

บทที่ 2

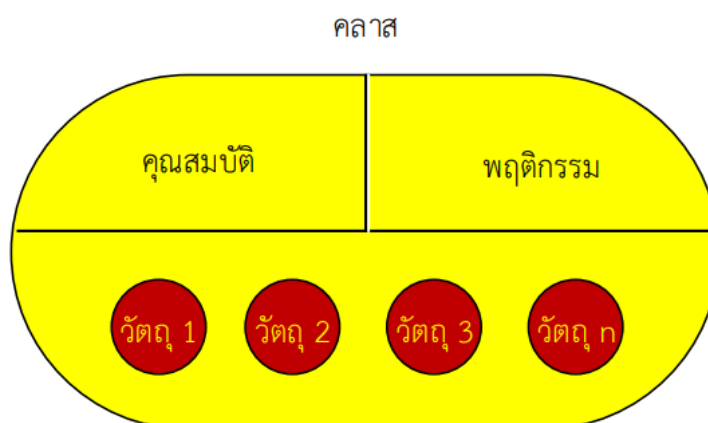
เอกสารและงานทฤษฎีที่เกี่ยวข้อง

การจัดทำโครงการ โครงงานพัฒนาระบบจัดการข้อมูลสาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏกาญจนบุรี ในการพัฒนาระบบจัดการข้อมูลนั้นต้องอาศัยหลักการและทฤษฎีต่าง ๆ รวมทั้งศึกษา แนวทางในการพัฒนาจากงานวิจัยที่เคยมีการพัฒนามาแล้วไปประยุกต์ใช้ในการพัฒนาระบบต่อไป ดังนั้นในบทนี้กล่าวถึง ทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบงาน โดยมุ่งเน้นการอธิบายทฤษฎีที่สำคัญ และเครื่องมือที่ทันสมัยสำหรับการพัฒนาซอฟต์แวร์โดยมีทั้งหมด 8 หัวข้อดังต่อไปนี้

1. การเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

การเขียนโปรแกรมเชิงวัตถุ [1] หมายถึง การมองการเขียนโปรแกรมเป็นลักษณะของวัตถุ มากกว่าการกระทำ การเขียนโปรแกรมเชิงวัตถุจะแยกปัญหาหรือระบบงานออกเป็น ส่วน เพื่อลดความซับซ้อนให้น้อยลง โดยมุ่งที่จะเพิ่มประสิทธิภาพในการสร้างโปรแกรม ทำให้การพัฒนาโปรแกรมด้วยวิธีนี้ใช้เวลาสั้นลงและต้นทุนต่ำลง โดยการนำชุดคำสั่งที่สร้างไปแล้วกลับมาใช้งานใหม่ได้ และยังหาจุดที่ต้องการแก้ไขในโปรแกรมได้ง่ายขึ้น เป็นวิธีการเขียนโปรแกรมที่น่าแนวคิด ในความเป็นจริงที่อยู่รอบตัวมาใช้ในการเขียนโปรแกรม

การเขียนโปรแกรมเชิงวัตถุ มีแนวคิดและองค์ประกอบมาจากความหมายของเชิงวัตถุประกอบไปด้วย 4 ส่วนสำคัญ ได้แก่ คลาส วัตถุ คุณสมบัติ และพฤติกรรม โดยมีรายละเอียดดังภาพที่ 2.1



ภาพที่ 2.1 ส่วนประกอบของวัตถุ

จากภาพที่ 2.1 ส่วนประกอบของวัตถุ โดยแนวคิดประกอบไปด้วยคลาส ภายใน คลาสนั้นจะประกอบด้วยคุณสมบัติและพฤติกรรม กำหนดคุณลักษณะของวัตถุวัตถุใดที่มีคุณสมบัติ หรือพฤติกรรมเดียวกับคลาส โดยส่วนประกอบของวัตถุมีรายละเอียดดังต่อไปนี้

1.1 คลาส (Class)

คลาส หมายถึงศัพท์ที่อธิบายโครงร่างหรือพิมพ์เขียวที่กำหนดขอบเขตกลุ่มของวัตถุ มีความหมายและคำจำกัดความกลุ่มของวัตถุกลุ่มใดกลุ่มหนึ่งมีรายละเอียดดังต่อไปนี้

แม่แบบ



คุณสมบัติ : แขน ขา ปาก หู ตา จมูก
รหัสบัตร ชื่อ เพศ อายุ

พฤติกรรม : วิ่ง หัวเราะ ปรบมือ ส่ายหัว พุด ชก กระโดด

ภาพที่ 2.2 คลาส

จากภาพที่ 2.2 ตัวอย่างคลาสมนุษย์ เมื่อมองในความหมายขอบเขตของมนุษย์ จะเป็นลักษณะของสิ่งมีชีวิตที่เป็นรูปธรรม เช่น แขน ขา หู ตา จมูก เป็นต้น หรือคุณสมบัติด้านนามธรรม เช่น ชื่อ อายุ เพศ น้ำหนัก ส่วนสูง เป็นต้น ในลักษณะของคลาสนั้นบ่งบอกลักษณะภายในตัวของคลาส และในส่วนที่สองเป็นส่วนของการมองในส่วนของคลาสนั้นสามารถเกิดการกระทำ จากตัวอย่าง จากภาพที่ 2.2 สามารถที่จะกระทำหรือแสดงพฤติกรรมออกมาได้ เช่น วิ่ง หัวเราะ ปรบมือ พุด กระโดด เป็นต้น ซึ่งเมื่อมองในความหมายภาพรวมเห็นได้ว่าเป็นลักษณะของมุมมองที่เป็นนามธรรมที่กำหนดขอบเขตของกลุ่มใดกลุ่มหนึ่ง

