

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องจะประกอบไปด้วย

2.1 ทฤษฎีด้านระบบฐานข้อมูล

- ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล
- โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล
- ฐานข้อมูลเชิงสัมพันธ์
- แผนผังความสัมพันธ์ของข้อมูล

2.2 ทฤษฎีกระบวนการผลิตซอฟต์แวร์

- โมเดลแบบก้นหอย (Spiral Model)

2.3 ทฤษฎีหรือเทคโนโลยีและเทคโนโลยีที่ใช้ในการพัฒนา

- ภาษาจาวา (JAVA)
- ภาษาเอสคิวแอล (SQL)
- แผนที่กูเกิล (Google Maps)
- เอพีไอ (Application Programming Interface : API)
- เฟสบุ๊ก (Facebook)
- ภาษาพีเอชพี (PHP)
- ภาษาจาวาสคริปต์ (JavaScript)
- CSS
- jQuery

2.4 งานวิจัยหรือผลงานที่เกี่ยวข้อง

- เกมอินเกรซ

2.1 ทฤษฎีด้านระบบฐานข้อมูล

ความรู้เบื้องต้นเกี่ยวกับฐานข้อมูล (INTRODUCTION TO DATABASE)

ความหมายของฐานข้อมูล

ฐานข้อมูล หมายถึง การรวบรวมข้อมูลที่เกี่ยวข้องกัน โดยกำหนดรูปแบบในการจัดเก็บอย่างเป็นระบบ เพื่อให้ผู้ใช้สามารถจัดการและเรียกใช้ข้อมูลได้อย่างมีประสิทธิภาพ

ระบบการจัดการฐานข้อมูล

จะทำหน้าที่ในการควบคุมดูแลการสร้างและการเรียกใช้ข้อมูลโดยไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างฐานข้อมูลซึ่งประกอบด้วยโครงสร้าง(Structure) และการใช้งาน (Operation)

- โครงสร้าง หมายถึง โครงสร้างของระบบฐานข้อมูล
- การใช้งาน หมายถึง วิธีการที่จะให้ผู้ใช้สามารถเรียก , แก้ไขข้อมูลในระบบฐานข้อมูลได้

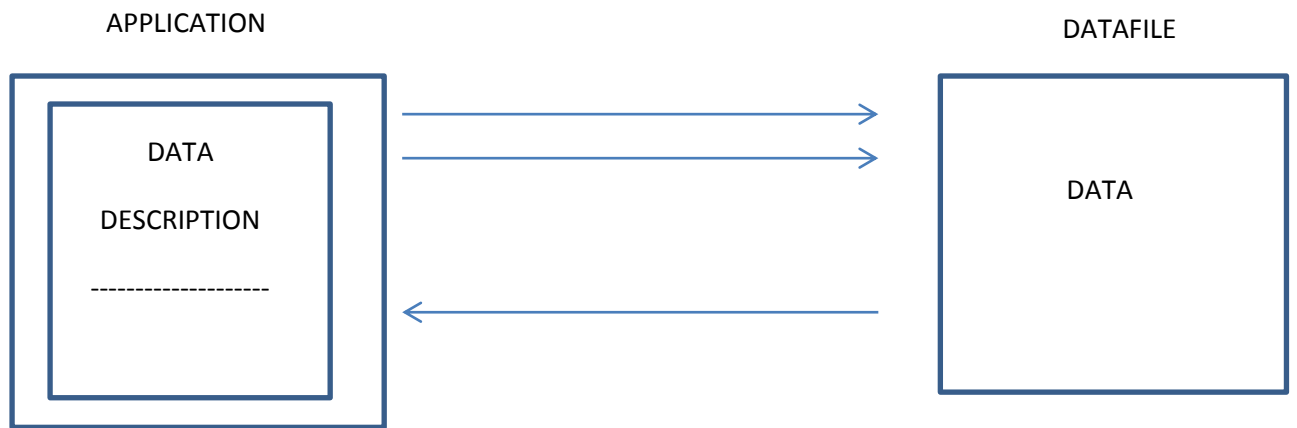
จุดมุ่งหมายของระบบจัดการฐานข้อมูล

เพื่อจัดการควบคุมฐานข้อมูล รวมถึงการพัฒนาข้อมูลให้ทันสมัย ตรงกับความต้องการของผู้ใช้งาน

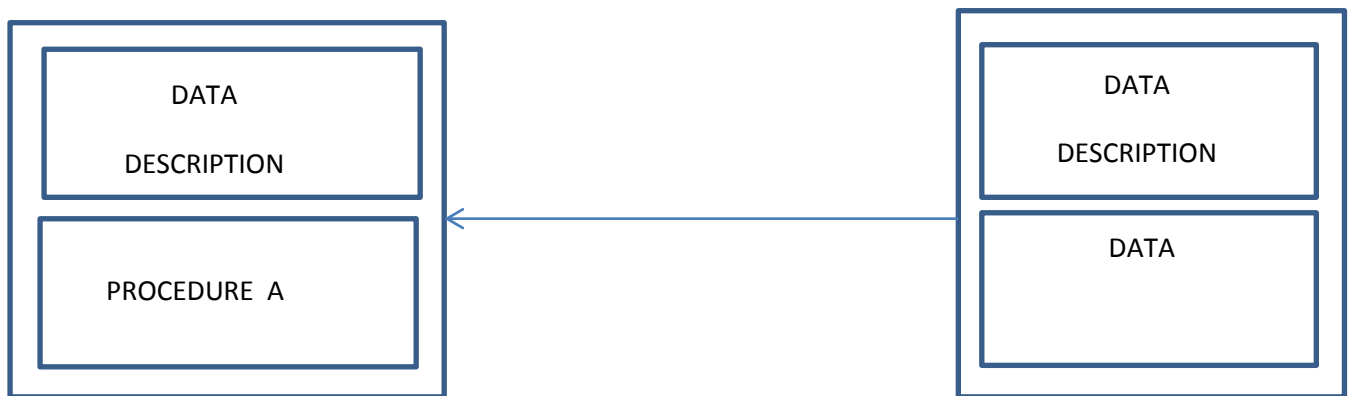
ลักษณะของการจัดการฐานข้อมูล

แบ่งได้ 2 ลักษณะใหญ่ๆ คือ

- 1.ลักษณะข้อมูลทั่วไปจะจัดเก็บแฟ้มข้อมูลธรรมดาที่มีเฉพาะค่าของข้อมูล ส่วนรายละเอียดที่เกี่ยวข้องกับตัวข้อมูลจะถูกระบุอยู่ในแฟ้มข้อมูลโปรแกรมซึ่งเขียนขึ้นโดยผู้เขียนโปรแกรมเอง
- 2.ลักษณะข้อมูลแบบเมทาเดตา (Meta data) จะจัดเก็บรายละเอียดข้อมูลเกี่ยวกับตัวข้อมูลไว้ที่เดียวกัน จัดแยกเป็น 2 ส่วน คือ ข้อมูลและเมทาเดตา(Meta data) ซึ่งบางโปรแกรม เรียกว่า พจนานุกรม (Data dictionary) โดยจะมีระบบจัดการฐานข้อมูลทำหน้าที่ควบคุมจัดการ



ภาพที่ 2.1 ลักษณะข้อมูลทั่วไป ในโปรแกรมประยุกต์ใช้งานทั่วไป



ภาพที่ 2.2 ลักษณะเมทาดาทา(Meta data) ในโปรแกรมประยุกต์ใช้ฐานข้อมูล

ประโยชน์ของฐานข้อมูล

-ลดความซ้ำซ้อนของข้อมูล

การจัดเก็บข้อมูลแบบแฟ้มข้อมูลอาจทำให้ข้อมูลถูกเก็บไว้หลายๆ แห่งทำให้เกิดความซ้ำซ้อน การนำข้อมูลรวบรวมเก็บไว้ในฐานข้อมูลเดียวกันจะช่วยลดปัญหาความซ้ำซ้อน

-หลีกเลี่ยงความขัดแย้งของข้อมูล

การจัดเก็บข้อมูลแบบแฟ้มข้อมูลหลายแฟ้มข้อมูลจะก่อให้เกิดความขัดแย้งของข้อมูลได้ ข้อมูลจะถูกจัดเก็บรวมกัน จึงสามารถเลี่ยงการซ้ำกันของข้อมูลได้

-สามารถใช้ข้อมูลร่วมกันได้

ผู้ใช้ข้อมูลสามารถใช้ข้อมูลเดียวกันได้ซึ่งอาจมาจากแฟ้มข้อมูลที่แตกต่างกัน โดยฐานข้อมูลจะมีลักษณะของการแชร์ข้อมูล (Share Data)

- การรักษาความถูกต้องเชื่อถือได้ของข้อมูล

ระบบการจัดการฐานข้อมูล สามารถที่จะระบุกฎเกณฑ์เพื่อควบคุมความผิดพลาดที่อาจเกิดขึ้นได้ในลักษณะข้อมูลที่ถูกต้อง (Valid Data)

- กำหนดระดับความปลอดภัยของข้อมูลได้

ผู้บริหารระบบสามารถกำหนดสิทธิ, ระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้

- กำหนดความเป็นมาตรฐานเดียวกัน

ผู้บริหารระบบฐานข้อมูลสามารถกำหนดสิทธิ, ระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้

- เกิดความเป็นอิสระของข้อมูลและโปรแกรม

ระบบฐานข้อมูล โปรแกรมประยุกต์จะทำงานอย่างมีระบบ โดยจะเชื่อมโยงฐานข้อมูลโดยที่จะสามารถใช้ข้อมูลได้อย่างอิสระ โดยวิธีการแปลงส่งและลักษณะวิว

ข้อเสีย

- มีต้นทุนสูง
- การเสี่ยงต่อการหยุดชะงักของระบบ

เนื่องจากข้อมูลอาจถูกเก็บไว้ในลักษณะเป็นศูนย์รวมความผิดพลาดของการทำงาน บางส่วนอาจมีผลกระทบกับระบบ ทำให้ระบบหยุดชะงักได้

องค์ประกอบของระบบฐานข้อมูล

1. ฮาร์ดแวร์ (Hardware)

ระบบฐานข้อมูล จะต้องมีส่วนฮาร์ดแวร์ที่มีประสิทธิภาพ ในเรื่องของหน่วยประมวลผลกลางที่ต้องการความเร็วสูง มีขนาดหน่วยความจำหลักที่มากพอ และหน่วยความจำสำรองที่จะจัดเก็บข้อมูลเพียงพอกับความต้องการของระบบ รวมถึงอุปกรณ์นำเข้าข้อมูลเข้าและออกรายงานตามคุณสมบัติที่ระบบต้องการ เพื่อให้ได้ระบบการประมวลผลมีประสิทธิภาพ

2. ซอฟต์แวร์ (Software)

ระบบฐานข้อมูลจะต้องมีโปรแกรมที่ทำหน้าที่ควบคุมดูแลการสร้างการสืบค้นหาข้อมูล ปรับเปลี่ยน การแก้ไขโครงสร้าง การจัดทำรายงาน โดยสื่อกลางระหว่างผู้ใช้กับระบบงานและโปรแกรมประยุกต์ต่างๆ ที่มีอยู่รวมถึงการสื่อสารข้อมูลระหว่างเครื่อง

คุณสมบัติและหน้าที่ของโปรแกรมการจัดการฐานข้อมูล

- สามารถกำหนดและเก็บโครงสร้างฐานข้อมูล
- สามารถรับและจัดเก็บข้อมูลไว้ในฐานข้อมูล
- สามารถดูแลรักษาข้อมูลในฐานข้อมูล
- สามารถประสานงานกับระบบปฏิบัติการ
- สามารถจัดทำข้อมูลสำรองและการกู้คืนข้อมูล
- สามารถควบคุมความปลอดภัยในระบบฐานข้อมูล
- สามารถทำพจนานุกรมข้อมูล
- สามารถติดต่อเชื่อมโยงในลักษณะของระบบเครือข่าย

1.ข้อมูล(Data)

ฐานข้อมูลเป็นการจัดเก็บข้อมูลที่เป็นข้อเท็จจริงขั้นต้น

2.บุคลากร (People)

-ผู้ใช้ (user)

เป็นบุคลากรที่ใช้ข้อมูลจากระบบฐานข้อมูล

-พนักงานปฏิบัติการ (operator)

เป็นผู้ป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์

-นักวิเคราะห์และออกแบบระบบ (system analysis)

เป็นผู้ทำหน้าที่วิเคราะห์ระบบฐานข้อมูล

-ผู้เขียนโปรแกรมประยุกต์ใช้งาน (programmer)

เป็นผู้ทำหน้าที่เขียนโปรแกรมประยุกต์ใช้งานต่างๆ ตามความต้องการ
ของผู้ใช้

-ผู้บริหารข้อมูล (database administration : DBA)

เป็นผู้ทำหน้าที่บริหาร และควบคุมระบบฐานข้อมูลทั้งหมดรวมถึงเป็น
ผู้ตัดสินใจในการเก็บ กำหนดโครงสร้างวิธีการเรียกใช้ข้อมูลเพื่อบริหาร
ฐานข้อมูลอย่างมีประสิทธิภาพ

ขั้นตอนการทำงานและหน้าที่ต่างๆในระบบฐานข้อมูล

- ผู้บริหารฐานข้อมูล (DBA) คือ ผู้ที่หน้าที่ควบคุมการบริหารงานของระบบ
ฐานข้อมูลทั้งหมด
- ตัดสินใจว่าจะรวบรวมงานของระบบฐานข้อมูลทั้งหมด
- วิเคราะห์และทำการตัดสินใจว่าควรเก็บข้อมูลด้วยวิธีใด และจะใช้เทคนิคใด
ในการเรียกใช้ข้อมูล
- ประสานงานกับผู้ใช้ และ คอยสำรวจผลการทำงานและตรวจตราความต้องการ
ของผู้ใช้
- กำหนดแบบความปลอดภัยและความคงสภาพของข้อมูล

โครงสร้างข้อมูลและสถาปัตยกรรมของระบบฐานข้อมูล

(DATA STRUCTURE AND DATABASE SYSTEM ARCHITECTURE)

ลักษณะของโครงสร้างข้อมูล สามารถแบ่งเป็นหน่วยข้อมูลได้ดังนี้

-บิต (bit)

