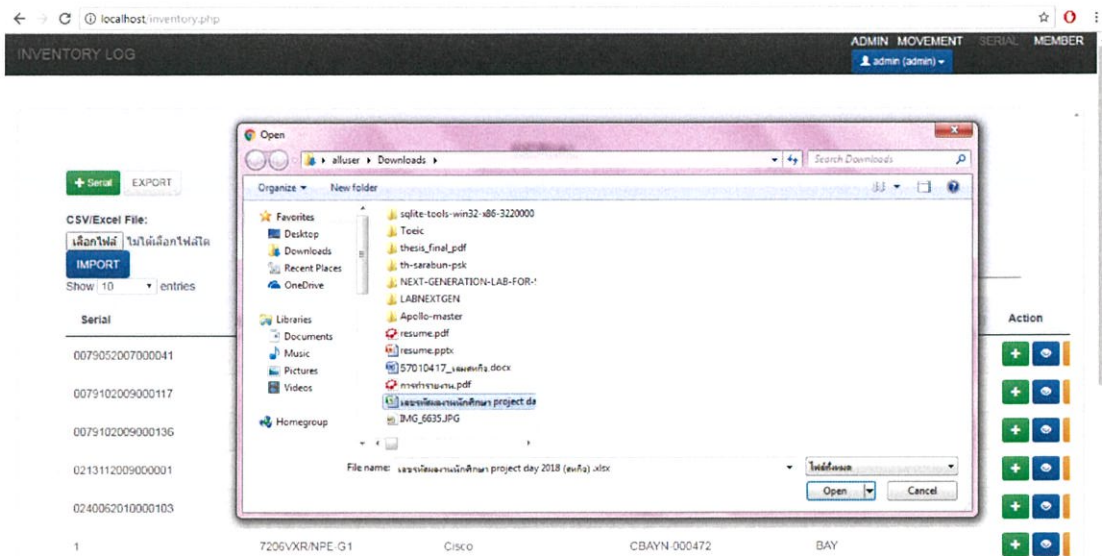
Serial Model Brand Asset Tag No Asset Owner Action

Showing 1 to 2 of 2 entries

Previous 1 Next

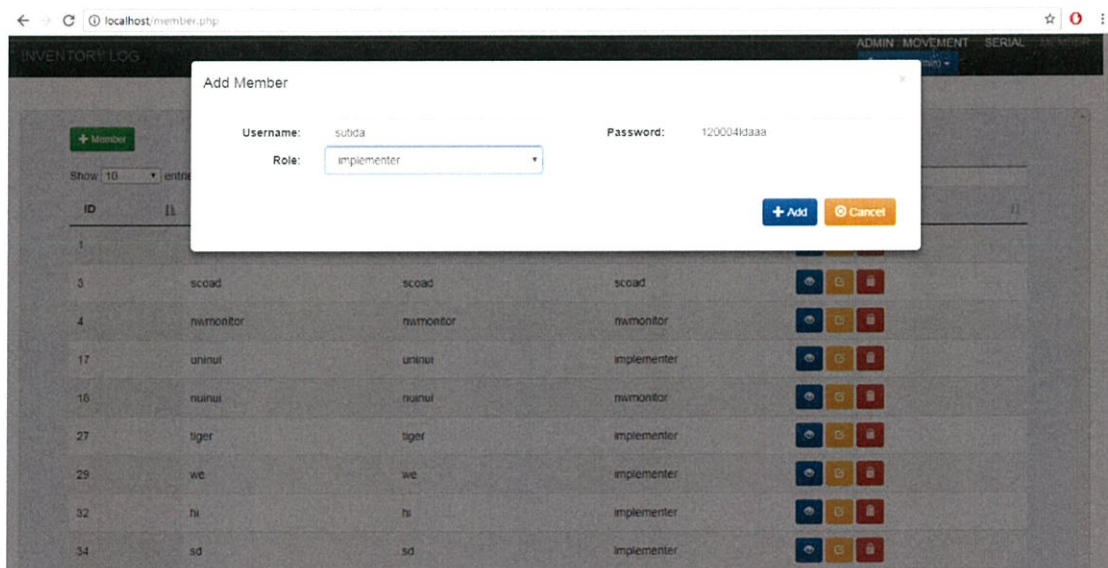
ภาพที่ 4.4 แสดงการค้นหาอุปกรณ์

- serial.php เป็นหน้าที่แสดงอุปกรณ์ที่มีทั้งหมด สามารถ import, export ไฟล์ที่เป็น CSV/Excel ได้ และสามารถเพิ่มข้อมูลไปที่หน้า movement ได้