1.2 คุณสมบัติ (Properties)

คุณสมบัติ หมายถึงองค์ประกอบภายในคลาส ใช้ในการกำหนดข้อมูลที่เก็บภายในคลาส หรือส่วนที่กำหนดคุณลักษณะข้อกำหนดของคลาสนั้น แต่ละคลาสจะกำหนดคุณสมบัติที่ซึ่งกำหนดสถานะของวัตถุที่เกิดจากคลาสนั้น มีรายละเอียดดังต่อไปนี้

แม่แบบ



คุณสมบัติ : แขน ขา ปาก หู ตา จมูก
รหัสบัตร ชื่อ เพศ อายุ

ภาพที่ 2.3 คุณสมบัติ

จากภาพที่ 2.3 ตัวอย่างคุณสมบัติที่เกิดจากคลาสมนุษย์ มนุษย์จะมีคุณสมบัติภายในคลาสของมนุษย์นั้น เช่น แขน ขา ชื่อ รหัสบัตร ที่อยู่ อายุ เพศ เป็นต้น คุณสมบัติเป็นองค์ประกอบที่สำคัญของคลาส โดยจะกำหนดคุณลักษณะที่มีของคลาสนั้น คุณสมบัติบางอย่างจะส่งผลกระทบต่อเนื่องไปยังพฤติกรรมที่เกิดขึ้นของคลาส

1.3 พฤติกรรม (Behavior)

พฤติกรรม หรือการกระทำ (Action) หมายถึงองค์ประกอบที่อยู่ภายในคลาสใช้ในการกำหนดการกระทำบางอย่างของคลาส โดยพฤติกรรมเป็นการกำหนดขอบเขตที่สามารถกระทำได้ของคลาส ส่วนมากจะเกิดจากคุณสมบัติที่มีอยู่ในคลาส มีรายละเอียดดังต่อไปนี้

แม่แบบ



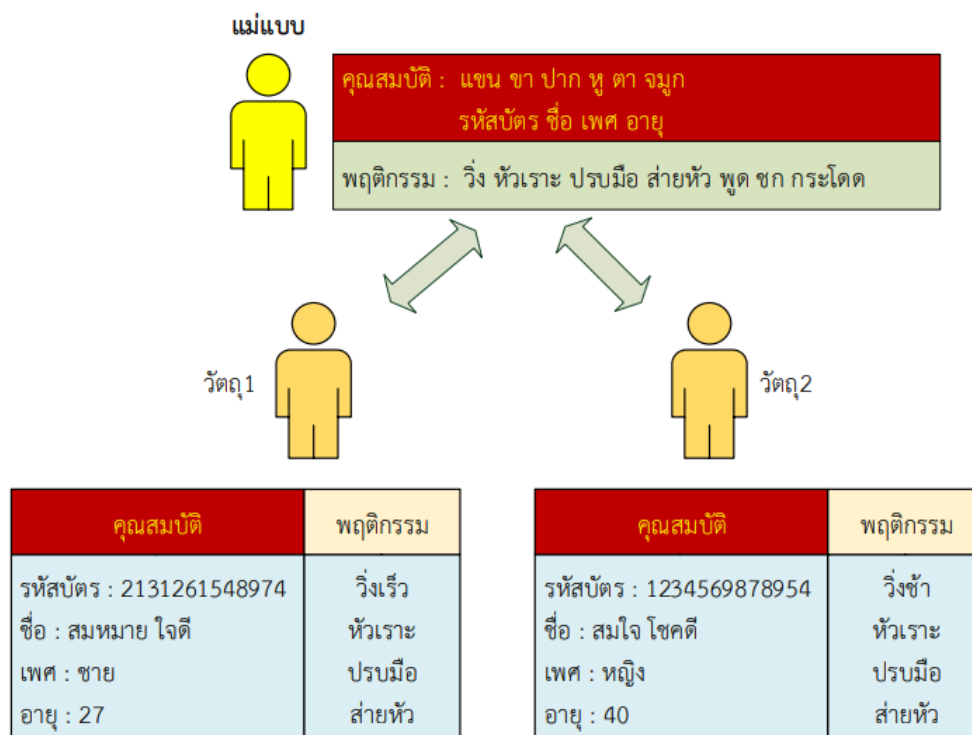
พฤติกรรม : วิ่ง หัวเราะ ปรบมือ ส่ายหัว พุด ชก กระโดด

ภาพที่ 2.4 พฤติกรรม

จากภาพที่ 2.4 ตัวอย่างพฤติกรรมที่เกิดจากคลาสมนุษย์ พฤติกรรมมักเกิดจากคุณสมบัติที่มีภายในคลาส เช่น คุณสมบัติมีขาจะสามารถวิ่งได้ ปากสามารถพูดได้ เป็นต้น พฤติกรรมเป็นกระบวนการที่เกิดขึ้นตั้งแต่รับเข้าข้อมูล ประมวลผลและเกิดผลลัพธ์

1.4 วัตถุ (Object)