หน่วยของข้อมูลที่มีขนาดเล็กที่สุดในระบบแฟ้มข้อมูล

-ไบต์ (byte)

หน่วยของข้อมูลที่ประกอบด้วยบิตหลายๆ บิตมารวมกันโดยมีขอบเขตที่แน่นอน ความหมาย หมายถึงสิ่งหนึ่ง เช่น รหัสประจำตัว เป็นตัวแทนค่าตัวเลขที่มีความหมายตามขอบเขต

ตัวอย่าง ID_NUMBER มีจำนวน 8 หลักเป็นต้น

-เรคคอร์ด (record)

หน่วยของข้อมูลที่ประกอบด้วยหลายๆ ฟิวด์มารวมกัน ที่มีความสัมพันธ์กัน เช่น รหัสประจำตัว ชื่อ ที่อยู่ เป็นต้น

-แฟ้มข้อมูล (file)

หน่วยของข้อมูลที่ประกอบด้วยหลายเรคคอร์ดรวมกัน

หน่วยการจัดเก็บของฐานข้อมูลเชิงสัมพันธ์สามารถเปรียบเทียบกับแฟ้มข้อมูล ดังนี้

ศัพท์แฟ้มข้อมูล	ศัพท์ฐานข้อมูล	ศัพท์เทคโนโลยี
แฟ้มข้อมูล (file)	ตาราง (table)	รีเลชัน (relation)
เรคคอร์ด (record)	แถว (row)	ทูเพิล (tuple)
ฟิวด์ (field)	คอลัมน์ (column)	แอททริบิวต์ (attribute)
จำนวนเรคคอร์ด (number of records)	จำนวนแถว (number of rows)	คาร์ดินาลิตี้ (cardinality)
กลุ่มข้อมูล (data type)	กลุ่มข้อมูล (data type)	โดเมน (domain)

ตารางที่ 2.1 หน่วยการจัดเก็บของฐานข้อมูลเชิงสัมพันธ์

-รีเลชั่น

การแทนข้อมูลของตารางในฐานะข้อมูลเชิงสัมพันธ์ ซึ่งอาจเกี่ยวกับคน สถานที่ สิ่งของ เช่น รีเลชั่นของข้อมูลสินค้า เป็นต้น

-ทูเพิล

การแทนค่าข้อมูลในแต่ละแถวหรืออาจเรียกกันว่าระเบียบ (record)

-แอททริบิวต์

การแทนค่าขอบเขตของข้อมูลในแต่ละคอลัมน์

-คาร์ดินาลิตี้

จำนวนแถวของข้อมูลในแต่ละตาราง

-โดเมน

ขอบเขตหรือค่าที่ควรจะเป็นของข้อมูลในแอททริบิวต์หนึ่งฐานข้อมูลเชิงสัมพันธ์ เป็นการจำกัดเก็บรวบรวมข้อมูลในลักษณะ 2 มิติ คือ แถวและคอลัมน์

โดยมีการจัดเก็บอย่างมีระบบและกฎเกณฑ์

รหัสวิชา	ชื่อวิชา	หน่วยกิต	จำนวนชั่วโมง	
2133301	ฐานข้อมูลเบื้องต้น	3	(2-2)	Tuple
2133302	การสื่อสารข้อมูล	3	(2-2)	
2133303	คอมพิวเตอร์กราฟฟิก	3	(2-2)	
	Attribute			

CARDINALITY

ตารางที่ 2.2 ตารางตัวอย่างของการแสดงข้อมูลของรายวิชา

สถาปัตยกรรมของฐานข้อมูล

ระบบฐานข้อมูลเป็นโครงสร้างข้อมูลที่ใช้หลายกลุ่ม ต้องการเรียกใช้ข้อมูลบางส่วนหรือทั้งหมดโดยจะแตกต่างกันไปตามงานที่ปฏิบัติลักษณะของผู้ใช้ก็จะมีหลากหลายโดยเฉพาะผู้ใช้ทั่วไป (end user) เป็นผู้ใช้ที่ไม่มีความจำเป็นต้องรู้รายละเอียดของการจัดเก็บข้อมูลดังนั้นเพื่อให้การทำงานเป็นไปตามระดับของผู้ใช้จึงสามารถแบ่งระดับของข้อมูลได้ 3 ระดับดังนี้

1.ระดับภายนอก (External Level)

เป็นระดับของข้อมูลที่ใช้แต่ละคนลงเก็บข้อมูล ตามระบบงานที่ผู้ใช้ได้รับผิดชอบงานนั้น ผู้ใช้ส่วนใหญ่เป็น operator หรือ โปรแกรมเมอร์

2.ระดับแนวคิด (Conceptual Level)

เป็นโครงร่างที่อธิบายถึงฐานข้อมูลโครงสร้างข้อมูลความสัมพันธ์ของข้อมูลกฎเกณฑ์และข้อจำกัดที่มีต่อฐานข้อมูล โดยจะผ่านการวิเคราะห์และออกแบบจากผู้บริหารฐานข้อมูล คือ DBA

3.ระดับภายใน (Internal /Physical Level)

เป็นโครงร่างที่อธิบายถึงการจัดเก็บข้อมูลจริง จัดเก็บในรูปแบบใด วิธีการเข้าถึงข้อมูลวิธีใด เช่น การแฮชชิง (Hashing Function) , อินเด็กซ์ (Indexing) เป็นต้นข้อมูลแต่ละระดับจะมีการเชื่อมโยงโดยจะถูกจัดการด้วยระบบการจัดการฐานข้อมูล โดยจะแปลงความหมายข้อมูลจากระดับหนึ่ง ไปยังอีกระดับหนึ่งที่เรียกกันว่าการแปลงส่ง

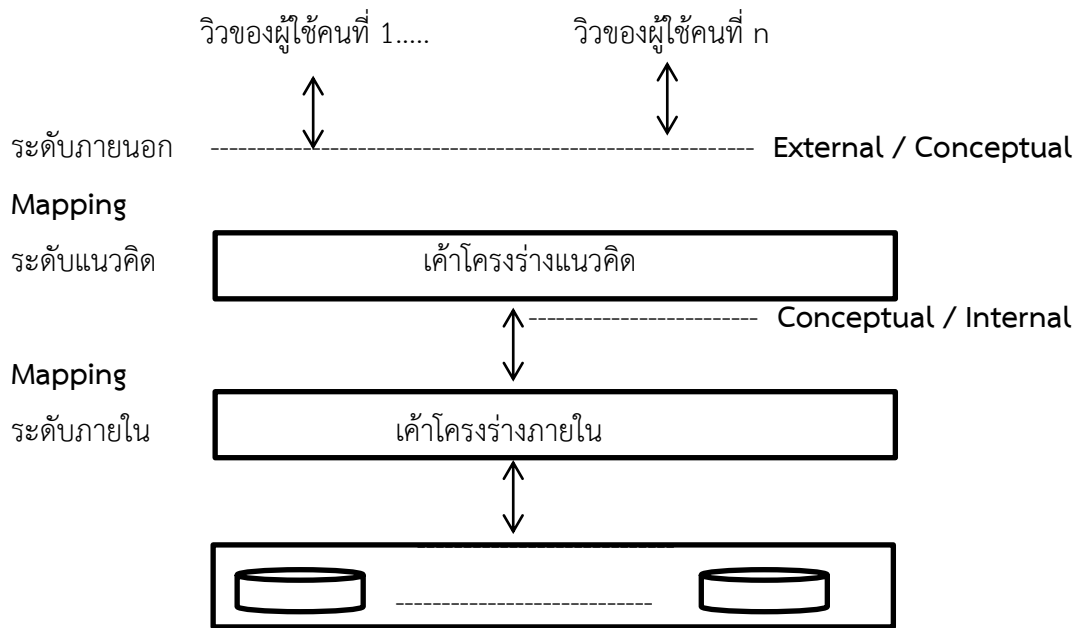
การแปลงส่ง (Mapping) แบบออกเป็น 2 ประเภท

1.การแปลงส่งระหว่างระดับภายนอกกับระดับแนวคิด

(External / Conceptual Mapping)

2.การแปลงส่งระหว่างระดับแนวคิดกับระดับภายใน

(Conceptual / Internal Mapping)



ภาพที่ 2.3 ฐานข้อมูล

การแบ่งระดับของข้อมูลทำให้เกิดความเป็นอิสระของข้อมูลโดยสามารถแบ่งเป็น 2 อย่าง

1. ความเป็นอิสระของข้อมูลเชิงตรรกะ (Logical Data Independence)

ความเป็นอิสระของข้อมูลในระดับภายนอกกับระดับแนวคิด โดยการเปลี่ยนแปลงข้อมูลที่เกิดขึ้นในระดับแนวคิด จะไม่ส่งผลกระทบต่อระดับภายนอก เช่น การเพิ่มแอททริบิวต์ไม่มีผลต่อโปรแกรมประยุกต์ที่เขียนขึ้น

2. ความเป็นอิสระของข้อมูลในเชิงกายภาพ (Physical Data Independence)

ความเป็นอิสระของข้อมูลในระดับแนวคิดกับระดับภายใน โดยการเปลี่ยนแปลงที่เกิดขึ้นในระดับภายใน จะไม่มีผลกระทบต่อระดับแนวคิด เช่น การเพิ่มประสิทธิภาพการเข้าถึงข้อมูลให้ไวขึ้น โดยการปรับปรุงเค้าโครงภายใน

ภาษาที่ใช้ในระบบฐานข้อมูล

แบ่งออกเป็น 3 ประเภท คือ

1. ภาษาที่ใช้ในการนิยามข้อมูล (Data Definition Language , DDL)

เป็นภาษาคำสั่งที่ใช้กำหนดโครงสร้างข้อมูลของฐานข้อมูล

2. ภาษาที่ใช้ในการดำเนินการจัดการข้อมูล (Data Manipulation Language , DML)

เป็นภาษาคำสั่งที่ใช้ในการเรียกใช้ข้อมูล การเปลี่ยนแปลงแก้ไขข้อมูล การเพิ่มหรือลบข้อมูล ภาษาที่ใช้เรียกกันว่า “ SQL ”

3. ภาษาที่ใช้ในการควบคุมข้อมูล (Data Control Language , DCL)

เป็นภาษาคำสั่งที่ใช้ในการควบคุม หรือป้องกันข้อมูล ที่เกิดจากการใช้ข้อมูลจากผู้ใช้หลายคน

ฐานข้อมูลเชิงสัมพันธ์ (RELATIONAL DATABASE)

ฐานข้อมูลเชิงสัมพันธ์

เป็นการจัดเก็บรวบรวมข้อมูลในลักษณะตาราง 2 มิติ คือ แถวและคอลัมน์ โดยมี การจัดเก็บอย่างเป็นระบบและกฎเกณฑ์ ซึ่งได้ผ่านกระบวนการทำรีเลชันให้เป็น บรรทัดฐาน(Normalized) เพื่อลดปัญหาที่จะเกิดขึ้นในการจัดเก็บและค้นหาข้อมูลการจัดเก็บฐานข้อมูลเชิงสัมพันธ์ จะทำให้ข้อมูลเป็นกลุ่มตาราง ผู้ใช้จะเรียนรู้และเข้าใจง่าย โดยไม่จำเป็นต้องรู้ว่า ข้อมูลจริงๆมีการจัดเก็บอย่างไร รวมถึงวิธีการเรียกใช้ข้อมูล ภาษาที่ใช้คือเอชคิวแอล (SQL) ซึ่งเป็นภาษาที่มีลักษณะคล้ายภาษาอังกฤษ ง่ายต่อการเขียนคำสั่งจึงเหมาะกับการใช้เป็นคำสั่งแบบสอบถาม

คุณสมบัติของตาราง

-ข้อมูลในแต่ละแถวจะไม่ซ้ำกันโดยระบบจัดการฐานข้อมูลจะควบคุมไม่ให้เกิดความซ้ำซ้อน

-ข้อมูลตารางในแต่ละแถวไม่มีความสำคัญในการเรียงลำดับ

-ในแต่ละคอลัมน์ของแถวๆ หนึ่งจะบรรจุข้อมูลได้เพียงค่าเดียว

-ค่าของข้อมูลในแต่ละคอลัมน์ จะบรรจุข้อมูลประเภทเดียวกัน

-คอลัมน์ทางเทคนิคจะใช้ Attribute

-แถวทางเทคนิค Tuple

ความหมายของคีย์ (key)

ข้อมูลเชิงสัมพันธ์ สามารถเชื่อมโยงข้อมูลด้วยการใช้คีย์ ซึ่งความหมายของคีย์ ก็จะใช้คุณสมบัติในการแบ่ง โดยแบ่งออกเป็น 2 ประเภทใหญ่ๆ ดังนี้

1.คีย์หลัก (Primary Key)

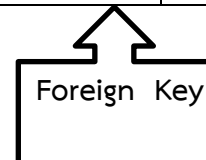
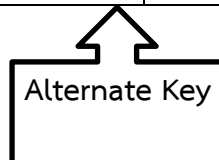
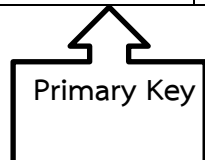
เป็นคีย์ที่เลือกจากแอททริบิวต์เดียวที่มีคุณสมบัติ คือ มีค่าที่ไม่ซ้ำซ้อนกัน
ในแต่ละแถว (Tuple) นั่นเอง

2.คีย์นอก (Foreign Key)

เป็นคีย์ที่ใช้อ้างอิง แอททริบิวต์เดียวกันในอีกตาราง (Relation) หนึ่ง
ซึ่งคีย์นอกนี้จะมีคุณสมบัติเป็นคีย์หลักในตารางอ้างอิง

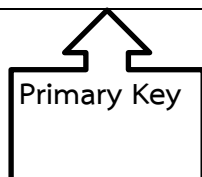
ตารางตัวอย่างของการแสดงข้อมูลพนักงาน

รหัสพนักงาน	ชื่อพนักงาน	นามสกุล	รหัสแผนก	เงินเดือน
4001001	ขจร	ชาติดี	01	7,500
4001002	ขจร	รุ่งทรัพย์	01	8,000
4001003	เล็ก	ใจดี	01	9,000
4002001	ปึก	พงศธร	02	8,500
4002002	โต	ใจร้อน	02	6,000



