

## บทที่ 2

### ทฤษฎีและระบบงานที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

##### 2.1.1 RESTful API คืออะไร

RESTful API [1] เป็นอินเทอร์เฟซที่ระบบคอมพิวเตอร์สองระบบใช้เพื่อแลกเปลี่ยนข้อมูลผ่านอินเทอร์เน็ตได้อย่างปลอดภัย แอปพลิเคชันทางธุรกิจส่วนใหญ่ต้องสื่อสารกับแอปพลิเคชันภายในอื่นๆ และของบุคคลที่สามเพื่อทำงานต่างๆ ตัวอย่างเช่น หากต้องการสร้างสลิปเงินเดือน ระบบบัญชีภายในของคุณต้องแบ่งปันข้อมูลกับระบบธนาคารของลูกค้าเพื่อออกไปแจ้งหนี้และสื่อสารกับแอปพลิเคชัน บันทึกเวลาปฏิบัติงานภายในโดยอัตโนมัติ RESTful API ให้การสนับสนุนการแลกเปลี่ยนข้อมูลนี้เพราะเป็นระบบที่มีมาตรฐานการสื่อสารระหว่างซอฟต์แวร์ที่ปลอดภัย เสถียร และมีประสิทธิภาพ

API คือ ส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface หรือ API) กำหนดกฎที่ต้องปฏิบัติตามเพื่อสื่อสารกับระบบซอฟต์แวร์อื่น โดยนักพัฒนาเปิดเผยหรือสร้าง API เพื่อให้แอปพลิเคชันอื่นสามารถสื่อสารกับแอปพลิเคชันของตนได้ทางโปรแกรม ตัวอย่างเช่น แอปพลิเคชันบันทึกเวลาปฏิบัติงานแสดง API ที่ขอชื่อเต็มของพนักงานและช่วงวันที่ เมื่อได้รับข้อมูลนี้แล้ว ระบบจะประมวลผลบันทึกเวลาปฏิบัติงานของพนักงานเป็นการภายใน และส่งกลับจำนวนชั่วโมงที่ทำงานในช่วงวันที่ดังกล่าวทั้งนี้คุณสามารถมองได้ว่า API เว็บเป็นเกตเวย์ระหว่างไคลเอ็นต์และทรัพยากรบนเว็บ

ไคลเอ็นต์คือผู้ใช้ที่ต้องการเข้าถึงข้อมูลจากเว็บ โดยไคลเอ็นต์อาจเป็นบุคคลหรือระบบซอฟต์แวร์ที่ใช้ API ก็ได้ ตัวอย่างเช่น นักพัฒนาสามารถเขียนโปรแกรมที่เข้าถึงข้อมูลสภาพอากาศจากระบบสภาพอากาศ หรือคุณสามารถเข้าถึงข้อมูลเดียวกันจากเบราว์เซอร์เมื่อคุณเยี่ยมชมเว็บไซต์รายงานสภาพอากาศได้โดยตรง

ทรัพยากรคือข้อมูลที่แอปพลิเคชันต่างๆ มอบให้แก่ไคลเอ็นต์ โดยทรัพยากรอาจเป็นรูปภาพ วิดีโอ ข้อความ ตัวเลข หรือข้อมูลประเภทใดก็ได้ ทั้งนี้เครื่องคอมพิวเตอร์ที่มอบทรัพยากรให้แก่ไคลเอ็นต์นั้นเรียกอีกอย่างว่าเซิร์ฟเวอร์ องค์กรต่างๆ ใช้ API เพื่อแบ่งปันทรัพยากรและให้บริการเว็บในขณะที่ยังคงดูแลรักษาความปลอดภัย การควบคุม และการรับรองความถูกต้องไปพร้อมกัน นอกจากนี้ API ยังช่วยให้ลูกค้าระบุได้ว่าไคลเอ็นต์ใดสามารถเข้าถึงทรัพยากรภายในที่เฉพาะเจาะจงได้

Representational State Transfer (REST) เป็นสถาปัตยกรรมซอฟต์แวร์ที่กำหนดเงื่อนไขว่า API ควรทำงานอย่างไร โดยแต่แรกเริ่มนั้น มีการสร้าง REST ขึ้นเพื่อเป็นแนวทางในการจัดการการสื่อสารบนเครือข่ายที่ซับซ้อน เช่น อินเทอร์เน็ต คุณสามารถใช้สถาปัตยกรรม REST เพื่อรองรับการสื่อสารที่มี

ประสิทธิภาพสูงและเชื่อถือได้ในทุกระดับ คุณยังสามารถใช้และปรับเปลี่ยนสถาปัตยกรรมได้อย่างง่ายดาย โดยนำความสามารถในการมองเห็นและการเคลื่อนย้ายข้ามแพลตฟอร์มมาสู่ทุกระบบ API

นักพัฒนา API สามารถออกแบบ API ได้โดยใช้สถาปัตยกรรมต่างๆ โดย API ที่เป็นไปตามรูปแบบสถาปัตยกรรม REST เรียกว่า REST API บริการเว็บที่ใช้สถาปัตยกรรม REST เรียกว่าบริการเว็บ RESTful คำว่า RESTful API โดยทั่วไปหมายถึง API เว็บแบบ RESTful อย่างไรก็ตาม คุณสามารถใช้คำว่า REST API และ RESTful API แทนกันได้ โดยหลักการบางประการของรูปแบบสถาปัตยกรรม REST มีดังต่อไปนี้:

อินเทอร์เน็ตเฟิร์มแวร์แบบเดียวกันถือเป็นพื้นฐานในการออกแบบบริการเว็บ RESTful ทุกประเภท ซึ่งระบุว่าเซิร์ฟเวอร์ถ่ายโอนข้อมูลในรูปแบบมาตรฐาน ทรัพยากรที่จัดรูปแบบเรียกว่าการแทนข้อมูลใน REST โดยรูปแบบนี้อาจแตกต่างจากการแทนข้อมูลภายในของทรัพยากรบนแอปพลิเคชันเซิร์ฟเวอร์ ตัวอย่างเช่น เซิร์ฟเวอร์สามารถจัดเก็บข้อมูลเป็นข้อความ แต่ส่งข้อมูลในรูปแบบการแทนข้อมูลด้วย HTML โดยอินเทอร์เน็ตเฟิร์มแวร์แบบเดียวกันกำหนดข้อจำกัดทางสถาปัตยกรรมไว้ 4 ประการ ได้แก่

1. คำขอควรระบุทรัพยากร ซึ่งสามารถทำเช่นนั้นได้โดยใช้ตัวระบุทรัพยากรรูปแบบเดียวกัน
2. โคลเอ็นต์มีข้อมูลเพียงพอในการแทนข้อมูลทรัพยากรเพื่อแก้ไขหรือลบทรัพยากรดังกล่าวหากต้องการ เซิร์ฟเวอร์เป็นไปตามเงื่อนไขนี้โดยการส่งข้อมูลเมตาที่อธิบายทรัพยากรเพิ่มเติม
3. โคลเอ็นต์ได้รับข้อมูลเกี่ยวกับวิธีการประมวลผลการแทนข้อมูลเพิ่มเติม เซิร์ฟเวอร์ดำเนินการเช่นนี้ได้โดยการส่งข้อความอธิบายตนเองที่มีข้อมูลเมตาเกี่ยวกับวิธีการที่โคลเอ็นต์สามารถใช้งานข้อมูลดังกล่าวได้ดีที่สุด
4. โคลเอ็นต์ได้รับข้อมูลเกี่ยวกับทรัพยากรอื่นๆ ที่เกี่ยวข้องทั้งหมดที่จำเป็นสำหรับการทำงานให้เสร็จสมบูรณ์ เซิร์ฟเวอร์ดำเนินการเช่นนี้ได้โดยการส่งไฮเปอร์ลิงก์ในการแทนข้อมูลเพื่อให้ลูกค้าสามารถค้นพบทรัพยากรเพิ่มเติมได้แบบไดนามิก

#### ความไร้สถานะ

ในสถาปัตยกรรม REST ความไร้สถานะหมายถึงวิธีการสื่อสารที่เซิร์ฟเวอร์ดำเนินการตามคำขอของโคลเอ็นต์ทั้งหมดโดยไม่ขึ้นกับคำขอก่อนหน้าทั้งหมด โดยโคลเอ็นต์สามารถร้องขอทรัพยากรในลำดับใดก็ได้ และทุกคำขอจะไร้สถานะหรือแยกออกจากคำขออื่นๆ ข้อจำกัดในการออกแบบ REST API นี้บ่งบอกว่าเซิร์ฟเวอร์สามารถเข้าใจและดำเนินการตามคำขอได้อย่างสมบูรณ์ทุกครั้ง

#### ระบบที่แบ่งออกเป็นชั้น

ในสถาปัตยกรรมระบบที่แบ่งออกเป็นชั้น โคลเอ็นต์สามารถเชื่อมต่อกับตัวกลางอื่นๆ ที่ได้รับอนุญาตระหว่างโคลเอ็นต์และเซิร์ฟเวอร์ได้ และจะยังคงได้รับการตอบสนองจากเซิร์ฟเวอร์ เซิร์ฟเวอร์ยัง

สามารถส่งต่อคำขอไปยังเซิร์ฟเวอร์อื่นได้อีกด้วย คุณสามารถออกแบบบริการเว็บ RESTful ให้ทำงานบนเซิร์ฟเวอร์หลายตัวที่มีหลายชั้นได้ เช่น ความปลอดภัย แอปพลิเคชัน และตรรกะทางธุรกิจ โดยทำงานร่วมกันเพื่อตอบสนองคำขอของไคลเอ็นต์ โดยไคลเอ็นต์ไม่สามารถมองเห็นชั้นต่างๆ เหล่านี้ได้

#### ความสามารถในการแคช

บริการเว็บ RESTful รองรับการแคช ซึ่งเป็นกระบวนการจัดเก็บการตอบสนองบางส่วนบนไคลเอ็นต์หรือตัวกลางเพื่อปรับปรุงเวลาตอบสนองของเซิร์ฟเวอร์ ตัวอย่างเช่น สมมติว่าคุณเยี่ยมชมเว็บไซต์ที่มีรูปภาพส่วนหัวและส่วนท้ายทั่วไปในทุกหน้า ทุกครั้งที่คุณเยี่ยมชมหน้าเว็บไซต์ใหม่ เซิร์ฟเวอร์จะต้องส่งภาพเดิมอีกครั้ง เพื่อหลีกเลี่ยงปัญหานี้ ไคลเอ็นต์จะแคชหรือจัดเก็บรูปภาพเหล่านี้ หลังจากการตอบสนองครั้งแรก จากนั้นจึงใช้รูปภาพโดยตรงจากแคช ทั้งนี้บริการเว็บ RESTful ควบคุมการแคชโดยใช้การตอบสนอง API ที่ระบุว่าสามารถแคชได้หรือไม่สามารถแคชได้