ภาพที่ 4.5 แสดงการ import excel file ในหน้า serial

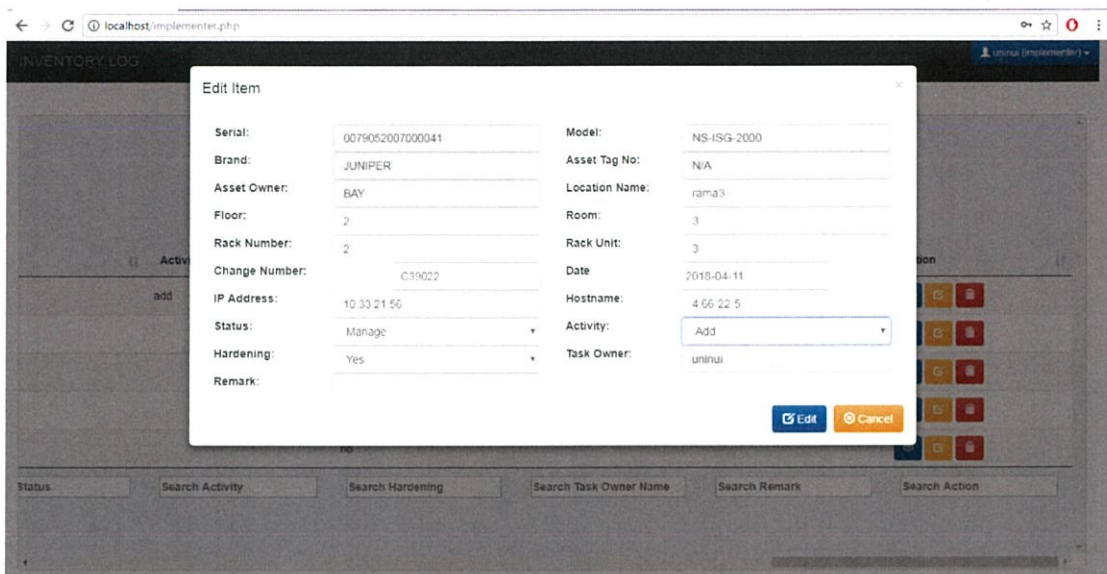
- member.php เป็นหน้าที่แสดงสมาชิกทั้งหมด admin สามารถเพิ่ม/แก้ไขข้อมูลของสมาชิกได้



ภาพที่ 4.6 แสดงการเพิ่มสมาชิกใหม่

## 4.2.2 Implementer

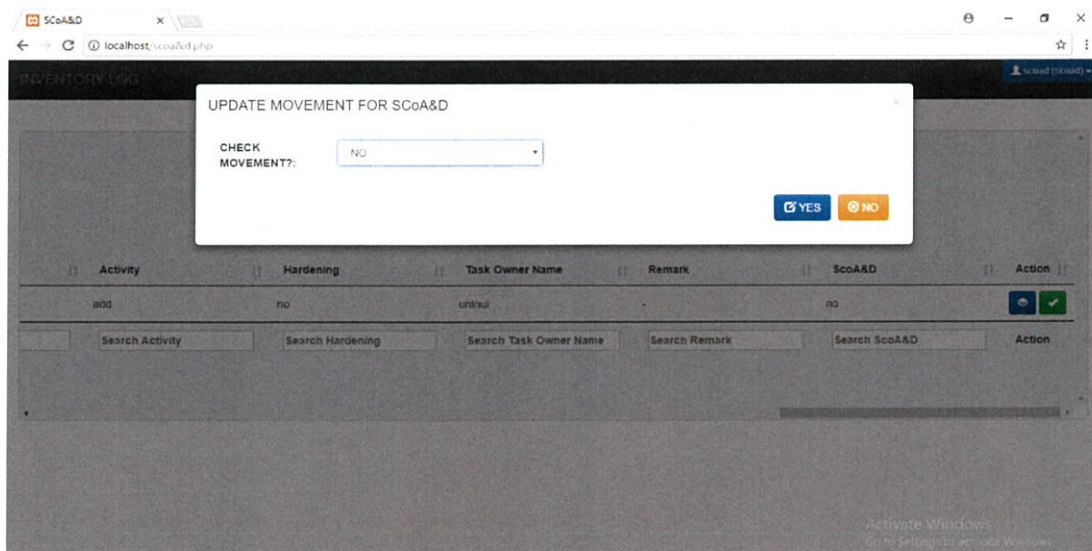
Implementer มีหน้าที่แก้ไขข้อมูลจากอุปกรณ์ที่ admin เพิ่มเข้ามา



