

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การพัฒนาระบบ มีความจำเป็นอย่างยิ่งที่จะต้องใช้ทฤษฎีหรืองานวิจัยต่างๆที่เกี่ยวข้องมาเพื่อประกอบการพัฒนาระบบ ตั้งแต่เริ่มการศึกษาปัญหา ออกแบบวิเคราะห์ระบบ การออกแบบโปรแกรม ตลอดจนการเขียนชุดคำสั่งควบคุมการทำงานของโปรแกรม และการประมวลต่างๆ จนโปรแกรมสามารถทำงานได้สมบูรณ์ ตามความต้องการ เอกสารและงานวิจัยที่เกี่ยวข้องในการพัฒนาระบบการจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ มีดังนี้

- 2.1 ทฤษฎีที่เกี่ยวข้องกับระบบจัดการข้อมูล
 - 2.1.1 ทฤษฎีเกี่ยวกับการจัดการข้อมูล
 - 2.1.2 ทฤษฎีเกี่ยวกับการออกแบบเว็บไซต์
- 2.2 ทฤษฎีที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์
 - 2.2.1 วงจรการพัฒนาซอฟต์แวร์ (System Development Life Cycle : SDLC)
 - 2.2.2 แบบจำลองกระบวนการพัฒนาระบบ
 - 2.2.3 การพัฒนาระบบเชิงวัตถุ
- 2.3 ทฤษฎีที่เกี่ยวข้องกับการวิเคราะห์และออกแบบระบบ
 - 2.3.1 การวิเคราะห์และออกแบบระบบเชิงวัตถุด้วย UML
- 2.4 ทฤษฎีที่เกี่ยวข้องกับการออกแบบฐานข้อมูล
 - 2.4.1 การออกแบบฐานข้อมูล (Database Design)
 - 2.4.2 ฐานข้อมูลเชิงสัมพันธ์ (Relational Database)
- 2.5 ทฤษฎีที่เกี่ยวข้องกับเครื่องมือและภาษาที่ใช้ในการพัฒนาระบบ
 - 2.5.1 ทฤษฎีแนวคิดเอ็มวีซี (MVC : Model View and Controller)
 - 2.5.2 พีเอชพี (Hypertext Preprocessor : PHP)
 - 2.5.3 จาวาสคริปต์ (JavaScript)
 - 2.5.4 ภาษาเอสคิวแอล (SQL : Standard Query Language)
- 2.6 เอกสารและงานวิจัยที่เกี่ยวข้อง
 - 2.6.1 ระบบนักศึกษาฝึกงานวิทยาลัยเทคนิคบ้านค่าย
 - 2.6.2 ระบบจัดการการฝึกงาน
- 2.7 สถิติที่ใช้ในการประเมินผล

2.1 ทฤษฎีที่เกี่ยวข้องกับการจัดการข้อมูล

2.1.1 ระบบการจัดการข้อมูล

ข้อมูลเป็นส่วนหนึ่งที่มีความสำคัญของระบบสารสนเทศคอมพิวเตอร์ การจัดการข้อมูล (Data Management) เป็นกลยุทธ์หนึ่งในการบริหารองค์การหรือหน่วยงานให้มีประสิทธิภาพ การจัดการและบริหารองค์การให้ประสบความสำเร็จ ดังนั้นระบบการจัดการข้อมูลจึงมีความสำคัญต่อองค์การหรือหน่วยงานอย่างมาก

การจัดการข้อมูล หมายถึง การจัดเก็บข้อมูล การเรียกใช้ข้อมูล รวมถึงการวิเคราะห์ข้อมูล เพื่อการใช้งานที่รวดเร็วจึงจัดเก็บในลักษณะของฐานข้อมูลเพื่อให้สะดวกต่อการเรียกใช้ข้อมูล

ชนิดของข้อมูลจะประกอบด้วย 2 ส่วนด้วยกัน คือ กลุ่มข้อมูล (Data) และโอเปอเรชัน (Operation)

โครงสร้างข้อมูล เป็นการรวมกันของข้อมูลเชิงเดี่ยวและข้อมูลเชิงประกอบเข้าด้วยกันกำหนดความสัมพันธ์ให้กับข้อมูลเหล่านั้น นอกจากนี้โครงสร้างข้อมูลก็ยังสามารถมีซ้อนกันได้ คือสามารถกำหนดโครงสร้างข้อมูลทีประกอบไปด้วยโครงสร้างข้อมูลอื่นๆ ทับซ้อนกันได้ เช่น โครงสร้างข้อมูลแบบอาร์เรย์ (Array) และ เรคอร์ด (Record)