#### การปรับแต่งโค้ดได้ตามความต้องการ

ในรูปแบบสถาปัตยกรรม REST เซิร์ฟเวอร์สามารถขยายหรือปรับแต่งฟังก์ชันการทำงานของไคลเอ็นต์ได้ชั่วคราวโดยการถ่ายโอนโค้ดการเขียนโปรแกรมซอฟต์แวร์ไปยังไคลเอ็นต์ ตัวอย่างเช่น เมื่อคุณรอกแบบฟอร์มลงทะเบียนบนเว็บไซต์ เบราวเซอร์ของคุณจะเน้นย้ำให้เห็นถึงข้อผิดพลาดของคุณทันที เช่น หมายเลขโทรศัพท์ที่ไม่ถูกต้อง ซึ่งระบบสามารถทำเช่นนี้ได้เนื่องจากโค้ดที่เซิร์ฟเวอร์ส่งไป

RESTful API มีข้อดีต่างๆ ดังต่อไปนี้:

#### 1. ความสามารถในการปรับขนาด

ระบบที่ใช้ REST API สามารถปรับขนาดได้อย่างมีประสิทธิภาพเนื่องจาก REST ปรับการโต้ตอบระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์ให้เหมาะสม ความไร้สถานะช่วยจัดการโหลดเซิร์ฟเวอร์เนื่องจากเซิร์ฟเวอร์ไม่จำเป็นต้องเก็บข้อมูลคำขอของไคลเอ็นต์ในอดีต การแคชที่มีการจัดการเป็นอย่างดีบางส่วนหรือทั้งหมดจะช่วยลดการโต้ตอบระหว่างไคลเอ็นต์กับเซิร์ฟเวอร์บางส่วน คุณสมบัติเหล่านี้ทั้งหมดจะสนับสนุนความสามารถในการปรับขนาดโดยไม่ทำให้เกิดปัญหาคอขวดในการสื่อสารซึ่งลดประสิทธิภาพการทำงาน

#### 2. ความยืดหยุ่น

บริการเว็บ RESTful รองรับการแยกไคลเอ็นต์และเซิร์ฟเวอร์โดยสมบูรณ์ ซึ่งลดความซับซ้อนและแยกส่วนประกอบเซิร์ฟเวอร์ต่างๆ เพื่อให้แต่ละส่วนสามารถพัฒนาได้อย่างอิสระ ทั้งนี้การเปลี่ยนแปลงแพลตฟอร์มหรือเทคโนโลยีที่แอปพลิเคชันเซิร์ฟเวอร์ไม่ส่งผลกระทบต่อแอปพลิเคชันไคลเอ็นต์

ความสามารถในการแบ่งชั้นการทำงานของแอปพลิเคชันช่วยเพิ่มความยืดหยุ่นยิ่งขึ้นไปอีก ตัวอย่างเช่น นักพัฒนาสามารถเปลี่ยนแปลงชั้นฐานข้อมูลได้โดยไม่ต้องเขียนตรรกะของแอปพลิเคชันชั้นใหม่

### 3. ความไม่ขึ้นกับระบบ

REST API ไม่ขึ้นอยู่กับเทคโนโลยีที่ใช้ คุณจึงสามารถเขียนแอปพลิเคชันไคลเอ็นต์และเซิร์ฟเวอร์ในภาษาการเขียนโปรแกรมต่างๆ ได้โดยไม่กระทบต่อการออกแบบ API นอกจากนี้ คุณยังสามารถเปลี่ยนเทคโนโลยีพื้นฐานในทั้งสองฝั่งได้โดยไม่มีผลกระทบต่อการสื่อสารอีกด้วย

### 4. RESTful API ทำงานอย่างไร

ฟังก์ชันพื้นฐานของ RESTful API จะเหมือนกับการท่องอินเทอร์เน็ต ไคลเอ็นต์จะติดต่อกับเซิร์ฟเวอร์โดยใช้ API เมื่อต้องใช้ทรัพยากร นักพัฒนา API อธิบายวิธีการที่ไคลเอ็นต์ควรใช้ REST API ในเอกสารประกอบ API ของแอปพลิเคชันเซิร์ฟเวอร์ โดยการเรียกใช้ REST API มีขั้นตอนทั่วไปดังนี้:

1. ไคลเอ็นต์ส่งคำขอไปยังเซิร์ฟเวอร์ ไคลเอ็นต์ปฏิบัติตามเอกสารประกอบ API เพื่อจัดรูปแบบคำขอในลักษณะที่เซิร์ฟเวอร์เข้าใจได้
2. เซิร์ฟเวอร์รับรองความถูกต้องของไคลเอ็นต์ และยืนยันว่าไคลเอ็นต์มีสิทธิ์ส่งคำขอดังกล่าว
3. เซิร์ฟเวอร์รับคำขอและประมวลผลเป็นการภายใน
4. เซิร์ฟเวอร์ส่งคืนการตอบสนองกลับไปยังไคลเอ็นต์ การตอบสนองมีข้อมูลที่บอกให้ลูกค้าทราบว่าคำขอดังกล่าวสำเร็จหรือไม่ การตอบสนองยังรวมถึงข้อมูลใดๆ ที่ไคลเอ็นต์ร้องขออีกด้วย

รายละเอียดคำขอและการตอบสนอง REST API จะแตกต่างกันเล็กน้อยโดยขึ้นอยู่กับวิธีการที่นักพัฒนา API ออกแบบ API

คำขอของไคลเอ็นต์ RESTful API มี RESTful API กำหนดให้คำขอมีส่วนประกอบหลักดังต่อไปนี้:

ตัวระบุทรัพยากรที่ไม่ซ้ำกัน

เซิร์ฟเวอร์ระบุทรัพยากรแต่ละรายการด้วยตัวระบุทรัพยากรที่ไม่ซ้ำกัน สำหรับบริการ REST โดยปกติแล้วเซิร์ฟเวอร์จะดำเนินการระบุทรัพยากรโดยใช้ตัวชี้แหล่งในอินเทอร์เน็ต (Uniform Resource Locator หรือ URL) โดย URL ระบุเส้นทางไปยังทรัพยากร ทั้งนี้ URL คล้ายคลึงกับที่อยู่เว็บไซต์ที่คุณป้อนลงในเบราว์เซอร์เพื่อเข้าชมหน้าเว็บต่างๆ นอกจากนี้ URL ยังเรียกอีกอย่างว่าตำแหน่งข้อมูลคำขอ และระบุอย่างชัดเจนต่อเซิร์ฟเวอร์ถึงสิ่งที่ลูกค้าต้องการ

วิธีการ

นักพัฒนามักใช้ RESTful API โดยใช้เกณฑ์วิธีขนส่งข้อความหลายมิติ (Hypertext Transfer Protocol หรือ HTTP) วิธีการ HTTP จะบอกให้เซิร์ฟเวอร์ทราบถึงสิ่งที่ต้องทำกับทรัพยากร โดยวิธีการ HTTP ทั่วไปมี 4 วิธีดังต่อไปนี้:



ภาพประกอบที่ 2.1 อธิบายหลักการของ RESTful API

ที่มา: [https://www.kodefin.com/storage/gallery/363468424\\_1609800313\\_1610436610.jpg](https://www.kodefin.com/storage/gallery/363468424_1609800313_1610436610.jpg)

### GET

ไคลเอ็นต์ใช้ GET เพื่อเข้าถึงทรัพยากรที่อยู่ URL ที่ระบุบนเซิร์ฟเวอร์ ซึ่งสามารถแคชค่าขอ GET และส่งพารามิเตอร์ในคำขอ RESTful API เพื่อสั่งให้เซิร์ฟเวอร์กรองข้อมูลก่อนส่ง

### POST

ไคลเอ็นต์ใช้ POST เพื่อส่งข้อมูลไปยังเซิร์ฟเวอร์ ซึ่งรวมถึงการแทนข้อมูลพร้อมกับคำขอ การส่งคำขอ POST เดียวกันหลายครั้งมีผลข้างเคียงเหมือนกับการสร้างทรัพยากรเดียวกันหลายครั้ง

### PUT

ไคลเอ็นต์ใช้ PUT เพื่ออัปเดตทรัพยากรที่มีอยู่บนเซิร์ฟเวอร์ การส่งคำขอ PUT เดียวกันหลายครั้งในบริการเว็บ RESTful จะให้ผลลัพธ์เหมือนกัน ซึ่งแตกต่างจาก POST

### DELETE

ไคลเอ็นต์ใช้คำขอ DELETE เพื่อลบทรัพยากรออก โดยคำขอ DELETE สามารถเปลี่ยนสถานะเซิร์ฟเวอร์ได้ อย่างไรก็ตาม หากผู้ใช้ไม่มีการรับรองความถูกต้องที่เหมาะสม คำขอจะล้มเหลว

### ส่วนหัว HTTP

ส่วนหัวของคำขอคือข้อมูลเมตาที่แลกเปลี่ยนระหว่างไคลเอ็นต์และเซิร์ฟเวอร์ ตัวอย่างเช่น ส่วนหัวของคำขอจะระบุรูปแบบของคำขอและการตอบกลับ ให้ข้อมูลเกี่ยวกับสถานะคำขอ และอื่นๆ

### ข้อมูล

คำขอ REST API อาจรวมถึงข้อมูลสำหรับวิธีการ POST, PUT และ HTTP อื่นๆ เพื่อให้ทำงานได้สำเร็จ

### พารามิเตอร์

คำขอ RESTful API อาจรวมถึงพารามิเตอร์ที่ให้รายละเอียดเพิ่มเติมกับเซิร์ฟเวอร์เกี่ยวกับสิ่งที่ต้องดำเนินการ โดยพารามิเตอร์ประเภทต่างๆ มีดังต่อไปนี้:

- พารามิเตอร์พาทที่ระบุรายละเอียด URL
- พารามิเตอร์การสอบถามที่ขอข้อมูลเพิ่มเติมเกี่ยวกับทรัพยากร