วัตถุ หมายถึงการมองสิ่งที่อยู่รอบข้างเป็นวัตถุเป้าหมาย ภายในวัตถุจะประกอบไปด้วยคุณสมบัติของวัตถุและพฤติกรรมหรือการกระทำ ที่ตอบสนองต่อเหตุการณ์ที่จะทำให้เกิดขึ้นของวัตถุ มีรายละเอียดดังต่อไปนี้



ภาพที่ 2.5 วัตถุ

จากภาพที่ 2.5 วัตถุ1 และวัตถุ2 จะมีคุณสมบัติเหมือนกัน คือ รหัสบัตร ชื่อ เพศและอายุ แต่ข้อมูลที่อยู่ภายในคุณสมบัตินี้ จะมีความแตกต่างกันที่บ่งบอกคุณลักษณะเฉพาะของวัตถุ

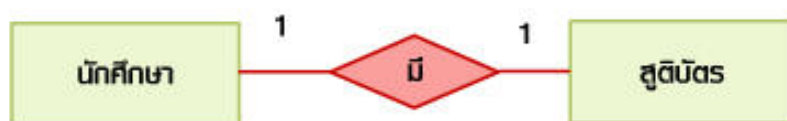
2. ระบบฐานข้อมูล (Database System)

ระบบฐานข้อมูล [2] คือ ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบ มีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล และเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่าง ผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (Database Management System) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่าย สะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

ความสัมพันธ์ [3] ของระบบฐานข้อมูล หมายถึง ความสัมพันธ์ระหว่างเอนทิตี เช่น ความสัมพันธ์ระหว่างนักศึกษา และวิชา เป็นลักษณะว่านักศึกษาแต่ละคนเรียนอยู่คณะวิชาหนึ่ง เป็นต้น ความสัมพันธ์แบ่งออกเป็น 3 ประเภทคือ

2.1 ความสัมพันธ์แบบ หนึ่งต่อหนึ่ง (One to one Relationships)

เป็นการแสดงความสัมพันธ์ของข้อมูล ของเอนทิตีหนึ่งว่ามีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่งในลักษณะ หนึ่ง ต่อ หนึ่ง หรือเขียนได้เป็น 1:1 แสดงว่าข้อมูลเพียงหนึ่งข้อมูลของเอนทิตีแรกมีความสัมพันธ์กับข้อมูลในเอนทิตีที่สองเพียงข้อมูลเดียว ดังรูปภาพที่ 2.6

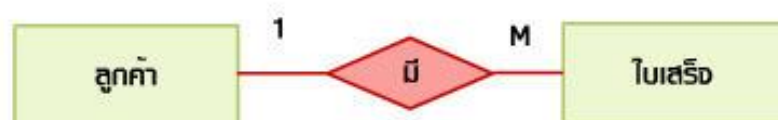


ภาพที่ 2.6 ความสัมพันธ์แบบหนึ่งต่อหนึ่ง

จากภาพที่ 2.6 นักศึกษาหนึ่งคนจะมีสมุดบัตรเพียงใบเดียวเท่านั้น และสมุดบัตรหนึ่งใบจะเป็นของนักศึกษาได้เพียงคนเดียวเท่านั้นเช่นกัน

2.2 ความสัมพันธ์แบบ หนึ่งต่อกลุ่ม (One to many Relationships)

เป็นการแสดงความสัมพันธ์ของข้อมูล ของเอนทิตีหนึ่งว่ามีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่งในลักษณะ หนึ่ง ต่อ กลุ่ม หรือเขียนได้เป็น 1:M แสดงว่าข้อมูลเพียงหนึ่งข้อมูลของเอนทิตีแรกมีความสัมพันธ์กับข้อมูลในเอนทิตีที่สองหลายข้อมูล ดังรูปภาพที่ 2.7



ภาพที่ 2.7 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม

จากภาพที่ 2.7 ลูกค้าหนึ่งคนมีใบเสร็จได้หลายใบ เนื่องจากลูกค้าหนึ่งคนสามารถซื้อสินค้าได้หลายครั้ง และใบเสร็จหนึ่งใบต้องเป็นของลูกค้าเพียงคนเดียวเท่านั้น

2.3 ความสัมพันธ์แบบ กลุ่มต่อกลุ่ม (Many to Many Relationships)

เป็นการแสดงความสัมพันธ์ของข้อมูล ของเอนทิตีหนึ่งว่ามีความสัมพันธ์กับข้อมูลในอีกเอนทิตีหนึ่งในลักษณะ กลุ่ม ต่อ กลุ่ม หรือเขียนได้เป็น M:M แสดงว่าข้อมูลเพียงหนึ่งข้อมูลของเอนทิตีแรก มีความสัมพันธ์กับข้อมูลในเอนทิตีที่สองหลายข้อมูล และทำนองเดียวกัน ข้อมูลเพียง

หนึ่งข้อมูลของเอนทิตีที่สองมีความสัมพันธ์ในทางกลับกันกับข้อมูลในเอนทิตีที่แรกหลายข้อมูล ดังรูปภาพที่ 2.8