การศึกษาทฤษฎีที่เกี่ยวข้องกับการจัดการข้อมูลนี้นำมาใช้ในส่วนของการออกแบบและจัดเก็บข้อมูลระบบสารสนเทศ เพื่อให้ได้ระบบการจัดการข้อมูลที่ดีและมีประสิทธิภาพ สามารถจัดเก็บข้อมูลที่ถูกต้องและครบถ้วนของข้อมูล

2.1.2 ทฤษฎีเกี่ยวกับการออกแบบเว็บไซต์

การสร้างเว็บไซต์ให้มีคุณภาพนั้นจะต้องเกี่ยวข้องกับข้อมูลมากมาย ซึ่งจะต้องมีการวิเคราะห์และตัดสินใจก่อนที่จะลงมือทำ ซึ่งมีขั้นตอนในการพัฒนาเว็บไซต์ ดังต่อไปนี้ กำหนดเป้าหมาย การวางแผน วิเคราะห์จัดโครงสร้างข้อมูล ออกแบบเว็บเพจ เตรียมข้อมูล ลงมือพัฒนาและทดสอบติดตั้ง และส่งเสริมให้เป็นที่รู้จักดูแลและปรับปรุงต่อเนื่องและอีกทั้งการออกแบบเว็บไซต์ซึ่งจะต้องมีการแบ่งออกแบบเจ็ทต่าง ๆ ซึ่งจะแบ่งออกเป็นฝั่งของไคลเอนต์กับเซิร์ฟเวอร์และกำหนดยูสเซอร์อินเทอร์เฟซของเว็บเพจ การแบ่งออกแบบเจ็ทในเว็บแอปพลิเคชันเป็นสิ่งที่มีความสำคัญอย่างมากซึ่งจะขึ้นอยู่กับสถาปัตยกรรมที่ผู้พัฒนาเลือกใช้

1) การออกแบบอินเว็บไคลเอนท์ (Thin Web Client) จะมีข้อกำหนดส่วนมากอยู่ที่เรื่องเว็บเพจ ซึ่งจะกำหนดให้ในแต่ละเพจจะต้องมีอีลิเมนต์ที่สนับสนุนเอชทีเอ็มแอล (Hyper Text Markup Language : HTML) การออกแบบ Thin Web Client จะใช้วิธีการนำโมเดลเชิงวิเคราะห์มาแปลงเป็นโมเดลเชิงออกแบบ จะมองไปที่ actor กับเซิร์ฟเวอร์เพจ โดยที่ actor จะติดต่อเฉพาะกับเพจของไคลเอนต์ ส่วนเซิร์ฟเวอร์เพจจะติดต่อเฉพาะกับริชเชอร์สทางฝั่งเซิร์ฟเวอร์เท่านั้น

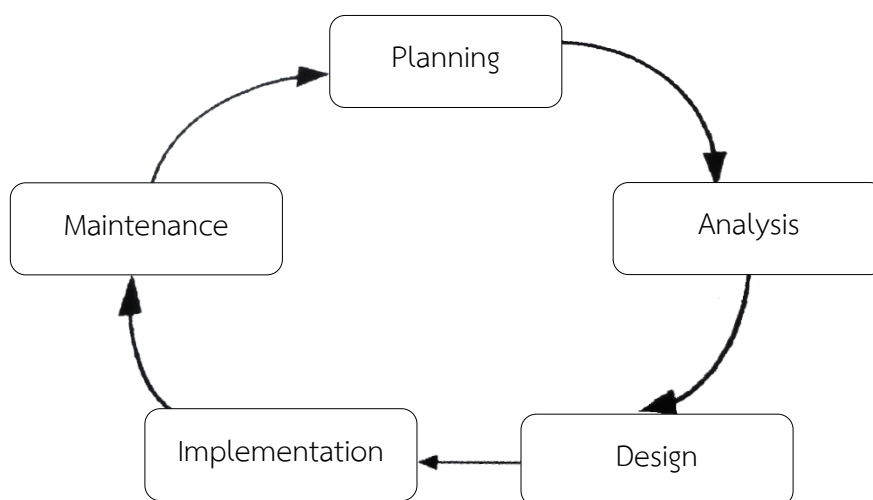
2) การออกแบบอีซีเว็บไคลเอนท์ (Thick Web Client) เป็นการออกแบบเว็บแอปพลิเคชัน ซึ่งจะมีส่วนที่เป็นไดนามิกเพจ ผู้ใช้ระบบสามารถกำหนดออบเจกต์ต่าง ๆ เองได้ ซึ่งในการออกแบบ Thick Web Client จะเริ่มที่การสร้าง ซีควেনซ์ ไดอะแกรม (Sequence Diagrams)

