บทที่ 2

แนวคิด ทฤษฎี เทคโนโลยี และระบบงานที่เกี่ยวข้อง

แนวคิด

1. ผู้ประกอบการธุรกิจพัฒนาอสังหาริมทรัพย์เพื่อขาย

ผู้ประกอบการธุรกิจพัฒนาอสังหาริมทรัพย์เพื่อขาย โครงการที่พัฒนาขึ้นนั้นล้วนแล้วดำเนินการไปในทิศทางเดียวกัน เช่น ทาวน์โฮม บ้านเดี่ยว และคอนโดมิเนียม ซึ่งมีทำเลที่ตั้งในเขตชุมชน เมืองหรือใกล้ศูนย์กลางทางธุรกิจ ทำให้ธุรกิจดังกล่าวมีความต้องการมากขึ้น ผู้ประกอบการจึงดำเนินการเดินหน้าขยายธุรกิจ เพื่อรองรับความต้องการของลูกค้ามากขึ้นไป ในการดำเนินธุรกิจของผู้ประกอบการธุรกิจพัฒนาอสังหาริมทรัย์นั้น ในด้านการพัฒนาโครงการต่าง ๆ เกินขึ้นจากแนวความคิดรูปแบบที่ทันสมัย อีกทั้งรูปแบบอสังหาริมทรัพย์ยังมีอยากหลายรูปแบบให้เลือกตามความต้องการของลูกค้า ผังโครงการ หรือที่เรียกกันอีกแบบว่า แบบบ้านนั้น ได้ถูกนำมาใช้ในด้านการขาย เพื่อให้ลูกค้าได้เลือกรูปแบบ ขนาดของแบบบ้านได้ตามต้องการ อีกทั้งผังโครงการ หรือแบบบ้านดังกล่าวยังนำไปใช้ในด้านการก่อสร้าง

จากข้อมูลพื้นฐานดังกล่าวที่มี ทั้งในส่วนของข้อมูลโครงการต่าง ๆ ข้อมูลการขาย ข้อมูลการก่อสร้าง อีกทั้งผังโครงการที่มี ได้ถูกนำมาใช้ในด้านข้อมูล และการนำเสนอของการพัฒนาระบบติดตามความคืบหน้าการก่อสร้าง และการขาย เพื่อให้ระบบดังกล่าวได้ตอบโจทย์แก่ผู้ประกอบการธุรกิจพัฒนาอสังหาริมทรัพย์เพื่อขาย โดยในส่วนของสถานะการก่อสร้างและการขายนั้น ผู้ประกอบการธุรกิจพัฒนาอสังริมทรัพย์เพื่อขายนั้นสามารถกำหนดขั้นตอนสถานะการก่อสร้างและการขายได้เองตามหลักพื้นฐานข้อมูล หรือข้อกำหนดเงื่อนไขสถานะได้เอง เช่น สถานะการก่อสร้าง ประกอบไปด้วย การออกแบบ การดำเนินการลงเสาเข็ม การดำเนินการงานโครงสร้าง การดำเนินการตกแต่ง เป็นต้น สถานะการขาย ประกอบด้วย สถานะว่าง จอง สัญญา โอน เป็นต้น โดยข้อมูลสถานะดังกล่าวผู้ประกอบการธุรกิจพัฒนาอสังหาริมทรัพย์ล้วนแล้วมีการตรวจสอบข้อมูลที่ต่างกัน ดังนั้น ในส่วนงานการดำเนินการติดตามความคืบหน้าการก่อสร้างและการขาย การกำหนดสถานะต่าง ๆ ขึ้นอยู่กับผู้ประกอบการนั้นเป็นตัวกำหนด

2. การคำนวณแบบคลาวด์ (Cloud Computing)

National Institute of Standards and Technology (NIST) ของสหรัฐอเมริกา ได้นิยามความหมายของการคำนวณแบบคลาวด์ไว้ว่า เป็นตัวแบบสำหรับการทำให้การเข้าถึงแหล่งรวมทรัพยากรของงคอมพิวเตอร์ที่ใช้ร่วมกัน และปรับแต่งโครงแบบได้ เช่น เครือข่าย, เครื่องบริการ, หน่วยเก็บ, โปรแกรมประยุกต์ และบริการ เป็นต้น ผ่านระบบเครือข่ายจากที่ใด ๆ ได้โดยสะดวกเมื่อทวงถาม ซึ่งทรัพยากรเหล่านี้สามารถจัดหาและคืนได้อย่างรวดเร็วโดยใช้ความพยายามในการจัดการ และการโต้ตอบกับผู้ให้บริการน้อยที่สุด

2.1 คุณสมบัติของบริการ Cloud Computing

2.1.1 บริการตนเองตามความต้องการ (On Demand Self Service) ผู้ใช้งานสามารถเข้าถึงระบบที่ให้บริการได้โดยอัตโนมัติ และสามารถปรับเปลี่ยนการใช้งาน เช่น server time และ storage ได้ตามความต้องการในช่วงเวลาใดก็ได้ ผ่านระบบบริหารจัดการบนเว็บไซต์ที่ผู้ให้บริการจัดหาไว้ให้

2.1.2 การเข้าถึงได้หลายช่องทาง (Broad Network Access) ผู้ใช้งานสามารถเข้าถึงระบบของผู้ให้บริการจากอุปกรณ์ประเภทใดก็ได้ เช่น สมาร์ตโฟน แท็บเล็ต โน้ตบุ๊ก และคอมพิวเตอร์ เป็นต้น โดยระบบต้องสามารถรองรับการใช้งานบนแพลตฟอร์มที่มีความหลากหลาย (Multi-platform)

2.1.3 การใช้ทรัพยากรร่วมกัน (Resource Pooling) ความสามารถในการบริหารจัดการระบบเพื่อให้บริการแก่ผู้ใช้งานจำนวนมากในเวลาเดียวกัน (Multi-tenants) โดยผู้ใช้ไม่จำเป็นต้องรู้ว่าข้อมูลต่าง ๆ มีการจัดเก็บที่ใด

2.1.4 ความยืดหยุ่นในการให้บริการสูง (Rapid Elasticity) ระบบที่มีความยืดหยุ่นสูงและหลากหลายตามความต้องการของผู้ใช้งาน ทำให้มีความสามารถในการเพิ่มหรือลดทรัพยากรได้อย่างรวดเร็ว และไม่มีข้อจำกัดเรื่องจำนวน ปริมาณและระยะเวลาในการใช้งาน

2.1.5 ระบบการวัดบริการ (Measured Service) ความสามารถในการบริหารจัดการและควบคุมการใช้ทรัพยากรที่เกี่ยวข้อง โดยการวัดปริมาณและคิดค่าบริการตามการใช้งานที่เกิดขึ้นจริง หรือ Pay-per-use

2.2 ตัวแบบการให้บริการคลาวด์ (Cloud service models)

2.2.1 บริการโครงสร้างพื้นฐานคลาวด์ (Infrastructure-as-a-Service: IaaS) หมายถึง การให้บริการทรัพยากรคอมพิวเตอร์พื้นฐาน เช่น ระบบประมวลผล ระบบจัดเก็บข้อมูล ระบบเครือข่าย ตลอดจนอุปกรณ์พื้นฐานที่เกี่ยวข้อง อาทิ สตอเรจ เซิร์ฟเวอร์ และระบบปฏิบัติการ

ที่ทำให้สามารถใช้งานซอฟต์แวร์แอปพลิเคชันได้อย่างมีประสิทธิภาพ โดยผู้ใช้ไม่ต้องบริหารจัดการโครงสร้างพื้นฐานเอง

2.2.2 บริการแพล็ตฟอร์มคลาวด์ (Platform-as-a-Service: PaaS) หมายถึง การให้บริการแพลตฟอร์ม และเครื่องมือเพื่อใช้ในการพัฒนาซอฟต์แวร์แอปพลิเคชัน เช่น โปรแกรมเบื้องต้น ฐานข้อมูล และระบบที่เอื้อต่อการสร้างแอปพลิเคชันโดยผู้ใช้ไม่ต้องบริหารจัดการระบบหรือเครื่องมือเองแต่ต้องติดตั้ง แก้ไข ปรับแต่งโปรแกรมประยุกต์ที่สร้างหรือพัฒนาขึ้นเอง

2.2.3 บริการซอฟต์แวร์คลาวด์ (Software-as-a-Service: SaaS) หมายถึง การให้บริการซอฟต์แวร์ที่มีความยืดหยุ่นต่อการเข้าถึงการใช้งานได้หลากหลาย โดยผู้ใช้ไม่จำเป็นต้องบริหารจัดการโครงสร้างพื้นฐานเอง ทั้งในส่วนของเครือข่าย เซิร์ฟเวอร์ ระบบปฏิบัติการ การจัดเก็บข้อมูล รวมถึงความสามารถของแอปพลิเคชันที่ใช้งาน

2.3 ตัวแบบการติดตั้งคลาวด์ (Cloud deployment models)

2.3.1 คลาวด์ส่วนบุคคล (Private Cloud) หมายถึง การคำนวณแบบคลาวด์สำหรับหน่วยงาน หรือองค์กรใดองค์กรหนึ่งเพียงองค์กรเดียว (แต่อาจมีผู้ใช้งานภายในองค์กรได้ไม่จำกัด) ซึ่งการบริหารจัดการระบบโดยส่วนใหญ่กระทำโดยบุคลากรภายในองค์กร

2.3.2 คลาวด์ชุมชน (Community Cloud) หมายถึง การคำนวณแบบคลาวด์ที่ดำเนินการร่วมกันโดยกลุ่มคนจากองค์กรต่าง ๆ ที่มีการรวมตัวกันในรูปแบบของการจัดตั้งเป็นสมาคม ชมรม หรือสหภาพ ทั้งเป็นทางการหรือไม่เป็นทางการ โดยมีวัตถุประสงค์ จุดมุ่งหมาย และความต้องการใช้บริการแบบเดียวกัน

2.3.3 คลาวด์สาธารณะ (Public Cloud) หมายถึง การคำนวณแบบคลาวด์ที่เปิดให้สาธารณชน และหน่วยงานต่าง ๆ ใช้งานทั่วไป โดยการบริหารจัดการ และการให้บริหารอาจเป็นบริษัท สถาบันการศึกษา หรือหน่วยงานภาครัฐ เป็นผู้ให้บริการ

2.3.4 คลาวด์แบบผสม (Hybrid Cloud) หมายถึง การคำนวณแบบคลาวด์ที่มีลักษณะผสมผสานรูปแบบการบริหารตั้งแต่ 2 แบบขึ้นไป การใช้งานแบบ Hybrid เป็นการใช้งานเฉพาะกิจ ซึ่งผู้ใช้งานจะต้องมีมาตรฐาน คุณสมบัติทางเทคนิค และเทคโนโลยีที่สามารถใช้งานข้อมูลและถ่ายโอนแอปพลิเคชัน สำหรับการใช้งานข้ามไปมาระหว่างรูปแบบแต่ละแบบที่เลือกใช้

จากคุณสมบัติทั้ง 5 ประการของการคำนวณแบบคลาวด์ช่วยให้การพัฒนาระบบบริการคลาวด์สำหรับระบบติดตามความคืบหน้าการก่อสร้าง และการขาย เพื่อสนับสนุนการดำเนินธุรกิจ และเพื่อให้บริการดังกล่าวสามารถเข้าถึงข้อมูล ได้อย่างทันที และตอบโจทย์ในด้านธุรกิจของผู้ประกอบการธุรกิจพัฒนาอสังหาริมทรัพย์ได้ (National Institute of Standards and Technology, 2011)