- พารามิเตอร์คูกี้ที่รับรองความถูกต้องของไคลเอ็นต์ได้อย่างรวดเร็ว

#### วิธีการรับรองความถูกต้อง RESTful API

บริการเว็บ RESTful ต้องรับรองความถูกต้องของคำขอก่อนที่จะส่งการตอบสนอง การรับรองความถูกต้องเป็นกระบวนการยืนยันตัวตน ตัวอย่างเช่น คุณสามารถพิสูจน์ตัวตนของคุณได้โดยแสดงบัตรประจำตัวประชาชนหรือใบขับขี่ ในทำนองเดียวกัน ไคลเอ็นต์บริการ RESTful เองก็ต้องพิสูจน์ตัวตนต่อเซิร์ฟเวอร์เพื่อสร้างความไว้วางใจ โดย RESTful API มีวิธีการรับรองความถูกต้องทั่วไป 4 วิธี ได้แก่

HTTP กำหนดรูปแบบการรับรองความถูกต้องบางส่วนที่คุณสามารถใช้ได้โดยตรงเมื่อคุณนำ REST API ไปใช้ โดยทั้งสองรูปแบบมีดังต่อไปนี้:

#### การรับรองความถูกต้องพื้นฐาน

ในการรับรองความถูกต้องพื้นฐาน ไคลเอ็นต์จะส่งชื่อผู้ใช้และรหัสผ่านในส่วนหัวของคำขอ ซึ่งจะเข้ารหัสด้วย base64 ซึ่งเป็นเทคนิคการเข้ารหัสที่แปลงทั้งคู่เป็นชุดอักขระ 64 ตัวเพื่อการรับส่งข้อมูลที่ปลอดภัย

คำว่ารับรองความถูกต้องของแบเรอร์หมายถึงกระบวนการให้การควบคุมการเข้าถึงแก่แบเรอร์ของโทเค็น โดยโทเค็นแบบแบเรอร์มักจะเป็นสตริงอักขระที่เข้ารหัสซึ่งเซิร์ฟเวอร์สร้างขึ้นเพื่อตอบสนองต่อคำขอเข้าสู่ระบบ จากนั้นไคลเอ็นต์จะส่งโทเค็นในส่วนหัวของคำขอเพื่อเข้าถึงทรัพยากร

คีย์ API เป็นอีกตัวเลือกหนึ่งสำหรับการรับรองความถูกต้องของ REST API ด้วยวิธีการนี้ เซิร์ฟเวอร์จะกำหนดค่าที่สร้างขึ้นเฉพาะให้กับไคลเอ็นต์ที่ใช้งานครั้งแรก เมื่อใดก็ตามที่ไคลเอ็นต์พยายามเข้าถึงทรัพยากร ก็จะใช้คีย์ API เฉพาะเพื่อยืนยันตัวตน ทั้งนี้คีย์ API มีความปลอดภัยน้อยกว่าเนื่องจากไคลเอ็นต์ต้องส่งคีย์ ซึ่งทำให้เสี่ยงต่อการถูกขโมยผ่านเครือข่าย

OAuth รวมรหัสผ่านและโทเค็นเข้าด้วยกันเพื่อการเข้าถึงการเข้าสู่ระบบที่มีความปลอดภัยสูงในทุกๆระบบ โดยเซิร์ฟเวอร์จะขอรหัสผ่านก่อนแล้วจึงขอโทเค็นเพิ่มเติมเพื่อให้กระบวนการรับรองความถูกต้องเสร็จสมบูรณ์ ซึ่งสามารถตรวจสอบโทเค็นได้ตลอดเวลาและเมื่อเวลาผ่านไปด้วยขอบเขตและระยะเวลาที่เฉพาะเจาะจง

#### การตอบสนองของเซิร์ฟเวอร์ RESTful API

หลักการ REST กำหนดให้การตอบสนองของเซิร์ฟเวอร์มีองค์ประกอบหลักดังต่อไปนี้:

#### 1. บรรทัดสถานะ

บรรทัดสถานะประกอบด้วยรหัสสถานะสามหลักที่แจ้งว่าคำขอสำเร็จหรือล้มเหลว ตัวอย่างเช่น รหัส 2XX ระบุถึงความสำเร็จ แต่รหัส 4XX และ 5XX ระบุถึงข้อผิดพลาด รหัส 3XX ระบุถึงการเปลี่ยนเส้นทาง URL โดยรหัสสถานะทั่วไปบางส่วนมีดังต่อไปนี้:

- 200: การตอบสนองเพื่อระบุถึงความสำเร็จทั่วไป
- 201: การตอบสนองเพื่อระบุถึงความสำเร็จของวิธีการ POST

- 400: คำขอที่ไม่ถูกต้องที่เซิร์ฟเวอร์ไม่สามารถประมวลผลได้
- 404: ไม่พบทรัพยากร

## 2. เนื้อความ

เนื้อความการตอบสนองประกอบด้วย การแทนข้อมูลทรัพยากร เซิร์ฟเวอร์จะเลือกรูปแบบการแทนข้อมูลที่เหมาะสมตามสิ่งที่อยู่ในส่วนหัวของคำขอ ไคลเอนต์สามารถขอข้อมูลในรูปแบบ XML หรือ JSON ได้ ซึ่งกำหนดวิธีการเขียนข้อมูลในรูปแบบข้อความธรรมดา ตัวอย่างเช่น หากไคลเอนต์ร้องขอชื่อและอายุของบุคคลชื่อว่า John เซิร์ฟเวอร์จะส่งคืนการแทนข้อมูล JSON ดังนี้:

```
{ "name": "John", "age": 30 }
```

## 3. ส่วนหัว

การตอบสนองยังมีส่วนหัวหรือข้อมูลเมตาเกี่ยวกับการตอบสนองอีกด้วย ซึ่งให้บริบทเพิ่มเติมเกี่ยวกับการตอบสนองและมีข้อมูลต่างๆ เช่น เซิร์ฟเวอร์ การเข้ารหัส วันที่ และประเภทเนื้อหา

AWS จะช่วยคุณในการจัดการ RESTful API

Amazon API Gateway เป็นบริการที่มีการจัดการเต็มรูปแบบ ซึ่งทำให้นักพัฒนาสามารถสร้าง เผยแพร่ บำรุงรักษา ฝ้าติดตาม และรักษาความปลอดภัยของ API ในทุกขนาดได้ง่าย การใช้ API Gateway ทำให้คุณสามารถสร้าง RESTful API สำหรับแอปพลิเคชันการสื่อสารสองทางแบบเรียลไทม์ได้ การใช้ API Gateway ทำให้คุณสามารถ:

- มอบประสิทธิภาพความเร็วสูงสำหรับทั้งคำขอและการตอบสนองของ API ให้แก่ผู้ใช้
- อนุมัติการเข้าถึง API ของคุณด้วย AWS Identity and Access Management (IAM) และ Amazon Cognito ซึ่งทั้งสองอย่างนี้รองรับ OAuth แบบเนทีฟ
- เรียกใช้ API เดียวกันได้หลายเวอร์ชันพร้อมกันด้วย API Gateway เพื่อทำซ้ำ ทดสอบ และเผยแพร่เวอร์ชันใหม่ได้อย่างรวดเร็ว
- ตรวจสอบตัวชี้วัดประสิทธิภาพและข้อมูลเกี่ยวกับการเรียก API, เวลาแฝงของข้อมูล และอัตราข้อผิดพลาดจาก API Gateway

### 2.1.2 การติดตั้งและใช้งาน Slim framework 4 เบื้องต้น

เนื้อหานี้จะแนะนำ Slim [2] ซึ่งเป็น PHP framework ขนาดเล็ก ที่จะช่วยพัฒนาเว็บแอปพลิเคชันและ API ได้อย่างง่ายและมีประสิทธิภาพ เนื้อหานี้จะไม่ได้ลงรายละเอียดอะไรมากนัก เป็นเพียงแนวทางการติดตั้งและใช้งานเบื้องต้น หลักๆ การทำงานของ slim ก็คือการรับคอยรับ HTTP request ที่ส่งเข้ามา แล้วกำหนดรูปแบบการจัดการ การทำงานของจากนั้นก็ส่งค่า HTTP response กลับออกมา แสดงถ้าใครได้ดูเนื้อหาเกี่ยวกับ NodeJs หรือการใช้งานเกี่ยวกับ HTTP RequestResponse ก็น่าจะ

คู่เคียงกับรูปแบบการทำงานลักษณะนี้คือ Slim เหมาะสำหรับนำมาใช้ในการพัฒนา API เพื่อจัดการกับข้อมูลเพื่อใช้งานร่วมกับการพัฒนาอื่นๆ ที่สำคัญก็คือ slim เขียนโค้ดน้อยและทำงานได้เร็ว

คุณสมบัติ

HTTP Router

Slim ให้เราเตอร์ที่รวดเร็วและทรงพลังที่แมปเส้นทางการเรียกกลับกับวิธีการร้องขอ HTTP และ URI ที่เฉพาะเจาะจง รองรับพารามิเตอร์และการจับคู่รูปแบบ

Middleware

สร้างแอปพลิเคชันของคุณด้วยมิดเดิลแวร์ที่รวมศูนย์เพื่อปรับแต่งคำขอ HTTP และอ็อบเจกต์การตอบสนองรอบๆ แอป Slim ของคุณ

PSR-7 Support

Slim รองรับการใช้งานข้อความ HTTP PSR-7 ดังนั้นคุณจึงสามารถตรวจสอบและจัดการเมธอดข้อความ HTTP, สถานะ, URI, ส่วนหัว, คุกกี้ และเนื้อความ

Dependency Injection

Slim รองรับ dependency injection ฉีดเพื่อให้คุณสามารถควบคุมเครื่องมือภายนอกได้อย่างสมบูรณ์ ใช้การใช้งาน PSR-11 ContainerInterface ใดๆ

### 2.1.3 Angular คือ อะไร

Angular [7] คือ เฟรมเวิร์ค (framework) สำหรับสร้างแอปพลิเคชันในฝั่งไคลเอ็นในรูปแบบของ HTML, CSS และ JavaScript/TypeScript ซึ่ง TypeScript จะถูก compile ไปเป็น JavaScript

ทำไมต้อง Angular?