ภาพที่ 4.7 แสดงการแก้ไขข้อมูลของ implementer

## 4.2.3 ScoA&D

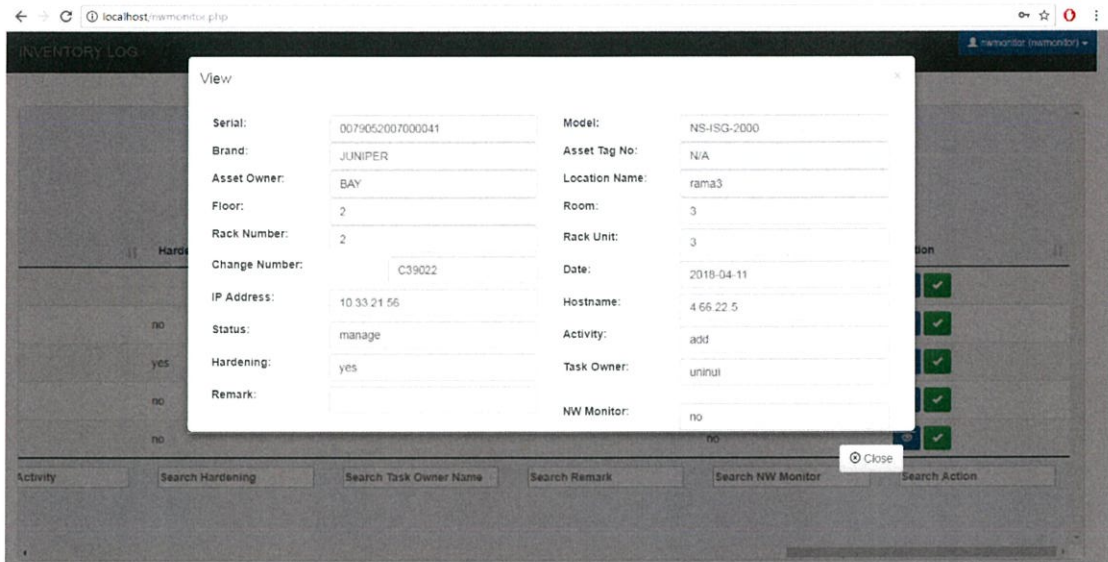
มีหน้าที่ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ แต่ไม่สามารถเข้ามาแก้ไขข้อมูลเองได้