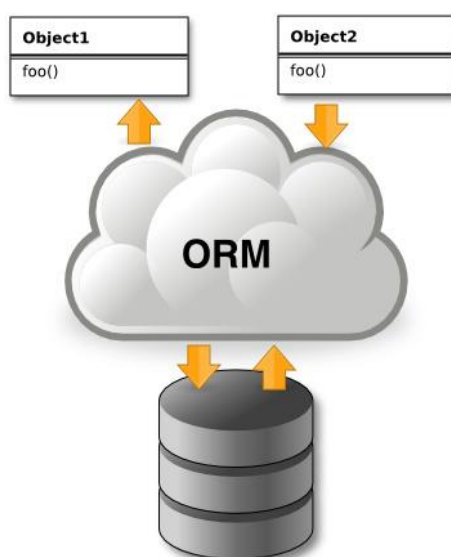
ภาพที่ 2.8 ความสัมพันธ์แบบกลุ่มต่อกลุ่ม

จากภาพที่ 2.8 นักศึกษาสามารถลงทะเบียนเรียนได้หลายวิชาพร้อมกันในแต่ละครั้ง และในทางตรงกันข้าม ในแต่ละวิชา 1 วิชา จะประกอบด้วยนักศึกษาหลายคนมาลงทะเบียนในวิชานั้น

3. โออาเอ็ม (ORM)

ORM (Object / Relational Mapping) [4] หมายถึง การ Map ข้อมูลในตารางข้อมูลของฐานข้อมูลให้อยู่ในรูปของ Object-Oriented Language ซึ่งจะเป็นการสร้าง Database แบบเสมือนขึ้นให้มาอยู่ในรูป Language Programming ทำให้ไม่ต้องไปยุ่งกับ SQL Statement ซึ่งถ้ามีการเปลี่ยนแปลงแก้ไขต้องแก้ไขที่ตัวโปรแกรมแทน และการกระทำต่าง ๆ ยังคงเป็นแบบ Relational แบบเดิม เช่น มีตารางชื่อ Product จะแปลงมาเป็น Class Product คุณลักษณะ (Attributes) ของตาราง Product ก็จะถูกแปลงมาเป็น Object ใน Class

ผู้ใช้ไม่จำเป็นต้องยุ่งในส่วนของ SQL (SELECT, INSERT, UPDATE, DELETE) แต่ผู้ใช้สามารถใช้ Framework มาช่วยจัดการแปลงจากภาษาโปรแกรมมิ่ง ไปเป็น SQL หรือ Database ได้ รายละเอียดดังภาพที่ 2.9



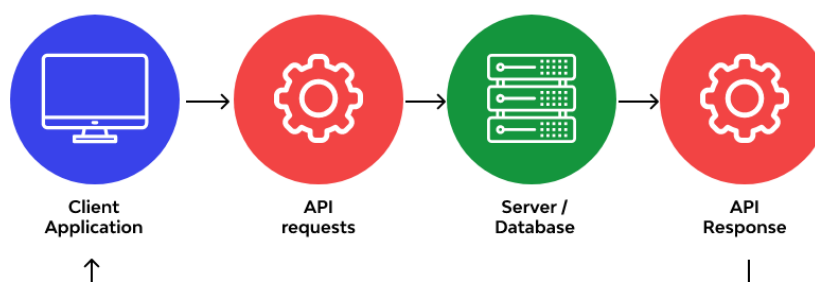
ภาพที่ 2.9 ORM: Object Relational Mapping

จากภาพที่ 2.9 ORM [5] เป็นส่วนที่ใช้สำหรับสลับระหว่าง ข้อมูลที่อยู่ในรูปแบบ Object แปลงไปเป็นข้อมูลที่อยู่ในรูปแบบ ความสัมพันธ์ และดึงข้อมูลที่อยู่ในรูปแบบความสัมพันธ์ กลับไปเป็นข้อมูลที่อยู่ในรูปแบบ Object และเครื่องมือที่พัฒนา ORM มีให้เลือกใช้หลายรูปแบบโดยที่ไม่จำเป็นต้องสร้างขึ้นใหม่

4. เอฟพีไอ (API: Application Programming Interface)

4.1 API [6]

เป็นบริการช่องทางการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลจากระบบหนึ่งไปสู่ระบบอื่น ๆ ที่มีความรวดเร็ว ปลอดภัย หน้าที่หลักของ API คือ รับคำสั่งจากฝั่งลูกข่าย (Client) ซึ่งก็คือ Application ต่าง ๆ เช่น Web Application, Mobile Application และ Desktop Application เป็นต้น เมื่อฝั่งลูกข่ายส่งคำสั่ง จะเรียกว่าการ Request จากนั้น API จะรับคำสั่งดังกล่าว นำไปประมวลผลและสรุปเป็นข้อมูลที่ตรงกับ Request และส่งข้อมูลเหล่านั้นกลับไปให้ฝั่งลูกข่าย หรือ Application เพื่อนำไปใช้งานต่อไป ดังภาพที่ 2.10