การศึกษาทฤษฎีที่เกี่ยวข้องกับการออกแบบเว็บไซต์นี้นำมาใช้ในส่วนของการออกแบบ การวางแผน ออกแบบหน้าจอของระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ

2.2 ทฤษฎีที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์

2.2.1 วงจรการพัฒนาซอฟต์แวร์ (System Development Life Cycle : SDLC)

การพัฒนาระบบ SDLC เป็นกระบวนการทางความคิด หรือการศึกษา วิเคราะห์ ถึงปัญหาที่เกิดขึ้นในระบบ พร้อมทั้งเสนอแนวทางแก้ไขตามความต้องการของผู้ใช้งานหรือหน่วยงาน ภายในวงจร SDLC แบ่งกระบวนการพัฒนาออกเป็นระยะได้แก่ การวางแผน การวิเคราะห์ การออกแบบ การพัฒนาและติดตั้ง และการบำรุงรักษา ดังภาพที่ 2-1



ภาพที่ 2-1 แสดงวงจรการพัฒนาซอฟต์แวร์ SDLC

(ที่มา: กิตติ ภักดีวัฒนกุล, 2551)

1) การวางแผน (Planning) เป็นระยะเริ่มแรกที่คุณพัฒนาจะต้องทำการสำรวจความต้องการของผู้ใช้งานระบบและนำมาวิเคราะห์เพื่อค้นหาโครงการพัฒนาระบบเพื่อตอบสนองความต้องการของผู้ใช้ได้ และกำหนดขอบเขตของระบบใหม่ ศึกษาความเป็นไปได้ของโครงการ จัดตารางดำเนินงาน วางแผนการใช้ทรัพยากร

2) การวิเคราะห์ (Analysis) เป็นขั้นตอนการศึกษาการดำเนินการของระบบงานเดิมเพื่อทำความเข้าใจกับปัญหาที่เกิดขึ้น รวบรวมความต้องการในระบบใหม่จากผู้ใช้งานระบบ จากนั้นนำความต้องการเหล่านั้นมาวิเคราะห์เพื่อหาทางแก้ไขปัญหา โดยนำแบบจำลองต่าง ๆ มาช่วยในการวิเคราะห์เพื่อนำไปออกแบบระบบในขั้นตอนต่อไป

3) การออกแบบ (Design) เป็นขั้นตอนการออกแบบระบบสารสนเทศที่จะนำมาใช้แก้ไขปัญหาหรือตอบสนองความต้องการที่ได้ทำการวิเคราะห์ไว้ ซึ่งจะมีรายละเอียดขององค์ประกอบส่วนต่าง ๆ ของระบบ

4) การพัฒนาและติดตั้ง (Implementation) เป็นขั้นตอนการพัฒนาหรือสร้างระบบ โดยการเขียนโปรแกรมเพื่อตอบสนองความต้องการของผู้ใช้ จากนั้นทำการทดสอบและติดตั้งระบบ จัดทำคู่มือประกอบการใช้งาน

5) การบำรุงรักษา (Maintenance) เป็นขั้นตอนหลังทำการทดสอบและติดตั้งเรียบร้อยแล้วจะต้องคอยดูแลการทำงานของระบบใหม่ให้ราบรื่น และมีประสิทธิภาพอยู่เสมอคอยให้ช่วยเหลือสนับสนุน แก้ไขปัญหาที่เกิดขึ้นระหว่างการใช้งาน

2.2.2 แบบจำลองกระบวนการพัฒนาระบบ

แบบจำลองกระบวนการพัฒนาระบบซึ่งเป็นแบบจำลองภาพของกระบวนการพัฒนาระบบ เพื่อให้ผู้พัฒนาได้เห็นการจัดโครงสร้างลำดับขั้นของแต่ละกระบวนการ ซึ่งแบบจำลองกระบวนการนี้เป็นการนำเสนอกระบวนการพัฒนาระบบในแบบนามธรรม ดังนั้นจึงมีรายละเอียดเพียงบางส่วน ดังนี้

2.2.2.1 วอเตอร์ฟอลล์ โมเดล (Waterfall Model) แบบจำลองกระบวนการแบบวอเตอร์ฟอลล์ โมเดล นี้เหมาะสมสำหรับการพัฒนาโปรแกรมสำเร็จรูปที่เป็นลักษณะที่ผลิตออกมาทีละเยาะๆ ซึ่งจะมีลักษณะการทำงานที่ไหลจากที่สูงลงสู่ที่ต่ำ โดยไม่สามารถไหลย้อนกลับได้ คือ การทำงานในขั้นตอนต่อไป จะเริ่มต้องได้ก็ต่อเมื่อขั้นตอนก่อนหน้าเสร็จสิ้น และจะไม่ย้อนกลับไปทำงานขั้นตอนเดิมอีก ขั้นตอนต่างๆ มีรายละเอียดดังแสดงในภาพที่ 2-2