หากเราต้องการใช้งาน JavaScript ในเว็บไซต์เราอาจเขียนด้วย JavaScript ปกติ หรืออาจใช้ jQuery เขียน จริงๆ ก็ทำงานได้นะ แต่เมื่อเว็บไซต์มีความซับซ้อนมากยิ่งขึ้นการเขียน JS ปกติ หรือ jQuery นั้นจะมีความยากลำบากในการ Maintain นั่นเอง

นอกจากนั้นยังมี JS Framework อื่นๆ ที่เป็นที่ยอมรับในขณะนี้คือ Angular, React และ Vuejs เป็นต้น

ประโยชน์เมื่อใช้ Angular

1. ให้ Application ที่มีโครงสร้างแบบ Clean Structure หรือ มีความง่ายต่อการ Maintain



2. มี re-usable code ต่างๆ มากมาย เช่น navigation หรือ browser history เป็นต้น
3. ทำให้ application มี test ที่หลากหลาย เช่น automatic test ซึ่งเป็นส่วนหนึ่งใน application

#### 2.1.4 TypeScript คืออะไร

TypeScript [3] ก็เป็นภาษาคอมพิวเตอร์ที่เป็น open-source อีกภาษาหนึ่ง ที่พัฒนาโดยบริษัทไมโครซอฟต์ เป็นภาษาที่เป็นแบบ High-Level และแตกต่างจากภาษาอื่นๆ ที่จะ Compile เป็น Machine Code แต่ TypeScript compile เป็นภาษา Javascript

Typescript ก็คือภาษา JavaScript ใน Version ที่ได้รับการ Upgrade นั้นเอง มันสามารถทำงานบน Node.js Environment หรือ Web Browser ต่าง ๆ ที่มีการรองรับ ECMAScript 3 ขึ้นไป TypeScript เป็น Statically Compiled Language ที่ได้จัดเตรียมทั้ง Static Typing, Classes และ Interface ไว้ให้แล้ว ช่วยให้คุณสามารถเขียน Code ของ JavaScript ที่เรียบง่ายและ Clean ได้อย่างสะดวกขึ้น ดังนั้น การใช้ TypeScript จะช่วยให้คุณสร้าง Software ที่ปรับใช้งานได้ง่ายและมีประสิทธิภาพมากยิ่งขึ้น

Javascript เป็นภาษาที่มี Performance ดีมาก อีกทั้งสามารถใช้ได้ทั้ง Client Side และ Server Side แต่ภาษา Javascript ยังมีปัญหาเรื่องคุณภาพของโค้ดอยู่ โดยทั้ง Google, Microsoft, Facebook ก็ได้มีการคุยกันและสรุปได้ว่าคอมไพเลอร์สามารถตรวจจับข้อผิดพลาดได้ดีกว่าที่จะให้สิ่งต่าง ๆ ที่ Error ในขณะรันไทม์

Javascript เป็นภาษาที่เป็น Interpreted Language ที่ไม่ต้องทำการ Compile ก่อน Run เช่นเดียวกับกับ Python , PHP อื่นๆ

Javascript เป็นภาษาที่เป็น Dynamic Typed คือประกาศตัวแปรโดยไม่ต้องบอก Type ว่าเป็น Int Float String แต่ถ้าเป็นใช้เป็น Static Typed ก็จะได้ดีกว่าเพราะ Argument ทุกตัวของ Method ต้องมีการบอกว่าเป็น Int Float String ทำให้คนที่เรียกใช้รู้ว่าต้องส่งค่าแบบไหนเข้าไปเพราะ Static variable ถือเป็นคู่มือที่ดีที่สุดในการเขียนโปรแกรมอีกอันหนึ่งเลยก็ว่าได้

#### ข้อดีของ TypeScript

- TypeScript มี Feature ที่มีประโยชน์อย่าง Code Navigation, Bug Prevention และ Maintainability of Code ไว้ให้แล้ว
- มันรองรับ Static Type Annotation หรือ Static Typing
- นอกจากนี้ยังรองรับ Interfaces, Sub-Interfaces, Classes, Subclasses และ ECMAScript 6 หรือ ES6

- มันมาพร้อมกับความสามารถใน Object-Oriented Programming พร้อมทั้งการ Inheritance ของ Private Members และ Interfaces
- มี IDE ให้ใช้อยู่มากมาย <https://medium.com/@nattaponra>

### 2.1.5 Visual Studio Code

Visual Studio Code หรือ VSCode [5] เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ

ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์มรองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย ไม่ว่าจะเป็น

1. การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go
2. Themes
3. Debugger
4. Commands เป็นต้น

### 2.1.6 WYSIWYG (What You See Is What You Get) คืออะไร

WYSIWYG [4] ย่อมาจาก What You See Is What You Get (วอท ยู ซี อีส วอท ยู เก็ท) แปลว่าคุณเห็นอย่างไรคุณก็ได้รับอย่างนั้นคือลักษณะของเอกสารและภาพกราฟิกที่คุณเห็นใน จอคอมพิวเตอร์

เมื่อพิมพ์ออกมาแล้วลักษณะของเอกสารหรือภาพกราฟิกที่คุณได้ก็จะเหมือนกับภาพที่ปรากฏในจอคอมพิวเตอร์ ถือเป็นสมรรถนะอย่างหนึ่งของโปรแกรมประเภทประมวลผลคำ หรือโปรแกรมการจัดพิมพ์ ซึ่งสามารถแสดงตัวอย่างเอกสารก่อนการพิมพ์ และสามารถทำให้เอกสารที่พิมพ์ออกมามีลักษณะเหมือนในจอคอมพิวเตอร์ นั่นคือ ถ้ามองเห็นบนจอภาพอย่างไร สิ่งพิมพ์ออกมา ก็จะได้ผลอย่าง ที่เห็นนั้น

ตัวอย่างเช่น ในการสร้างและออกแบบ web site หรือ web page มักจะใช้โปรแกรมประเภท Web Design ซึ่งมีคุณสมบัติในการใช้งานในแบบ WYSIWYG เพื่อให้การทำ web site หรือ web page นั้น ทำมาอย่างไร ก็แสดงผลอย่างนั้น ไม่ผิดเพี้ยนไป อีกทั้งยังง่ายต่อการใช้ เพราะผู้ใช้ไม่จำเป็นต้องมา

นั่งเขียนโค้ด HTML เอง แบบว่าไม่มีความรู้เรียกการเขียนโปรแกรมก็ใช้ได้ เช่น โปรแกรม Dreamweaver , WYSIWYG Web Builder, FrontPage และ XSitePro เป็นต้น

### 2.1.7 PHP

PHP [8] ย่อมาจาก PHP Hypertext Preprocessor แต่เดิมย่อมาจาก Personal Home Page Tools PHP คือภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่งตัวอย่างของภาษาสคริปต์ก็ เช่น JavaScript , Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language นั่นคือในทุกๆ ครั้งก่อนที่เครื่องคอมพิวเตอร์ซึ่งให้บริการเป็น Web server จะส่งหน้าเว็บเพจที่เขียนด้วย PHP ให้เรา มันจะทำการประมวลผลตามคำสั่งที่มีอยู่ให้เสร็จก่อน แล้วจึงค่อยส่งผลลัพธ์ที่ได้ให้เรา ผลลัพธ์ที่ได้นั้นก็คือเว็บเพจที่เราเห็นนั่นเองถือว่า PHP เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้าง Dynamic Web pages (เว็บเพจที่มีการโต้ตอบกับผู้ใช้) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

PHP เป็นผลงานที่เติบโตมาจากกลุ่มของนักพัฒนาในเชิงเปิดเผยรหัสต้นฉบับ หรือ OpenSource ดังนั้น PHP จึงมีการพัฒนาไปอย่างรวดเร็ว และแพร่หลายโดยเฉพาะอย่างยิ่งเมื่อใช้ร่วมกับ Apache Web server ระบบปฏิบัติการอย่างเช่น Linux หรือ FreeBSD เป็นต้น ในปัจจุบัน PHP สามารถใช้ร่วมกับ Web Server หลากๆตัวบนระบบปฏิบัติการอย่างเช่น Windows 95/98/NT เป็นต้น

ลักษณะเด่นของ PHP

1. ใช้ได้ฟรี
2. PHP เป็นโปรแกรมวิ่งข้าง Sever ดังนั้นขีดความสามารถไม่จำกัด
3. Conlatfun นั่นคือPHP วิ่งบนเครื่อง UNIX,Linux,Windows ได้หมด
4. เรียนรู้ง่าย เนื่องจาก PHP ผั่งเข้าไปใน HTML และใช้โครงสร้างและไวยากรณ์ภาษาง่ายๆ
5. เร็วและมีประสิทธิภาพ โดยเฉพาะเมื่อใช้กับ Apach Xerve เพราะไม่ต้องใช้โปรแกรมจากภายนอก
6. ใช้ร่วมกับ XML ได้ทันที
7. ใช้กับระบบเพิ่มข้อมูลได้
8. ใช้กับข้อมูลตัวอักษรได้อย่างมีประสิทธิภาพ
9. ใช้กับโครงสร้างข้อมูล แบบ Scalar,Array,Associative array
10. ใช้กับการประมวลผลภาพได้

เว็บไซต์ต่างๆได้มีการพัฒนาในด้านต่างๆ เช่น การบริการข่าวสารข้อมูลที่ทันสมัย เป็นสื่อกลางในการติดต่อ และสิ่งหนึ่งที่กำลังได้รับความนิยม ถือได้ว่าเป็นการปฏิวัติรูปแบบการขายของก็คือ E-Commerce ซึ่งเจ้าของสินค้าต่างๆ ไม่จำเป็นต้องมีร้านค้าจริงและไม่จำเป็นต้องจ้างคนขายของอีกต่อไป ร้านค้าและตัวสินค้านั้น จะไปปรากฏอยู่บนเว็บไซต์แทน และการซื้อขายก็เกิดขึ้นบนโลกของ Internet แล้ว PHP ช่วยเราให้เป็นเจ้าของร้านบน Internet และ PHP เป็นภาษาสคริปต์ที่มีความสามารถสูงสำหรับการพัฒนา Web Site และความสามารถที่โดดเด่นอีกประการหนึ่งของ PHP คือ database enabled webpage ทำให้เอกสารของ HTML สามารถที่จะเชื่อมต่อกับระบบฐานข้อมูล ได้อย่างมีประสิทธิภาพและรวดเร็ว จึงทำให้ความต้องการในเรื่องการจัดการสินค้าและรับรายการสั่งของตลอดจนการจัดเก็บ ข้อมูลต่างๆ ที่สำคัญผ่านทาง Internet เป็นไปได้อย่างง่ายดาย