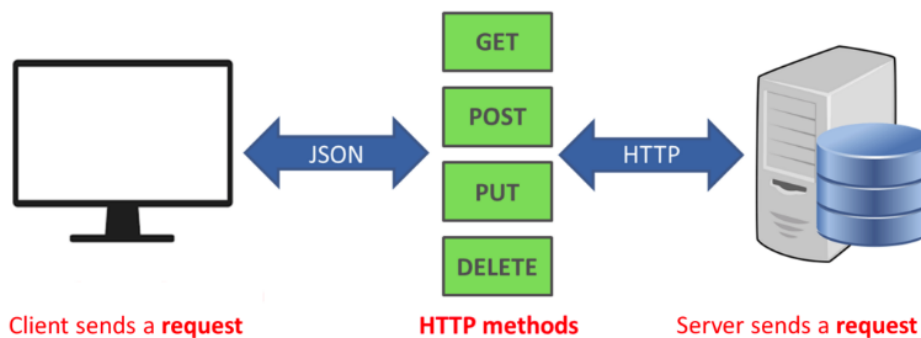
ภาพที่ 2.10 การทำงานของ API

จากภาพที่ 2.10 API เป็นช่องทางในการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลจากระบบหลักไปสู่ระบบลูกข่าย อื่นๆ หน้าที่หลักของ API จะคอยรับคำสั่งจากฝั่งลูกข่าย ซึ่งก็คือแอปพลิเคชันต่าง ๆ จะมีการรับคำสั่งไปประมวลผลและสรุปข้อมูลให้ตรงกับ Request ที่ต้องการ ก่อนจะส่งข้อมูลนั้นกลับไปให้ฝั่งลูกข่าย

4.2 REST API [7]

REST (Representational State Transfer) เป็นรูปแบบการส่งข้อมูลระหว่าง Server-Client รูปแบบหนึ่งซึ่งอยู่บนพื้นฐานของ HTTP Protocol เป็นการสร้าง Web Service เพื่อแลกเปลี่ยนข้อมูลกันผ่านแอปพลิเคชันวิธีหนึ่ง ซึ่งส่งข้อมูลได้หลายชนิด เช่น Text, XML และ JSON โดยส่วนใหญ่แล้วจะเลือกชนิด JSON ที่สามารถรองรับได้หลายรูปแบบไม่ว่าจะเป็น Browser หรือ Mobile เพื่อระบุและทำงานกับทรัพยากร โดยที่เมธอดที่สำคัญใน REST API ประกอบด้วย

- 1) GET: ใช้สำหรับเรียกดูข้อมูลทรัพยากร
- 2) POST: ใช้สำหรับสร้างข้อมูลใหม่ทรัพยากร
- 3) PUT: ใช้สำหรับอัปเดตข้อมูลของทรัพยากรที่มีอยู่แล้ว
- 4) DELETE: ใช้สำหรับลบทรัพยากร



ภาพที่ 2.11 REST API

จากภาพที่ 2.11 REST API อาศัยรูปแบบของ HTTP Method เช่น GET, POST, PUT และ DELETE ในการทำงาน และจะส่งค่ากลับมาเป็น JSON ส่งผลให้สามารถรับส่งข้อมูลไปมาข้าม Platform ได้

5. เว็บแอปพลิเคชัน (Web application)

เว็บแอปพลิเคชัน คือ โปรแกรมประยุกต์ที่จะเข้าถึงด้วยโปรแกรม Internet Browser ซึ่งทำให้เหมาะสำหรับงานที่ต้องการข้อมูลแบบ Real Time จะพบข้อดีของเว็บแอปพลิเคชัน คือ ข้อมูลต่างๆ ที่อยู่ในระบบที่มีการไหลเวียนในแบบ Online จึงสามารถโต้ตอบกับผู้ใช้บริการแบบ Real Time ทำให้เกิดความประทับใจ รวมทั้งสามารถใช้งานได้ง่ายโดยไม่ต้องติดตั้ง Client Program จะทำให้ไม่ต้อง Upgrade Client Program และสามารถใช้งานผ่าน Internet Connection ที่มีความเร็วต่ำส่งผลให้ผู้ใช้สามารถใช้โปรแกรมได้จากทุกที่ ตัวอย่างระบบออนไลน์ที่เหมาะสมกับเว็บ แอปพลิเคชัน เช่น ระบบการจองสินค้าหรือบริการต่าง ๆ ระบบงานบุคลากร ระบบงานแผนการตลาด ระบบการสั่งซื้อแบบพิเศษ และระบบงานในโรงเรียน เป็นต้น

6. รีแอกต์ (React)

React คือ JavaScript Library [8] โดยเป็นเครื่องมือที่ใช้สร้างเว็บสำหรับสร้าง UI (User Interface) หรือ หน้าจอแอปพลิเคชัน จุดเด่นของ React ที่เหมาะนำมาใช้งานคือ ระบบเลขในตัวทำให้หน้าเว็บของผู้พัฒนามีการตอบสนองที่เร็ว และในการเขียน React ผู้พัฒนายังสามารถแยกองค์ประกอบของหน้าเว็บออกเป็นส่วน ๆ เรียกว่า Component แล้วนำมาประกอบกันเป็นหน้าเว็บ

ซึ่งทำให้เราสามารถนำ Component ไปใช้ซ้ำที่อื่นได้ และผู้พัฒนาไม่จำเป็นต้องเขียนขึ้นใหม่ โดยสามารถแบ่งส่วนประกอบหลัก ๆ ของ React ได้ดังต่อไปนี้

6.1 คอมโพเนนต์ (Component)

คอมโพเนนต์ เป็นส่วนที่ใช้สร้างตัวอย่างของ UI ใน React โดยจะเป็นฟังก์ชันหรือคลาสที่จะคืนค่า UI ที่กำหนดรูปแบบและสิ่งที่จะแสดงบนหน้าเว็บ ผู้พัฒนาสามารถสร้างคอมโพเนนต์ใหม่ ๆ และนำมาใช้ซ้ำในแอปพลิเคชันได้

```
function MyComponent(props) {
  return (
    <div>
      <h1>Hello, {props.name}!</h1>
      <p>{props.message}</p>
    </div>
  );
}
```