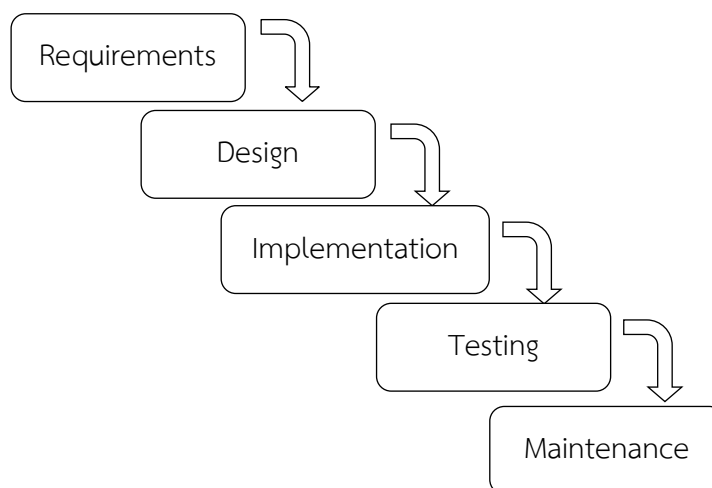
1) รวบรวมความต้องการของผู้ใช้ระบบ (Requirement) เป็นขั้นตอนในการเก็บความต้องการของผู้ใช้โปรแกรมว่า ผู้ใช้โปรแกรมต้องการโปรแกรมอะไร ทำงานอย่างไร

2) ขั้นตอนการวิเคราะห์และออกแบบระบบ (Analysis and Design) เป็นขั้นตอนในการเอาความต้องการของผู้ใช้ระบบมาวิเคราะห์ถึงความเป็นไปได้และความเหมาะสม เลือกเครื่องมือที่จะทำการพัฒนาโปรแกรม แล้วทำการออกแบบโปรแกรม

3) การเขียนโปรแกรม (Coding) เป็นขั้นตอนของการเขียนโปรแกรมตามที่เราก่อแบบ

4) การทดสอบโปรแกรม (Testing) เป็นขั้นตอนการเอาโปรแกรมที่เสร็จเรียบร้อยแล้วมาทำการทดสอบหาข้อผิดพลาด

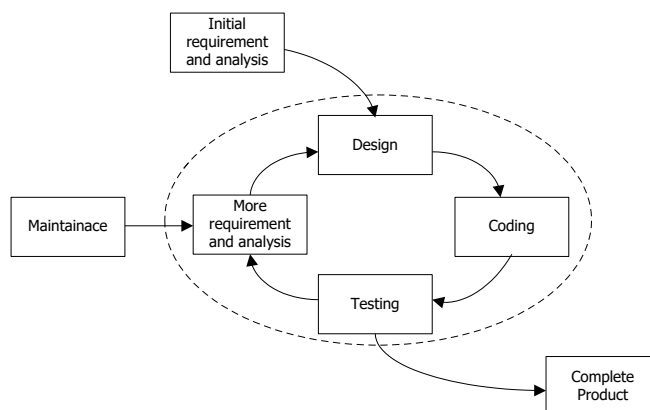
5) การบำรุงรักษาระบบ (Maintenance) เป็นขั้นตอนการดูแลผู้ใช้โปรแกรมว่ามีปัญหาในการใช้งานหรือไม่อย่างไร พร้อมให้คำปรึกษาและรับฟังข้อคิดเห็นเพื่อจะนำมาพัฒนาโปรแกรมในรุ่นถัดไป



ภาพที่ 2-2 แสดงกระบวนการพัฒนาระบบแบบ Waterfall Model

(ที่มา: <http://waterwaterfall.blogspot.com/2014/12/waterfall-software-development.html>)