ทฤษฎี

1. การวิเคราะห์และออกแบบระบบ (System Analysis and Design)

ระบบ (System) คือกลุ่มขององค์ประกอบต่าง ๆ ที่มีความสัมพันธ์กัน แต่ละองค์ประกอบจะประสานการทำงานร่วมกัน เพื่อบรรลุสู่เป้าหมายเดียวกัน ระบบที่ดีจำเป็นต้องมีองค์ประกอบของระบบที่เรียกว่า ระบบย่อย (Subsystem) ที่สามารถประสานการทำงานร่วมกันภายในระบบได้เป็นอย่างดี เพื่อนำไปสู่ภาพใหญ่ของระบบให้สามารถทำงานได้บรรลุตามเป้าหมาย ตัวอย่างเช่น ระบบงานทางคอมพิวเตอร์ที่ประกอบด้วยระบบย่อยอยู่ 3 ส่วนหลัก ๆ ด้วยกันคือ ฮาร์ดแวร์ ซอฟต์แวร์ และบุคลากร ส่วนประกอบทั้ง 3 เหล่านี้ จะต้องประสานการทำงานร่วมกันเพื่อบรรลุถึงเป้าหมายเดียวกัน เพื่อให้ได้มาซึ่งผลลัพธ์ตรงตามความต้องการ หากมีส่วนใดขัดข้อง ก็ย่อมส่งผลกระทบต่อระบบโดยรวม และหากผลกระทบได้พอกพูนมากขึ้น ก็อาจนำไปสู่ความล้มเหลวของระบบได้ในที่สุด (โอภาส เอี่ยมสิริวงศ์, 2555, หน้า 18)

ระบบสารสนเทศ (Information Systems: IS) หมายถึง ระบบงานที่นำเทคโนโลยีสารสนเทศ (Information Systems: IS) มาใช้เพื่อจัดเก็บ ประมวลผล และเรียกดูข้อมูล โดยเทคโนโลยีสารสนเทศมีบทบาทสำคัญต่อการเพิ่มผลผลิต (Productivity) การสร้างความได้เปรียบในเชิงแข่งขัน รวมถึงการสร้างผลกำไรให้แก่องค์กร อย่างไรก็ตาม หากพิจารณาในรายละเอียดแล้วเทคโนโลยีมิได้เป็นตัวช่วยเพิ่มผลผลิต หรือสร้างผลกำไรให้แก่องค์กรได้โดยลำพัง แต่กลับเป็นมนุษย์ต่างหากที่พัฒนาระบบขึ้นมา ด้วยการนำความสามารถของเทคโนโลยีเข้ามาประยุกต์ใช้ เพื่อควบคุมระบบและกระบวนการทางธุรกิจให้ดำเนินการอย่างมีประสิทธิภาพ ดังนั้น กุญแจที่นำไปสู่ความสำเร็จของธุรกิจสมัยใหม่ที่ต้องการนำระบบสารสนเทศมาใช้เพื่อแก้ไขปัญหาแก่องค์กรให้ถูกทาง ก็คือการวิเคราะห์และออกแบบระบบ

2. วงจรการพัฒนาระบบ

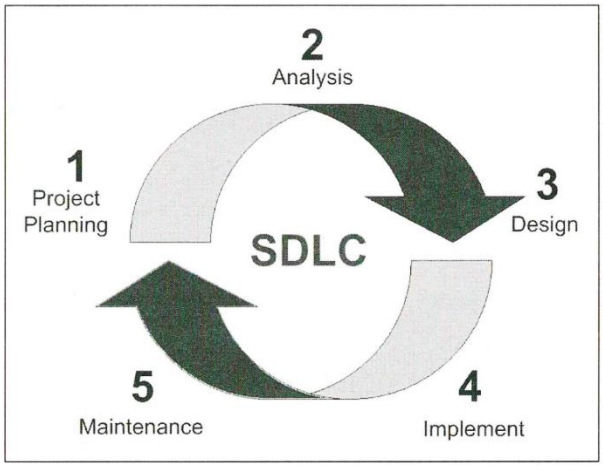
ขั้นตอนการพัฒนาซอฟต์แวร์แบบพื้นฐานที่เรียกว่า วงจรการพัฒนาระบบสารสนเทศ (Systems Development Life Cycle) หรือ SDLC ที่มักถูกนำไปใช้ในหลาย ๆ องค์กรด้วยกัน ซึ่งโดยทั่วไป การพัฒนาซอฟต์แวร์ มักจะประกอบไปด้วยกลุ่มกิจกรรม 3 ส่วนหลัก ๆ ด้วยกันคือ

1) การวิเคราะห์

2) การออกแบบ

3) การนำไปใช้

โดยกิจกรรมทั้งสามเหล่านี้ สามารถนำมาใช้งานได้ดีกับโครงการซอฟต์แวร์ขนาดเล็ก ในขณะที่โครงการซอฟต์แวร์ขนาดใหญ่ มักจำเป็นต้องใช้แบบแผนการพัฒนาซอฟต์แวร์ตามแนวทางของ SDLC จนครบทุกกิจกรรม ซึ่งแสดงไว้ดังภาพประกอบที่ 2-1



ภาพประกอบที่ 2-1 วงจรการพัฒนาระบบ หรือ SDLC (โอภาส เอี่ยมสิริวงศ์, 2555, หน้า 50)

ระยะที่ 1: การวางแผนโครงการ (Project Planning)

การวางแผนโครงการ จัดเป็นกระบวนการพื้นฐานของความเข้าใจว่า ทำไม (Why) ระบบสารสนเทศจึงสมควรที่จะสร้างขึ้นมา และจะต้องกำหนดทีมงานขึ้นมาเพื่อดำเนินการสร้างระบบนี้ได้อย่างไร ในช่วงของการเริ่มโครงการ (Project Initiate)

ระยะที่ 2: การวิเคราะห์ (Analysis)

ระยะการวิเคราะห์จะตอบคำถามเกี่ยวกับสิ่งต่อไปนี้คือ ใคร (Who) เป็นผู้ใช้ระบบ มีอะไรบ้าง (What) ที่จะต้องทำ และทำที่ไหน (Where) เมื่อไร (When) โดยในระยะนี้ ทีมงานจะทำการศึกษาระบบงานปัจจุบัน พร้อมระบุแนวทางในการปรับปรุงกระบวนการที่ดีขึ้น เพื่อพัฒนาเป็นแนวคิดสำหรับระบบใหม่ขึ้นมา

สิ่งสำคัญของระยะนี้ก็คือ การรวบรวมความต้องการ (Requirements Gathering) ซึ่งนักวิเคราะห์ระบบสามารถรวบรวมความต้องการต่าง ๆ ได้จากการสังเกตการณ์ทำงานของผู้ใช้ การสัมภาษณ์ การจัดทำแบบสอบถาม การอ่านเอกสารเกี่ยวกับการปฏิบัติงานของระบบงานปัจจุบัน และระเบียบกฎเกณฑ์ต่าง ๆ ของบริษัท ซึ่งตลอดระยะเวลาของการรวบรวมความต้องการ ก็จะได้พบปะกับผู้ใช้ในระดับต่าง ๆ ที่ทำให้ทราบถึงกระบวนการทำงาน ปัญหาที่เกิดขึ้น และแนวทางการแก้ไขปัญหาที่แนะนำโดยผู้ใช้ ดังนั้น การรวบรวมความต้องการ จึงเป็นกิจกรรมสำคัญเพื่อค้นหาความจริง และต้องทำความเข้าใจซึ่งกันและกัน เพื่อสรุปออกมาเป็นข้อกำหนด (Requirements Specification) ที่ชัดเจน

ระยะที่ 3: การออกแบบ (Design)

ระยะการอออกแบบ จะเป็นการตัดสินใจว่า ระบบจะดำเนินการไปได้อย่างไร (How) ในด้านของการจัดหาอุปกรณ์ฮาร์ดแวร์ ซอฟต์แวร์ โครงสร้างเครือข่ายที่จะนำมาใช้ การปฏิสัมพันธ์ระหว่างผู้ใช้กับระบบ รวมถึงแบบฟอร์มและรายงานต่าง ๆ ที่จะต้องถูกนำมาใช้ นอกจากนี้ ยังรวมถึงโปรแกรม ฐานข้อมูล และแฟ้มข้อมูลที่จำเป็น การออกแบบนั้น จะมุ่งประเด็นเกี่ยวกับวิธีการดำเนินงานระบบด้วยการนำแบบจำลองเชิงตรรกะ (Logical Model) ที่ได้จากระยะการวิเคราะห์มาพัฒนาเป็นแบบจำลองเชิงกายภาพ (Physical Model) มาใช้งานให้เกิดผลได้อย่างไร โดยที่การวิเคราะห์มุ่งเน้นการแก้ปัญหาอะไร (What) การออกแบบมุ่งเน้นการแก้ปัญหาอย่างไร (How)

ระยะที่ 4: การนำไปใช้ (Implementation)

กิจกรรมต่าง ๆ ในระยะการนำไปใช้จะเกี่ยวข้องกับการสร้างระบบ การทดสอบ และการติดตั้งระบบ โดยมีจุดประสงค์หลักที่ไม่ใช่มีแค่เพียงการสร้างผลิตภัณฑ์ให้มีความน่าเชื่อถือ และระบบสารสนเทศจะต้องสามารถตอบสนองฟังก์ชั่นการทำงานทางธุรกิจตามหน่วยงานต่าง ๆ ได้อย่างสมบูรณ์เท่านั้น แต่จะต้องรวมถึงความมั่นใจว่าผู้ใช้ระบบทุก ๆ คนได้ผ่านการฝึกอบรมใช้งานเพื่อเตรียมความพร้อมต่อการใช้ระบบสารสนเทศให้เกิดประโยชน์ต่อองค์กรดังที่คาดหวัง

ระยะที่ 5: การบำรุงรักษา (maintenance)

ระยะนี้จะใช้เวลายาวนานที่สุดเมื่อเทียบกับระยะอื่น ๆ ที่ผ่านมา เนื่องจากระบบจะต้องได้รับการบำรุงรักษาตลอดระยะเวลาที่มีการใช้งาน โดยสิ่งที่คาดหวังขององค์กรก็คือ ระบบจะสามารถใช้งานได้ยาวนานหลายปี และรองรับเทคโนโลยีใหม่ ๆ ในอนาคตได้ ดังนั้น ในช่วงระยะของการบำรุงรักษา จึงสามารถเพิ่มเติมคุณสมบัติใหม่ ๆ เข้าไปเพื่อเพิ่มประสิทธิภาพในการทำงานให้กับระบบได้ ซึ่งคุณสมบัติใหม่ ๆ เหล่านี้อาจมาจากความต้องการของผู้ใช้เอง (โอภาส เอี่ยมสิริวงศ์, 2555, หน้า 50-57)

3. การพัฒนาระบบเชิงวัตถุ (Object-Oriented System Development)

Object Orientation หมายถึง การใช้แนวคิดเกี่ยวกับวัตถุ (Object) เป็นแนวทางในการพิจารณาความเป็นจริงต่าง ๆ ที่เกิดขึ้นในโลก (Real World)