### 2.1.8 MySQL

MySQL [6] เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์โอเพนซอร์สบนพื้นฐานของ SQL ซึ่ง MySQL ได้รับการออกแบบและปรับให้เหมาะสมสำหรับเว็บแอปพลิเคชันและสามารถทำงานบนแพลตฟอร์มใดก็ได้ MySQL ทำงานเป็นดาต้าเบสเซิร์ฟเวอร์ และอนุญาตให้ผู้ใช้หลายคนจัดการและสร้างฐานข้อมูลจำนวนมาก เป็นองค์ประกอบสำคัญใน LAMP (LAMP ย่อมาจาก Linux, Apache, MySQL และ PHP) เมื่อความต้องการใหม่และแตกต่างกันเกิดขึ้นกับอินเทอร์เน็ต MySQL กลายเป็นแพลตฟอร์มทางเลือกสำหรับนักพัฒนาเว็บไซต์และเว็บแอปพลิเคชัน เนื่องจากได้รับการออกแบบมาเพื่อประมวลผลการค้นหาบนล้าน และในการติดตั้ง WordPress ส่วนใหญ่ใช้ LAMP stack เพราะเป็นโอเพนซอร์สและทำงานร่วมกับ WordPress ได้อย่างราบรื่น MySQL เป็นตัวเลือกยอดนิยมสำหรับธุรกิจอีคอมเมิร์ซที่ต้องการจัดการการเงินหลายครั้ง ซึ่งมีความยืดหยุ่นตามความต้องการตามคุณสมบัติหลักของ MySQL

คำสั่ง Sql กับ MySql

SQL คือ Structured Query Language คือคำสั่งที่ใช้บริหารจัดการ database ซึ่งเป็นภาษา programming ที่ออกแบบมาเพื่อทำการจัดการข้อมูล ค้นหาข้อมูล ปรับปรุง เปลี่ยนแปลง เพิ่ม และ ลบ ข้อมูลนั่นเอง ซึ่งข้อมูลจะถูกเก็บอยู่ในฐานข้อมูลในรูปแบบตารางที่มีลักษณะเป็นคอลัมน์และแถว

ความหมายอย่างเป็นทางการ เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้ภาษา SQL โดย MySQL เป็นซอฟต์แวร์โอเพนซอร์ส โดยมีการพัฒนาโดยบริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ฟรี และแบบที่ใช้ในเชิงธุรกิจ สรุปลง่าย ๆ คือ MySQL ก็คือ เป็นฐานข้อมูล

MySQL เป็นฐานข้อมูล โดยมีภาษา SQL เป็นภาษาในการสื่อสารกันเชื่อมโยงกันนั่นเอง ซึ่งภาษา SQL ยังเป็นภาษาที่ใช้ติดต่อกับฐานข้อมูลอื่นๆ อีกด้วย

ภาษา SQL จะใช้จัดการ 2 กลุ่มหลัก ๆ ได้แก่

1. จัดการ database คือ คำสั่งจัดการฐานข้อมูล

Create Database คือ การสร้างฐานข้อมูล

Drop Database คือ การลบตารางออกจากฐานข้อมูล

Select Database คือ การเลือกใช้งานฐานข้อมูลที่ต้องการ

2. จัดการ table คือ คำสั่งจัดการตาราง

Select Query คือ การแสดงข้อมูลในตารางของฐานข้อมูล

Insert Query คือ การเพิ่มแถวข้อมูลลงในตารางของฐานข้อมูล

Update Query คือ การเปลี่ยนแปลงข้อมูลในตารางของฐานข้อมูล

Delete Query คือ การลบข้อมูลในตารางของฐานข้อมูล

Where Clause คือ การสร้างเงื่อนไขเพื่อเลือกข้อมูลที่ต้องการ

AND & OR Clauses คือ การเชื่อมความสัมพันธ์ของเงื่อนไข

Group By คือ การจัดกลุ่มข้อมูลตาม column ที่กำหนด

Order By คือ การจัดเรียงข้อมูลที่แสดงผลตามลำดับตัวอย่างเอกสารแนบการ

ประเมินเว็บไซต์หน่วยงานภายใน

### 2.1.9 การออกแบบฐานข้อมูลด้วยแผนภาพ E-R (Entity-Relationship Diagrams)

ปี ค.ศ. 1976 Peter Chen [14] ได้พัฒนาแบบจำลอง E-R (Entity-Relationship Model , ERM) ขึ้นมาใช้เป็นเครื่องมือในการนำเสนอโครงสร้างของฐานข้อมูลใน ระดับ ความคิด (Conceptual Level) ออกมาในลักษณะของแผนภาพ (Diagram) ที่เข้าใจได้ ง่ายในการสื่อความหมายระหว่างนักออกแบบฐานข้อมูล และผู้ใช้ข้อมูลเกี่ยวกับ ความสัมพันธ์ของเอนทิตีกับเอนทิตี และความสัมพันธ์ของเอนทิตีกับแอททริบิวต์ และ นับจากนั้นเป็นต้นมา แผนภาพ E-R ก็ได้มีการนำไปใช้งานอย่างกว้างขวาง มีการ พัฒนารูปแบบที่หลากหลาย แต่ลัทธิรูปแบบของแผนภาพ E-R ที่หลากหลายดังกล่าว ล้วนอยู่บนพื้นฐานแนวความคิด

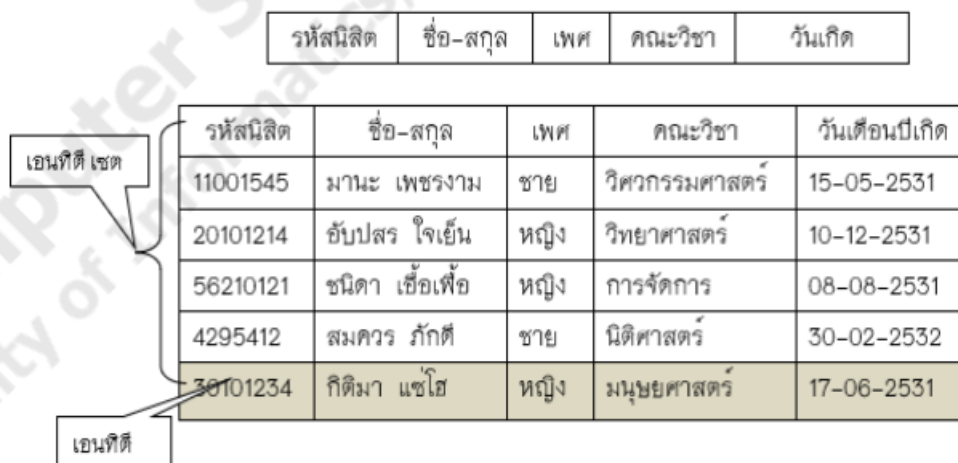
แผนภาพ E-R (E-R Diagram) เป็นแบบจำลองที่ใช้อธิบายโครงสร้างของ ฐานข้อมูลซึ่งเขียนออกมาในลักษณะของรูปภาพ การอธิบายโครงสร้างและ ความสัมพันธ์ของข้อมูล ความสัมพันธ์ของเอนทิตี ที่ช่วยในการออกแบบฐานข้อมูล และได้รับความนิยมเป็นอย่างมาก แผนภาพ E-R เป็นแบบจำลองเชิงแนวคิด (Conceptual Data Model) ที่แสดงออกมาในลักษณะของแผนภาพ โดยใช้หลักการจาก โมเดลฐานข้อมูลเชิงสัมพันธ์ในการแสดงลักษณะโดยรวมของข้อมูลในระบบ ช่วย สื่อสารให้เกิด

ความเข้าใจร่วมกันระหว่างผู้วิเคราะห์และผู้ใช้ได้เป็นอย่างดี แผนภาพที่ นิยมใช้ในการนำเสนอโครงสร้างฐานข้อมูลที่นิยม คือแผนภาพ E-R ซึ่งประกอบไปด้วยเอนทิตี แอททริบิวต์ของแต่ละเอนทิตี ความสัมพันธ์ระหว่างเอนทิตีและคีย์ของ ความสัมพันธ์ (Degree Of A Relationship)

แผนภาพ E-R มีความสำคัญต่อการพัฒนาระบบงานฐานข้อมูล แอปพลิเคชัน (Applications) ต่าง ๆ ที่ต้องการการเก็บข้อมูลอย่างมีระบบ แผนภาพ E-R จึงใช้เพื่อ เป็นเอกสารในการสื่อสารระหว่างนักออกแบบระบบ และนักพัฒนาระบบ เพื่อให้ สื่อสารอย่างตรงกัน และเป็นสากล

### 1. องค์ประกอบของแผนภาพ E-R

1.1 เอนทิตี (Entity) เอนทิตีหมายถึงสิ่งที่มีอยู่จริง จับต้องได้ หรืออาจจะเป็นจินตภาพที่แสดงความเป็นหนึ่งเดียว ซึ่งเมื่อกล่าวถึงแล้วทุกคนเข้าใจตรงกัน แต่โดยทั่วไปแล้วเอนทิตี มักจะ อยู่ในรูปของนาม ซึ่งอาจเป็นสิ่งที่ป็นรูปธรรมคือสามารถมองเห็นได้ด้วยตา หรืออยู่ ในรูปของนามธรรมคือไม่สามารถมองเห็นได้ด้วยตาก็ได้ เช่น บุคคล (นิสิต อาจารย์ ลูกค้า พนักงาน คนไข้) สถานที่ (โรงเรียน ห้องเก็บสินค้า คลังสินค้า ร้านค้า) วัตถุ (หนังสือ เครื่องจักรกล เครื่องยนต์ สินค้า วัตถุดิบ) เหตุการณ์ (การยืม การคืน ลงทะเบียน การฝาก-การถอนเงิน) หรือ แนวคิด (วัน วิชา บัญชี หลักสูตร ความชำนาญ) ที่ทำให้เกิดกลุ่มของข้อมูลที่ต้องการจัดเก็บรวมทั้งสามารถบ่งชี้ถึงความเป็น เอกลักษณ์เฉพาะตัวได้ เอนทิตีมีคุณสมบัติได้หลายอย่าง และเอนทิตีเป็นสิ่งที่ผู้ใช้งาน ฐานข้อมูลจะต้องเกี่ยวข้องกับเมื่อมีการออกแบบระบบฐานข้อมูล สัญลักษณ์รูปสี่เหลี่ยมผืนผ้าแทนเอนทิตีหนึ่งเอนทิตีและมีชื่อเอนทิตีกำกับอยู่ ภายใน ชื่อที่ใช้จะต้องเป็น คำนาม และโดยปกติมักจะเขียน ตัวพิมพ์ใหญ่