ตารางที่ 2.3 ตารางบุคลากร

รหัสแผนก	ชื่อแผนก	จำนวน
01	การเงิน	2
02	บัญชี	3
03	คอมพิวเตอร์	2



ตารางที่ 2.4 ตารางแผนก

การสร้างคีย์สามารถทำได้ในแต่ละตาราง ซึ่งแอททริบิวต์ใดไม่สามารถเป็นคีย์หลักได้ ก็จะทำกรรวมแอททริบิวต์ที่เป็นคีย์รอง (ALTERNATE KEY) มารวมกัน เพื่อให้เป็นคีย์หลักซึ่งมีค่าเป็นค่าที่ไม่ซ้ำกัน

(UNIQUE) โดยมีวิธีทำอยู่ 3 วิธี ดังนี้

1.แบบคีย์ผสม (COMPOSITE)

.เป็นการนำคีย์รอง (ALTERNATE KEY) ที่มากกว่า 2 คีย์ขึ้นไป นำมารวมกัน ทำให้เป็นคีย์หลัก (PRIMARY KEY)

2.แบบคีย์คู่แข่ง (CANDIDATE)

เป็นการนำคีย์หลัก (PRIMARY KEY) และคีย์รอง (ALTERNATE KEY) มารวมกัน และเมื่อตัดเอาคีย์รอง (ALTERNATE KEY) มารวมกันและเมื่อตัดคีย์รอง (ALTERNATE KEY) ออก ค่าก็ยังเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) การนำมารวมกัน เพื่อเป็นประโยชน์ในการค้นหา

3.แบบคีย์ประกอบ (COMBINED)

เป็นการนำคีย์รอง (ALTERNATE KEY) ที่มีมากกว่า 2 คีย์ขึ้นไป มารวมกัน เพื่อเป็นค่าที่ไม่ซ้ำกัน (UNIQUE) ทำเป็นคีย์หลัก (PRIMARY KEY) โดยหากขาดตัวใดไปแล้วค่าจะไม่เป็นค่าที่ไม่ซ้ำกัน (UNIQUE)

ฐานข้อมูลเชิงสัมพันธ์กับกฎของคีย์

-กฎความบูรณภาพของเอนทิตี (THE ENTITY INTEGRITY RULE)

คอลัมน์ใดที่เป็นคีย์หลัก จะมีค่าว่าง (NULL) ไม่ได้ เพราะจะไม่ทราบค่าที่แน่นอน

-กฎความบูรณภาพอ้างอิง (THE REFERENT INTEGRITY RULE)

การอ้างอิงของคีย์นอก จะต้องสามารถอ้างอิงให้ตรงกับค่าของคีย์หลัก เพื่อเชื่อมโยงข้อมูลระหว่างตารางได้

หลักการแก้ไขและลบข้อมูล

-การแก้ไขหรือลบข้อมูลแบบมีข้อจำกัด (RESTRICT)

การแก้ไขหรือลบข้อมูลได้ก็ต่อเมื่อ ไม่มีข้อมูลที่ถูกอ้างอิงโดยคีย์นอกของอีกหนึ่งตาราง

-การแก้ไขหรือลบข้อมูลแบบต่อเรียง (CASCADE)

การแก้ไขหรือลบข้อมูลแบบต่อเรียงของคีย์หลัก ระบบจะทำการลบข้อมูลของคีย์นอกในตารางที่มีคีย์นอกอ้างอิงอยู่

-การแก้ไขหรือลบข้อมูลโดยเปลี่ยนเป็นว่าง (NULLIFY)

การแก้ไขหรือลบข้อมูลโดยเปลี่ยนเป็นว่าง จะทำได้ก็ต่อเมื่อ มีการเปลี่ยนแปลงค่าคีย์นอกในข้อมูลที่ถูกอ้างอิงเป็นว่างเสียก่อน

-การแก้ไขหรือลบข้อมูลแบบใช้ค่าโดยปริยาย (DEFAULT)

จะเป็นการปรับค่าคีย์นอกที่กำหนดขึ้น ในกรณียกเลิกวิชาที่เปิดสอน เป็นปรับเปลี่ยนค่าเป็น “0” สำหรับค่าที่ไม่มีการเปิดวิชาเรียน

ประเภทตาราง

-ตารางของข้อมูลเชิงสัมพันธ์

จะมีตารางหลักที่ถูกกำหนดขึ้น เพื่อเก็บข้อมูลจริงจะทำการเรียกใช้ข้อมูลภายหลัง

-ตารางของข้อมูลเชิงสัมพันธ์แบบวิว (VIEW)

เป็นตารางที่ถูกสร้างขึ้นตามความต้องการของผู้ใช้แต่ละคน ตามงานที่ผู้ใช้กำหนดขึ้น หรือที่เราเรียกว่า “ ตารางสมมติ ”

แผนผังความสัมพันธ์ของข้อมูล (E-R DIAGRAM)

จะใช้เป็นตัวโครงร่างในการออกแบบฐานข้อมูล ซึ่งแบ่งเป็น 2 ส่วน ดังนี้

1. เอนทิตี (ENTITY)






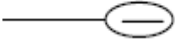
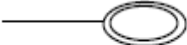

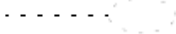


เป็นอ็อบเจกทีฟ (OBJECTIVE) อะไรก็ได้ที่เราสนใจที่จะนำมาเป็นหลักในการทำแฟ้มข้อมูลของเรื่องนั้น ซึ่งสามารถแบ่งออกได้ว่าเป็นเอนทิตี (ENTITY) เช่น โต๊ะ จะมีแอททริบิวต์เป็นหมายเลขครุภัณฑ์ , หนังสือ จะมีแอททริบิวต์เป็นหมายเลขหมวดหมู่

2. กลุ่มของเอนทิตี (ENTITY SET)

เป็นกลุ่ม (SET) ของเอนทิตี (ENTITY) ที่เป็นชนิดเดียวกัน เช่น กลุ่มของเอนทิตี(ENTITY SET)ของหนังสือ เป็นต้น

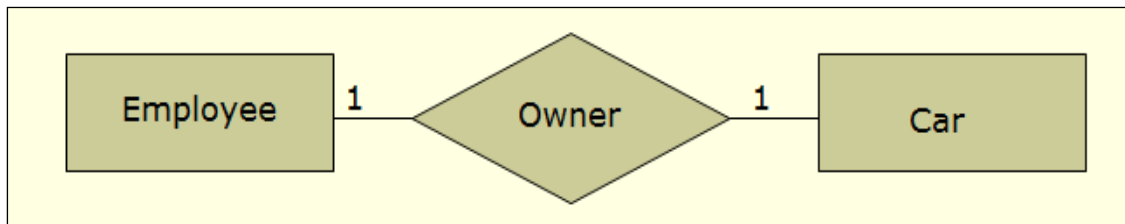
วิธีการเขียนแผนผังความสัมพันธ์ของข้อมูล (E-R DIARAM)

สัญลักษณ์ที่ใช้เขียนแผนผังความสัมพันธ์ของข้อมูล มีดังนี้

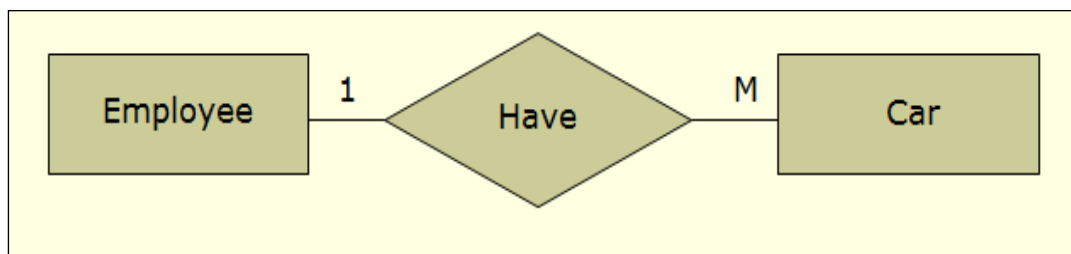
สัญลักษณ์	ความหมาย
	ENTITY
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERIVED ATTRIBUTE
	TOTAL PARTICIPATION of E_2 in R
	CARDINALITY RATIO 1:N สำหรับ E_1 : E_2 ใน R

ภาพที่ 2.4 แสดง วิธีการเขียนแผนผังความสัมพันธ์ของข้อมูล (E-R DIARAM)

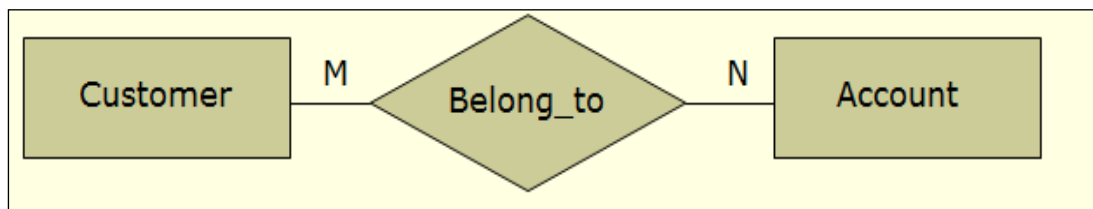
การเขียนแผนผังความสัมพันธ์ของข้อมูล (E-R DIARAM)



รูปที่ 2.5 ความสัมพันธ์ของสมาชิกของใน Entity หนึ่งไปสัมพันธ์กับ
สมาชิกของใน Entity อีกอันหนึ่ง แบบ 1:1



รูปที่ 2.6 เป็นความสัมพันธ์ของสมาชิกของใน Entity หนึ่งไปสัมพันธ์กับ
สมาชิกของใน Entity อีกอันหนึ่ง แบบ 1: M



รูปที่ 2.7 เป็นความสัมพันธ์ของสมาชิกมากกว่า 1 สมาชิก ของใน Entity หนึ่งไปสัมพันธ์กับ
สมาชิกของใน Entity อีกอันหนึ่ง แบบ M:N

ตัวอย่าง รูปแบบการเขียนโครงร่างเอนทิตี (ENTITY)

-EMPLOYEE (CODE,NAME,SURNAME,DEP,SARALY)

-CAR (COLOR, GENERATION , PRICE)

การเขียนรูปแบบความสัมพันธ์

การเขียนไดอะแกรม (DIAGRAM) แบบความสัมพันธ์โดยทั่วไปสามารถเขียนได้ 2 แบบ ดังนี้

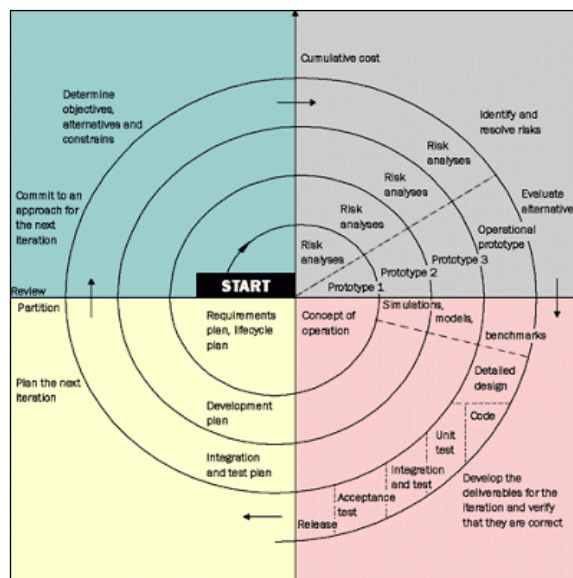
1. ความสัมพันธ์แบบทั่วไป (GENERALIZATION)
2. ความสัมพันธ์แบบประยุกต์ (APPLICATION)

2.2 ทฤษฎีกระบวนการผลิตซอฟต์แวร์

โมเดลแบบก้นหอย (Spiral Model)

กระบวนการพัฒนาซอฟต์แวร์ (Software Development Process)

ที่ถูกพัฒนาขึ้นโดยเอาจุดแข็งของการพัฒนาโมเดล (Development Model) อื่นที่ดีอยู่แล้วมาประยุกต์โมเดลแบบน้ำตก (Waterfall Model) โดยเพิ่มเติมส่วนของการวิเคราะห์และตีค่าความเสี่ยงที่เกิดขึ้นเพื่อจะได้ทราบว่าจุดใดมีความเสี่ยงมากน้อยเพียงใดจะได้หาวิธีลดความเสี่ยงซึ่งความเสี่ยงเป็นสาเหตุที่ทำให้การพัฒนาไม่ประสบความสำเร็จ การวิเคราะห์หรือต้นเหตุของความเสี่ยง เพื่อที่จะหาวิธีการที่จะทำให้เกิดความเสี่ยงน้อยที่สุด รวมถึงวิธีการแก้ไขเมื่อเกิดเหตุการณ์ที่คาดคิดเกิดขึ้น ถ้าความเสี่ยงน้อยลง ก็ทำให้ต้นทุน (Cost) หรือต้นทุนที่ใช้ ก็จะลดลงตามไปด้วย

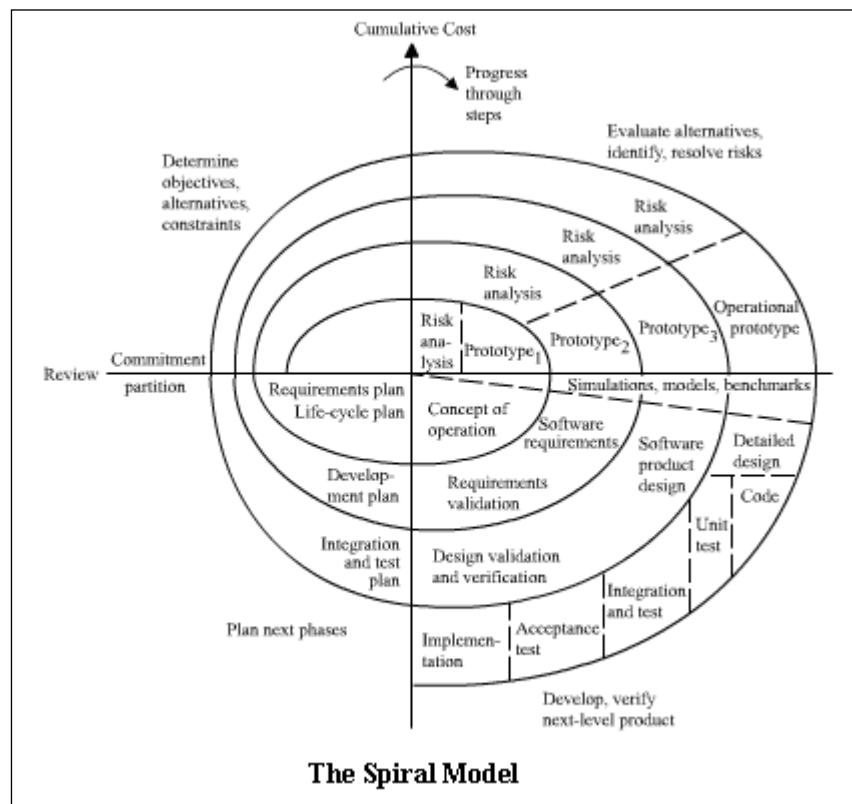


ภาพที่ 2.8 รูปแสดงหลักการทำงานของระบบโมเดลแบบก้นหอย (Spiral Model)