การนำแนวคิดเชิงวัตถุมาปรับใช้กับการพัฒนาระบบคอมพิวเตอร์นั้นสามารถทำได้โดยจะต้องมองทุกสิ่งในระบบ (Problem Domain) ที่อยู่ในโลกความจริงให้เป็น Object เสียก่อน แล้วใส่แนวคิดเข้าไปว่า Object นั้น ๆ มีคุณลักษณะและพฤติกรรมอย่างไรบ้าง ผลจากการให้แนวคิดดังกล่าว ทำให้เกิดการจัดกลุ่มของ Object เรียกว่า “Class” แล้วจำลองระบบด้วยแผนภาพ (Diagram) ชนิดต่าง ๆ ที่แสดงถึงลักษณะและพฤติกรรมของ Object แต่การทำงานของระบบในคอมพิวเตอร์จะเกิดขึ้นได้ ต้องมีการสร้างปฏิสัมพันธ์ระหว่าง Object จากนั้นจึงเขียนโค้ดโปรแกรมคอมพิวเตอร์ เพื่อสร้าง Object ขึ้นมาทำงาน (พนิดา พานิชกุล, 2552, หน้า 9)

ออบเจ็กต์ (Object)

ออบเจ็กต์หรือวัตถุ คือหน่วยสนใจของระบบที่ทำให้เกิดเหตุการณ์บางอย่างขึ้น ซึ่งสามารถเป็นได้ทั้งบุคคล สถานที่ สิ่งของ หรือเหตุการณ์ กล่าวคือ ออบเจ็กต์อาจเป็นสิ่งที่เราสามารถจับต้องได้ หรือจับต้องไม่ได้ โดยออบเจ็กต์จะมีข้อมูลที่ใช้บ่งบอกถึงความเป็นตัวตน และยังสามารถแสดงพฤติกรรมของตัวเองออกมาให้เห็นได้ในขณะเดียวกัน ตัวอย่างเช่น รถยนต์ซึ่งจัดเป็นออบเจ็กต์หนึ่ง โดยรถยนต์แต่ละคันก็จะมีข้อมูลหรือคุณลักษณะของตัวเองที่แตกต่างกันได้ เช่น รถยนต์คันนั้นเป็นสีขาว เปิดประทุน หรือรถยนต์คันนั้นเป็นสีแดง มีหลังคา ส่วนพฤติกรรมของรถยนต์ก็คือ การวิ่งเคลื่อนที่ การเลี้ยว และการหยุด เป็นต้น

ออบเจ็กต์จะมี คุณสมบัติ (Attributes) ที่ใช้อธิบายถึงลักษณะของออบเจ็กต์นั้น ๆ รวมถึง วิธีการ (Method) ซึ่งจะปฏิบัติตามหรือตอบสนองต่อ เมสเสจ (Message) ที่ร้องขอเข้ามา เช่น รถยนต์ได้เตรียมวิธีการ “Drive” เอาไว้ เพื่อส่งเมสเสจไปยังระบบเกียร์ เพื่อขับเคลื่อนรถยนต์ให้เคลื่อนที่ไปข้างหน้า หรือที่ทำการไปรษณีย์ ได้เตรียมวิธีการส่งสินค้า เพื่อส่งเมสเสจไปยังระบบการขนส่ง เพื่อจัดส่งพัสดุไปยังปลายทาง เป็นต้น

คลาส (Class)

คลาสและออบเจ็กต์มีความคล้ายกันมาก จนทำให้หลายคนอาจสงสัยว่าเป็นสิ่งเดียวกันหรือไม่ ในความเป็นจริงแล้ว คลาส เป็นนามธรรม (Abstract) ในขณะที่ ออบเจ็กต์ เป็นสิ่งที่มีตัวตน (Concrete) กล่าวคือคลาสก็คือตัวแบบ (Template) ของออบเจ็กต์ โดยที่คลาสไม่สามารถทำงานได้ ในขณะที่ออบเจ็กต์สามารถทำงานได้ การทำงานของออบเจ็กต์จะเป็นไปตามคุณสมบัติที่กำหนดไว้ในคลาส และออบเจ็กต์ทุกตัวก็จะต้องอยู่ในคลาส ซึ่งคลาสกับออบเจ็กต์จะเป็นสิ่งที่คู่กันเสมอ โดยเราสามารถดูคุณสมบัติของออบเจ็กต์ได้ด้วยการดูที่คลาส

นอกจากนี้ คลาสยังสามารถนำมาใช้จัดกลุ่มออบเจ็กต์ต่าง ๆ ที่มีโครงสร้างพื้นฐานพฤติกรรมเดียวกันมาอยู่รวมกลุ่มภายใต้คลาสเดียวกันได้ ดังนั้นออบเจ็กต์ใด ๆ ก็ตามที่มีคุณสมบัติเป็นไปในลักษณะเดียวกัน ก็จะถูกรวมกลุ่มเข้าด้วยกันภายในคลาสนั้น ๆ ในขณะเดียวกันออบเจ็กต์ทุกตัวที่อยู่ภายใต้คลาสดังกล่าว ก็จะได้รับคุณสมบัติต่าง ๆ ตามคลาสเหล่านั้น จึงสรุปได้ว่าคลาสก็คือตัวแบบข้อมูล ที่มีไว้เพื่อสร้างออบเจ็กต์นั่นเอง

นิยามความหมายของหลักการพัฒนาระบบเชิงวัตถุ

1. OOA (Object-Oriented Analysis) เป็นขั้นตอนการวิเคราะห์เพื่อรับรู้และทำความเข้าใจเกี่ยวกับรายละเอียดของปัญหา ว่ามีปัญหาอะไรบ้างที่ต้องได้รับการแก้ไข โดยในขั้นตอนนี้จะมีการสร้างแผนภาพขึ้นมาที่เรียกว่า Use Case Diagram

2. OOD (Object-Oriented Design) เป็นขั้นตอนการออกแบบกระบวนการ ด้วยการสร้างแบบจำลองเชิงวัตถุ ที่สามารถแสดงความหมายออกมาทั้งในเชิงตรรกะและเชิงกายภาพ ว่าเราจะแก้ไขปัญหาเหล่านั้นได้อย่างไร โดยจะมีการกำหนดชนิดของวัตถุเพิ่มเติม ให้มีส่วนสำคัญต่อการสื่อสารกับมนุษย์และอุปกรณ์ในระบบ ออบเจ็กต์มีการโต้ตอบกันอย่างไรจนกระทั่งงานนั้นเสร็จสมบูรณ์ ต่อมาก็จะปรับออบเจ็กต์แต่ละตัวให้มีความชัดเจนยิ่งขึ้น เพื่อนำไปเป็นแบบสำหรับการเขียนโปรแกรมต่อไป

3. OOP (Object-Oriented Programming) เป็นขั้นตอนการนำสิ่งที่ได้วิเคราะห์และออกแบบมาทั้งหมด มาแปลงเป็นระบบจริงขึ้นมา ด้วยการเขียนโปรแกรมภาษาคอมพิวเตอร์เพื่อกำหนดหรือสั่งให้ออบเจ็กต์แต่ละตัวมีหน้าที่อะไร รวมถึงเมสเสจที่มีการสื่อสารโต้ตอบระหว่างกัน

UML (Unified Modeling Language)

Grady Booch, Ivar Jacobson และ James Rumbaugh ที่ได้ร่วมมือพัฒนา UML ขึ้นมา ได้นิยามไว้ว่า UML เป็นสัญลักษณ์ (Notation) ที่ใช้อธิบาย แสดงรายละเอียด จำลองการสร้าง และจัดการกับเอกสารต่าง ๆ ในระบบ เพื่อให้การออกแบบซอฟต์แวร์สามารถทำได้โดยง่ายและปรับปรุงวิธีการทำงานให้ดีขึ้น

UML Diagram

UML Diagram ประกอบไปด้วยไดอะแกรมต่าง ๆ มากมาย แต่ละไดอะแกรมต่างก็ให้มุมมองที่แตกต่างกัน เพื่อให้เกิดความเข้าใจในระบบงานมากขึ้น แต่ในการพัฒนาระบบงานอาจไม่จำเป็นต้องใช้ทุกไดอะแกรมก็ได้ ซึ่งอาจพิจารณาเพียงไดอะแกรมที่เหมาะสม และเพียงพอต่อความต้องการ

1. ยูสเคส (Use Case)

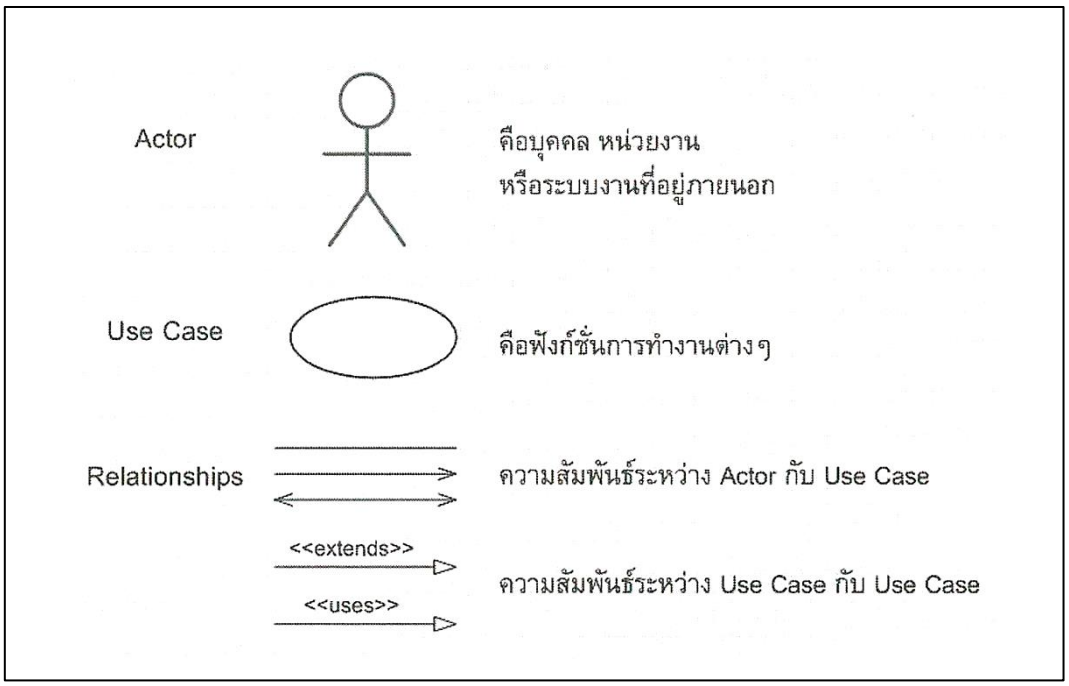
แผนภาพยูสเคสถือเป็นจุดเริ่มต้นของการรวบรวมความต้องการของผู้ใช้ โดยเป็นแผนภาพที่ถูกนำมาใช้เพื่อแสดงภาพรวมของระบบ ด้วยการอธิบายถึงพฤติกรรมหรือฟังก์ชั่นการทำงานของระบบว่ามีอะไรบ้าง เกี่ยวข้องกับใคร ซึ่งช่วยให้นักพัฒนาสามารถแยกแยะกิจกรรมที่เกิดขึ้นในระบบ ว่ามีฟังก์ชั่นการทำงานอะไร โดยฟังก์ชั่นการทำงานเหล่านี้ จะต้องได้รับการดำเนินงานโดยระบบ

แผนภาพยูสเคส ประกอบด้วย Actor, Use Case และ Relationship โดยที่

1.1 Actor มีสัญลักษณ์เป็นรูปคน หมายถึง ผู้เกี่ยวข้องกับระบบ เป็นองค์ประกอบที่แสดงถึงเอนทิตี้ที่อยู่ภายนอกระบบ และมีการปฏิสัมพันธ์กับระบบ รวมถึงแสดงความสัมพันธ์กับ Use Case