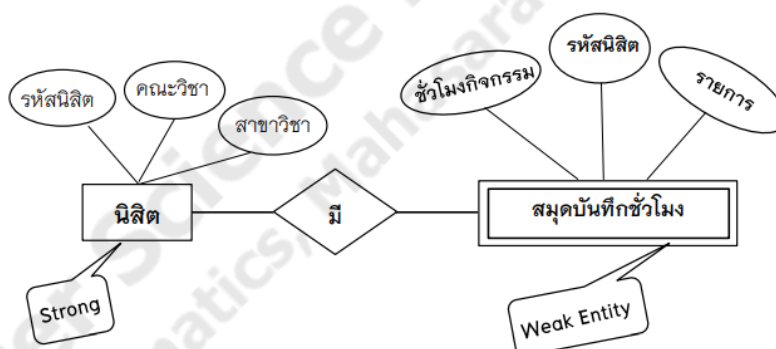


ภาพประกอบที่ 2.2 เอนทิตีและเอนทิตีเซต

เอนทิตีสามารถแบ่งออกได้หลายประเภทด้วยกัน เช่น

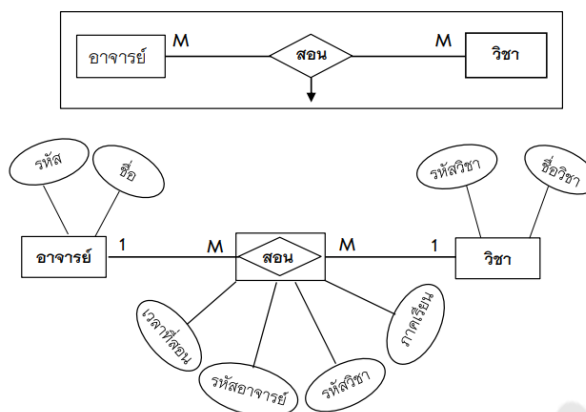
1) เอนทิตีปกติ (Strong Entity หรือ Regular Entity) เป็นเอนทิตีที่มี คุณสมบัติเฉพาะ (Identity) ในตัวเอง สามารถเกิดขึ้นได้ด้วยตัวเอง การคงอยู่ของ เอนทิตีจะไม่ขึ้นกับเอนทิตีอื่น เช่น นิสิต หรือ อาจารย์ หรือ สินค้า สัญลักษณ์ที่ใช้คือ รูปสี่เหลี่ยมผืนผ้า

2) เอนทิตีอ่อนแอ (Weak Entity) เป็นเอนทิตีที่ไม่สามารถเกิดขึ้นได้ ตามลำพัง จะขึ้นอยู่กับ เอนทิตีอื่น จะมีคีย์หลักจากการสืบทอดเอนทิตีที่อ้างอิงอยู่มา ใช้เป็นคีย์หลักหรือส่วนหนึ่งของคีย์หลัก และจะถูกลบออกไปด้วยเมื่อเอนทิตีหลักถูก ลบ สัญลักษณ์ที่ใช้คือรูปสี่เหลี่ยมผืนผ้าเส้นคู่ เอนทิตีทั้ง 2 ประเภทอธิบายได้ดังนี้ เช่น ในมหาวิทยาลัยแต่ละแห่ง นิสิตแต่ละ คนจะมี รหัสนิสิต ชื่อ-สกุล หมายเลข โทรศัพท์ ที่ไม่ซ้ำกัน ดังนั้นเอนทิตีนิสิตจะ จัดเป็น เอนทิตีปกติ และนิสิตในมหาวิทยาลัยแต่ละคน จะมี สมุดบันทึกชั่วโมงกิจกรรม นิสิตแต่ละคนอาจจะมีรายการกิจกรรม หรือมีชั่วโมงกิจกรรมที่เหมือนกัน หรือไม่ เหมือนกันก็ได้ ดังนั้นถ้าไม่มีเอนทิตีนิสิต ก็จะไม่สามารถทราบว่ามีนิสิตคนใด ใด ๆ กิจกรรมอะไร จำนวนชั่วโมงกิจกรรมเป็นเท่าใด ดังนั้นเอนทิตีสสมุดบันทึกชั่วโมง กิจกรรม จึงจัดเป็นเอนทิตีอ่อนแอ เพราะเอนทิตีนี้จะคงอยู่ได้ต้องอาศัยเอนทิตี นิสิต



ภาพประกอบที่ 2.3 เอนทิตีปกติ (Strong Entity) และเอนทิตีอ่อนแอ (Weak Entity)

3) คอมโพสิตเอนทิตี หรือเอนทิตีเชิงความสัมพันธ์ (Composite / Associate Entity) เป็น เอนทิตีที่สร้างขึ้นมาแปลงความสัมพันธ์ของเอนทิตีสองเอนทิตีที่มีความสัมพันธ์แบบ  $M : N$  ให้เป็นแบบ  $1 : M$  เพื่อให้เข้าใจได้ง่าย โดยการนำเอาคีย์ หลักของเอนทิตีทั้งสองมารวมกันกับแอททริบิวต์อื่นๆ ที่ สนใจ สัญลักษณ์ที่ใช้คือ สี่เหลี่ยม ผืนผ้าที่ภายในมีรูปสี่เหลี่ยมขนมเปียกปูน เช่น อาจารย์หนึ่งคนสอนได้ หลาย วิชา และวิชาหนึ่งรายวิชามีอาจารย์สอนได้หลายคน



ภาพประกอบที่ 2.4 คอมโพสิตเอนทิตี หรือเอนทิตีเชิงความสัมพันธ์

### 1.2 แอททริบิวต์ หรือ พรอพเพอร์ตี (Property)

เอนทิตีหนึ่งๆ จะประกอบไปด้วยคุณสมบัติ หรือ ลักษณะที่แตกต่างกัน ออกไป ข้อมูลที่แสดงถึงคุณลักษณะของเอนทิตีเรียกว่าแอททริบิวต์ ดังนั้นความหมาย ของแอททริบิวต์ก็คือ คุณสมบัติของเอนทิตีหรือข้อมูลของเอนทิตีที่เราสนใจจะเก็บ หรืออาจจะหมายถึงกลุ่มของค่าความจริงใด ๆ ที่เป็นรายละเอียดของเอนทิตีซึ่งแสดง ลักษณะและคุณสมบัติของเอนทิตี แต่ละเอนทิตีสามารถมีคุณสมบัติต่าง ๆ ได้ มากมาย ขึ้นอยู่กับว่าต้องการจัดเก็บข้อมูลอะไรบ้าง เช่น คุณสมบัติหรือข้อมูลของ นิสิต ประกอบไปด้วย ชื่อ, ที่อยู่, หมายเลขโทรศัพท์ รหัสนิสิต, คณะวิชา, สาขาวิชา, วัน เดือนปีเกิด, เพศ, ส่วนสูง, น้ำหนัก ฯลฯ

แอททริบิวต์สามารถแบ่งออกได้หลายประเภทเช่น

1) ซิมเพิลแอททริบิวต์ (Simple Attribute) เป็นแอททริบิวต์แบบ ง่าย ๆ มีเพียงองค์ประกอบเดียวหรือข้อมูลเดียว ภายในแอททริบิวต์ไม่สามารถ แบ่งย่อยได้อีก เช่น ข้อมูล เพศ, อายุ, ส่วนสูง, น้ำหนัก, รหัสนิสิต, คณะวิชา, สาขาวิชา เป็นต้น จะประกอบด้วยค่าข้อมูลเดียว สัญลักษณ์ที่ใช้แทนแอททริบิวต์ ชนิดนี้ คือวงรีที่มีเส้นเชื่อมต่อไปยังเอนทิตีของแอททริบิวต์ นั้นๆ โดยมีชื่อแอททริบิวต์ อยู่ในวงรี

2) คอมโพสิต แอททริบิวต์ (Composite Attribute) หมายถึง แอททริบิวต์ที่มีองค์ประกอบหรือข้อมูล ซึ่งจะตรงกันข้ามกับซิมเพิลแอททริบิวต์ ค่าภายในแอททริบิวต์ สามารถแบ่งย่อยออกมาได้อีก แต่มีอิสระต่อกัน เช่น

- แอททริบิวต์ “ที่อยู่” จะประกอบด้วย บ้านเลขที่ หมู่ที่ ตำบล อำเภอ และจังหวัด
- แอททริบิวต์ “ชื่อ” จะประกอบด้วย คำนำหน้าชื่อ ชื่อ ตัว และ นามสกุล

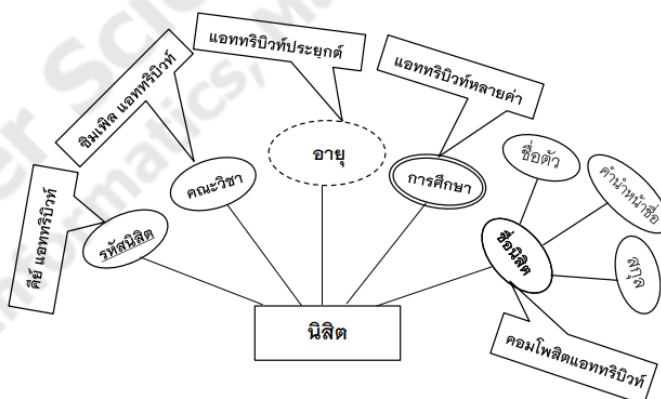


3) แอททริบิวต์ ค่าเดียว (Single-value Attribute) คือ แอททริบิวต์ที่มีค่าข้อมูลที่เก็บเพียงค่าเดียว เช่น รหัสนิสิต ชื่อ ภาควิชา ในเอนทิตี “นิสิต” ซึ่งนิสิต แต่ละคนก็จะมีค่าข้อมูลเหล่านี้เพียงค่าเดียวเท่านั้น หรือเอนทิตี “เพศ” ก็จะมีค่าข้อมูลที่เก็บเพียง 1 ค่า คือ หญิง หรือ ชาย เท่านั้นเช่นกัน

4) แอททริบิวต์ หลายค่า (Multivalued Attribute) คือ แอททริบิวต์ที่สามารถมีได้หลายค่า เช่น คนหนึ่งคนสามารถมี “วุฒิการศึกษา” ได้หลายระดับ เช่น ระดับปริญญาตรี ระดับปริญญาโท และระดับปริญญาเอก หรือ คนหนึ่งคนอาจจะ มี “หมายเลขโทรศัพท์” ได้หลายหมายเลข หรือข้อมูล “โรงเรียนที่เรียนจบ” ตั้งแต่ระดับ อนุบาล ประถมศึกษา มัธยมศึกษา และอุดมศึกษา อาจจะมาจากหลายโรงเรียน เป็นต้น สัญลักษณ์ของแอททริบิวต์ หลายค่า คือรูปวงรีสองวงซ้อนกัน

5) แอททริบิวต์ ประยุกต์ (Derived Attribute) หมายถึง แอททริบิวต์ที่ไม่ได้เก็บอยู่ในฐานข้อมูล แต่ได้จากการประยุกต์ใช้แอททริบิวต์อื่น เช่น แอททริบิวต์ รายได้รวมของพนักงานขายของ ได้มาจากการรวมเงินเดือน ค่าโบนัส ค่าบำนาญ (Commission) ของแต่ละเดือน หรือ แอททริบิวต์ อายุ ได้จากการคำนวณจากวันเดือน ปีเกิด สัญลักษณ์ของแอททริบิวต์ประยุกต์ เป็นรูปวงรีแบบเส้นประ