กระบวนการพัฒนาของโมเดลแบบก้นหอย

ถูกพัฒนามาจากโครงสร้างพื้นฐานของโมเดลแบบน้ำตก (Waterfall Model) ที่มีการแบ่งแยกขั้นตอน เช่น ขั้นตอนแนวคิดของการดำเนินงาน , ขั้นตอนความต้องการของซอฟต์แวร์,ขั้นตอนการออกแบบ,ขั้นตอนการเขียนคำสั่ง,ขั้นตอนการรวบรวม,ขั้นตอนการใช้ เป็นต้นเนื่องจากในโมเดลแบบน้ำตก (Waterfall Model) สามารถส่งผลลัพธ์ที่ได้กลับไปยังขั้นตอนก่อนหน้าโดยไม่ต้องมีการแก้ไขทุกขั้นตอนใหม่หมด แต่โมเดลแบบน้ำตก

(Waterfall Model) ในแต่ละขั้นตอนจะเกิดการตอบกลับ(Feedback) บ่อยครั้ง โมเดลแบบก้นหอย(Spiral Model) จึงถูกค้นพบความเสี่ยงและความเป็นไปได้ที่เกิดขึ้นตลอดจนหาแนวทางแก้ไขเมื่อเกิดข้อผิดพลาด



ภาพที่ 2.9 รูปกระบวนการซอฟต์แวร์ของโมเดลแบบก้นหอย

(Spiral Model of the Software Process)

โครงสร้างของโมเดลแบบก้นหอย (Spiral Model)

แสดงในรูปโดย

- รัศมีของวงกลม หมายถึง ต้นทุน (Cost) ที่เกิดขึ้นในกระบวนการพัฒนาซอฟต์แวร์ ถ้าจำนวนรอบ (Cycle) ที่มากขึ้นก็จะหมายถึง ต้นทุนของการพัฒนา ก็จะเพิ่มมากขึ้นเช่นกัน

- มุมของวงกลม หมายถึง ความก้าวหน้าในการปฏิบัติแต่ละขั้นตอนในแต่ละรอบได้สำเร็จ ส่วนต่างๆ ในแต่ละรอบของโมเดลแบบก้นหอย ประกอบด้วย

1.กำหนดระยะ (Determine Phase) เป็นส่วนที่ทำหน้าที่กำหนด

1.1 วัตถุประสงค์ (Objective) กำหนดผลลัพธ์ที่จะได้รับ

1.2 ทางเลือก (Alternative) ที่เป็นไปได้ของการนำไปใช้

(Implement) ทางเลือกอาจมีการนำสิ่งที่มีอยู่แล้วมาใช้ใหม่ (Reuse) , ทางเลือกในการจัดซื้อ

1.3 เงื่อนไข (Constraint) เป็นเงื่อนไขที่เกิดขึ้นในการพัฒนา

1.3.1 ขั้นตอนการประเมิน (Evaluate Phase)

จากทางเลือกทั้งหมดที่กำหนดเอาไว้ในขั้นตอนการประเมิน (Determine Phase) และเงื่อนไขที่ได้ตั้งไว้ เพื่อทราบว่า จุดใดในขบวนการที่เป็นจุดสำคัญที่ทำการวิเคราะห์ หรือ ประเมินค่าความเสี่ยงอาจทำได้โดยการทำต้นแบบ ประสิทธิภาพของความเสี่ยง (Performance Risk) และ ความต้องการของผู้ใช้ (User Requirement) , การจำลองสถานการณ์ (Simulate) เพื่อหาประสิทธิภาพในการทำต้นแบบ (Prototypes) เบื้องต้น เป็นความพยายามในการแก้ไขความเสี่ยงที่เกิดขึ้น ในกรณีของความเสี่ยงการเชื่อมต่อกับผู้ใช้ (User Interface Risk) และการควบคุมความเสี่ยง (Control Risk)

1.3.2 ขั้นตอนการพัฒนา

ตรวจสอบ (Develop , verify Phase) หลังจากการทำต้นแบบ (Prototype) เพื่อวิเคราะห์หาความเสี่ยงในเฟส(Phase) ถึงขั้นตอนในการพัฒนาซอฟต์แวร์ (Software Development Phase) ให้เป็นไปตามวัตถุประสงค์ที่ว่าความเสี่ยงที่เกิดจากหน้าประสานงานกับผู้ใช้งาน (User Interface) หรือตีค่าความเสี่ยงและวางแผนสำหรับรอบ (Cycle) ต่อไปของการพัฒนา ซึ่งจุดนี้เองทำให้ในขั้นตอน

การเปรียบเทียบโมเดลแบบกันหอยกับโมเดลกระบวนการพัฒนาซอฟต์แวร์อื่น

ข้อดี

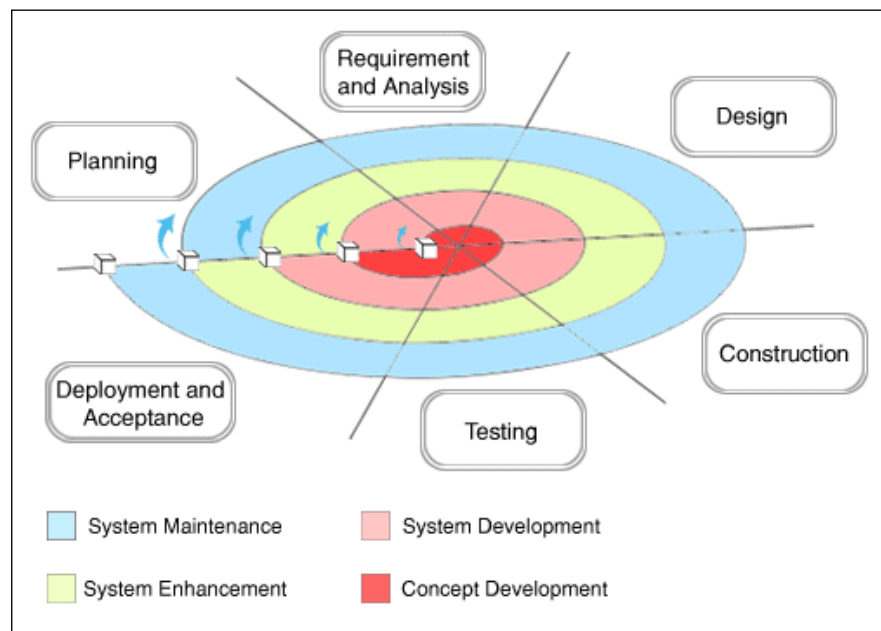
- ถ้าในโครงการ (Project) มีความเสี่ยงต่ำในด้านของหน้าประสานงานกับผู้ใช้ (User Interface) หรือมีประสิทธิภาพ (Performance) และมีความเสี่ยงสูงในแง่ของงบประมาณ (Budget) และระยะเวลายาวเหมือนกับเป็นโมเดลแบบน้ำตก (Waterfall Model)
- ถ้าความต้องการของซอฟต์แวร์ (Software Requirement) มีค่าค่อนข้างคงที่ คือไม่เปลี่ยนแปลงบ่อยจะเหมือนกับเป็นโมเดลแบบสองขา (Two-leg Model)
- ถ้าในโครงการ (Project) มีความเสี่ยงต่ำในแง่ของงบประมาณ (Budget) แต่มีความเสี่ยงสูงในแง่ของหน้าประสานผู้ใช้ (User Interface) ว่าจะไม่ตรงกับความต้องการจะเหมือนกับโมเดลแบบการประเมิน (Evaluation Model)
- ถ้าสามารถเปลี่ยนจากการประยุกต์ (Application) ไปเป็นซอฟต์แวร์ (Software) หรือคำสั่ง (Code) ได้จะเหมือนกับเป็นโมเดลแบบเปลี่ยน (Transform Model)
- ถ้ามีความเสี่ยงในหลายปัจจัยข้างต้น โมเดลแบบกันหอย (Spiral Model) จะช่วยให้เสี่ยงน้อยที่สุด คือมีความเหมาะสมในหลายๆปัจจัย

ข้อดีของโมเดลแบบก้นหอย (Spiral Model)

- 1.สนับสนุนการนำซอฟต์แวร์ (Software) กลับมาใช้อย่างเต็มตัว
- 2.ในแต่ละรอบ (Cycle) มีขั้นตอนประมวลผลที่สิ้นสุดภายในรอบเดียว
- 3.การวางแผนเพื่อกำหนดทางเดินของกระบวนการซอฟต์แวร์ (Software Process) ในรอบต่อไป
- 4.เนื่องจากการพัฒนาอยู่บนพื้นฐานของวัตถุประสงค์ที่ตั้งไว้ทำให้ผลลัพธ์ของผลิตภัณฑ์ซอฟต์แวร์ (Software Product) ตรงตามความต้องการของผู้ใช้
- 5.แก้ไขข้อผิดพลาดที่เกิดขึ้นตั้งแต่เนิ่นๆ
- 6.มีความเป็นอิสระต่อกัน ทางด้านการพัฒนาและการแก้ไข

ข้อเสียของโมเดลแบบก้นหอย (Spiral Model)

เนื่องจากโมเดลแบบก้นหอย (Spiral Model) ทุกรอบ (Cycle) ของการพัฒนา มีการวิเคราะห์และตีค่า ถ้าการวิเคราะห์เกิดความผิดพลาด จะทำให้ผลิตภัณฑ์ซอฟต์แวร์ (Software Produce) ที่ออกมาผิดพลาดทำให้เกิดค่าใช้จ่ายที่เพิ่มขึ้น



ภาพที่ 2.10 แสดงการทำงานของแต่ละส่วนของโมเดลแบบก้นหอย (Spiral Model)

2.3 ทฤษฎีหรือเทคโนโลยีและเทคโนโลยีที่ใช้ในการพัฒนา

ภาษาจาวา (JAVA)

ภาษาจาวา (Java) ย่อมาจาก “ Java programming language ” เป็นภาษาโปรแกรมเชิงวัตถุ ถูกพัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ของบริษัท Sun Microsystems ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยมีรูปแบบที่เพิ่มเติมขึ้น คล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน ซึ่งจุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้ ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

คุณลักษณะ

- โปรแกรมมีขนาดเล็กในระบบเครือข่าย การคอมไพล์จะแปลงโปรแกรมเป็น Java bytecode ซึ่งสามารถเรียกใช้งานได้ทุกที่ภายในเครือข่าย Java virtual machine เป็นตัวแปร bytecode ให้เป็นโปรแกรมเพื่อใช้งานบนเครื่องคอมพิวเตอร์ มีความหมายว่า platform ที่แตกต่างกันของคอมพิวเตอร์ สามารถใช้โปรแกรมนี้ได้
- คำสั่งเป็นแบบ "Robust" มีความหมายว่า อ็อบเจกต์ของ Java ไม่มีการอ้างอิงข้อมูลหรืออ็อบเจกต์จากภายนอกซึ่งแตกต่างจาก C++ และภาษาอื่น ๆ เป็นการทำให้มั่นใจไม่มีการเก็บตำแหน่งของข้อมูล ในโปรแกรมประยุกต์อื่น หรือในระบบปฏิบัติการที่ทำให้โปรแกรมไม่ทำงาน Java virtual machine ทำการตรวจสอบแต่ละอ็อบเจกต์ที่ใช้ในโปรแกรม
- Java เป็นเหมือนกับอ็อบเจกต์หนึ่ง สามารถได้รับประโยชน์จาก class หรือคำสั่งเนื่องจากอ็อบเจกต์มีคุณสมบัติเป็น "นาม" ซึ่งทำให้ติดต่อกับผู้ใช้ได้ ในขณะนี้ภาษาดั้งเดิมมีคุณสมบัติเป็น "กริยา" ดังนั้น method จะได้รับการรับรู้เป็นความสามารถของอ็อบเจกต์หรือพฤติกรรม
- การประมวลผลทำที่เครื่องลูกข่าย ดังนั้น Java applet มีคุณลักษณะในการออกแบบให้ทำงานได้เร็ว Java ง่ายกว่า C++ โดยเปรียบเทียบ

ข้อดีของ ภาษา Java

- ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้คำหรือชื่อต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น
- โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารถูก compile และ run บนเครื่องพีซีธรรมดาได้
- ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอน compile time และ runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ debug โปรแกรมได้ง่าย
- ภาษาจาวามีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น
- ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะ Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificates ของ
- มี IDE, application server, และ library ต่าง ๆ มากมายสำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ tool และ s/w ต่างๆ

ข้อเสียของ ภาษา Java

- ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง)หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่งทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา
- tool ที่มีในการใช้พัฒนาโปรแกรมจาวามักไม่ค่อยเก่ง ทำให้หลายอย่างโปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ tool ทำไม่ได้ ถ้าเราดู tool ของ MS จะใช้งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า (แต่เราต้องซื้อ tool ของ MS และก็ต้องรันบน platform ของ MS)

ตัวแปรและชนิดข้อมูล

ตัวแปร (Variable) จะเป็นชื่อของหน่วยความจำสำหรับใช้เก็บข้อมูลชนิดต่างๆ และสามารถนำออกมาใช้ได้ ตัวแปรจะประกอบด้วย