ภาพที่ 2.12 React Component

จากภาพที่ 2.12 ตัวอย่างของฟังก์ชันคอมโพเนนต์ใน React โดยมีชื่อว่า MyComponent ซึ่งรับพารามิเตอร์ props ภายในฟังก์ชันในส่วนของการแสดงผล ของคอมโพเนนต์นี้มี HTML และ JSX ที่กำหนดรูปแบบและเนื้อหาที่จะแสดงบนหน้าเว็บ

6.2 สเตต (State)

สเตต ใน React เป็นข้อมูลที่ถูกเก็บไว้ภายในคอมโพเนนต์ แต่ละคอมโพเนนต์ สามารถมีสเตตของตัวเองได้โดยอิสระ ซึ่งสเตตนี้สามารถเปลี่ยนแปลงได้และจะทำให้ React รีเรนเดอร์หรืออัปเดตส่วนของ UI ที่เกี่ยวข้อง

```
function Counter() {
  const [count, setCount] = useState(0);

  const increment = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={increment}>Increment</button>
    </div>
  );
}
```

ภาพที่ 2.13 React State

จากภาพที่ 2.13 ตัวอย่างนี้เราใช้ useState hook เพื่อสร้างสเตตที่ชื่อ count และ setCount สำหรับอัปเดตค่า count เมื่อมีการคลิกที่ปุ่มจะเพิ่มค่า count ผ่านฟังก์ชันที่ชื่อ increment

6.3 พรอปส์ (Props: Properties)

พรอปส์ ใน React คือข้อมูลที่ถูกส่งจากคอมโพเนนต์หนึ่งไปยัง คอมโพเนนต์อื่น โดยใช้ตัวแปรที่สามารถรับค่าได้เป็นพารามิเตอร์ของฟังก์ชันคอมโพเนนต์นั้น ๆ ซึ่งพรอปส์ถูกใช้เพื่อส่งข้อมูลระหว่างคอมโพเนนต์แม่ และคอมโพเนนต์ลูก เพื่อให้คอมโพเนนต์ลูกสามารถใช้ข้อมูลจากคอมโพเนนต์แม่ได้

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}

function App() {
  return <Greeting name="John" />;
}
```

ภาพที่ 2.14 React Props

จากภาพที่ 2.14 ตัวอย่างนี้มีการสร้างฟังก์ชันคอมโพเนนต์ ชื่อ Greeting ที่มีพรอปส์ชื่อ name เพื่อแสดงข้อความ โดยข้อความจะขึ้นอยู่กับค่า name ที่ถูกส่งมาในพรอปส์ของคอมโพเนนต์นี้ และในตัวอย่างนี้เราสร้างฟังก์ชันคอมโพเนนต์ชื่อ App ซึ่งนำคอมโพเนนต์ Greeting มาใช้และส่งพรอปส์ name ให้กับคอมโพเนนต์ Greeting โดยในตัวอย่างนี้มีการส่งค่า "John" ในพรอปส์ name

6.4 MobX

เป็นเครื่องมือสำหรับจัดการสเตต [9] (State Management) ที่สามารถนำมาใช้ร่วมกับ React ในการจัดการข้อมูล และสถานะของแอปพลิเคชันอย่างมีประสิทธิภาพและง่ายต่อการใช้งาน MobX ช่วยให้สามารถประกาศตัวแปรแบบ observable ที่จะติดตามการเปลี่ยนแปลงของได้โดยอัตโนมัติ และจะทำการอัปเดตคอมโพเนนต์ที่พึ่งพาข้อมูลนั้น ๆ เมื่อมีการเปลี่ยนแปลงของข้อมูลดังกล่าวเกิดขึ้น นอกจากนี้ยังมีหลักการงานเพิ่มเติมอีกหลายอย่างเช่น computed values และ actions ซึ่งช่วยให้สามารถควบคุมและจัดการการเปลี่ยนแปลงข้อมูลได้อย่างมีประสิทธิภาพ



ภาพที่ 2.15 หลักการทำงานของ MobX

จากภาพที่ 2.15 โดยทั่วไปแล้วแนวคิดของ MobX สามารถอธิบายได้ด้วยหลายส่วนประกอบ ซึ่งเป็นตัวบ่งบอกในกระบวนการจัดการสแตต ดังนี้

- 1) Actions คือฟังก์ชันหรือเมธอดที่ถูกเรียกใช้เพื่อเปลี่ยนแปลงสแตตหรือข้อมูลในแอปพลิเคชัน สิ่งนี้สามารถทำได้โดยการดำเนินการกับ observable หรือผ่านฟังก์ชันที่ถูกต้องเท่านั้น
- 2) Observable คือตัวแปรหรือวัตถุที่ถูกติดตามการเปลี่ยนแปลงของ MobX โดยอัตโนมัติ ซึ่งคือสแตตที่ต้องการจัดการในแอปพลิเคชัน เมื่อมีการเปลี่ยนแปลงใด ๆ ที่เกี่ยวข้องกับ observable MobX จะทำการอัปเดตและแจ้งเตือนถึงส่วนที่เกี่ยวข้องให้รีเ็นเดอร์หรืออื่น ๆ
- 3) Computed Values คือค่าที่คำนวณขึ้นมาจากสแตตอื่น ๆ ในแอปพลิเคชัน โดย MobX จะตรวจสอบการเปลี่ยนแปลงในสแตตที่เกี่ยวข้องและอัปเดตค่า computed values ตามต้องการ
- 4) Side-Effect เป็นการกระทำที่เกิดขึ้นหลังจากการเปลี่ยนแปลงในสแตต โดยทั่วไปจะเป็นการทำงานเพิ่มเติมที่เกี่ยวข้องกับการแสดงผล หรือการกระทำภายนอก เช่น การรีเ็นเดอร์คอมโพเนนท์ React หรือการส่งคำขอไปยังเซิร์ฟเวอร์ เพื่ออัปเดตข้อมูลหรือดึงข้อมูลใหม่ ซึ่งเมื่อเกิดการเปลี่ยนแปลงในสแตตที่เกี่ยวข้อง การกระทำเสริมจะถูกเรียกใช้เพื่อดำเนินการตามที่ต้องการ และอาจมีผลกระทบต่อสแตตหรือข้อมูลอื่น ๆ ในแอปพลิเคชัน

7. ภาษาซีชาร์ป (C# Programming Language)

ภาษาคอมพิวเตอร์ประเภท Object-Oriented Programming [10] พัฒนาโดย Microsoft โดยมีจุดมุ่งหมายในการรวมความสามารถคำนวณของภาษาซีพลัสพลัส (C++) ด้วยการโปรแกรมง่ายกว่าของ Visual Basic โดยภาษาซีชาร์ปมีพื้นฐานจากภาษาซีพลัสพลัส และเก็บส่วนการทำงานคล้ายกับภาษาจาวา (Java) โครงสร้างของภาษาซีชาร์ป ประกอบด้วยส่วนต่าง ๆ ดังนี้

7.1 คลาส

คลาส ในภาษาซีชาร์ปใช้สำหรับกำหนดโครงสร้างและพฤติกรรมของวัตถุที่จะถูกสร้างขึ้นตามแม่แบบที่กำหนดไว้ โดยสามารถขยายและเพิ่มคุณสมบัติตามความต้องการ ตัวอย่างดังภาพที่