6) คีย์ แอททริบิวต์ (Key Attributes) หรือ คีย์หลัก หรือ กุญแจหลัก เป็นแอททริบิวต์ที่บอกเอกลักษณ์ของเอนทิตีนั้นๆได้ จะมีค่าของข้อมูลที่ไม่ซ้ำกัน เช่น รหัสนิสิต จัดเป็นคีย์แอททริบิวต์ ใช้ระบุความแตกต่างของแต่ละคนในเอนทิตี สัญลักษณ์ของคีย์หลักคือรูปวงรีภายในมีชื่อแอททริบิวต์ที่มีการขีดเส้นใต้แทนคีย์หลัก



ภาพประกอบที่ 2.5 แสดงแอททริบิวต์ชนิดต่าง ๆ

### 1.3 ความสัมพันธ์ระหว่างเอนทิตี (Entity Relationship)

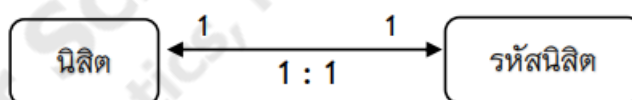
ในปี ค.ศ. 1976 Chen ได้เสนอโมเดลข้อมูลเชิงสัมพันธ์ (Entity Relationship Model) หรือ E-R Model และได้รับการพัฒนาจนสมบูรณ์ในปี ค.ศ. 1979 เพื่อใช้ในการ อธิบายฐานข้อมูลในรูปแบบของแผนภาพ เรียกว่า “E-R diagram” ซึ่งจะใช้พื้นฐานของ โมเดลข้อมูลเชิงตรรกะที่ Fleming ที่ได้เสนอไว้ในปี ค.ศ. 1989 เป็นการสร้างโครงร่าง วิวของผู้ใช้ จะแสดงข้อมูลในขอบเขตที่ผู้ออกแบบสนใจ โดยมีสิ่งที่ต้องกำหนดเป็น พื้นฐานได้แก่ ความสัมพันธ์ระหว่างเอนทิตีและแอททริบิวต์ ในฐานข้อมูลนั้น

ความสัมพันธ์ หรือ Relationship จะหมายถึงความสัมพันธ์ระหว่างเอนทิตี หรือความสัมพันธ์ระหว่างข้อมูล เป็นลักษณะของความสัมพันธ์ระหว่างเอนทิตีหนึ่งกับ ตัวของมันเองหรือกับเอนทิตีอื่น (มากกว่า 2 เอนทิตีก็ได้) ว่ามีความสัมพันธ์ของข้อมูล เป็นแบบใด ความสัมพันธ์ระหว่างเอนทิตีในแผนภาพ E-R จะแทนด้วยรูปสี่เหลี่ยมข้าวหลามตัดที่มีชื่อของความสัมพันธ์อยู่ภายในและมีเส้นเชื่อมระหว่างเอนทิตีและ ความสัมพันธ์ระหว่างเอนทิตี

ความสัมพันธ์ระหว่างข้อมูลที่เรารู้จัก ซึ่งพิจารณาได้ 2 แบบ คือ ความสัมพันธ์ แบบจำเป็น หรือแบบบังคับ (Mandatory) หมายถึงความสัมพันธ์ที่ จะต้อง มี หรือ จะต้องเกิดขึ้นอย่างแน่นอน นิยมแทนด้วยสัญลักษณ์เส้นตรงทึบ หรือ เส้น ตรงสองเส้น และความสัมพันธ์แบบไม่จำเป็น (Optional) เป็นความสัมพันธ์ที่ อาจจะมีหรือไม่มีก็ได้ นิยมแทนด้วยสัญลักษณ์เส้นตรงประหรือเส้นตรงหนึ่งเส้น

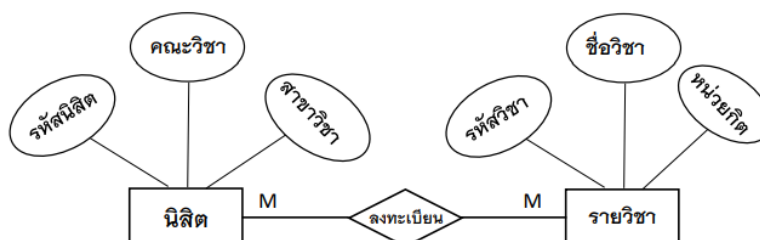
### 1.3.1 ชนิดความสัมพันธ์ระหว่างข้อมูล

1) ความสัมพันธ์แบบหนึ่งต่อหนึ่ง (One to One Relationships ; 1 : 1) เป็นการแสดงถึงความสัมพันธ์ของข้อมูลของเอนทิตีหนึ่งว่ามีความสัมพันธ์กับ ข้อมูลในอีกเอนทิตีหนึ่งในลักษณะหนึ่งต่อหนึ่ง หมายถึงข้อมูลหนึ่งข้อมูลของเอนทิตีที่ 1 มีความสัมพันธ์กับข้อมูลในเอนทิตีที่ 2 เพียงข้อมูลเดียวเท่านั้น ซึ่งใช้ตัวเลขกำกับที่ เส้นเพื่อระบุความสัมพันธ์ เช่น ความสัมพันธ์ระหว่างข้อมูลนิสิตกับรหัสนิสิต : นิสิต 1 คน จะมีรหัส นิสิตได้เพียง 1 รหัสเท่านั้น ในขณะที่ตัวกันรหัสนิสิต 1 รหัส จะใช้กับนิสิตได้เพียง 1 คน เช่นกัน



ภาพประกอบที่ 2.6 ความสัมพันธ์ระหว่างข้อมูลนิสิตกับรหัสนิสิต แบบ 1:1

3) ความสัมพันธ์แบบกลุ่มต่อกลุ่ม (Many to Many Relationships, M:N) เป็นความสัมพันธ์แบบหลายรายการระหว่างสองเอนทิตี เป็น ความสัมพันธ์ของข้อมูลของเอนทิตีหนึ่งมีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่ง ใน ลักษณะกลุ่มต่อกลุ่ม เช่น ความสัมพันธ์ระหว่างนิสิตกับรายวิชาที่เปิดสอน : นิสิต 1 คน ลงทะเบียนเรียนได้หลายวิชาและวิชาหนึ่ง ๆ จะมีนิสิตลงทะเบียนได้มากกว่า 1 คน

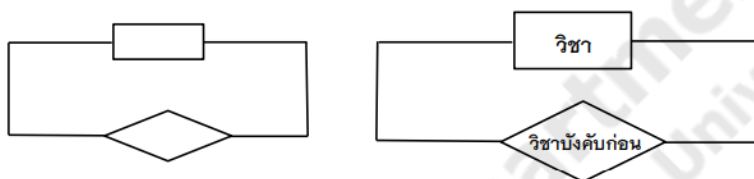


ภาพประกอบที่ 2.7 แสดงความสัมพันธ์แบบกลุ่มต่อกลุ่ม หรือ M : N

### 1.3.2 ดีกรีของความสัมพันธ์ (Degree of Relationship)

ดีกรีความสัมพันธ์ หรือระดับชั้นของความสัมพันธ์ หมายถึงจำนวนเอนทิตีที่มี ส่วนร่วมกับความสัมพันธะนั้น ๆ ดีกรีความสัมพันธ์แบ่งออกได้เป็น 3 ประเภท คือ

1) ความสัมพันธ์แบบยูนารี (Unary Relationship) หรือ ความสัมพันธ์แบบรีเคอร์ซีฟ (Recursive Relationship) หรือความสัมพันธ์เอนทิตีเดียว เป็นความสัมพันธ์ระหว่างเอนทิตีชนิดเดียวกัน หรือความสัมพันธ์ในตัวเอง เช่น ความสัมพันธ์ระหว่างเอนทิตี “วิชาที่บังคับเรียนก่อน” และ “วิชา” (วิชาเรียนบางวิชาบังคับให้ต้องเรียนบางวิชาก่อนจึงจะสามารถเรียนวิชาดังกล่าวได้)



ภาพประกอบที่ 2.8 ความสัมพันธ์แบบยูนารี

2) ความสัมพันธ์แบบไบนารี หรือ ความสัมพันธ์ระหว่างสอง เอนทิตี (Binary Relationship) เป็นความสัมพันธ์เป็นความสัมพันธ์แบบ 2 ทางระหว่าง เอนทิตี 2 เอนทิตี คือ เอนทิตีหนึ่งมีความสัมพันธ์กับเอนทิตีอื่นที่ต่างประเภทกัน เป็น ความสัมพันธ์ที่พบได้บ่อยที่สุด เช่น นิสิต คณะวิชา อาจารย์ นิสิต



ภาพประกอบที่ 2.9 แสดงตัวอย่างความสัมพันธ์แบบไบนารี

3) ความสัมพันธ์ที่มีเอนทิตีมาเกี่ยวข้องตั้งแต่สามเอนทิตีขึ้นไป (N-ary Relationship) เป็นความสัมพันธ์แบบสามทาง หรือเป็นความสัมพันธ์ระหว่าง เอนทิตีต่างประเภทกันตั้งแต่ 3 เอนทิตีขึ้นไป โดยที่ N จะหมายถึงจำนวนเอนทิตีที่มา สัมพันธ์กับความสัมพันธ์หนึ่ง

ตารางเปรียบเทียบ สัญลักษณ์ที่ใช้ใน E-R Diagram

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

ภาพประกอบที่ 2.10 สัญลักษณ์ที่ใช้ใน ER Diagram

## 2.2 ตัวอย่างเอกสารแนบการประเมินเว็บไซต์หน่วยงานภายใน

### 2.2.1 ส่วนที่ 1 ข้อกำหนด/แนวปฏิบัติพื้นฐาน

1. การแสดงผล
  - มีการแสดงผลอย่างน้อย 1 ภาษา คือภาษาไทย และภาษาอังกฤษ
2. URL เว็บไซต์คณะ/หน่วยงาน
  - เว็บไซต์คณะ/หน่วยงานอยู่ภายใต้โดเมนเนม (Domain Name) ของมหาวิทยาลัย คือ (.msu.ac.th)
  - เว็บไซต์คณะ/หน่วยงานมีการติดตั้ง SSL Certificate (<https://xxx.msu.ac.th>)
3. เกี่ยวกับหน่วยงาน
  - ประวัติความเป็นมา
  - วิสัยทัศน์ พันธกิจ
  - โครงสร้างหน่วยงาน ผู้บริหาร อำนาจหน้าที่
  - การกิจ และหน้าที่รับผิดชอบของหน่วยงาน