-ชื่อ

-ชนิดข้อมูล

-ค่า

ก่อนที่จะใช้ตัวแปรนั้นต้องประกาศก่อนว่าเป็นตัวแปรชนิดข้อมูลอะไร แล้วจึงกำหนดค่าให้มัน จาวามีตัวแปรสามชนิดคือ

- 1.ตัวแปรของ instance (instance variables) หรือตัวแปรของออบเจกต์
- 2.ตัวแปรของคลาส (class variables)
- 3.ตัวแปรประเภทโลคอลล (local variables) หรือตัวแปรประเภทท้องถิ่น

การประกาศตัวแปร (Declaring Variables)

การใช้ตัวแปรใดๆ ในโปรแกรมของจาวาต้องประกาศก่อนที่จะใช้มัน การประกาศตัวแปรจะประกอบด้วยชนิดข้อมูลและชื่อตัวแปรดังนี้

```
String Name;           // ประกาศ Name เป็นตัวแปรชนิดสตริง
int Salary;            // ประกาศ Salary เป็นตัวแปรชนิดจำนวนเต็ม
int myAge, mySalary;   // ประกาศ myAge และ my Salary เป็นตัวแปรชนิด
                        // จำนวนเต็ม

int myAge=15; // ประกาศ myAge เป็นตัวแปรชนิดจำนวนเต็มและให้มีค่า 15
```

การตั้งชื่อตัวแปร (Notes on Variable Names)

การตั้งชื่อตัวแปรในภาษาจาวามีกฎเกณฑ์ดังนี้

- เริ่มต้นตัวอักษรหรือ underscore (_) หรือเครื่องหมายดอลลาร์ (\$)
- ห้ามเริ่มชื่อตัวแปรด้วยตัวเลข
- ภายหลังจากใช้อักษรเริ่มตัวแรกขอลตัวแปรแล้ว หลังจากนั้นสามารถใช้ตัวอักษรใดๆ หรือตัวเลขหรือสัญลักษณ์ต่างๆ ได้
- การใช้อักษรตัวพิมพ์ใหญ่และอักษรตัวพิมพ์เล็กมีความแตกต่างกัน

ชนิดของตัวแปร (Variable Types)

ชื่อของตัวแปรแต่ละชนิดต้องระบุชนิดของตัวแปรเมื่อเราประกาศก่อนนำไปใช้เก็บข้อมูลแต่ละชนิด

ชนิดของตัวแปรแบ่งเป็น 4 ประเภทใหญ่ๆ ดังนี้

- จำนวนเต็ม (integers) ได้แก่ byte, short, int, long
- เลขทศนิยม (floating point numbers) ได้แก่ floating, double
- อักขระ (characters) ได้แก่ char
- บูลีน (boolean) ได้แก่ boolean จะให้ค่า true หรือ false

นิพจน์และตัวดำเนินการ (Expressions and Operators)

- นิพจน์ (Expressions) เป็นรูปแบบของคำสั่งแบบธรรมดาในภาษาจาวาซึ่งช่วยดำเนินการส่งค่ากลับ

- ตัวดำเนินการ (operators) เป็นสัญลักษณ์พิเศษที่นำไปใช้ในนิพจน์

ตัวดำเนินการเลขคณิต (Arithmetic Operators)

ตัวดำเนินการ (operator)	ความหมาย	ตัวอย่าง
+	การบวก	1 + 5
-	การลบ	6 - 4
*	การคูณ	2 * 4
/	การหาร (Division)	12 / 4 (ผลหาร 3)
%	การหาร (Modulus)	15 % 7 (ผลหาร 1 คัดเฉพาะเศษ)

ตัวดำเนินการเปรียบเทียบ (Comparison Operators)

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
==	เท่ากัน	X == 100;
!=	ไม่เท่ากัน	X != 9;
<	น้อยกว่า	X < 9;
>	มากกว่า	X > 2;
<=	น้อยกว่าหรือเท่ากับ	X <= 15;
>=	มากกว่าหรือเท่ากับ	X >= 20;

ตัวดำเนินการตรรก (Logical Operators)

ตัวดำเนินการ	ความหมาย	ตัวอย่าง
&&	AND	X > 2 && x < 10
	OR	X > 10 x < 2
!	NOT	!(x > 6)

ตัวดำเนินการ	ความหมาย
&	Bitwise AND
	Bitwise OR
^	Bitwise XOR
<<	Left shift
>>	Right shift
>>>	Zero fill right shift
~	Bitwise complement
<<=	Left shift assignment ($x = x \ll y$)
>>=	Right shift assignment ($x = x \gg y$)
>>>=	Zero fill right shift assignment ($x = x \ggg y$)
$X \&= y$	AND assignment ($x = x \& y$)
$X = y$	OR assignment ($x = x y$)
$X \wedge= y$	XOR assignment ($x = x \wedge y$)

ตารางที่ 2.5 แสดงตัวดำเนินการบิตไวส์ (Bitwise Operators) ใช้ดำเนินการกับแต่ละบิตในจำนวนเต็ม

อะเรย์ (Array)

อะเรย์เป็นกลุ่มของตัวแปรต่างมิติเดียวหรือหลายมิติเป็นวิธีการที่จะบรรจุรายการของ
 ออปเจกต์หรือชนิดข้อมูลดั้งเดิม แต่ละอะเรย์จะบรรจุแต่ละสมาชิกที่สามารถแทนค่าหรือเปลี่ยนแปลง
 เราไม่สามารถบรรจุชนิดข้อมูลที่แตกต่างกันในเดียวกัน อะเรย์มีขั้นตอนการสร้างดังนี้

การประกาศตัวแปรต่างๆของอะเรย์ (Declare Array Variables)

```
String str[]; หรือ String[] str; หรือ String []str;
int temp[]; หรือ int[] temp; หรือ int []temp;
```

การสร้างออบเจ็กต์ของอะเรย์ (Crewting Arrays Objects)

```
String[] str;
Str = new String[10];
หรือ
String[] str = new String[10];
```

ตัวดำเนินการ new จะช่วยจองพื้นที่ในหน่วยความจำให้ออบเจ็กต์ของอะเรย์ str[] ใหม่ของสตริง พร้อมกำหนดให้ 10 ที่สำหรับบรรจุสมาชิกต่างๆ การเข้าถึงสมาชิกต่างๆของอะเรย์ (Accessing Array Elements) สามารถทดสอบและเปลี่ยนแปลงค่าในแต่ละห้องของอะเรย์นั้นๆการนำค่าออกมาจากภายในอะเรย์ จะต้องใช้นิพจน์ของสับสคริป (subscript) ของอะเรย์ ดังนี้

```
anyArray[subscript];
```

anyArray เป็นส่วนหนึ่งของนิพจน์นี้ ซึ่งเป็นตัวแปรบรรจุออบเจ็กต์ของอะเรย์แม้ว่ามันสามารถเป็นนิพจน์ที่ให้ผลลัพธ์ในอะเรย์ นิพจน์นี้จะระบุห้องต่างๆในอะเรย์ที่จะเข้าถึงสับสคริป (subscript) ของอะเรย์จะเริ่มต้นด้วย 0 ดังนั้นเราสามารถเข้าถึงแต่ละสมาชิกของอะเรย์ได้โดยใช้สับสคริปที่เป็นเลขจำนวนเต็มระบุลำดับของสมาชิก

```
String[] myArr = new String[9];
myArr[0], myArr[1],..., myArr[8]
```

การเปลี่ยนสมาชิกต่างๆของอะเรย์ (Changing Array Elements)

```
numArray[1] = 14;
MyString[0] = "Hello";
MyString[10] = MyString[0];
```

อะเรย์หลายมิติ (Multidimensional Arrays)

```
int point[][] = new int[10][10]; // ประกาศตัวแปรอะเรย์ point พร้อมกำหนด
                                หน่วยความจำบรรจุสมาชิกของอะเรย์จำนวน
                                100 สมาชิก
point[0][0] = 4;                // กำหนดค่าให้สมาชิกของอะเรย์
```

คำสั่งบล็อก (Block Statements)

คำสั่งบล็อกเป็นกลุ่มของคำสั่งอื่นๆที่ล้อมรอบด้วยวงเล็บ({}) ท่านสามารถใช้บล็อกกับคำสั่งต่างๆในโปรแกรม สามารถสร้างบล็อกใหม่ของคำสั่งบรรจุในบล็อกเดิมได้ นั่นคือสามารถประกาศและใช้ตัวแปรแบบโลคอลภายในบล็อกและตัวแปรต่างๆเหล่านั้นจะถูกทำลายเมื่อบล็อกนั้นสิ้นสุดการประมวลผล

เงื่อนไขของ if (if Conditionals)

เงื่อนไขของ if เขียนอยู่ในรูปแบบต่อไปนี้

```
if (นิพจน์ที่เป็นเงื่อนไข) // ถ้าเงื่อนไขเป็นจริง (true) จะทำคำสั่ง 1 แต่
ถ้าคำสั่ง 1;              // เป็นเท็จ (false) จะข้ามคำสั่ง 1 ไปทำคำสั่งถัดไป
```

หรือ

```
if (นิพจน์ที่เป็นเงื่อนไข) // ถ้าเงื่อนไขเป็นจริง (true) จะทำคำสั่งที่อยู่ในบล็อก
{                          // แต่ถ้าเป็นเท็จ (false) จะกระโดดออกจากบล็อก
คำสั่ง 1;                  // ไปทำคำสั่งถัดไป
คำสั่ง 2;
...
}
```

เงื่อนไข switch (Switch Conditionals)

คำสั่ง switch จะช่วยเลือกทำคำสั่งใดคำสั่งหนึ่งตามต้องการโดยมีทางเลือกให้ทำคำสั่งหลายๆทาง

```
switch (oper)                // เลือกทำงานตามค่าของ oper
{
    // เริ่มต้นคำสั่ง switch
    case '+' :
        add(val1,val2);      // กรณีที่ oper เป็น '+' ให้เรียกใช้ add() method
        break;               // ยุติการประมวลผลแล้วข้ามไปที่คำสั่งถัดจาก

    // คำสั่ง switch
    case '-':
        sub(val1,val2);
        break;

        default:
            System.out.println("Unknown operator"); // เลือกทำคำสั่งนี้เมื่อ
                                                    oper ไม่อยู่ในกรณีใดๆ

    }                        // สิ้นสุดคำสั่ง switch
```

ลูป for (for Loops)

ช่วยทำคำสั่งหรือบล็อกของคำสั่งที่ต้องการซ้ำแล้วซ้ำอีก จนพบเงื่อนไขที่กำหนดจึงออกจากลูป

```
for ( ค่าเริ่มต้น; เงื่อนไข; ค่าเพิ่มขึ้นหรือค่าลดลง)
{
    คำสั่งต่างๆ;
}
```

การเริ่มต้นลูป for มี 3 ส่วนดังนี้

-ค่าเริ่มแรกของลูป for เป็นนิพจน์ เช่น `i=0` หรือ `int i=0` ตัวแปรที่ประกาศในส่วนนี้ของลูป for เป็นประเภทโลคอล จะถูกทำลายทิ้งเมื่อสิ้นสุดการประมวลผลของลูป

-เงื่อนไข (condition) จะเกิดขึ้นภายหลังที่ลูป for วนรอบทำงานในแต่ละครั้ง เงื่อนไขต้องเป็นนิพจน์หรือฟังก์ชันที่ให้ค่าชนิด Boolean

-ค่าเพิ่มขึ้นหรือลดลง เป็นนิพจน์ใดๆ

การออกนอกลูป (Breaking Out of Loops)

ลูปทั้งหมดจะสิ้นสุดการทำงานตามเงื่อนไขที่กำหนดไว้ บางกรณีอาจต้องออกจากลูปในขณะที่ลูปกำลังวนรอบทำงาน สามารถใช้คีย์เวิร์ด

`break;` และ `continue;`

การจัดการเหตุการณ์ธรรมดาและการโต้ตอบการกดปุ่มเมาส์และการปล่อยปุ่มเมาส์ (`mouseDown` and `mouseUp`) ใน applet method ทั้งสองจะถูกเรียกใช้ทันทีที่เกิดเหตุการณ์ ต่อไปนี้เป็นรูปแบบของ `mouseDown()` method เมื่อเกิดเหตุการณ์ `mouseDown`

```
public boolean mouseDown(Event myEvent, int x, int y)
{
    // กิจกรรม
    ...
}
```

`myEvent` คือ เหตุการณ์ต่างๆของระบบทั้งหมดทำให้เกิด instance ของคลาส `Event` `x` และ `y` คือ พิกัดของตำแหน่งที่เราคลิกเมาส์บนจอภาพ `mouseUp()` method ถูกเรียกใช้งาน เมื่อผู้ใช้ปล่อยปุ่มของเมาส์ รูปแบบการเขียนจะคล้ายกับ `mouseDown()` method

ภาษาเอสคิวแอล (SQL)

ภาษาที่กลายเป็นภาษามาตรฐานสำหรับระบบฐานข้อมูลคือ ภาษาเอสคิวแอล (SQL) ย่อมาจาก (Standard Query Language) / ซีควอล (SE-QUEL) ซึ่งเป็นภาษาที่พัฒนาขึ้นมาโดยบริษัทไอบีเอ็ม ภาษา SQL (Standard Query Language) เป็นส่วนหนึ่งของระบบฐานข้อมูลแบบรีเลชันเนล (Relational Database) ที่ได้รับความนิยมมาก เพราะง่ายต่อความเข้าใจและอยู่ในรูปภาษาอังกฤษ

ภาษาSQL แบ่งออกเป็น 2 ส่วน

- 1.ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)
- 2.ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)
 - 2.1ภาษาควบคุม (Control Language)
 - 2.2ภาษาในการเลือกข้อมูล (Data Query Language)