2.2.2.2 ไอเทอเรทีฟ โมเดล (Iterativemodel) เนื่องจากวอเตอร์ฟอลล์ โมเดลมีการวางแผนที่ดีในการวิเคราะห์และออกแบบ เพราะการทำงานแบบ วอเตอร์ฟอลล์ โมเดลนั้นไม่สามารถย้อนกลับไปทำขั้นตอนก่อนหน้านี้ได้จึงทำให้มีความเสี่ยงต่อการพัฒนาและทั้งยังขาดการรับฟังข้อคิดเห็นจากผู้ใช้โปรแกรมเพื่อที่จะนำมาปรับปรุงโปรแกรมโดยทันทีทำให้โปรแกรมที่ออกมาไม่ตรงกับความต้องการของผู้ใช้โปรแกรม จึงทำให้เกิดโมเดล ในการพัฒนาโปรแกรมแบบใหม่ที่เรียก ไอเทอเรทีฟ โมเดล ซึ่งมีรายละเอียดดังภาพที่ 2-3



ภาพที่ 2-3 แสดงกระบวนการพัฒนาระบบแบบ Iterative model

(ที่มา: <http://softwaretesting-pvn.blogspot.com/2010/06/software-development-process-models.html>)

- 1) การวิเคราะห์ความต้องการของระบบ (Initial Requirement and Analysis) เป็นการเก็บรวบรวมความต้องการของผู้ใช้โปรแกรมจากนั้นนำมาวิเคราะห์ พร้อมทั้งหาเครื่องมือที่จะใช้ในการพัฒนาโปรแกรม
- 2) ขั้นตอนการออกแบบระบบ (Design) เป็นขั้นตอนในการออกแบบโปรแกรมตามที่ได้วิเคราะห์ไว้
- 3) การเขียนโปรแกรม (Coding) เป็นขั้นตอนในการพัฒนาโปรแกรมตามที่เราได้วิเคราะห์และออกแบบไว้
- 4) การทดสอบโปรแกรม (Testing) เป็นขั้นตอนการทำการทดสอบโปรแกรมที่ได้ทำการพัฒนาขึ้นเพื่อหาข้อผิดพลาดต่างๆ ของโปรแกรม ซึ่งผู้ที่ทำการตรวจสอบโปรแกรมจะมีทั้งการตรวจสอบภายในทีมเขียนโปรแกรมและการตรวจสอบจากผู้ใช้โปรแกรม
- 5) วิเคราะห์ความต้องการของระบบเพิ่มเติม (More Requirement and Analysis) ถ้าผู้ใช้โปรแกรมตรวจสอบโปรแกรมแล้วยังไม่เป็นที่น่าพอใจหรือยังพบข้อผิดพลาดของโปรแกรม ซึ่งผู้พัฒนาจะนำเอาข้อเสนอเหล่านั้นจากผู้ที่ทำการตรวจสอบมาทำการแก้ไขเพิ่มเติม โดยทำการออกแบบใหม่อีกรอบหนึ่ง เพื่อได้โปรแกรมที่มีประสิทธิภาพความต้องการของผู้ใช้งาน
- 6) นำมาใช้งานจริง (Complete Product) ถ้าการทดสอบเป็นที่น่าพอใจแล้วจึงเริ่มใช้ผลิตภัณฑ์กับงานระบบจริง
- 7) การบำรุงรักษาระบบ (Maintenance) ถึงกระนั้นผลิตภัณฑ์ที่ใช้งานจริงย่อมมีปัญหาเกิดขึ้นเสมอผู้ใช้งานอาจมีความต้องการเพิ่ม จึงนำเอาความต้องการเหล่านั้นเข้าสู่ขั้นตอนการพัฒนาโปรแกรมอีกครั้งซึ่งจะอยู่ในขั้นตอนของการวิเคราะห์ความต้องการของระบบเพิ่มเติม

ผู้พัฒนาระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ ได้นำทฤษฎีที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์มาประยุกต์ใช้กับขั้นตอนการศึกษาถึงปัญหาของระบบงานและความเป็นไปได้ของงาน รวมของการนำไปวิเคราะห์ ออกแบบระบบ ขั้นตอนการพัฒนา ระบบ ขั้นตอนการดำเนินการและวิธีการบำรุงรักษา เพื่อให้การทำงานเป็นไปตามขั้นตอนและมีประสิทธิภาพมากยิ่งขึ้น

2.2.3 กระบวนการพัฒนาระบบเชิงวัตถุ

การพัฒนาระบบเชิงวัตถุ (Agile Development) มีกระบวนการทำงานที่คล้ายกระบวนการพัฒนาระบบโดยทั่วไป ซึ่งจะมีการวิเคราะห์ความต้องการของระบบ การวิเคราะห์ระบบ การออกแบบระบบ การพัฒนา ติดตั้งระบบ และการบำรุงรักษาระบบ กระบวนการพัฒนาระบบเชิงวัตถุจะมุ่งเน้นที่ลำดับของการดำเนินการและผลลัพธ์ที่เกิดจากกระบวนการ เพื่อช่วยให้ประสิทธิภาพในการดำเนินการนอกจากนี้ยังมีกระบวนการที่เรียกว่า “Artifact” ได้แก่ เอกสาร โปรแกรม หน้าตาเว็บไซต์ โครงสร้างฐานข้อมูล แผนภาพ และโมเดลต่าง ๆ เป็นต้น

การศึกษาทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบเชิงวัตถุนี้นำมาใช้ในส่วนของการวิเคราะห์ระบบเชิงวัตถุ ในการวางแผน และออกแบบระบบการทำงาน ของระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ

2.3 ทฤษฎีที่เกี่ยวข้องกับการวิเคราะห์และออกแบบเชิงวัตถุ

2.3.1 การวิเคราะห์และออกแบบเชิงวัตถุด้วย UML

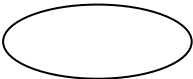



การวิเคราะห์และออกแบบเชิงวัตถุ (Object-Oriented Analysis Design: OOAD) เป็นการจำลองแบบเชิงวัตถุ ซึ่งจะเป็นตัวแทนของระบบสารสนเทศ เพราะออกแบบเจ็ดทำหน้าที่ปฏิบัติงานและเป็นตัวโต้ตอบกับระบบ การใช้วัตถุ เป็นตัวหลักในการพิจารณาความเป็นจริงต่าง ๆ ที่เกิดขึ้น โดยมองทุกสิ่งเป็นวัตถุทั้งหมด และกิจกรรมที่เกิดขึ้นนั้นเกิดจากความสัมพันธ์และปฏิสัมพันธ์ระหว่างวัตถุ

ยูเอ็มแอล (Unified Modeling Language : UML) เป็นภาษาที่ใช้ในการอธิบายแบบจำลองต่าง ๆ หรือเป็นแผนภาพสำหรับสร้างตัวแบบเชิงวัตถุ และเป็นภาษามาตรฐานสำหรับสร้างแบบพิมพ์เขียวให้แก่ระบบงาน ซึ่งสามารถใช้ในการสร้างมุมมอง กำหนดรายละเอียดในการสร้างระบบงาน และจัดทำเอกสารอ้างอิงให้แก่ระบบงานได้ เนื่องจากยูเอ็มแอล เป็นภาษาที่มีการใช้สัญลักษณ์รูปภาพ

แบบจำลอง (Modeling) เป็นวิธีการวิเคราะห์ออกแบบ (Analysis and Design) อย่างหนึ่งที่น่าสนใจการใช้งานแบบจำลองเป็นหลัก ซึ่งแบบจำลองที่สร้างขึ้นมาจะสามารถช่วยให้เข้าใจในปัญหาได้ง่ายขึ้น อีกทั้งยังสามารถนำแบบจำลองมาเป็นเครื่องมือในการสื่อสารถ่ายทอดความคิดกับบุคคลอื่น ๆ ที่เกี่ยวข้องในระบบงานได้ ส่วนแบบจำลองภาพ คือการใช้สัญลักษณ์รูปภาพในการสร้างแบบจำลองของระบบงาน ที่จะพัฒนาเพื่อทำความเข้าใจกับความต้องการของผู้จัดทำระบบ การออกแบบระบบที่เป็นไปได้อย่างชัดเจนขึ้นและการบำรุงรักษาที่ง่ายยิ่งขึ้น แบบจำลองเกิดขึ้นโดยการนำเสนอส่วน

ต่าง ๆ ของระบบซึ่งมีเพียงส่วนที่สำคัญโดยไม่คำนึงถึงรายละเอียดปลีกย่อยต่าง ๆ ในการพัฒนาระบบซอฟต์แวร์ที่ซับซ้อน ผู้พัฒนาจึงต้องทำความเข้าใจกับมุมมองด้านต่าง ๆ ของระบบก่อนทำการพัฒนาจริง โดยการสร้างแบบจำลองอันเปรียบเสมือนพิมพ์เขียวที่แสดงถึงภาพรวมทั้งหมดของระบบ แบบจำลองที่สร้างขึ้นจะต้องมีความสอดคล้องกับความต้องการของผู้ใช้งานระบบเป็นสำคัญ ในส่วนของรายละเอียดต่าง ๆ จะค่อย ๆ ถูกเพิ่มเติมลงไปในตัวแบบจำลองและในที่สุดแบบจำลองจะถูกนำไปพัฒนาขึ้นเป็นระบบจริง แบบจำลองที่ผู้พัฒนาได้นำมาใช้ในการวิเคราะห์และออกแบบระบบ ได้แก่ ยูสเคส ไดอะแกรม (Use Case Diagram) คลาสไดอะแกรม (Class Diagram) ซีควเอนซ์ไดอะแกรม (Sequence Diagram)