- ข้อมูลบุคลากรภายในหน่วยงาน ประกอบด้วย ชื่อ-สกุล, ตำแหน่ง, อีเมล (อย่างน้อย)
  - ข้อมูลการบริการของหน่วยงาน (หน่วยงานสนับสนุน) /หลักสูตรที่เปิดสอน (คณะ/หน่วยงานที่มีการเรียนการสอน)
  - แผนกลยุทธ์หน่วยงาน (เฉพาะเว็บไซต์ฉบับภาษาไทย)
  - แผนปฏิบัติราชการ (เฉพาะเว็บไซต์ฉบับภาษาไทย)
  - ข้อมูลการติดต่อ ประกอบด้วย ที่อยู่ เบอร์โทรศัพท์ โทรสาร และแผนที่ตั้งหน่วยงาน เป็นต้น
  - ที่อยู่ไปรษณีย์อิเล็กทรอนิกส์ (e-Mail Address) ของบุคคลภายในหน่วยงานที่รับผิดชอบข้อมูล เช่น ผู้ดูแลเว็บไซต์(Webmaster) เป็นต้น
4. ข่าวประชาสัมพันธ์
- ข่าวประชาสัมพันธ์ทั่วไป
  - ข่าวสารและประกาศของหน่วยงาน เช่น ประกาศรับสมัครงาน การจัดซื้อจัดจ้าง การจัดฝึกอบรม เป็นต้น
  - ปฏิทินกิจกรรมของหน่วยงาน
5. e-Form/Online Form
- ส่วนที่ให้บริการบันทึกข้อมูลลงใน e-Form/Online Form ต่าง ๆ ของหน่วยงาน
6. เว็บลิงค์
- ส่วนงานภายใน
  - หน่วยงานภายนอก (หน่วยงานราชการ สถานศึกษา รัฐวิสาหกิจ)
7. เอกสารเผยแพร่
- ส่วนที่ให้บริการสำหรับ Download แบบฟอร์มต่าง ๆ ของหน่วยงาน
  - ผลงานวิจัย บทความ หรือข้อมูลเผยแพร่ความรู้
8. ผังเว็บไซต์ (Site map)
- ส่วนที่แสดงแผนผังเว็บไซต์ทั้งหมด
9. ช่องทางการติดต่อสื่อสารกับผู้ใช้บริการ

- ช่องทางสอบถามข้อมูล หรือข้อสงสัยมายัง ผู้ให้บริการ และได้รับคำตอบจากผู้ให้บริการ และผู้ให้บริการ สามารถแจ้งข่าวสาร หรือแจ้งเตือน ผู้ใช้บริการ ผ่านช่องทางต่าง ๆ เช่น
- ช่องทาง ถาม - ตอบ (Q & A)
- ช่องทางการติดต่อหน่วยงานในรูปแบบเครือข่ายสังคมออนไลน์ (Social Networking)
- ช่องทางการรับเรื่องร้องทุกข์ ร้องเรียน และช่องทางติดตามสถานะเรื่องร้องเรียน
- ช่องทางแสดงความคิดเห็น หรือข้อเสนอแนะ เช่น e-mail, Web board เป็นต้น
- ช่องทางการแบ่งปัน (Share) ข้อมูลไปยัง Social Networking ต่าง ๆ เช่น Facebook, Line เป็นต้น

#### 10. ระบบการค้นหาข้อมูลภายในเว็บไซต์

- ฟังก์ชันค้นหาข้อมูลภายในเว็บไซต์
- แสดงผลการค้นหา ประกอบด้วย หัวข้อข้อมูลที่ทำการค้นหา พร้อมแสดงผลเป็นข้อความ รายละเอียด หรือเส้นเชื่อม (Link) สำหรับคลิกเพื่อเข้าถึงข้อมูล

#### 11. ส่วนล่างของเว็บไซต์ (Page Footer)

- เมนูหลักในรูปแบบข้อความ
- ข้อมูลติดต่อหน่วยงาน ได้แก่ ชื่อและที่อยู่หมายเลขโทรศัพท์ หมายเลขโทรสาร และที่อยู่ไปรษณีย์อิเล็กทรอนิกส์
- เส้นเชื่อมกลับไปยังหน้าหลักของเว็บไซต์
- คำสงวนลิขสิทธิ์ (Copyright)

#### 2.2.2 ส่วนที่ 2 ข้อกำหนด/แนวปฏิบัติเพิ่มเติม จำนวนรวม 3 ข้อ

1. เครื่องมือสนับสนุนการใช้งาน
  - มีระบบ Navigation ที่ชัดเจน ง่ายต่อการเข้าใจ
2. เครื่องมือสำหรับเก็บข้อมูลการเยี่ยมชมเว็บไซต์ (Web Analytic)
  - มีเครื่องมือสำหรับเก็บข้อมูล การเยี่ยมชมเว็บไซต์ของผู้ใช้บริการ เช่นจำนวน ครั้ง จำนวนหน้า ความสนใจระยะเวลา เป็นต้น ทั้งนี้

เพื่อให้หน่วยงานมี ข้อมูลในการวิเคราะห์พฤติกรรมกรรมการเยี่ยมชม  
เว็บไซต์ (Web Behavior)

3. เส้นเชื่อม (Link)
  - เส้นเชื่อม (Link) มีความพร้อมใช้งาน (ใช้เครื่องมือ  
<https://www.deadlinkchecker.com/> ในการตรวจสอบเส้น  
เชื่อม Link)
4. แบนเนอร์ขอความยินยอมจาก User ในการใช้ข้อมูลที่ได้จากคุกกี้  
(Cookie Consent)
  - มีการติดตั้งแบนเนอร์ขอความยินยอมจาก User ในการใช้ข้อมูลที่ได้  
จากคุกกี้ (Cookie Consent) ตัวอย่างโค้ดการติดตั้ง Cookie  
Consentสามารถดูรายละเอียดได้ที่  
<https://cc.msu.ac.th/th/cookieweb.php>

2.2.3 ส่วนที่ 3 แนวปฏิบัติสำหรับมหาวิทยาลัย เพื่อการพัฒนาเว็บไซต์ให้สอดคล้องการจัด  
อันดับมหาวิทยาลัยโดยเว็บโอบเมตริกซ์และ มาตรฐานอื่น ๆ จำนวนรวม 4 ข้อ

1. เว็บไซต์มีความยืดหยุ่นในการแสดงผล
  - เว็บไซต์ต้องมีความยืดหยุ่น (Responsive) ในการแสดงผลสามารถ  
รองรับอุปกรณ์หลากหลายรูปแบบ (ใช้เครื่องมือ  
<https://www.responsinator.com/>)
2. เว็บไซต์ที่มีการเผยแพร่ผลงานวิชาการ
  - เว็บไซต์ที่มีการเผยแพร่ผลงานทางวิชาการจะต้องอยู่ภายใต้โดเมน  
เนม (Domain Name) เดียวกันของมหาวิทยาลัย คือ .msu.ac.th
3. กำหนด Metadata ให้เอกสารดิจิทัลไฟล์
  - มีการกำหนด Metadata ให้กับเอกสารดิจิทัลไฟล์ pdf doc docx  
ppt pptx ตลอดจนไฟล์รูปภาพต่างๆ เพื่อเป็นตัวแทน (Agent) ใน  
การสืบค้น จาก เสิร์ชเอนจิน
4. กำหนด Meta data บน Web Page
  - กำหนดคำอธิบายในโค้ดหรือการกำหนดเป็นชื่อรูปภาพในคำสั่ง  
html
  - กำหนดไตเติ้ล (title) ให้กำหนดเป็นทั้งภาษาไทยและภาษาอังกฤษ
  - กำหนด Meta Language Tag

- กำหนด Meta Content Type Tag
- กำหนด Meta Keyword Tag
- กำหนด Meta Description Tag
- กำหนด Meta Meta Revisit-After Tag
- กำหนด Meta Robots Tag

## 2.3 ตารางเปรียบเทียบโรงพยาบาลสัตว์

ตารางที่ 2.1 ข้อแตกต่างระหว่างโรงพยาบาลสัตว์

การทำงานของระบบ	เว็บไซต์โรงพยาบาลสัตว์ มหาวิทยาลัยขอนแก่น	เว็บไซต์โรงพยาบาลสัตว์ มหาวิทยาลัยเกษตรศาสตร์	เว็บไซต์โรงพยาบาลสัตว์ เล็ก มหาวิทยาลัยเชียงใหม่	เว็บไซต์โรงพยาบาลสัตว์ เพื่อการเรียนรู้การสอนคณะสัตวแพทยศาสตร์ มมส.
การแสดงผลภาษาอย่างน้อย 2 ภาษา		✓	✓	
ประวัติโรงพยาบาล		✓		✓
วิสัยทัศน์ พันธกิจ		✓	✓	✓
โครงสร้างหน่วยงานผู้บริหาร อำนาจหน้าที่				✓
ข้อมูลบุคลากร	✓	✓	✓	✓
ข้อมูลบริการ	✓	✓	✓	✓
ที่อยู่ไปรษณีย์อิเล็กทรอนิกส์			✓	✓
ข่าวประชาสัมพันธ์	✓		✓	✓
ข้อมูลการติดต่อ	✓	✓	✓	✓
ปฏิทินกิจกรรมของหน่วยงาน			✓	✓
กิจกรรมเพื่อสังคม	✓		✓	✓



ตารางที่ 2.1 ข้อแตกต่างระหว่างโรงพยาบาลสัตว์ (ต่อ)

การทำงานของระบบ	โรงพยาบาลสัตว์ มหาวิทยาลัยขอนแก่น	โรงพยาบาลสัตว์ มหาวิทยาลัยเกษตรศาสตร์	โรงพยาบาลสัตว์เล็ก มหาวิทยาลัยเชียงใหม่	โรงพยาบาลสัตว์เพื่อการ เรียนการสอนคณะสัตว แพทยศาสตร์ มมส.
เอกสารเผยแพร่	✓	✓	✓	✓
e-Form/Online Form	✓			✓
ผังเว็บไซต์				✓
ช่องค้นหาข้อมูลภายในเว็บไซต์ได้	✓	✓		✓
เวรสัตว์แพทย์	✓			✓
ข้อมูลการรักษา	✓	✓	✓	✓
แกลลอรี่ความรู้		✓		✓
เก็บข้อมูลการเยี่ยมชมเว็บไซต์				✓
ส่วนล่างของเว็บไซต์	✓	✓	✓	✓