1.2 Use Case ใช้สัญลักษณ์รูปวงรี เพื่อแสดงฟังก์ชั่นการทำงานของระบบ หรือสิ่งที่ระบบต้องทำในมุมมองของผู้ใช้งาน

1.3 Relationships คือความสัมพันธ์ ซึ่งสามารถเป็นความสัมพันธ์ที่เกิดขึ้นระหว่าง Use Case กับ Use Case, Use Case กับ Actor หรือ Actor กับ Actor โดยจะมีรูปแบบความสัมพันธ์ต่าง ๆ เช่น Association, Aggregation, Composition และ Generalization



ภาพประกอบที่ 2-2 สัญลักษณ์ Use Case Diagram (โอภาส เอี่ยมสิริวงศ์, 2555, หน้า 410)

ความสัมพันธ์ระหว่างยูสเคส

ตามปกติแล้ว แบบจำลองของระบบใดระบบหนึ่ง มักจะมียูสเคสมากกว่า 1 ยูสเคส และแต่ละยูสเคสก็ยังมีความสัมพันธ์ระหว่างกันได้ โดยความสัมพันธ์ระหว่างยูสเคสที่มักพบเห็นกันอยู่บ่อย ๆ มีอยู่ 2 ชนิดด้วยกันคือ

<<uses>> หรือ <<include>> เป็นความสัมพันธ์แบบเรียกใช้งานยูสเคส อื่น ๆ ตามที่ต้องการ เพื่อทำงานร่วมกัน

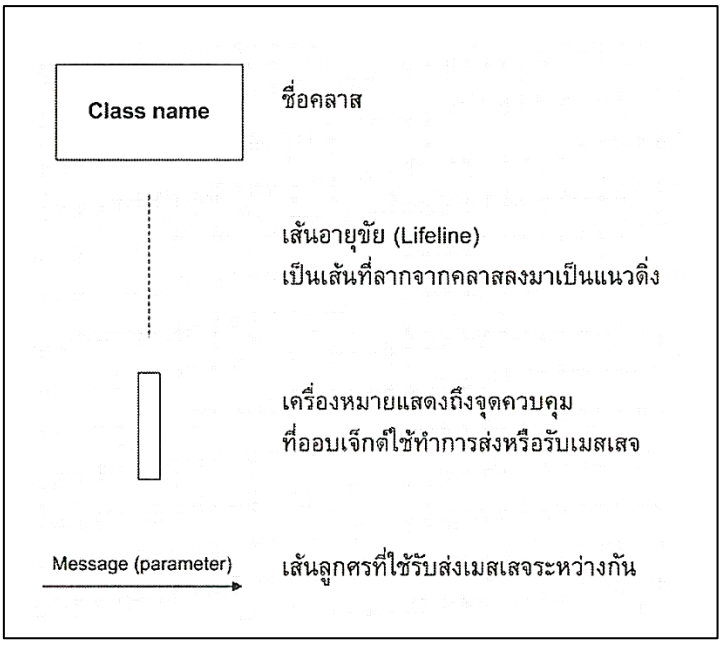
<<extended>> เป็นความสัมพันธ์แบบทางเลือก โดยยูสเคสที่เป็นส่วนขยายนั้น จะมีกระบวนการทำงานเพิ่มเติมจากยูสเคสหลัก ดังนั้น กรณีต้องการใช้ส่วนขยายเพิ่มเติม ก็ต้องเชื่อมความสัมพันธ์ของส่วนขยายเหล่านั้น ไปยังยูสเคสหลัก เพื่อให้สามารถรองรับการทำงานตามเงื่อนไขดังกล่าว

โดยทั้ง uses และ extended จะต้องเขียนอยู่ภายใต้สัญลักษณ์ << … >> ที่เรียกว่า Stereotype

2. Sequence Diagram

Sequence Diagram เป็นแผนภาพที่ใช้อธิบายการทำงานของยูสเคสนั้น ๆ ว่าออบเจ็กต์แต่ละตัวมีการสื่อสารกันอย่างไร มีลำดับขั้นตอนการทำงานอย่างไร นอกจากนี้ Sequence Diagram ยังเกี่ยวข้องกับเงื่อนไขเวลาที่ใช้ในการทำงานด้วย เช่น หากเวลาได้มีการเปลี่ยนแปลงขั้นตอนการทำงานก็จะเปลี่ยนไปด้วย

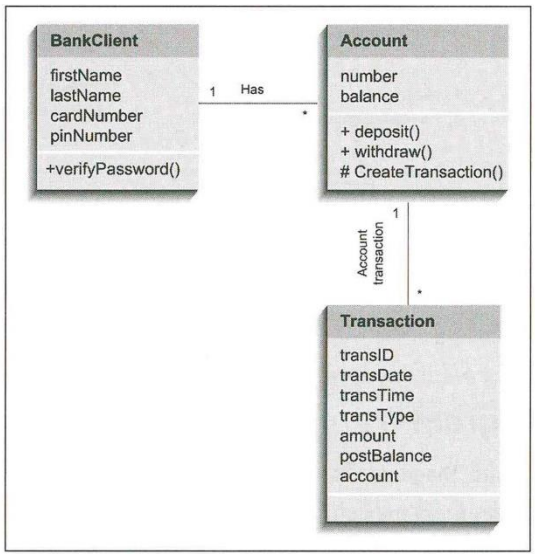
Sequence Diagram จะแสดงในรูปแบบ 2 มิติ โดยเส้นประแนวตั้ง จะนำเสนอเป็นแกนเวลา ส่วนเส้นแนวนอน จะเป็นการโต้ตอบระหว่างออบเจ็กต์หรือคลาส ทั้งนี้เส้นแนวตั้งหรือแนวดิ่งที่เป็นเส้นประ จะเรียกว่า เส้นอายุขัย (Lifeline) ที่แสดงช่วงเวลาของแต่ละคลาสโต้ตอบกัน ตั้งแต่เริ่มต้นถูกสร้างจนกระทั่งถูกทำลาย โดยสัญลักษณ์ต่าง ๆ ที่ใช้ใน Sequence Diagram เป็นไปดังภาพประกอบที่ 2-3



ภาพประกอบที่ 2-3 สัญลักษณ์ Sequence Diagram (โอภาส เอี่ยมสิริวงศ์, 2555, หน้า 414)

3. Class Diagram

Class Diagram เป็นแผนภาพที่ใช้แสดงโครงสร้างของระบบ ซึ่งประกอบไปด้วยคลาสต่าง ๆ และความสัมพันธ์ระหว่างคลาส โดยอาจเป็นได้ทั้งความสัมพันธ์แบบ Association, Aggregation, Composition และ Generalization



ภาพประกอบที่ 2-4 ตัวอย่าง Class Diagram (บางส่วน) ของระบบธนาคาร (โอภาส เอี่ยมสิริวงศ์, 2555, หน้า 415)

4. ระบบฐานข้อมูล (Database System)

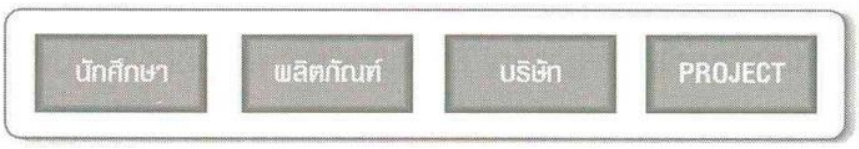
4.1 แบบจำลองอีอาร์

แบบจำลองอีอาร์ (Entity-Relationship Model: ER Model) เป็นภาษาแบบจำลองข้อมูลที่นิยมใช้ในการออกแบบเค้าร่างเชิงแนวคิดสำหรับฐานข้อมูลเชิงสัมพันธ์อย่างแพร่หลาย จนกลายเป็นมาตรฐานโดยพฤตินัย ถูกพัฒนาขึ้นโดยศาสตราจารย์ Peter Pin-Shin Chen ในปี ค.ศ. 1976 มีคุณลักษณะพื้นฐานเช่นเดียวกับภาษามนุษย์คือ ส่วนประกอบปฐมฐาน (Primitive Constructs) และวากยสัมพันธ์ (Syntax) ส่วนประกอบปฐมฐานของแบบจำลองอีอาร์เป็นสัญลักษณ์รูปภาพที่ใช้แทนหลักการสำคัญ 3 ประการคือ เอนทิตี้ แอตทริบิวต์ และความสัมพันธ์

4.1.1 เอนทิตี้ (Entity)

เอนทิตี้ หมายถึง สิ่งต่าง ๆ ที่มีแอตทริบิวต์ซึ่งต้องการเก็บในฐานข้อมูลเชิงสัมพันธ์ เอนทิตี้อาจเป็นสิ่งที่จับต้องได้ เช่น นักศึกษา ลูกค้า ผลิตภัณฑ์ และบ้าน หรือสิ่งที่เป็นนามธรรม เช่น บริษัท โครงการ มหาวิทยาลัย และบัญชีธนาคาร กล่าวอีกนัยหนึ่ง เอนทิตี้คือคลาสที่สร้างด้วยวิธีกำหนดสาระสำคัญแบบการแยกประเภทของกลุ่มออบเจ็คข้อมูล ซึ่งแอตทริบิวต์เหล่านี้จะถูกบันทึกไว้ในฐานข้อมูลเชิงสัมพันธ์ เช่น เอนทิตี้นักศึกษา คือ คลาสนักศึกษาที่เกิดจากการกำหนดสาระสำคัญแบบการแยกประเภทออบเจ็ค นายเก่ง นางสาวฤดี และนายสมหมาย เป็นต้น แอตทริบิวต์ของเอนทิตี้นักศึกษา คือ รหัสนักศึกษา ชื่อ ที่อยู่ และหมายเลขโทรศัพท์ เป็นต้น

เอนทิตี้ถูกแสดงในแบบจำลองอีอาร์ด้วยสัญลักษณ์สี่เหลี่ยม ภายในสี่เหลี่ยมมีชื่อที่ใช้ระบุเอนทิตี้ หากใช้ชื่อเป็นภาษาอังกฤษจะนิยมใช้ตัวพิมพ์ใหญ่ทั้งหมด สิ่งสำคัญคือในแบบจำลองอีอาร์เดียวกันห้ามมีเอนทิตี้ที่มีชื่อซ้ำกัน



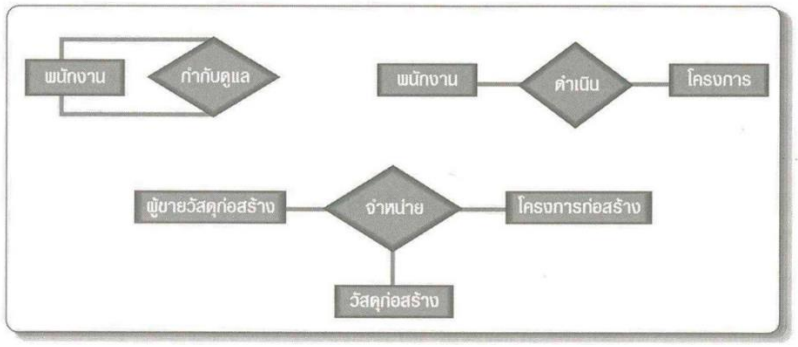
ภาพประกอบที่ 2-5 ตัวอย่างสัญลักษณ์เอนทิตี้ในแบบจำลองอีอาร์ (เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554, หน้า 47)

4.1.2 ความสัมพันธ์ (Relationship)