รูปแบบการใช้คำสั่ง SQL สามารถแบ่งเป็น 2 รูปแบบ

- 1.คำสั่ง SQL ที่ใช้เรียกดูข้อมูลได้ทันที (Interactive SQL)
เป็นการเรียกใช้คำสั่ง SQL สั่งงานบนจอภาพ เพื่อเรียกดูข้อมูลในขณะที่ทำงานได้ทันที เช่น

```
SELECT CITY
FROM SUPPLIER
WHERE SNO = 'SE';
```
- 2.คำสั่ง SQL ที่ใช้เขียนรวมกันกับโปรแกรมอื่นๆ (Embedded SQL)
เป็นคำสั่ง SQL ที่ใช้รวมกับคำสั่งของโปรแกรมภาษาต่างๆ เช่น PL/1 PASCAL ฯลฯ หรือแม้แต่กับคำสั่งในโปรแกรมที่ระบบจัดการฐานข้อมูลนั้นมีใช้เฉพาะ เช่น ORACLE มี PL/SQL (Procedural Language /SQL) ที่สามารถเขียนโปรแกรมและนำคำสั่ง SQL มาเขียนรวมด้วย เป็นต้น

ตัวอย่างการใช้คำสั่ง SQL ในภาษา PL/1

```
EXEC SQL SELECT CITY
INTO :XCITY
FROM SUPPLIER
WHERE SNO = 'S4';
```


ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)

Data Definition Language (DDL) เป็นภาษาที่ใช้นิยามโครงสร้างข้อมูลเพื่อเปลี่ยนแปลง หรือยกเลิกโครงสร้างข้อมูล ตามที่ออกแบบไว้โครงสร้างดังกล่าว คือ สคีมา (Schema) นั้นเอง

ตัวอย่างเช่น การกำหนดให้ฐานข้อมูลประกอบด้วยตารางอะไรบ้าง ชื่ออะไร ประเภทใดมีอินเด็กซ์ (Index)

ภาษา DDL ประกอบด้วย 3 คำสั่งคือ

1. คำสั่งการสร้าง (Create) ได้แก่ การสร้างตารางและอินเด็กซ์

```
CREATE TABLE <Table name>
( Attribute 1 Type 1,
Attribute 2 Type 2 ,
)
CREATE Unique Index on X<Table name>
เช่น
```

```
CREATE TABLE S11
(SNO CHAR(5) Not NULL,
SNAME CHAR(10) ,
STATUS integer
)
CREATE Unique Index XS11 on S11(SNO)
```

2. คำสั่งเปลี่ยนแปลงโครงสร้าง

```
ALTER TABLE < ชื่อตารางที่ตั้งขึ้น >
<คำสั่งการเปลี่ยนแปลง> (<ชื่อคอลัมน์ประเภทข้อมูล>);
เช่น
```

```
ALTER TABLE SUPPLIER
ADD (LAST_SNAME Char(10));
```

3. คำสั่งยกเลิก (Drop) ต่างๆ

```
การลบโครงสร้างตาราง
DROP TABLE < ชื่อตารางที่ตั้งขึ้น >
```

ภาษาดังกล่าวคือ ภาษาที่ใช้สร้างฐานข้อมูลลงในคอมพิวเตอร์หลังจากที่เราได้ออกแบบแล้วว่าฐานข้อมูลมีกี่รีเลชั่น แต่ละรีเลชั่นมีความสัมพันธ์อย่างไร จากนั้นการใช้ภาษา DDL นี้แปลงรีเลชั่นต่างๆ ให้อยู่ในรูปภาษาสำหรับนิยามข้อมูล เพื่อบันทึกเข้าสู่ระบบฐานข้อมูล เพื่อสร้างฐานข้อมูลที่แท้จริง ให้เกิดขึ้นในคอมพิวเตอร์ ภาษา DDL สามารถสรุปคำสั่งต่างๆได้ดังต่อไปนี้

คำสั่ง	ความหมาย
CREATE TABLE	นิยามโครงสร้างข้อมูลในรูปตารางบนฐานข้อมูล
DROP TABLE	ลบโครงสร้างตารางข้อมูลออกจากระบบ
ALTER TABLE	แก้ไขปรับปรุงโครงสร้างตาราง
CREATE INDEX	สร้างดัชนีของตาราง
DROP INDEX	ลบ ดัชนีของตารางออกจากระบบ
CREATE VIEW	กำหนดโครงสร้างวิวของผู้ใช้
DROP VIEW	ลบโครงสร้างวิวออกจากระบบ

ตารางที่ 2.6 แสดงคอมพิวเตอร์ภาษา DDL สามารถสรุปคำสั่งต่างๆ

คำสั่งนิยามโครงสร้างตาราง

การสร้างตารางในฐานข้อมูลแบบรีเลชันเนลโดยเฉพาะฐานข้อมูลขนาดใหญ่บนระบบ UNIX จะทำได้ด้วยการป้อนคำสั่งในลักษณะเท็กซ์โหมด (Text Mode) เข้าไปในระบบฐานข้อมูล

CREATE TABLE <ชื่อตาราง >

(<ชื่อคอลัมน์ประเภทของข้อมูล>[,<ชื่อคอลัมน์ประเภทของข้อมูล>]....);

ประเภทของข้อมูล

ประเภทของข้อมูลแบ่งเป็น 5 ประเภทใหญ่ๆ ขึ้นอยู่กับซอฟต์แวร์ระบบฐานข้อมูลที่ใ้ใช้คืออะไร ตัวอย่างเช่น CHAR, INTEGER, DATE ฯลฯ

1. คำสั่งการลบโครงสร้างตาราง

DROP TABLE <ชื่อตารางที่ต้องการลบ>

2. คำสั่งการเปลี่ยนแปลงโครงสร้างตาราง

ในกรณีที่ต้องการเปลี่ยนแปลงโครงสร้างตารางที่เคยนิยามไว้สามารถใช้คำสั่งต่อไปนี้

ALTER TABLE <ชื่อตารางที่ต้องการเปลี่ยนแปลง>

<คำสั่งการเปลี่ยนแปลง> <[,<ชื่อคอลัมน์ประเภทของข้อมูล>]>

3. คำสั่งดัชนี

ดัชนี (INDEX) มีความสำคัญมากต่อฐานข้อมูลเชิงสัมพันธ์ เนื่องจากระบบฐานข้อมูล แบบรีเลชันเนล (RDBMS) จะใช้ดัชนีในการค้นหาระเบียนที่ต้องการได้อย่างรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้น จะเก็บไว้แยกจากตารางในพื้นที่ต่างหากของคอมพิวเตอร์ โดยปกติถ้าไม่มีการประกาศดัชนีไว้ การค้นหาข้อมูลในตาราง นั้นจะต้องทำแบบเรียงลำดับจากแถวที่หนึ่งจนถึงแถวสุดท้ายการสร้างดัชนีสำหรับตารางใดๆ จะทำได้โดยการเลือกคอลัมน์ใดคอลัมน์หนึ่งจากตารางมาเป็นดัชนีและตารางหนึ่งๆ สามารถมีได้หลายดัชนี นอกจากเพิ่มความรวดเร็วในการดึงข้อมูลแล้ว ยังสามารถนำไปใช้ในการควบคุม คอลัมน์ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลที่ไม่ซ้ำกัน (Unique) อีกด้วย

ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML)

หลังจากที่เราสร้างโครงสร้างฐานข้อมูลขึ้นแล้วคำสั่งต่อไปในการป้อนข้อมูลลงในฐานข้อมูลและเปลี่ยนแปลงข้อมูลในฐานข้อมูลโดยใช้ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language -DML) ใช้จัดการข้อมูลภายในตารางภายในฐานข้อมูลและภาษาแก้ไขเปลี่ยนแปลงตาราง

แบ่งออกเป็น 4 Statement คือ

- 1.Select Statement : การเรียกหา(Retrieve)ข้อมูลจากฐานข้อมูล
- 2.Insert Statement : การเพิ่มเติมข้อมูลในตาราง(Table)จากฐานข้อมูล
- 3.Delete Statement:การลบข้อมูลออกจากตาราง(Table)จากฐานข้อมูล
- 4.Update Statement: การเปลี่ยนแปลงข้อมูลในตาราง(Table)จากฐานข้อมูล

คำสั่ง	ความหมาย
SELECT	เรียกค้นข้อมูลในตาราง
INSERT	เพิ่มแถวข้อมูลลงในตาราง
DELETE	ลบแถวข้อมูลในตาราง
UPDATE	ปรับปรุงแถวข้อมูลในตาราง

ตารางที่ 2.7แสดงชุดคำสั่งStatement

คำสั่งค้นหาข้อมูล (Query Statement)

คำสั่ง SELECT เป็นคำสั่งการเรียกดูข้อมูลหรือค้นข้อมูลตามเงื่อนไขที่ระบุ เนื่องจากคำสั่งSELECT เป็นคำสั่งที่มีรูปแบบการใช้งานที่ง่าย เพื่อช่วยในการค้นหาข้อมูลที่ซับซ้อน มีรูปแบบดังนี้

```

SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>
FROM <ชื่อตาราง>
WHERE <เงื่อนไขตามระบุ>
  
```

SELEC

เป็นคำสั่งในการเรียกดูข้อมูลในคอลัมน์ที่ระบุซึ่งอาจจะมากกว่าหนึ่งก็ได้ และถ้ามีมากกว่าหนึ่งคอลัมน์ ต้องคั่นด้วยเครื่องหมายคอมมา (,) และ นอกจากนี้ยังสามารถใช้เครื่องหมายดอกจัน (*) เพื่อแสดงถึงการขอข้อมูลทั้งหมดได้อีกด้วย

FROM

เป็นส่วนประกอบของคำสั่งที่บอกถึงตารางที่ต้องการดูซึ่งอาจจะมีมากกว่าหนึ่งตารางก็ได้ที่จะถูกเรียกใช้จากคำสั่ง SELECT

WHERE

เป็นส่วนประกอบของคำสั่ง ที่ใช้บ่งบอกเงื่อนไขที่จะใช้ในการค้นหาข้อมูล ขึ้นมาจากตารางใด ๆ ที่อยู่หลัง FROM นี้

การเรียกดูแบบซ้อนทับกัน (Nested SELECT Statement)

```
SELECT <ชื่อคอลัมน์>
FROM <ชื่อตาราง>
WHERE <ชื่อคอลัมน์> IN
( SELECT <ชื่อคอลัมน์>
FROM <ชื่อตาราง>
WHERE <ชื่อคอลัมน์> )
```

คำสั่งเติมข้อมูล (Insert Statement)

```
INSERT INTO < ชื่อตาราง >
VALUES (< ชื่อคอลัมน์_1> [<ชื่อคอลัมน์_2>]...);
```

คำสั่งแก้ไขและลบแถว (Update Statement)

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

ภาษาควบคุม (Control Language)

เป็นภาษาที่ใช้ควบคุมระบบรักษาความปลอดภัยของฐานข้อมูล ประกอบด้วย 2 คำสั่งคือ

1. คำสั่ง GRANT เป็นคำสั่งที่ใช้กำหนดสิทธิให้กับผู้ใช้แต่ละคนให้มีสิทธิกระทำการใดกับข้อมูล เช่น การเพิ่มข้อมูล การแก้ไข หรือ การลบข้อมูลในตารางใดบ้าง

2. คำสั่ง REVOKE เป็นคำสั่งให้มีการยกเลิกสิทธินั้นหลังจากที่ได้ GRANT แล้ว

คาบูลีน (Boolean-Type Data)

- AND - OR
- NOT - =
- > หรือ >= - < หรือ =<
- <> - Built-In Function
- COUNT - SUM
- AVG - MAX
- MIN - DISTINCT

ชุดคำสั่ง GROUP BY, ORDER BY และ HAVING

เนื่องจากข้อมูลที่สนใจมักจะเป็นผลสรุปหรือข้อมูลที่ใช้ในการตัดสินใจ ภาษา SQL จึงได้ออกแบบให้มีชุดคำสั่งพิเศษ ที่ทำหน้าที่แบ่งออกเป็นกลุ่มๆ ได้แก่

1.1 ชุดคำสั่ง GROUP BY

GROUP BY เป็นคำสั่งให้มีการจัดกลุ่มแถวข้อมูลตามคอลัมน์ โดยข้อมูลที่เหมือนกันจะถูกจัดให้อยู่ในกลุ่มเดียวกัน

```
SELECT S# , MAX( QTY)
```

```
FROM SPJ
```

```
GROUP BY S#
```

ผลลัพธ์

S#	QTY
S1	700
S2	800
S3	500
S4	300
S5	800

ตารางที่ 2.8 แสดงผลลัพธ์ของชุดคำสั่ง GROUP BY

1.2 ชุดคำสั่ง ORDER BY

ORDER BY เป็นคำสั่งให้มีการเรียงลำดับข้อมูลในแถวข้อมูลตามคอลัมน์ โดยข้อมูลที่เหมือนกันจะถูกจัดเรียงจากน้อยไปมากถ้าไม่การระบุแต่ถ้าระบุว่าเป็น DESC จะเรียงจากมากไปน้อย ดังนั้นจากตัวอย่างต่อไปนี้

```
SELECT S# , MAX( QTY)
```

```
FROM SPJ
```

```
GROUP BY S#
```

```
ORDER BY 2 DESC
```

S#	QTY
S5	800
S2	800
S1	700
S3	500
S4	300

ตารางที่ 2.9 แสดงผลลัพธ์ของชุดคำสั่ง ORDER BY

หมายเหตุ ตัวเลข 2 ที่อยู่หลัง ORDER BY จะเป็น การบอกระบบให้ทำการเรียงลำดับข้อมูลตามลำดับคอลัมน์ที่ 2 ของคำสั่ง SQL ซึ่งก็คือ MAX(QTY) นั่นเอง

1.3 ชุดคำสั่ง HAVING

HAVING เหมือนกับคำสั่ง WHERE ที่ต้องตามด้วยเงื่อนไข แต่ HAVING จะใช้ในกรณีที่มีการจัดกลุ่มหรือการใช้ GROUP BY