```

class MyClass
{
    // ตัวแปรสมาชิก (Member Variables)
    private int myVariable;

    // คอนสตรัคเตอร์ (Constructor)
    public MyClass()
    {
        // การกำหนดค่าเริ่มต้นสำหรับตัวแปรสมาชิก
        myVariable = 0;
    }

    // เมธอด (Methods)
    public void MyMethod()
    {
        // การดำเนินการต่าง ๆ
        Console.WriteLine("Hello, World!");
    }
}

```

ภาพที่ 2.16 โครงสร้างคลาสในภาษาซีชาร์ป

จากภาพที่ 2.16 รายละเอียดโครงสร้างดังนี้

- 1) MyClass คือ ชื่อของคลาส
- 2) myVariable คือ ตัวแปรสมาชิกที่เป็นตัวแปรที่ถูกเก็บเฉพาะภายในคลาส
- 3) MyClass() คือ คอนสตรัคเตอร์ (Constructor) ที่ใช้สร้างวัตถุของคลาส ในตัวอย่างนี้จะกำหนดค่าเริ่มต้นของ myVariable เป็น 0
- 4) MyMethod() คือ เมธอดที่ใช้สำหรับการดำเนินการต่าง ๆ ภายในคลาส

7.2 วัตถุ

วัตถุเป็นตัวแทนของคลาสที่ถูกสร้างขึ้นจากแม่แบบคลาส โดยมีคุณสมบัติและพฤติกรรมตามที่กำหนดไว้ในคลาสนั้น โดยการสร้างวัตถุจะใช้คีย์เวิร์ด new ตามด้วยชื่อของคลาสและวงเล็บว่างเปล่า () ที่เรียกว่าคอนสตรัคเตอร์ของคลาส ตัวอย่างดังภาพที่ 2.17

```

class Person
{
    public string name;
    public int age;

    public Person(string n, int a)
    {
        name = n;
        age = a;
    }

    public void SayHello()
    {
        Console.WriteLine("Hello, my name is " + name + " and I am " + age
    }
}

Person person1 = new Person("John", 25);

person1.SayHello();

```

ภาพที่ 2.17 การสร้างวัตถุในภาษาซีชาร์ป

จากภาพที่ 2.17 ตัวอย่างคลาสชื่อ Person ที่มีคุณสมบัติ name และ age และเมธอด SayHello() เมื่อมีสร้างวัตถุของคลาส Person โดยใช้คอนสตรัคเตอร์ Person() และกำหนดค่า name และ age ตามลำดับ จากนั้นจะสามารถเรียกใช้เมธอด SayHello() ของวัตถุ person1

7.3 เมธอด

เมธอด เป็นส่วนที่ใช้ในการกำหนดการกระทำหรือการดำเนินการที่คลาสหรือวัตถุสามารถทำได้ โดยสามารถรับพารามิเตอร์เข้ามาเพื่อประมวลผลและส่งค่ากลับ ดังภาพที่ 2.18