ความสัมพันธ์ในที่นี้คือความสัมพันธ์ระหว่างเอนทิตี้ หมายถึง การกำหนดสาระสำคัญแบบการรวมกลุ่มระหว่างเอนทิตี้ตั้งแต่ 2 เอนทิตี้ขึ้นไป เช่น การดำเนินโครงการเป็นความสัมพันธ์ระหว่างเอนทิตี้พนักงานและเอนทิตี้โครงการ และการดำเนินโครงการจะเกิดขึ้น

ไม่ได้หากขาดเอนทิตี้ใดเอนทิตี้หนึ่งไป ดังนั้นการดำเนินโครงการจึงเป็นการกำหนดสาระสำคัญแบบรวมกลุ่มระหว่างเอนทิตี้ทั้งสอง

ความสัมพันธ์ถูกแสดงในแบบจำลองอีอาร์ด้วยสัญลักษณ์สี่เหลี่ยมขนมเปียกปูน ภายในมีชื่อที่ระบุความสัมพันธ์ ซึ่งหากใช้ภาษาอังกฤษจะนิยมใช้ตัวพิมพ์ใหญ่ทั้งหมด และใช้เส้นใต้อักขระ (“_”) แทนการเว้นวรรค ชื่อความสัมพันธ์แสดงถึงบทบาท (Role) ของเอนทิตี้ที่เกี่ยวข้อง ในแบบจำลองอีอาร์เดียวกันห้ามมีความสัมพันธ์ที่มีชื่อซ้ำกัน นอกจากนี้ ความสัมพันธ์จะมีเส้นตรงเชื่อมต่อกับเอนทิตี้ที่เกี่ยวข้องทั้งหมด โดยนิยมจัดเรียงเอนทิตี้ที่เกี่ยวข้อง และความสัมพันธ์ในลักษณะที่สามารถอ่านได้ในทิศทางจากซ้ายไปขวา หรือบนลงล่าง



ภาพประกอบที่ 2-6 ตัวอย่างสัญลักษณ์ความสัมพันธ์และเอนทิตี้ที่เกี่ยวข้อง (เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554, หน้า 48)

เนื่องจากความสัมพันธ์มีพื้นฐานอยู่บนหลักการกำหนดสาระสำคัญแบบการรวมกลุ่มที่ได้อธิบายไว้ก่อนนี้ ความสัมพันธ์จึงสามารถแบ่งตาม ดีกรี ได้เป็น 4 ระดับเช่นกันคือ

4.1.2.1 ความสัมพันธ์แบบเอกภาค (Unary Relationship) ดังภาพประกอบที่ 2-6 บน (ซ้าย) ซึ่งมีความหมายว่า พนักงานที่เป็นหัวหน้ากำกับดูแลพนักงานที่เป็นผู้ใต้บังคับบัญชา

4.1.2.2 ความสัมพันธ์แบบทวิภาค (Binary Relationship) ดังภาพประกอบที่ 2-6 บน (ขวา)

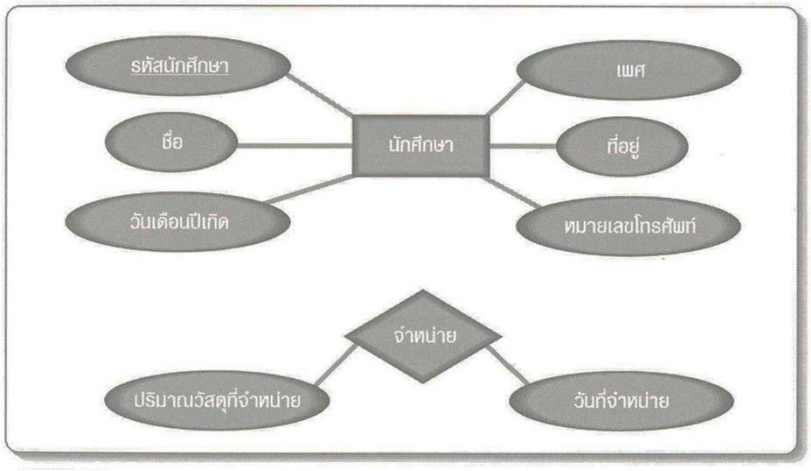
4.1.2.3 ความสัมพันธ์แบบไตรภาค (Ternary Relationship) ดังภาพประกอบที่ 2-6 ล่าง

4.1.2.4 ความสัมพันธ์แบบพหุภาค (N-Ary Relationship) ซึ่งแสดงด้วยสัญลักษณ์สี่เหลี่ยมขนมเปียกปูนที่มีเส้นตรงเชื่อมแต่ละเอนทิตี้ที่เกี่ยวข้องทั้งหมดมากกว่า 3 เอนทิตี้

4.1.3 ลักษณะประจำ (Attribute)

ลักษณะประจำ หรือแอตทริบิวต์ คือ คุณลักษณะเฉพาะของเอนทิตี้หรือความสัมพันธ์ เช่น เอนทิตี้นักศึกษามีแอตทริบิวต์ คือ รหัสนักศึกษา ชื่อนักศึกษา วันเกิด เพศ ที่อยู่ และหมายเลขโทรศัพท์ติดต่อ เป็นต้น เอนทิตี้โครงการประกอบด้วยแอตทริบิวต์ รหัสโครงการ ชื่อโครงการ ระยะเวลาดำเนินโครงการ รหัสผู้ดูแลโครงการ และสถานที่ดำเนินโครงการ เป็นต้น

แอตทริบิวต์ถูกแสดงในแบบจำลองอีอาร์ด้วยรูปวงรี ซึ่งเชื่อมต่อกับเอนทิตี้หรือความสัมพันธ์ที่เป็นเจ้าของแอตทริบิวต์ด้วยเส้นตรง ภายในวงรีกำกับด้วยชื่อแอตทริบิวต์ ซึ่งถ้าใช้ภาษาอังกฤษจะนิยมเขียนด้วยตัวพิมพ์เล็ก ยกเว้นอักษรตัวแรกที่เป็นตัวพิมพ์ใหญ่ และเพื่อป้องกันความสับสนที่อาจเกิดขึ้นภายหลังระหว่างการออกแบบฐานข้อมูล แอตทริบิวต์ในแบบจำลองอีอาร์เดียวกันควรมีชื่อที่ต่างกัน



ภาพประกอบที่ 2-7 ตัวอย่างแอตทริบิวต์ของเอนทิตี้และความสัมพันธ์ในแบบจำลองอีอาร์
(เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554, หน้า 51)

สิ่งสำคัญอีกประการหนึ่งคือ ทุกเอนทิตี้ต้องมีกุญแจหลัก 1 ชุดเสมอ เพื่อใช้เป็นดรรชนีระบุแยกแยะกรณีตัวอย่างเอนทิตี้ทั้งหมด แอตทริบิวต์ที่เป็นกุญแจหลักจะถูกแสดงในแบบจำลองอีอาร์ด้วยการขีดเส้นใต้ที่ชื่อ (เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554, หน้า 47-51)

4.2 การทำให้เป็นบรรทัดฐาน

การทำให้เป็นบรรทัดฐาน หรือนอร์ลมัลไลเซชั่น (Normalization) ซึ่งมีวัตถุประสงค์ดังนี้

4.2.1 เพื่อลดความซับซ้อนของการนำเสนอตารางความสัมพันธ์ (Simplify Relation Representation) และความซับซ้อนของคำสั่งคิวรี่ (คำสั่งฐานข้อมูลที่ใช้ในการค้นคืนข้อมูลจากฐานข้อมูล)

4.2.2 เพื่อลดความซ้ำซ้อนของข้อมูล เป็นการเพิ่มประสิทธิภาพในการเพิ่ม, การลบ และการปรับข้อมูลในฐานข้อมูล ส่งผลให้ความไม่ต้องกันของข้อมูลลดลง

4.2.3 เพื่อป้องกันการสูญหายของสารสนเทศและ ความผิดปกติของการปรับข้อมูล (Update Anomalies) ซึ่งจำแนกได้ 3 แบบคือ ความผิดปกติของการเพิ่มข้อมูล, การผิดปกติของการลบข้อมูล และความผิดปกติของการเปลี่ยนแปลงข้อมูล

การทำให้เป็นบรรทัดฐานเป็นกระบวนการวิเคราะห์ ภาวะการพึ่งพิงซึ่งกันและกันของแอตทริบิวต์ (Attribute Interdependency) ในแต่ละตารางความสัมพันธ์ โดยทดสอบเปรียบเทียบกับ เกณฑ์ของการเป็นรูปแบบบรรทัดฐาน (Normal Form Criteria) ซึ่งที่ใช้งานกันทั่วไปมีทั้งหมด 6 ระดับหรือ 6 รูปแบบบรรทัดฐาน ได้แก่ รูปแบบบรรทัดฐานที่ 1 (1NF), รูปแบบบรรทัดฐานที่ 2 (2NF), รูปแบบบรรทัดฐานที่ 3 (3NF), รูปแบบบรรทัดฐาน BCNF รูปแบบบรรทัดฐานที่ 4 (4NF), รูปแบบบรรทัดฐานที่ 5 (5NF) (เทพฤทธิ์ บัณฑิตวัฒนาวงศ์, 2554, หน้า 83)

4.3 ระบบฐานข้อมูล (Database System)

ฐานข้อมูล (Database) หมายถึง ชุดของข้อมูลที่มีความสัมพันธ์กันที่ถูกนำมาจัดเก็บไว้ด้วยกัน เพื่อให้สามารถใช้ข้อมูลเหล่านั้นร่วมกันได้อย่างรวดเร็วและมีประสิทธิภาพ ตัวอย่างของฐานข้อมูลอย่างง่าย ๆ และเกี่ยวข้องกับชีวิตประจำวันของเรา ได้แก่ สมุดโทรศัพท์ ซึ่งเป็นการจัดเก็บรวบรวมรายชื่อและเบอร์โทรศัพท์ของผู้ที่เราต้องการติดต่อด้วย หรือ การจัดเก็บข้อมูลค่าใช้จ่ายภายในครอบครัว เป็นต้น การจัดเก็บข้อมูลจะมีประสิทธิภาพได้ก็ต่อเมื่อมีวิธีการจัดการข้อมูลที่ดี กล่าวคือ วิธีการจัดเก็บและค้นคืนข้อมูลต้องเป็นไปอย่างมีประสิทธิภาพและรวดเร็ว เช่น มีการจัดเก็บรายชื่อแบ่งตามลำดับตัวอักษร เป็นต้น (วิเชียร เปรมชัยสวัสดิ์, 2546)

ระบบจัดการฐานข้อมูล (Database Management System: DBMS) หมายถึง ซอฟต์แวร์ระบบที่ใช้ในการจัดการฐานข้อมูล โดยมีวัตถุประสงค์หลักคือ การสร้างสภาวะแวดล้อมที่สะดวกและมีประสิทธิภาพในการเข้าถึงและจัดเก็บข้อมูลของฐานข้อมูล ระบบจัดการฐานข้อมูลจะทำหน้าที่ในการแปลความต้องการของผู้ใช้ให้อยู่ในรูปแบบที่สามารถทำงานได้กับฐานข้อมูลตามวัตถุประสงค์ของผู้ใช้