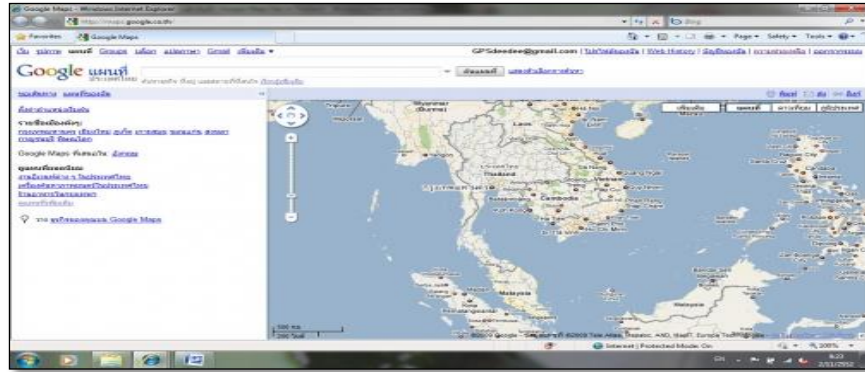
สรุปชุดคำสั่งมาตรฐานของ SQL

```
1. CREATE TABLE [[database.]owner.]table_name
  (column_name datatype [not null | null] IDENTITY [(seed,
  increment)][constraint]
  [, column_name datatype [not null | null] IDENTITY [(seed,
  increment)]]].
  [constraint] ...)
  [ON segment name]
```


2. DELETE [FROM] table_name
WHERE column_name = "value"
3. ALTER TABLE [[<database.>]<owner.>]<table_name>
ADD <column_name><datatype>NULL [Canstsaint]
[WITH nocmeck]
{drop []
4. UPDATE table_name
SET column_name= value
WHERE column_name =operator_value
5. CREATE VIEW View_name AS
SELECT column
FROM table_1
WHERE table_key_1=table_key_2

แผนที่กูเกิล (Google Maps)

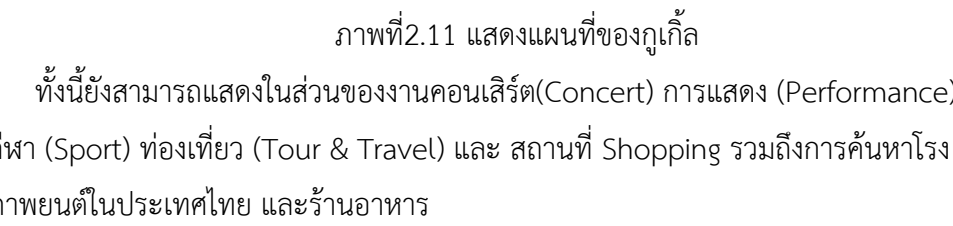
Google Maps คือ บริการแผนที่ออนไลน์จาก Google แผนที่ออนไลน์นี้สามารถใช้งานได้อย่างหลากหลาย เช่น หาตำแหน่งพิกัด ตรวจสอบสภาพการจราจร



ภาพที่ 2.10 แสดงแผนที่ของกูเกิล

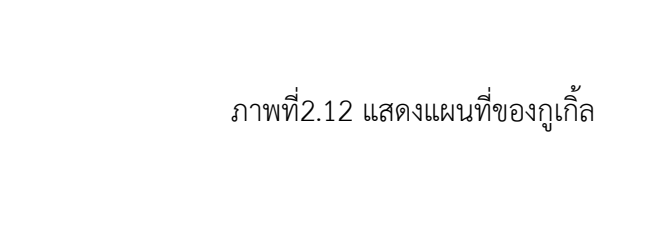
Google Maps ได้รวมบริการหลายอย่างเข้าไว้ด้วยกัน เช่น ภาพแผนที่บนถนนหนทางต่างๆ ภาพถ่ายดาวเทียม ภาพถ่ายจากอากาศยานที่มีความละเอียดของภาพสูง บริการค้นหาสถานที่ห้างร้านต่างๆ บริการค้นหาเส้นทางจากสถานที่ต้นทางไปยังสถานที่ปลายทาง บริการภาพถ่ายจากถนนในเมืองสำคัญที่ทำให้ผู้ใช้เห็นสภาพแวดล้อมและอาคารบ้านเรือนที่อยู่สองฝั่งถนน เป็นต้น

นอกจากนี้ Google Maps ประเทศไทยยังแสดงข้อมูลที่เฉพาะเจาะจงสำหรับผู้ใช้งานคนไทยในกรอบด้านซ้ายมือได้แก่ รายชื่อเมืองต่างๆ แผนที่ยอดนิยมและลิงก์สำหรับเพิ่มรายชื่อธุรกิจของคุณบน Google Maps หัวข้อรายชื่อเมืองต่างๆ แสดงชื่อจังหวัดและสถานที่แหล่งท่องเที่ยวที่มีชื่อเสียงของไทย ซึ่งผู้ใช้สามารถคลิกเพื่อไปยังสถานที่เหล่านั้นได้ทันที ส่วนหัวข้อแผนที่ยอดนิยมจะแสดง โดยในตอนนี้มีโปรแกรมสำหรับแสดงตำแหน่งของสถานที่จัดงานอีเวนต์ต่างๆ เช่น สถานที่จัดคอนเสิร์ต ดาเอนโดรฟิน ดังรูป โดยงานอีเวนต์ต่างๆ เหล่านี้จัดทำโดย ไทยทิคเก็ตเมเจอร์



ภาพที่ 2.12 แสดงแผนที่ของภูเก็ต

ภาพที่ 2.12 แสดงแผนที่ของภูเก็ต



เอพีไอ (Application Programming Interface : API)

API คือ ช่องทางการเชื่อมต่อ , ช่องทางหนึ่งที่จะเชื่อมต่อกับเว็บไซต์ผู้ให้บริการ API จากที่อื่น เป็นตัวกลางที่ทำให้โปรแกรมประยุกต์เชื่อมต่อกับโปรแกรมประยุกต์อื่น หรือเชื่อมการทำงานเข้ากับระบบปฏิบัติการ ตัวอย่าง เช่น Twitter มีหลายเว็บ ที่มีการเชื่อมโยงข้อมูลกับ twitter ทั้งเป็นการอ่านข้อมูลจาก twitter หรือ ส่งข้อมูลเข้า twitter เองก็ตาม ซึ่งล้วนอาศัยการเชื่อมต่อแลกเปลี่ยนข้อมูลกัน ด้วย API นั้นเอง และ อีกหนึ่งตัวอย่าง เช่น Google Maps API คือบริการของ Google อีกรูปแบบหนึ่งที่เราสามารถนำข้อมูลของ Google Maps ที่ทาง Google ให้บริการโดยส่วนมากจะนำมาใช้ กับเว็บไซต์ ของบริษัทหรือเว็บไซต์ห้างร้านต่างๆ เพื่อเป็นอีกช่องทางที่ให้ลูกค้ารู้ว่า บริษัทหรือห้างร้าน

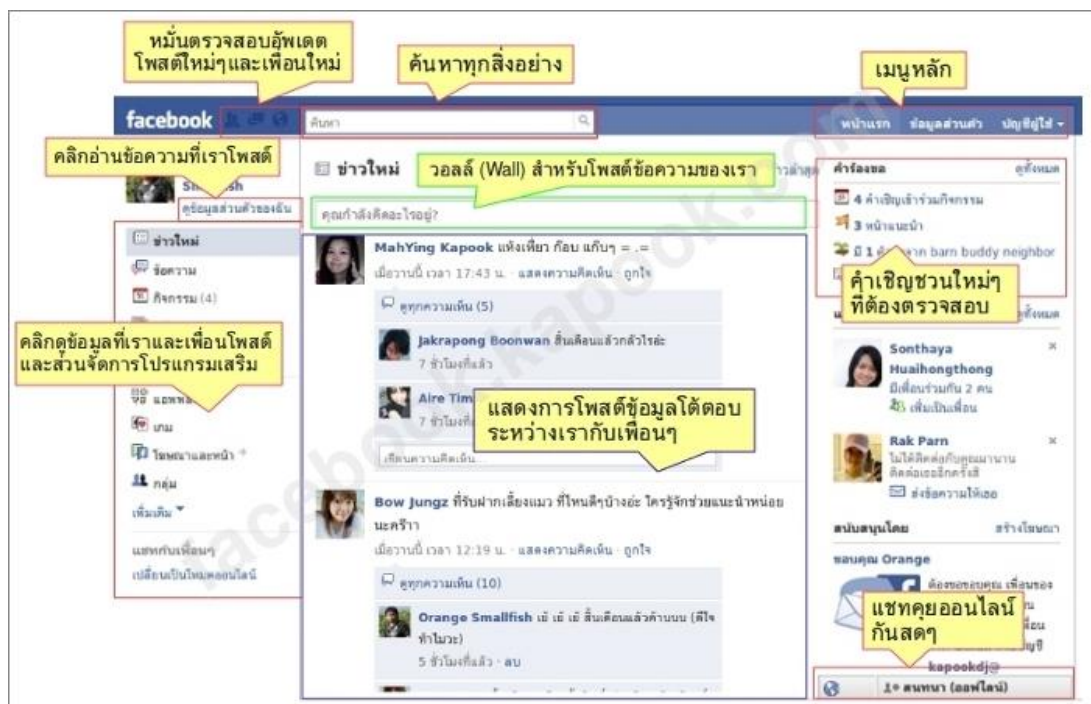
ประโยชน์ของ API

- 1.สามารถรับส่งข้อมูลข้าม Server ได้
- 2.ไม่จำเป็นต้องเข้าหน้าเว็บหลัก ก็มีข้อมูลของเว็บหลัก จากเว็บที่ดึง API เอ พีโอ แบ่งเป็น
- 3.เอพีไอที่ขึ้นกับภาษา (language-dependent API) คือ เอพีไอ ที่สามารถ เรียกใช้จากโปรแกรมที่เขียนขึ้นด้วยภาษาเพียงภาษาใดภาษาหนึ่ง
- 4.เอพีไอไม่ขึ้นกับภาษา (language-independent API) คือ เอพีไอ ที่สามารถ เรียกได้จากโปรแกรมหลายๆภาษา

API ถือเป็นกลุ่มของฟังก์ชัน ขั้นตอน หรือคลาส (Class) ที่ระบบปฏิบัติการ (OS) หรือ ผู้ให้บริการ สร้างขึ้นมาเพื่อรองรับการเรียกขอข้อมูล จากโปรแกรมอื่น ๆ ทั้งนี้ API สามารถ ใช้งานได้กับภาษาในการเขียนโปรแกรมที่รองรับเท่านั้น ซึ่งมันจะถูกจัดทำให้อยู่ในรูปแบบ Syntax หรือ element ที่สามารถนำไปใช้ได้อย่างสะดวกสบาย

เฟสบุ๊ก (Facebook)

Facebook (เฟสบุ๊ก) คือ บริการบนอินเทอร์เน็ตบริการหนึ่ง ที่จะทำให้ผู้ใช้สามารถติดต่อสื่อสารและร่วมทำกิจกรรมใดกิจกรรมหนึ่งหรือหลายๆ กิจกรรมกับผู้ใช้ Facebook คนอื่นๆได้ ไม่ว่าจะเป็นการตั้งประเด็นถามตอบในเรื่องที่สนใจ , โพสต์รูปภาพ , โพสต์คลิปวิดีโอ , เขียนบทความหรือบล็อก , แชทคุยกันแบบสดๆ , เล่นเกมส์แบบเป็นกลุ่ม (เป็นที่นิยมกันอย่างมาก) และยังสามารถทำกิจกรรมอื่นๆ ผ่านแอปพลิเคชันเสริม (Applications) ที่มีอยู่อย่างมากมาย ซึ่งแอปพลิเคชันดังกล่าวได้ถูกพัฒนาเข้ามาเพิ่ม เต็มอยู่เรื่อยๆ



ภาพที่ 2.13 แสดงส่วนต่างๆ ของเฟสบุ๊ก

พีเอชพี (Personal Home Page : PHP)

พีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ซ ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจที่มีความตอบโต้ได้อย่างรวดเร็ว ภาษาพีเอชพี ในชื่อภาษาอังกฤษว่า PHP ซึ่งใช้เป็นคำย่อแบบกล่าวซ้ำ จากคำว่า PHP Hypertext Preprocessor หรือชื่อเดิม Personal Home Page พีเอชพีรุ่นล่าสุดคือ PHP 5.4.0 ส่วนรุ่นพัฒนาคือ PHP 6.0.0

ประวัติความเป็นมาของ PHP

PHP นั้นถูกคิดค้นขึ้นมาในปี ค.ศ.1994 โดยนาย Rasmus Lerdorf แต่เป็นเวอร์ชันที่ไม่เป็นทางการหรือรุ่นทดลองนั่นเอง ซึ่งเวอร์ชันนี้ได้มีการทดสอบกับเครื่องของเขาเอง โดยใช้ตรวจสอบติดตามเก็บสถิติข้อมูล ผู้ที่เข้าเยี่ยมชมประวัติส่วนตัวบนเว็บเพจของเขาเท่านั้น ต่อมา PHP เวอร์ชันแรกได้ถูกพัฒนาและเผยแพร่ให้กับผู้อื่นที่ต้องการใช้ศึกษา ในปี ค.ศ.1995 ซึ่งถูกเรียกว่า "Hypertext Preprocessor" ซึ่งเป็นที่มาของคำว่า PHP นั่นเอง ซึ่งในระยะเวลาที่นั้น PHP ยังไม่มีความสามารถอะไรที่โดดเด่นมากมาย จนกระทั่งเมื่อประมาณกลางปี ค.ศ.1995 นาย Rasmus ได้คิดค้นและพัฒนาให้ PHP/PI หรือ PHP เวอร์ชัน 2 ให้มีความสามารถจัดการเกี่ยวกับแบบฟอร์มข้อมูลที่ถูกสร้างมาจากภาษา HTML และสนับสนุนการติดต่อกับโปรแกรมจัดการฐานข้อมูล mSQL จึงทำให้ PHP เริ่มถูกใช้มากขึ้นอย่างรวดเร็ว และเริ่มมีผู้สนับสนุนการใช้งาน PHP มากขึ้น โดยในปลายปี ค.ศ.1996 PHP ถูกนำไปใช้ประมาณ 15,000 เว็บทั่วโลก และเพิ่มจำนวนขึ้นเรื่อยๆ นอกจากนี้ในราวกลางปี ค.ศ.1997 PHP ได้มีการเปลี่ยนแปลงและพัฒนาจากเจ้าของเดิมคือ นาย Rasmus ที่พัฒนาอยู่เพียงผู้เดียว มาเป็นทีมงาน โดยมีนาย Zeev Suraski และนาย Adni Gutmans ทำการวิเคราะห์พื้นฐานของ PHP/FI และได้้นำโค้ดมาพัฒนาให้เป็น PHP เวอร์ชัน 3 ซึ่งมีความสามารถที่มีความสมบูรณ์มากขึ้น ในราวกลางปี ค.ศ.1999 PHP เวอร์ชัน 3 ได้ถูกพัฒนาจนสามารถทำงานร่วมกับ C2's StrongHold Web Server และ Red Hat Linux ได้