```

class Calculator
{
    public int Add(int a, int b)
    {
        int result = a + b;
        return result;
    }

    public int Multiply(int a, int b)
    {
        int result = a * b;
        return result;
    }
}

```

ภาพที่ 2.18 เมธอดในภาษาซีชาร์ป

จากภาพที่ 2.18 ตัวอย่างเมธอดในคลาส Calculator ที่มีเมธอด Add และ Multiply ในตัวอย่างนี้ จะมีคลาส Calculator ที่มีเมธอด Add และ Multiply ซึ่งรับพารามิเตอร์ a และ b เพื่อประมวลผลและ ส่งค่าผลลัพธ์กลับ โดยสามารถเรียกใช้เมธอดเหล่านี้ได้ดังภาพที่ 2.19

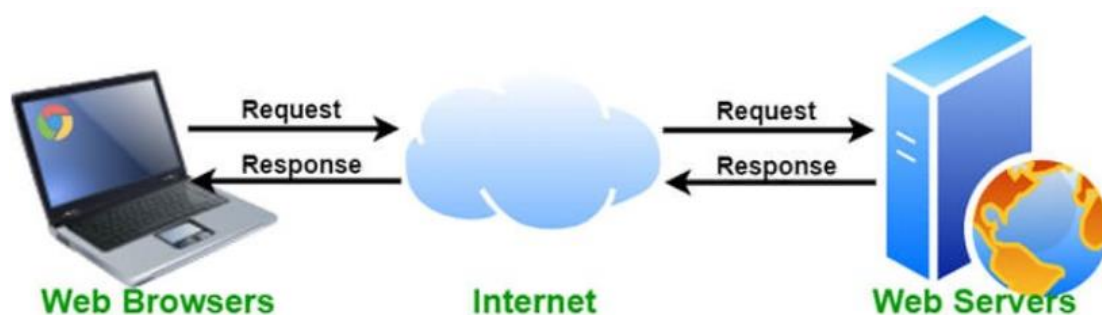
```
Calculator calculator = new Calculator();
int sum = calculator.Add(5, 3);
int product = calculator.Multiply(4, 2);
```

ภาพที่ 2.19 ตัวอย่างการใช้งานเมธอด

จากภาพที่ 2.19 สร้างวัตถุ calculator จากคลาส Calculator และเรียกใช้เมธอด Add และ Multiply เพื่อทำการบวกและคูณตัวเลข กลับมาเก็บผลลัพธ์ในตัวแปร sum และ product ตามลำดับ

8. เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์ [11] คือ เครื่องคอมพิวเตอร์ ที่ติดตั้งโปรแกรมคอมพิวเตอร์ ซึ่งทำหน้าที่ให้บริการ ข้อมูลแก่ เครื่องคอมพิวเตอร์ที่ขอรับบริการ ในรูปแบบสื่อผสม ผ่านระบบเครือข่าย โดยผู้ใช้งาน ติดต่อสื่อสารกับเว็บเบราว์เซอร์ ผ่านทางโปรแกรมเว็บเซิร์ฟเวอร์ และมีการเชื่อมต่อผ่านอินเทอร์เน็ต ดังภาพที่ 2.20



ภาพที่ 2.20 การทำงานของเว็บเซิร์ฟเวอร์

จากภาพที่ 2.20 ขั้นตอนการทำงานของระบบ โดยเริ่มจากผู้ใช้งานได้ทำการร้องขอบริการด้าน ข้อมูลกับเซิร์ฟเวอร์ซึ่งผู้ใช้งานสามารถติดต่อสื่อสารผ่านทางระบบอินเทอร์เน็ต และในปัจจุบัน สามารถเรียกใช้เว็บเซิร์ฟเวอร์ผ่านทางโทรศัพท์เคลื่อนที่ได้

บรรณานุกรม

- [1] เอกราช ธรรมษา. (2559). การเขียนโปรแกรมเชิงวัตถุ. http://www.com-tech.ubru.ac.th/~ekarach/book/pcp/09_OOP.pdf.
- [2] Mindphp.com. (2565). ระบบฐานข้อมูลคืออะไร. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2055-database-คืออะไร23.html>.
- [3] Thaksin University. (2563). ความสัมพันธ์ของระบบฐานข้อมูล. https://cst.tsu.ac.th/courseonline/course/computer_it/database/relationship.html.
- [4] ณัฐคนัย อุปทอง. (2563). ความหมายของ ORM. <https://www.mindphp.com/คู่มือ/73-คืออะไร/3247-orm-doctrine.html>.
- [5] Box Single. (2556). ORM (Object Relational Mapping). [https://www.boxsingle.com/3/ORM-\(Object-Relational-Mapping\)-คืออะไร](https://www.boxsingle.com/3/ORM-(Object-Relational-Mapping)-คืออะไร)
- [6] Hayatee e-pong. (2565). API คืออะไร. <https://www.mindphp.com/คู่มือ/73-คืออะไร/8082-what-is-an-api.html>.
- [7] Settawat Janpuk. (2562). ทำความรู้จักกับ REST API. <https://medium.com/@settawatjanpuk/beginner-มาทำความรู้จักกับ-rest-api-กันดีกว่า-มีตัวอย่าง-c855d93beaf3>.
- [8] Mos Programmer. (2562). รู้จัก React. <https://mosprogramer.medium.com/มารู้จัก-react-กันเถอะ-eb04ab8b8ec1>.
- [9] hilla. (2565). Application State Management With MobX. <https://hilla.dev/docs/tutorials/in-depth-course/application-state-management-with-mobx>
- [10] Mindphp.com. (2560). ซีชาร์ป คือ ภาษาคอมพิวเตอร์ ซึ่งพัฒนามาจากภาษา C++. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2184-c-ชาร์ป-คืออะไร.html>.
- [11] Mindphp.com. (2560). Web Server คืออะไร. <https://www.mindphp.com/คู่มือ/73-คืออะไร/2053-web-server-คืออะไร.html>.