ภาพที่ 4.8 แสดงการตรวจสอบข้อมูลของ ScoA&D

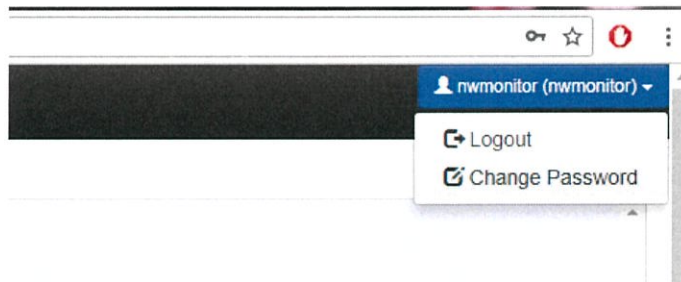
#### 4.2.4 Network Monitor

มีหน้าที่ตรวจสอบข้อมูลว่าถูกต้องหรือไม่ แต่ไม่สามารถเข้ามาแก้ไขข้อมูลเองได้

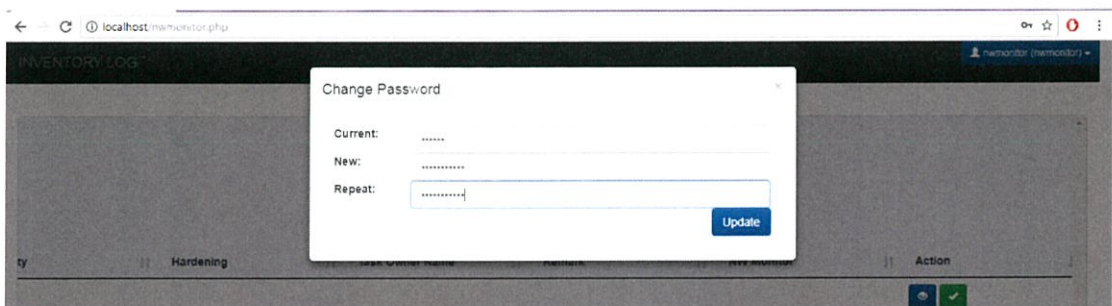


ภาพที่ 4.9 แสดง Network monitor ไม่สามารถแก้ไขข้อมูลได้

#### 4.3 การเปลี่ยนรหัส



ภาพที่ 4.10 แสดงปุ่ม change password



ภาพที่ 4.11 แสดงการ change password

## บทที่ 5

### สรุปผลดำเนินการ และข้อเสนอแนะ

การทำเว็บไซต์เก็บข้อมูลสินค้าคงคลังเพื่อมาแทนที่การเก็บข้อมูลใน Microsoft Excel สามารถสรุปผลการดำเนินการและข้อเสนอแนะได้ดังนี้

#### 5.1 การดำเนินการจัดทำโครงการ

##### 5.1.1 วัตถุประสงค์ของโครงการ

- เพื่อความสะดวกสบายในการค้นหาข้อมูล
- เพื่อความถูกต้องแม่นยำของข้อมูลลดปัญหาข้อมูลไม่อัปเดต
- เพื่อเพิ่มประสิทธิภาพในการทำงานให้ดีขึ้น ช่วยลดเวลาในการทำงาน

##### 5.1.2 วัสดุ อุปกรณ์ เครื่องมือหรือโปรแกรมที่ใช้ในการพัฒนา

- คอมพิวเตอร์พร้อมเชื่อมต่ออินเทอร์เน็ต
- ซอฟต์แวร์ ได้แก่ XAMPP, MySQL, Sublime text 2

#### 5.2 สรุปผลการดำเนินโครงการ

เว็บไซต์เก็บข้อมูลสินค้าคงคลังนี้ช่วยเพื่อความสะดวกสบายในการตรวจสอบจำนวนสินค้า และตำแหน่งของสินค้าให้ถูกต้องแม่นยำมากขึ้น และช่วยเพิ่มประสิทธิภาพในการทำงาน โดยที่ทุกคนในองค์กรสามารถเข้าถึงข้อมูลได้ตลอดเวลา ทำให้มีความรวดเร็วในการค้นหาและตรวจสอบอุปกรณ์

นอกจากนี้ยังช่วยลดปัญหาข้อมูลไม่อัปเดต ซึ่งในอดีตที่เก็บข้อมูลลงในโปรแกรม Microsoft Excel นั้นเกิดปัญหาคือเมื่อมีการเคลื่อนย้ายหรือเพิ่มอุปกรณ์ลงในคลังสินค้าแล้ว ไม่ได้แก้ไขข้อมูลลงในไฟล์รวม ทำให้เกิดความผิดพลาดของข้อมูลได้ จึงเกิดเป็นระบบเว็บไซต์ซึ่งง่ายต่อการแก้ไขข้อมูล โดยการกดแก้ไขเพียงครั้งเดียวข้อมูลก็จะอัปเดตไปในทุกๆฐานข้อมูล

และยังสามารถเข้าถึงข้อมูลได้ง่าย เนื่องด้วยการเชื่อมต่อทางอินเทอร์เน็ตเป็นสิ่งที่สามารถเชื่อมทุกคนจาก ทุกมุมโลกให้สามารถติดต่อสื่อสารกันผ่านเครือข่ายเน็ตเวิร์ก การมีเว็บไซต์เพื่อเก็บข้อมูลสินค้าคงคลังจึงเป็นการเพิ่มความสะดวกสบายให้กับผู้ใช้ เพราะไม่จำเป็นต้องเข้าไปในคลังสินค้าเองให้เสียเวลา

### 5.3 ข้อเสนอแนะและปัญหาอุปสรรค

เว็บไซต์นี้เหมาะสำหรับการค้นหาตำแหน่งของอุปกรณ์ที่ต้องการ และการอัปเดตข้อมูลของอุปกรณ์ว่าอยู่ที่ใด ซึ่งถ้ามีข้อมูลอยู่จำนวนมากจะใช้เวลาค่อนข้างนานในการอัปเดตข้อมูล หรือแสดงผลข้อมูล

## เอกสารอ้างอิง

- [1] “Download XAMPP”[ออนไลน์]. เข้าถึงได้จาก:  
<https://www.apachefriends.org/download.html>
- [2] “Download phpmyadmin”[ออนไลน์]. เข้าถึงได้จาก:  
<https://www.phpmyadmin.net/downloads/>
- [3] “Download Sublime Text”[ออนไลน์]. เข้าถึงได้จาก :  
<https://www.sublimetext.com/2>