คุณสมบัติของภาษา PHP

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือ และคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะ CGI คุณสมบัติอื่น เช่น การประมวลผลตามบรรทัดคำสั่ง (Command Line Scripting) ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP Parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ในยูนิกซ์หรือลินุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple Text Processing Tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แพลช (โดยใช้ Libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและเข้าสู่เอกสาร XML เรารองรับมาตรฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML เมื่อใช้พีเอชพีในการทำอีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercash payment, CyberMUT, VeriSign Payflow Pro และ CCVS Functions เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน

การรองรับภาษา PHP

คำสั่งของพีเอชพี สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น โน้ตแพด หรือ Vi ซึ่งทำให้การทำงานของพีเอชพี สามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผล Apache, Microsoft Internet Information Services (IIS), Personal Web Server, Netscape และ iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, และอื่นๆ อีกมากมาย สำหรับส่วนหลักของ PHP ยังมี Module ในการรองรับ CGI มาตรฐาน ซึ่ง PHP สามารถทำงานเป็นตัวประมวลผล CGI ด้วย และด้วย PHP มีอิสรภาพในการเลือก ระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ นอกจากนี้คุณยังสามารถใช้สร้างโปรแกรมโครงสร้างสร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่าความสามารถของคำสั่ง OOP มาตรฐานในเวอร์ชันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ (รวมถึง PEAR Library) ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ OOP เท่านั้น

พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ ออราเคิล, dBase, PostgreSQL, IBM, DB2, MySQL, Informix, ODBC โครงสร้างของฐานข้อมูลแบบ DBX ซึ่งทำให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้ และ PHP ยังรองรับ ODBC (Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย คุณสามารถเชื่อมต่อกับฐานข้อมูลต่างๆ ที่รองรับมาตรฐานโลกนี้ได้ พีเอชพียังสามารถรองรับการสื่อสารกับการบริการในโพรโทคอลต่างๆ เช่น LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM(บนวินโดวส์) และอื่นๆ อีกมากมาย คุณสามารถเปิด Socket บนเครือข่ายโดยตรงและตอบโต้โดยใช้ โพรโทคอลใดๆ ก็ได้ PHP มีการรองรับสำหรับการแลกเปลี่ยนข้อมูลแบบ WDDX Complex กับ Web Programming อื่นๆ ทั่วไปได้พูดถึงในส่วน Interconnection, พีเอชพีมีการรองรับสำหรับ Java objects ให้เปลี่ยนมันเป็น PHP Object แล้วใช้งาน คุณยังสามารถใช้ รูปแบบ CORBA เพื่อเข้าสู่ Remote Object ได้

CSS (Cascading Style Sheets)

CSS ซึ่งย่อมาจากคำว่า Cascading Style Sheets นั้นเอง บางทีอาจจะเรียกว่า Style Sheets หรือ CSS ซึ่งจริงๆแล้วมันคือตัวเดียวกัน

การทำงานของ CSS

จะทำงานร่วมกับ HTML โดยจะกำหนดการแสดงผลของสิ่งต่างบนเว็บ เช่น สีอักษร สีพื้นหลัง ขนาดตัวอักษร จัดการเลย์เอ๊าท์ ให้สวยงามและอื่นๆ

ข้อดีสำคัญของ CSS

1. ทำมห้ขนาดไฟล์แต่ละหน้าเล็กลงกว่าเดิม เพราะใช้โค้ดน้อยกว่า
2. มีความยืดหยุ่นสูง ในการปรับแต่งแก้ไขในอนาคต
3. สามารถกำหนดแยกไว้ต่างหากจาก ไฟล์เอกสาร html และสามารถนำมาใช้ร่วมกับเอกสารหลายไฟล์ได้ สำหรับการแก้ไขก็แก้เพียง จุดเดียวก็มีผลกับเอกสารทั้งหมดได้
4. สามารถจัดการเลย์เอ๊าท์ได้อย่างละเอียด แม่นยำ
5. ง่ายในการเรียกดู Source
6. ใช้ดีกับระบบเสิร์ชเอนจิน ซึ่งระบบเสิร์สเอนจินต่างๆ

jQuery

jQuery คือ

jQuery นั้นเป็น JavaScript Library ที่บรรจุเอา Function และ คำสั่งต่างๆ ที่จะทำให้เราๆท่านๆ ไม่ต้องมาเขียนเองใหม่ทั้งหมดตั้งแต่ต้น เราสามารถที่จะเขียน ajax ได้แบบง่ายๆ เพียง code ไม่กี่บรรทัด หรือจะเขียน javascript เพื่อดัก Event (เหตุการณ์) ต่างๆที่เขาต้องการ เช่น การ click, rollover, mouse moved อื่นๆอีกมากมาย ซึ่งถ้าเขียนเองทั้งหมดตั้งแต่ต้นละก็ รับลองได้เลย หลายบรรทัดแน่นอน เวลามาก็ code ที่หลังนี้เล่นเอาปวดกระบานไปหลายวันเลยละ (จากประสบการณ์จริงเลย)

jQuery เอามาทำอะไรได้บ้าง

เมื่อถามว่ามันทำอะไรได้บ้าง ก็ทำได้หลายอย่างมากหลายเลยละ ไม่ว่าจะเป็นการ

-ความสามารถในการทำงานแบบ ajax

การสร้าง animation ได้แบบง่ายๆเลย ไม่ว่าจะเป็นทำรูปให้เคลื่อนที่ หรือ DIV เช่นเอา mouse จับ DIV ลากไปมา

ความสามารถในการ binds หรือการผูก หรือจับ function ที่เขาเขียนขึ้นให้ทำงานร่วมกับ function อื่นๆ

สามารถจัดการกับ css (style sheet) ของ element นั้นๆได้

ค้นหา element ที่เราต้องการและจัดการ เพิ่มหรือลบ Attributes ที่เราต้องการได้

ทำ Effect ต่างๆกับ Element ที่เราต้องการ เช่นการ hide DIV ที่เราต้องการ

การดักเหตุการณ์ต่างๆ

jQuery

นั้นมี Plugin มากหลายเลยที่เราสามารถนำมาใช้ในงานของเราได้ ผมเองก็ได้ประโยชน์จาก jQuery Plugin เยอะพอสมควรเลยสำหรับงานริบๆ ไม่ว่าจะเป็น Plugin สำหรับการเช็ค Form เช่น เช็ค email ว่ารูปแบบถูกต้องหรือไม่ หรือ plugin ปฏิทิน อันนี้ใช้บ่อยมาก ใช้สำหรับให้ user คลิกเลือกวันที่เอา โดยไม่ต้องมากรอกเอง หรือไม่ว่าจะเป็นการดึงเอาข้อมูล XML , JSON, TEXT ก็สามารถทำได้ง่ายๆเลย

ส่วนวิธีการใช้งาน หรือการติดตั้งนั้น เพื่อนๆสามารถอ่านได้ในบทความ ต่อไปที่จะเขียนอธิบายถึงวิธีติดตั้ง jQuery เพื่อนำมาใช้งาน และ บทความ ที่จะอธิบายคำว่า DOM คืออะไร เพื่อให้เข้าใจการทำงานของ jQuery ได้ดียิ่งขึ้น

ภาษาจาวาสคริปต์ (Java script)

JavaScript เป็นภาษาโปรแกรม (programming language) ประเภทหนึ่ง ที่เรียกกันว่า "สคริปต์" (script) ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) ภาษานี้เดิมมีชื่อว่า LiveScript ได้รับการพัฒนาขึ้นโดย Netscape ด้วยวัตถุประสงค์ เพื่อที่จะช่วยให้เว็บเพจสามารถแสดงเนื้อหา ที่มีการเปลี่ยนแปลงไปได้ตามเงื่อนไขหรือสภาพแวดล้อมต่าง ๆ กัน หรือสามารถโต้ตอบกับผู้ชมได้มากขึ้น ทั้งนี้เพราะภาษา HTML แต่เดิมนั้น เหมาะสำหรับใช้แสดงเอกสาร ที่มีเนื้อหาคงที่แน่นอน และไม่มีลูกเล่นอะไรมากมายนัก

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงตามความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA ซึ่งเราจะพบว่าปัจจุบัน จะหาเว็บเพจที่ไม่ใช้ JavaScript เลยนั้นได้ยากเต็มที

การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ตาม สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆ ออกมาด้วย (ปัจจุบันคือรุ่น 1.5) ดังนั้น ถ้านำโค้ดของเวอร์ชันใหม่ ไปรันบนบราวเซอร์รุ่นเก่าที่ยังไม่สนับสนุน ก็อาจจะทำให้เกิด error ได้

การทำงานของ JavaScript เกิดขึ้นบนบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้นไม่ว่าคุณจะใช้เซิร์ฟเวอร์อะไร หรือที่ไหน ก็ยังคงสามารถใช้ JavaScript ในเว็บเพจได้ ต่างกับภาษาสคริปต์อื่น เช่น Perl, PHP หรือ ASP ซึ่งต้องแปลความและทำงานที่ตัวเครื่องเซิร์ฟเวอร์ (เรียกว่า server-side script) ดังนั้นจึงต้องใช้บนเซิร์ฟเวอร์ที่สนับสนุนภาษาเหล่านั้นเท่านั้น อย่างไรก็ตาม จากลักษณะดังกล่าวก็ทำให้ JavaScript มีข้อจำกัด คือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริง JavaScript ที่ทำงานบนเซิร์ฟเวอร์เวอร์ก็มี ซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก)

การทำงานของ JavaScript จะมีประสิทธิภาพมาก ถ้ามันสามารถดัดแปลงคุณสมบัติ ขององค์ประกอบต่างๆ บนเว็บเพจ (เช่น สี หรือรูปแบบของข้อความ) และสามารถรับรู้เหตุการณ์ ที่ผู้ชมเว็บเพจโต้ตอบกับองค์ประกอบเหล่านั้น (เช่น การคลิก หรือเลื่อนเมาส์ไปวาง) ได้ ดังนั้นจากภาษา HTML เดิม ที่มีลักษณะสถิต (static) ใน HTML เวอร์ชันใหม่ๆ จึงได้มีการพัฒนาให้มีคุณสมบัติบางอย่างเพิ่มขึ้น และมีลักษณะเป็นอ็อบเจ็ค "object" มากขึ้น การทำงานร่วมกันระหว่างคุณสมบัติใหม่ของ HTML ร่วมกับ JavaScript นี้เอง ทำให้เกิดเป็นสิ่งที่เรียกว่า Dynamic HTML คือภาษา HTML ที่สามารถใช้สร้างเว็บเพจที่มีลักษณะพลวัต (dynamic) ได้นั่นเอง

นอกจากนี้ อีกองค์ประกอบหนึ่งที่เกี่ยวข้อง ก็คือ Cascading Style Sheet (CSS) ซึ่งเป็นภาษาที่ช่วยให้เราควบคุมรูปแบบ ขององค์ประกอบต่างๆ บนเว็บเพจ ได้อย่างมีประสิทธิภาพมากกว่าคำสั่ง หรือแท็ก (tag) ปกติของ HTML เนื่องจาก JavaScript สามารถดัดแปลงคุณสมบัติของ CSS ได้เช่นกัน ดังนั้นมันจึงช่วยให้เราควบคุมเว็บเพจ ได้อย่างมีประสิทธิภาพมากยิ่งขึ้นไปอีก

2.4งานวิจัยหรือผลงานที่เกี่ยวข้อง

เกมอินเกรซ (Ingress)

เกมอินเกรซเป็นเกมสำหรับสมาร์ทโฟน ซึ่งเป็นผลงานการพัฒนาาระบบให้บริการแบบอิงตำแหน่งผู้ใช้งานจากทีม Niantic Labs เน้นการเล่นโดยอาศัยข้อมูลจากตำแหน่งจริงของผู้เล่นทั่วโลก

Ingress สมมติเหตุการณ์ว่านักวิทยาศาสตร์ค้นพบพลังงานลึกลับที่สามารถเปลี่ยนความคิดของผู้คนได้ โดยพลังงานลึกลับที่ว่าเป็นตัวกำหนดผู้เล่นออกเป็น 2 กลุ่ม ซึ่งต้องแข่งกันเพื่อแย่งชิงอำนาจเหนืออีกฝ่าย ระหว่างฝ่ายที่มีชื่อว่า Enlightened และ ฝ่าย Resistance

Ingress เน้นการเล่นโดยอิงตำแหน่งของผู้ใช้งานซึ่งระบุตำแหน่งจากเสาสัญญาณในบริเวณใกล้เคียง ควบคุมกับระบบ GPS โดยผู้เล่นต้องเคลื่อนที่ไปยังจุดต่างๆ เพื่อสร้างปฏิสัมพันธ์กับตัวเกม เช่น ค้นหาแหล่งพลังงาน, ยึดครองเขตแดน, เก็บสิ่งของเพื่อใช้ในการกิจ และเข้าร่วมกับผู้เล่นอื่นเพื่อเสริมกำลังให้กลุ่มของตน

ตัวเกมถูกพัฒนามาสำหรับสมาร์ทโฟนโดยเฉพาะ โดยเกมนี้มีผู้เล่นร่วมเล่นคือคนจากทั่วทุกมุมโลก ซึ่งล่าสุด Ingress เข้าสู่ขั้นตอนการทดสอบในกลุ่มปิดเฉพาะผู้ที่ได้รับคำเชิญ โดยปรากฏหน้าเว็บของเกมใน Google Play เพื่อให้ผู้เข้าร่วมการทดสอบสามารถดาวน์โหลดเกมดังกล่าวได้ฟรี ทั้งนี้ตามข้อมูลที่ระบุในหน้าเว็บ สามารถใช้ได้กับ Android 2.3 ขึ้นไป



ภาพที่2.14 แสดงเกมอินเกรซ



ภาพที่2.15 แสดงเกมอินเกรซ