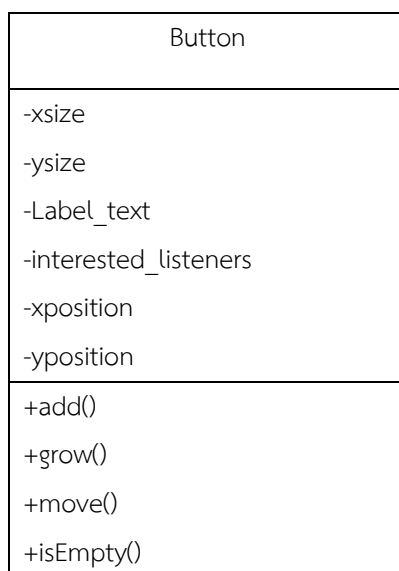
ยูสเคส ไดอะแกรมเป็นแผนภาพแสดงการใช้งานจากผู้ใช้งานระบบ (Actor) หรือ Use Case และความสัมพันธ์ระหว่าง Use Case และผู้ใช้งานระบบ ซึ่งสัญลักษณ์ของยูสเคส ไดอะแกรมมีดังแสดงในภาพที่ 2-4

สัญลักษณ์	รายละเอียด
1. use case name 	Use case เป็นหน้าที่ ที่ระบบจะต้องทำ
2. actor name 	Actor เป็นผู้ที่เกี่ยวข้องกับระบบ ทำหน้าที่ผลักดันให้เกิดกิจกรรมของระบบ หรือ ทำหน้าที่ดูแลกิจกรรมของระบบ
3. System name 	System boundary เป็นเส้นแบ่งขอบเขตระหว่างระบบและ actor
4. Connection 	Connection เป็นเส้นเชื่อมระหว่าง actor และ use case

ภาพที่ 2-4 แสดงสัญลักษณ์ของ Use Case Diagram

(ที่มา : http://communicationdiagram.blogspot.com/2014_05_01_archive.html)

แผนภาพคลาส (Class Diagram) เป็นแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแง่ต่าง ๆ ระหว่าง Class เหล่านั้น ซึ่งในแต่ละคลาสจะมีการจัดเก็บข้อมูลและมีวิธีในการจัดการกับข้อมูลที่จัดเก็บ โดยคลาสจะเป็นรูปสี่เหลี่ยมที่ถูกแบ่งออกเป็นสามส่วน โดยชื่อคลาสจะอยู่ส่วนบนสุด และแอตทริบิวต์จะอยู่ตรงกลาง และโอเปอเรชันจะอยู่ล่างสุด ดังแสดงในภาพที่ 2-5



ภาพที่ 2-5 แสดงสัญลักษณ์ของ Class Diagram
(ที่มา: สุนทริน วงศ์ศิริกุล, 2550)

1) แอตทริบิวต์ (Attribute) คือข้อมูลที่เป็นคุณสมบัติของคลาส คือข้อมูลที่จะจัดเก็บและนำมาใช้ในระบบ ซึ่งสามารถกำหนดระดับของการเข้าถึงข้อมูลได้ โดยการใส่เครื่องหมายดังต่อไปนี้ไว้ข้างหน้าของแอตทริบิวต์

2) สาธารณะ (Public) หมายถึงการอนุญาตให้คลาสอื่น ๆ สามารถมองเห็นและใช้งานข้อมูลที่อยู่ในแอตทริบิวต์นี้ได้ ลักษณะที่ใช้แทนคือ (+)

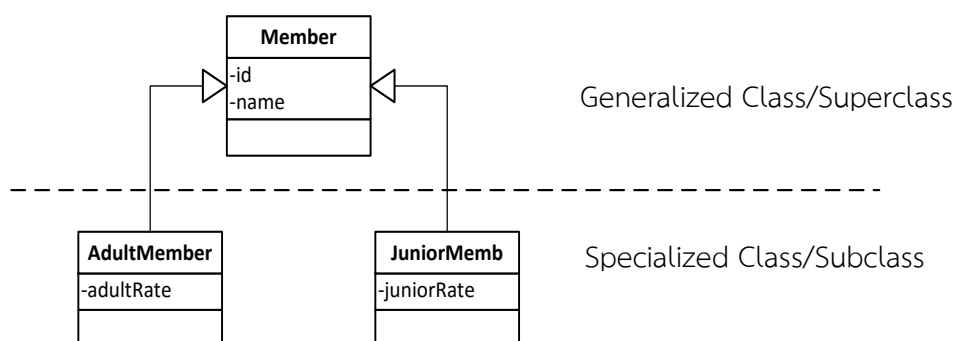
3) ปกป้อง (Protected) หมายถึงการอนุญาตให้คลาสอื่น ๆ สามารถมองเห็นแอตทริบิวต์นี้ได้แต่ไม่อนุญาตให้ใช้งานแอตทริบิวต์นี้ได้ ลักษณะที่ใช้แทนคือ (#)