ระบบจัดการฐานข้อมูลจึงประกอบไปด้วยแฟ้มข้อมูลต่าง ๆ ที่มีความสัมพันธ์กัน และกลุ่มของโปรแกรม ที่ใช้เพื่อการเข้าถึงและเปลี่ยนแปลงแก้ไขข้อมูลในแฟ้มข้อมูลเหล่านั้น และเพื่อทำให้ระบบฐานข้อมูลง่ายต่อการใช้งานของผู้ใช้จึงมีการให้บริการผู้ใช้ด้วยข้อมูลเชิงนามธรรม (data abstraction คือ ข้อมูลที่มองเห็นในทางตรรกะ) โดยซ่อนรายละเอียดในส่วนของการจัดการข้อมูลที่มีความยุ่งยากไว้ภายในไม่ให้ผู้ใช้เห็น จึงง่ายต่อการเข้าใจและการใช้งาน (วิเชียร เปรมชัยสวัสดิ์, 2546, หน้า 13-14)

เทคโนโลยีที่เกี่ยวข้อง

1. ภาษาจาวา (JAVA)

ภาษาจาวา (JAVA) หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ “จาวา” ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้

ภาษาจาวาเป็นภาษาที่ใช้สำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

ประวัติภาษา JAVA

ภาษาจาวา เป็นภาษาโปรแกรมเชิงวัตถุที่พัฒนาขึ้นโดย “เจมส์ กอสลิง” และทีมวิศวกรของเขา ซึ่งบริษัทซันไมโครซิสเต็ม ต้องการนำภาษาจาวามาใช้แทนภาษา C++ ชื่อของ “จาวา” มาจากชื่อกาแฟที่ทีมวิศวกรของซันดื่มตอนที่ร่วมกันพัฒนาภาษาจาวาขึ้นมา Java ถูกคิดค้นและสร้างโดย บริษัท Sun Microsystems ซึ่งเป็นบริษัทผู้ขายระบบ Unix ที่มีชื่อว่า Solaris ซึ่งจุดเด่นของภาษา Java อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตนด้วย Java ได้พัฒนาขึ้นโดยทีมวิจัยของ บริษัท ซันไมโครซิสเต็ม (Sun Microsystems)พัฒนามาจากโครงการที่ต้องการพัฒนาระบบซอฟต์แวร์เพื่อควบคุมเครื่องใช้ไฟฟ้าขนาดเล็กภายในบ้านชื่อเดิมคือภาษา Oak ต่อมาเปลี่ยนชื่อเป็นภาษาจาวาภาษาจาวาเริ่มเป็นที่นิยม

แพร่หลายในปี ค.ศ. 1995 ภาษาจาวาเป็นภาษาที่ไม่ขึ้นกับแพลตฟอร์ม (platform independent) JDK 1.0 ประกาศใช้เมื่อปี 1996 JDK เวอร์ชันปัจจุบันคือ Java 8

ข้อดีของ ภาษาจาวา

– ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้คำหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น

– โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารถถูก 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 andprivate key management, access control และ certificatesของ

– มี IDE, application server, และ library ต่าง ๆ มากมายสำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย

ข้อเสียของ ภาษาภาษา

– ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่ง ทีล่ะคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา

– เครื่องมือที่มีในการใช้พัฒนาโปรแกรมจาวามักไม่ค่อยเก่ง ทำให้หลายอย่างโปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ เครื่องมือทำไม่ได้ ถ้าเราดู

เครื่องมือของ MS จะใช้งานได้ง่ายกว่า และพัฒนาได้เร็วกว่า (แต่เราต้องซื้อ tool ของ MS และก็ต้องรันบน platform ของ MS)

คุณลักษณะเด่นของภาษา Java

– ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์

– โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารถถูก compile และ run บนเครื่องพีซีธรรมดาได้

– เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่าโปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ถึง 4 เท่า และใช้เวลาในการเขียนโปรแกรม น้อยกว่าประมาณ 2 เท่า

– Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public andprivate key management, access control และ certificatesของภาษาจาวา

จุดเด่นของภาษาจาวา

– ความง่าย (simple)

– ภาษาเชิงออปเจ็ค (object oriented)

– การกระจาย (distributed)

– การป้อ้องกันการผิดพลาด (robust)

– ความปลอดภัย (secure)

– สถาปัตยกรรมกลาง (architecture neutral)

– เคลื่อนย้ายง่าย (portable)

– อินเตอร์พ์พรีต (interpreted)

– ประสิทธิภาพสูง (high performance)

– มัลติเธรด (multithreaded)

– พลวัต (dynamic)

รูปแบบของภาษาจาวา

ภาษาจาวา เป็นภาษาที่ไม่กำหนดแบบการเขียนโปรแกรม ในแต่ละบรรทัด แต่ละบรรทัดสามารถเขียนคำสั่งได้หลายคำสั่งสามารถแทรกคำอธิบาย (comment) จาวาเป็นภาษาที่บังคับอักขระตัวพิมพ์ใหญ่ ตัวพิมพ์เล็ก (Case Sensitiv) จาวามีตัวดำเนินการ(operators) หลายชนิด ให้ใช้งาน นอกจากคำสั่งนั้นเป็นคำสั่งที่ผู้ใช้สร้างขึ้นมาใหม่ อาจกำหนดเป็นตัวพิมพ์ใหญ่หรือตัวเล็กก็ได้ และ

สามารถเขียนชุดคำสั่งที่ประกอบด้วยตัวดำเนินการหลายตัวที่ต่างชนิดกันในชุดคำสั่งหนึ่งๆได้ โดยภาษา Java จะจัดลำดับการประมวลผลตามลำดับการทำงานของตัวดำเนินการ

รูปแบบคำสั่ง(statements) Java คือ ส่วนประมวลผล(Execute) เพื่อให้ได้ผลลัพธ์ออกมา ทุกคำสั่งจะต้องจบด้วยเครื่องหมาย เซมิโคลอน(;)

รูปแบบของ script

ในการเขียน script สามารถเขียน โดยในรูปแบบที่ 1 ได้โดยไม่ต้องระบุภาษาก็ได้ แต่ต้องเขียน tag ของ script ดังนี้

```
<Script>
        JavaScript statements;
</Script>
<Script>
        document.write(‘Khomkrit);
</Script>
```

ในการเขียน script ตามรูปแบบที่ 2 โดยระบุภาษาเป็น javascript และเขียนใน tag ของ script ดังนี้

```
<Script Language=”JavaScript”>
        JavaScript statements;
</Script>
```

การคำสั่งแสดงผล single quote (‘ ‘)

ในการเขียนการแสดงผลข้อมูลที่อยู่หลังคำสั่ง document นั้นสามารถเขียนใช้เครื่องหมายในแบบ single quote (‘ ‘) ก็ได้ดังนี้

```
<Script Language=”JavaScript”>
        document.write(‘Khomkrit);
</Script>
```

การใช้ HTML ร่วมกับ script ขึ้นบรรทัดใหม่ โดยใช้

การกำหนดให้ขึ้นบรรทัดใหม่ ส่วนใหญ่จะใช้รูปแบบของ tag HTML คือ
 โดยการใส่ไว้หลังคำสั่ง document อาจจะเป็นข้างหน้า หรือข้างหลังก็ได้

```
<Script Language=”JavaScript”>
        document.write(‘Khomkrit<br>’);
```

```
document.write('<font color="red">Khomkrit</font>');
</Script>
```

Source Code

ใน Java จะมี Source Code เป็น File ที่มีนามสกุล เป็น *.java เมื่อ ผ่านการ Compile แล้ว จะมี File เพิ่มมาเป็น File ที่มีนามสกุลเป็น *.class System.out.println ("…."); เป็นคำสั่งที่ใช้การแสดงตัวอักษรซอร์สโค้ดโปรแกรมจาวาอยู่ในแฟ้มที่มีนามสกุล java (ทศพล ธนะทิพานนท์ และ วรเศรษฐ สุวรรณิก, 2558)

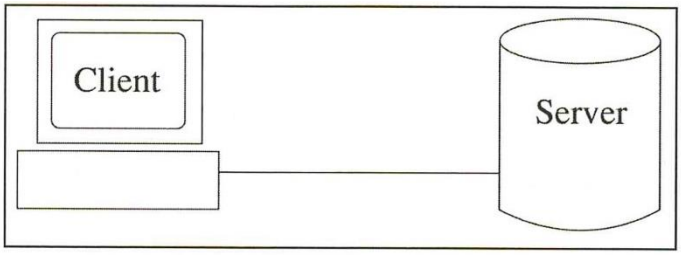
2. มายเอสคิวแอล (MySQL)

มายเอสคิวแอล (MySQL) เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต สาเหตุก็เพราะว่า MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในตลาดปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบฐานข้อมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถ ความรวดเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมหาศาล ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย ไม่ว่าจะเป็น Unix, OS/2, Mac OS หรือ Windows ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหลาย ไม่ว่าจะเป็น C, C++, Java, Perl, PHP, Python, Tcl หรือ ASP ก็ตามที ดังนั้นจึงไม่เป็นที่น่าแปลกใจเลยว่า ทำไม MySQL จึงได้รับความนิยมอย่างมากในปัจจุบัน และมีแนวโน้มสูงยิ่งขึ้นต่อ ๆ ไปในอนาคต

MySQL จัดเป็นซอฟต์แวร์ประเภท Open Source Software สามารถดาวน์โหลด Source Code ต้นฉบับได้จากอินเทอร์เน็ต โดยไม่เสียค่าใช้จ่ายใด ๆ การแก้ไขก็สามารถกระทำได้ตามความต้องการ MySQL ยึดถือสิทธิบัตรตาม GPL (GNU General Public License) ซึ่งเป็นข้อกำหนดของซอฟต์แวร์ประเภทนี้ส่วนใหญ่ โดยจะเป็นการชี้แจงว่า สิ่งใดทำได้ หรือทำไม่ได้ สำหรับการใช้งานในกรณีต่าง ๆ

2.1 สถาปัตยกรรมของ MySQL

สถาปัตยกรรม หรือโครงสร้างภายในของ MySQL ก็คือ การออกแบบการทำงานในลักษณะของ Client/Server นั่นเอง ซึ่งประกอบด้วยส่วนหลัก ๆ 2 ส่วน คือ ส่วนของผู้ให้บริการ (Server) และส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนก็จะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของตน



ภาพประกอบที่ 2-8 ส่วนประกอบการทำงานในลักษณะของ Client/Server (สงกรานต์ ทองสว่าง, 2545, หน้า 19)

หลักการทำงานในลักษณะ Client/Server มีดังนี้

2.1.1 ที่ฝั่งของ Server จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอคอยการร้องขอการใช้บริการจาก Client

2.1.2 เมื่อมีการร้องขอการใช้บริการเข้ามา Server จะทำการตรวจสอบตามวิธีการของตน เช่น อาจจะมีการให้ผู้ใช้บริการระบุชื่อและรหัสผ่าน และสำหรับ MySQL สามารถกำหนดได้ว่าจะอนุญาตหรือปฏิเสธ Client ใด ๆ ในระบบที่จะเข้าใช้บริการอีกด้วย

2.1.3 ถ้าผ่านการตรวจสอบ Server ก็จะอนุมัติการให้บริการแก่ Client ที่ร้องขอการใช้บริการนั้น ๆ ต่อไป และถ้าในกรณีที่ไม่ได้รับการอนุมัติ Server ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไปที่ Client ที่ร้องขอการใช้บริการนั้น

เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client หรือ Server อาจจะอยู่บนเครื่องเดียวกันหรือแยกเครื่องกันก็ได้ ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบ ตามปกติถ้าเป็นการทำงานในลักษณะ Web-based มีการใช้ฐานข้อมูลขนาดไม่ใหญ่มากนัก ตัว MySQL Server และ Client มักจะอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน (เช่นเนื้อที่ฮาร์ดดิสก์, RAM เป็นต้น) มากพอสมควร แต่สำหรับการทำงานจริง (Real-world Application) ก็มักจะแยก Client และ Server ออกเป็นคนละเครื่องกัน เพราะสามารถรองรับงานได้ดีกว่า มากกว่า ดังนั้น ผู้บริหารระบบ หรือผู้กำหนดนโยบายสำหรับการทำงานเครือข่าย จะต้องคำนึงถึงเรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะทำให้ระบบมีการทำงานรองรับการให้บริการแก่ผู้ใช้อย่างมีประสิทธิภาพ และข้อมูลมีความปลอดภัยมากที่สุด (สงกรานต์ ทองสว่าง, 2545, หน้า 17-23)

3. ภาษาจาวาสคริปต์ (JavaScript)

คือ ภาษายุคใหม่สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง เราสามารถเขียนภาษาจาวาเพิ่มเข้าไปในเว็บเพจ เพื่อใช้ประโยชน์สำหรับงานด้านต่าง ๆ ทั้งการคำนวณ, การแสดงผล, การรับ-ส่งข้อมูล, และที่สำคัญคือ สามารถโต้ตอบกับผู้ใช้ได้อย่างทันที นอกจากนี้ยังมีความสามารถด้านอื่น ๆ อีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับเว็บเพจของเราได้อย่างมาก ภาษาจาวาสคริปต์ถูกพัฒนาโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า ไลฟ์สคริปต์ (Live Script) ออกมาพร้อมกับเน็ตสเคปเนวิเกเตอร์ 2.0 (Netscape Navigator 2.0) เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบไลฟ์ไวร์ (Live Wire) ต่อมาเน็ตสเคปจึงได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของเบราว์เซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุงไลฟ์สคริปต์ใหม่เมื่อปี 2538 แล้วตั้งชื่อใหม่ว่าจาวาสคริปต์ (อนุสรา ลาทอง, 2551)

ลักษณะการทำงานของภาษาจาวา

เป็นภาษาสคริปต์เชิงวัตถุ หรือเรียกว่า อ็อบเจกต์โอเรียลเต็ด โปรแกรมมิ่ง (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนเอกสารด้วยภาษาเอชทีเอ็มแอล สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับภาษาแอชทีเอ็มแอล และภาษาจาวาทำงานได้ทั้งทางฝั่งผู้ใช้ และทางฝั่งเซิร์ฟเวอร์ โดยมีลักษณะการทำงานดังนี้

1. เนวิเกเตอร์ จาวาสคริปต์ (Navigator JavaScript) ซึ่งหมายถึง จาวาสคริปต์ที่ถูกแปลทางฝั่งไคลเอนต์ (ฝั่งเครื่องคอมพิวเตอร์ของผู้ใช้ ไม่ว่าจะเป็นเครื่องพีซี เครื่องแมคอินทอช หรืออื่น ๆ) จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่

2. ไลฟ์ไวร์ จาวาสคริปต์ (LiveWire JavaScript) เป็นเซิร์ฟเวอร์-ไซด์ จาวาสคริปต์ (Server-Side JavaScript) ซึ่งหมายถึง จาวาสคริปต์ที่ถูกแปลทางฝั่งเซิร์ฟเวอร์ (ฝั่งเครื่องคอมพิวเตอร์ของผู้ให้บริการเว็บ โดยอาจจะเป็นเครื่องของซันซิลิคอมกราฟิกส์ หรืออื่น ๆ) สามารถใช้ได้เฉพาะกับไลฟ์ไวร์ของเน็ตสเคปโดยตรง

4. Java Server Faces Framework

JSF เป็นเทคโนโลยีใหม่ของ java ที่น่าสนใจในการพัฒนาบนเว็บ ซึ่ง java เองได้มีแนวการพัฒนาเทคโนโลยีใหม่ ๆ ออกมาเรื่อย ๆ เพราะมีกลุ่ม partner อยู่มากมาย JSF ก็เช่นเดียวกันที่ได้พัฒนาขึ้นมาอีกขั้นหนึ่งต่อจาก JSP ซึ่งมีลักษณะพิเศษเป็น Framework เพื่อแก้ปัญหาการเขียนโปรแกรมที่ซ้ำแล้วซ้ำอีก และมีความน่าเบื่อของ JSP นอกจากนี้ยังคงมีลักษณะการทำงานแบบ Form และใช้วิธีของ HTTP request และ Response ข้อมูลไปมายังฐานข้อมูลได้อย่างง่ายได้ และมี user interface ที่ให้นักพัฒนาบนเว็บใช้ได้อย่างง่าย เช่น button text box check box และให้โปรแกรมเมอร์สามารถเขียนคำสั่งเพิ่มเติมได้สะดวก

JSF สร้างโดย Java Community Process ซึ่งกลุ่มผู้พัฒนามาจากหลากหลายบริษัท ได้แก่ Jakarta, Struts, Oracle, Sun, IBM และ ATG เพื่อแก้ปัญหาที่เกิดขึ้นในแบบเก่า การเขียนโปรแกรมบนเว็บด้วย JSF เป็นการเขียนที่แตกต่างการพัฒนาด้วยโปรแกรมอื่น ๆ เช่น PHP JSP และ ASP เพราะใช้หลักการของ Java ในการเขียน เว็บเพจเพียงหนึ่งเพจจะต้องมีไฟล์ที่เกี่ยวข้องอย่างน้อย 3 ไฟล์ ได้แก่ ไฟล์ .jsp .xml .java ต่างทำหน้าที่ในส่วนของตนเอง JSF มี features หลัก ๆ หลายอย่างดังนี้

- Page authors ผู้ที่เคยใช้คำสั่ง Tag HTML หรือ JSP ในการใช้ออกแบบเว็บเพจยังคงสามารถใช้แบบเดิมต่อไปได้ และยังสามารถใช้ Tag library ของ JSF แทรกเพิ่มเข้าไปได้

- Application Developers ผู้พัฒนาชุดคำสั่งเพื่อควบคุมการทำงาน สามารถบูรณาการเอาเว็บเพจที่ออกแบบตามข้อที่ 1 มาโปรแกรมเพิ่มเติมตามรูปแบบของ JSF อันได้แก่ ความสามารถทางด้าน Object จัดการ Event Converter และ Validator

- Component writers โปรแกรมเมอร์ที่เชี่ยวชาญแล้ว สามารถสร้าง UI component ของตนเองไว้ใช้ได้เพราะ JSF ยอมให้ใช้การ extensible และจัดการด้วยตนเองในเบื้องต้นอยู่แล้ว เราสามารถสร้าง Component เป็นของตนเองโดยตรงจาก UI Component class

MVC and JSF

MVC เป็นสถาปัตยกรรมการพัฒนาซอฟต์แวร์ใหม่ล่าสุดที่กล่าวถึงกันมาก ไม่ว่าจะเป็นโปรแกรม Ruby on Rail หรือโปรแกรมอื่น ๆ ต่างกล่าวถึงกัน ย่อมาจาก Model-View-Controller ซึ่งเป็นส่วนหนึ่งของ design pattern ที่ช่วยในการแยกส่วนของของการพัฒนาโปรแกรมออกเป็น 3 ส่วน 3 ไฟล์ คือ คำสั่งสำหรับอินเทอร์เฟสที่ผู้ใช้มองเห็นเป็นส่วนของ View เมื่อต้องการแก้ไขเราจะแก้ไขหน้าตาของโปรแกรมเฉพาะไฟล์นี้ สำหรับ Model ใช้สำหรับการเขียนคำสั่งทางด้าน Business logic หรือ และสุดท้ายได้แก่ Controller เป็นตัวควบคุมการประสานคำสั่งต่าง ๆ ที่เราต้องการคำนวณ java เรียกว่า Backing Bean (Nascent , 2556)

5. Application Server

Application Server คือ เซิร์ฟเวอร์ที่รันโปรแกรมประยุกต์ได้ด้วย โดยการทำงานสอดคล้องกับไคลเอ็นต์ เช่น Mail Server (รัน MS Exchange Server) Proxy Server (รัน Proxy Server) หรือ Web Server (รัน Web Server Program เช่น Xitami , Apache) ใช้ในการประมวลคือมันจะมาประมวลผลที่ Client ทำให้ลดภาระการประมวลผลที่ Server ลงได้ เหมาะกับงานที่ไม่ต้องใช้ประสิทธิภาพ Server สูงมาก Application Server ที่มีประสิทธิภาพในปัจจุบันมีหลากหลาย ซึ่งมุ่งเน้นพัฒนาในแต่ละด้านต่างกันมี Application Server ที่ต้องเสียค่าใช้จ่ายในด้านของค่าลิขสิทธิ์ software หากต้องการนำมาติดตั้ง ใช้งาน Application Server บางชิ้นไม่สามารถทำงานได้ในหลากหลาย platform หลายชิ้นมีประสิทธิภาพในการทำงานไม่เพียงพอต่อความต้องการของผู้ใช้งาน

JBoss มีคุณสมบัติต่าง ๆ ที่น่าสนใจสำหรับนำมาใช้งานที่สามารถตอบสนองความต้องการได้ อย่างมากมายไม่ว่าจะเป็นทางด้าน web service การออกแบบที่เป็น Enterprise มีกลไกการทำงานที่ไม่สลับซับซ้อนแต่ได้ประสิทธิภาพสูง มี service ต่าง ๆ รองรับการใช้งาน รวมทั้งมีคุณสมบัติที่น่าสนใจอื่น ๆ ทำให้ JBoss เป็น Application Server ที่น่านำมาศึกษาและทดลองใช้งาน

JBoss application server เป็นมาตรฐานทีเป็นพื้นฐานของ J2EE Platform ซึ่งเป็นอิสระในการใช้ ไม่ว่าจะเป็นผู้ใช้ทั่วไปหรือจะเป็น Independent - software vendor (ISV) ไม่คำนึงถึงขนาดของการ deploy JBoss application server กลายมาเป็น platform ที่ได้รับความนิยมมากสำหรับผู้พัฒนาและ ISV และเติบโตอย่างรวดเร็วในรูปแบบของ production deployments ด้วย

คุณลักษณะของ JBoss

1. Open source,no cost product licenses: JBoss เป็น open source ซึ่งไม่ต้องเสียค่าลิขสิทธิ์ เนื่องจากสามารถ download ได้ฟรี

2. Performance: JBoss อำนวยต่อการปรับปรุง และ การนำ J2EE Application server ไปใช้ให้เกิดประโยชน์

3. Customizable Footprint: JBoss มีอำนวยต่อการนำมาปรับปรุงใหม่ได้และ มีสถาปัตยกรรมที่ดีและยืดหยุ่นในการสร้าง application ซึ่งมี component model ที่จัดเตรียมไว้ให้ตามที่ต้องการ

4. Services-Oriented Architecture: JBoss มีการบริการสำหรับองค์กรที่แตกต่างออกไป รวมไปถึง Transaction Management, messaging, mail services, security, connection pooling Service ต่างๆเหล่านี้สามารถเพิ่มหรือนำออกไปได้ตามความต้องการ Service ทั้งหมดนี้มี package สำหรับจัดเก็บไว้อย่างเรียบร้อย โดยที่สามารถทำการสร้างและเพิ่ม service เข้าไปได้เอง

5. Enterprise-Class Services for any Java Object: JBoss Application Server สามารถใช้ Aspect-Oriented Programming (AOP) - model ในการส่งต่อไปยัง EJB คล้ายกับการส่ง function

6. Application-level features such as No compilation: ได้รับความนิยมมากจากผู้พัฒนาที่ใช้ภาษา Java ระดับสูง

7. Full security implementation: มีความปลอดภัยสูง ประกอบไปด้วยการผสมผสานกันของ JAAS (Anutsara, 2552)

6. IFT-IPC (Java Class Library)

IFT-IPC คือ Java Class Library (Package) ที่ถูกพัฒนาโดยบริษัทไอ เฟิร์สเทคจำกัด ทำงานในส่วนของการประมวลผลรูปภาพ โดยใช้หลักการของฟัซซีลอจิก (fuzzy logic) ในการพัฒนา การทำงานของ IFT-IPC จะทำการอ่านข้อมูลที่ถูกกำหนดไว้บนรูปภาพ โดยอ่านตัวอักษรเปรียบเทียบกับรูปแบบที่ถูกกำหนดไว้ใน IFT-IPC ขนาดตัวอักษรที่จะ๔อ่านได้ต้องกำหนดเป็นรูปแบบตัวอักษร Tahoma ขนาด 9-10 เท่านั้น เมื่ออ่านพบอักษรดังกล่าว IFT-IPC จะทำการอ่านพื้นที่โดยรอบของตัวอักษรนั้น เพื่อทราบถึงขนาดพื้นที่ทั้งหมดโดยคำนวณจากจุดสิ้นสุดของเส้นขอบที่พบ (i-firsttech, 2558)

ระบบงานที่เกี่ยวข้อง

ระบบติดตามความคืบหน้าการก่อสร้าง และการขาย iMAP เป็นซอฟต์แวร์ระบบใช้สำหรับการติดตามความคืบหน้าสถานะการก่อสร้าง และการขาย ของโครงการ ทาวน์เฮ้าส์ บ้านเดียว และคอนโดมิเนียม ของบริษัท เอพี (ไทยแลนด์) จำกัด (มหาชน)

1. แนวคิด

1.1 การติดตามความคืบหน้าการก่อสร้าง และการขาย เป็นไปตามหลักข้อมูลพื้นฐานของระบบของหน่วยงานในการติดตามความคืบหน้าสถานะการก่อสร้างของโครงการต่าง ๆ หรือการติดตามสถานการณ์ขายของโครงการต่าง ๆ

1.2 การติดตามความคืบหน้าการก่อสร้าง และการขาย เป็นไปตามหลักเกณฑ์และวิธีการบริหารกิจการขององค์กร ซึ่งมีหลักในการบริหารการจัดการเพื่อให้บรรลุเป้าหมายโดยมีการติดตามความคืบหน้าการก่อสร้าง และการขาย อย่างสม่ำเสมอ

1.3 การติดตามความคืบหน้าการก่อสร้าง และการขายกำหนดแนวทางตามประเภทของส่วนงานคือ

1.3.1 การติดตามความคืบหน้าส่วนงานก่อสร้าง

1.3.2 การติดตามความคืบหน้าส่วนงานขาย

2. วัตถุประสงค์ของติดตามความคืบหน้าการก่อสร้าง และการขาย

ติดตามความคืบหน้าการก่อสร้าง และการขาย มีวัตถุประสงค์ให้ได้ข้อมูลเกี่ยวกับการปฏิบัติงานของพนักงาน เพื่อประกอบการพิจารณาในเรื่องต่าง ๆ แล้วแต่กรณี ดังนี้

2.1 การดำเนินงานในส่วนของการก่อสร้างโครงการต่าง ๆ

2.2 การดำเนินงานในส่วนของการขายโครงการต่าง ๆ

2.3 การติดตามการดำเนินงานของพนักงาน

2.4 การปฏิบัติงานในแต่ละส่วนงาน

2.5 การประเมินผลความสำเร็จของโครงการต่าง ๆ

3. แนวทางการติดตามความคืบหน้าการก่อสร้าง และการขาย

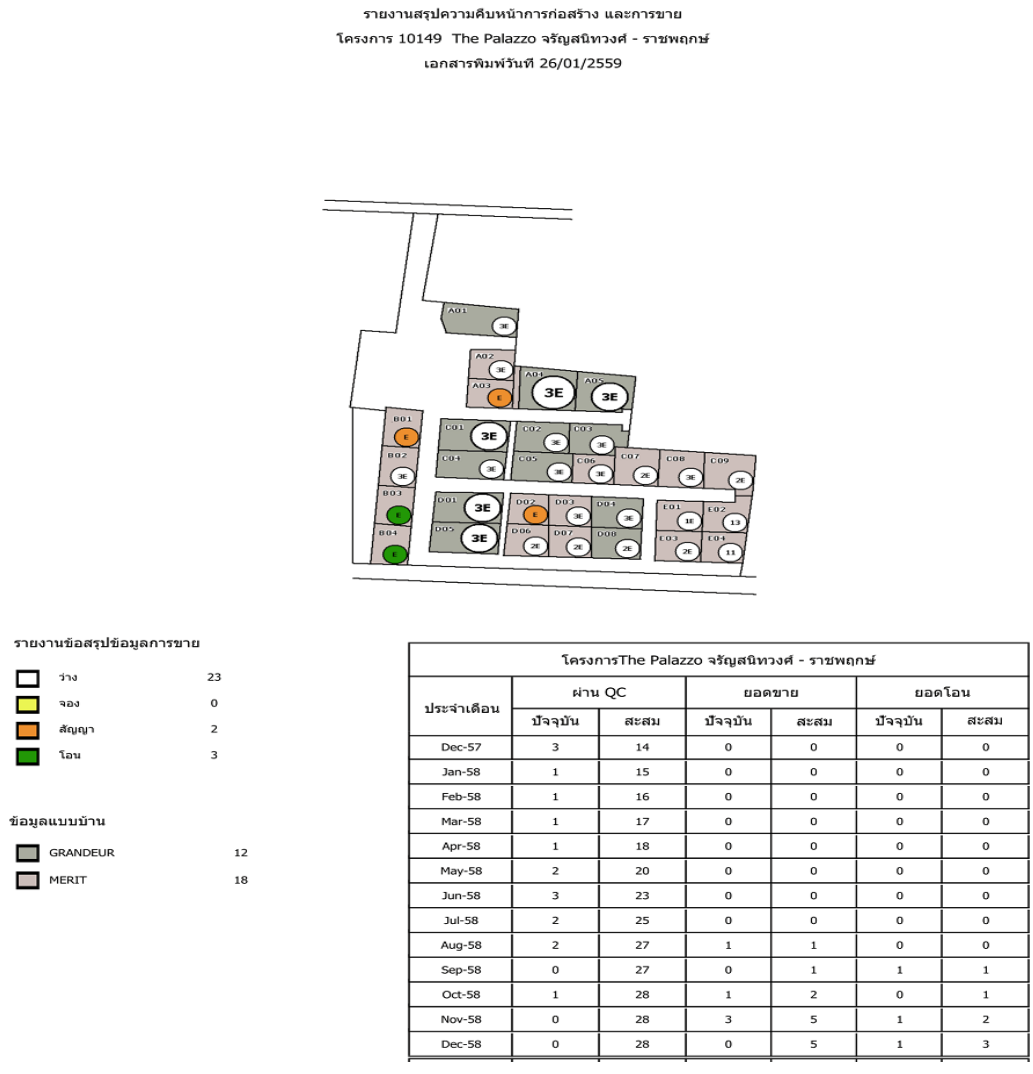
การติดตามความคืบหน้าการก่อสร้าง และการขาย เป็นแนวทางการติดตามความคืบหน้าการก่อสร้าง และการขาย กลุ่มงานบริการด้านการขาย และกลุ่มงานด้านการก่อสร้าง ซึ่งมีแนวทางการดำเนินการดังต่อไปนี้

3.1 การนำผลการติดตามการก่อสร้าง และการขาย ไปใช้สำหรับการประเมินสถานะการณ์ความเสี่ยงจะสำเร็จของโครงการ

3.2 การประเมินผลการทำงานของพนักงาน หรือผู้ที่เกี่ยวข้องเพื่อทราบถึงแนวทางการปฏิบัติงานให้สามารถดำเนินการได้อย่างมีประสิทธิภาพ

ระบบใช้สำหรับการติดตามความคืบหน้าสถานะการก่อสร้าง และการขาย ของโครงการต่างๆ ของบริษัท เอพี(ไทยแลนด์) จำกัด (มหาชน) มีลักษณะการทำงานเป็น Intranet มีพื้นขนาดปริมาณจำกัดในการเก็บข้อมูล และหน่วยประมวลผลค่อนข้างจำกัดในการใช้งานด้านการประมวลผล ทำให้พบปัญหาการใช้งานซอฟร์แวร์เกิดการหยุดทำงานเนื่องจากหน่วยความจำมีปริมาณไม่เพียงพอ มีความซับซ้อนด้านการจัดการข้อมูลโครงการ รวมถึงข้อมูลยูนิตต่างๆ ของโครงการ ในด้าน สถานะการก่อสร้าง และสถานะการขาย ทำให้เกิดข้อผิดพลาดด้านการประมวลผลพอสมควร อีกทั้งไม่รองรับการทำงานนอกพื้นที่องค์กร หน่วยงาน หรือบริษัท ด้านข้อมูล ระบบดังกล่าวไม่สามารถเพิ่ม หรือแก้ไขในส่วนของประเภทของโครงการได้ในกรณีที่องค์กร หน่วยงาน หรือบริษัท ต้องการขยายฐานธุรกิจขององค์กร หน่วยงาน หรือบริษัทให้มากยิ่งขึ้น หรือลดลงตามความต้องการขององค์กร หน่วยงาน หรือบริษัทได้ ผู้ศึกษาจึงได้นำแนวทางการปฏิบัติปัญหาดังกล่าวมาใช้สำหรับการพัฒนาระบบเพื่อลดข้อผิดพลาดดังกล่าว และเพื่อเป็นการเพิ่มประสิทธิภาพด้านการทำงาน ด้านการประมวลผล ผู้ศึกษาได้นำการคำนวณแบบคลาวด์มา

ใช้ เพื่อรองรับการขยายตัวของปริมาณการใช้งานได้อย่างไร้ขีดจำกัด รองรับข้อมูลมหาศาลที่จะเกิดขึ้นจากการประมวลผลของระบบ อีกทั้งยังสามารถเข้าถึงใช้งานระบบได้ทุกที่ทุกเวลา และง่ายต่อการจัดการ รองรับการขยายตัวของธุรกิจขององค์กร หน่วยงาน หรือบริษัท ได้ในอนาคต (AP Thailand , 2558)



โครงการThe Palazzo จรัญสนิทวงศ์ - ราชพฤกษ์						
ประจำเดือน	ผ่าน QC		ยอดขาย		ยอดโอน	
	ปัจจุบัน	สะสม	ปัจจุบัน	สะสม	ปัจจุบัน	สะสม
Dec-57	3	14	0	0	0	0
Jan-58	1	15	0	0	0	0
Feb-58	1	16	0	0	0	0
Mar-58	1	17	0	0	0	0
Apr-58	1	18	0	0	0	0
May-58	2	20	0	0	0	0
Jun-58	3	23	0	0	0	0
Jul-58	2	25	0	0	0	0
Aug-58	2	27	1	1	0	0
Sep-58	0	27	0	1	1	1
Oct-58	1	28	1	2	0	1
Nov-58	0	28	3	5	1	2
Dec-58	0	28	0	5	1	3

ภาพประกอบที่ 2-9 ผลการติดตามการก่อสร้างและการขายระบบ iMAP