4) ซ่อนไว้ (Hidden) หมายถึงคลาสอื่น ๆ ไม่สามารถที่จะมองเห็นและใช้งานแอตทริบิวต์นี้ได้ ลักษณะที่ใช้แทนคือ (-)

โอเปอเรชัน (Operation) คือหน้าที่การทำงานที่คลาสสามารถกระทำได้ โดยโอเปอเรชันจะตามด้วยเครื่องหมาย () ต่อท้ายแต่ละโอเปอเรชัน ซึ่งหมายถึงการระบุพารามิเตอร์ (Parameter) ที่จะใช้ส่งผ่านกันระหว่างโอเปอเรชันไว้ใน () บางโอเปอเรชันอาจไม่มีพารามิเตอร์ที่ต้องส่งแต่ก็ต้องใส่เครื่องหมายวงเล็บ () ไว้เช่นกัน โดยปล่อยให้ค่าภายใน () นั้นเป็นค่าว่างไว้ ซึ่งโอเปอเรชันจะมีสถานะเป็นสาธารณะ (Public) โดยการระบุเครื่องหมายบวก + ไว้ที่ข้างหน้าของแต่ละโอเปอเรชัน

ความสัมพันธ์ระหว่างคลาส (Class Relationship) เป็นการประสานการทำงานของแต่ละคลาสเข้าด้วยกัน ซึ่งรูปแบบความสัมพันธ์ระหว่างคลาส (Class relationship) ของคลาสไดอะแกรมสามารถแบ่งออกได้เป็นรูปแบบต่าง ๆ ดังนี้

1. Generalization หรือ Inheritance เป็นความสัมพันธ์ระหว่างคลาสในลักษณะของการจำแนกชนิด การจำเพาะเจาะจงรายละเอียด หรือการหาลักษณะร่วมกันของคลาสที่ต่างชนิดกัน เพื่อสร้างคลาสที่เป็นตัวแทนของกลุ่มคลาสเหล่านั้น ดังภาพที่ 2-6



ภาพที่ 2-6 แสดงความสัมพันธ์แบบ Generalization/Specialization

(ที่มา: กิตติ ภัคดีวัฒนกุล, 2551)

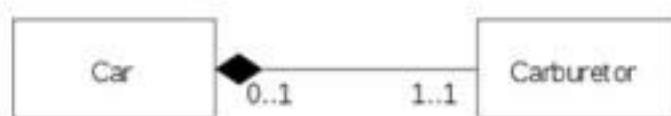
2. การเป็นส่วนหนึ่งของ (Aggregation) เป็นความสัมพันธ์แบบต่างระดับในลักษณะของการเป็นองค์ประกอบโดยคลาสที่เป็นองค์ประกอบเรียกว่า Part Class คลาสที่เกิดจากการรวมกันขององค์ประกอบต่าง ๆ เรียกว่า Whole Class ข้อสังเกตของความสัมพันธ์ชนิดนี้คือ Part Class มีอิสระไม่จำเป็นต้องพึ่งพา Whole Class และเมื่อ Whole Class ถูกทำลายหรือเสียหาย Part Class ก็ยังคงอยู่แสดงสัญลักษณ์ดังภาพที่ 2-7



ภาพที่ 2-7 แสดงสัญลักษณ์การเป็นส่วนหนึ่งของ (Aggregation)

(ที่มา : <http://projectumobjectdiagram.blogspot.com/p/class-diagram-2-class-relationship.html>)

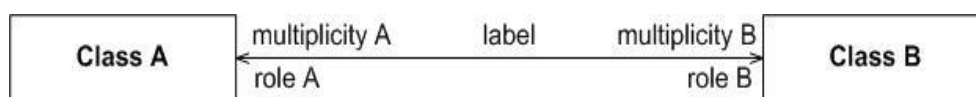
3. การเป็นองค์ประกอบ (Composition) ความสัมพันธ์นี้จะมีลักษณะคล้าย ๆ กับความสัมพันธ์แบบเป็นส่วนหนึ่งของ คือคลาสย่อยเป็นส่วนหนึ่งของคลาสหลัก แต่ต่างกันที่ส่วนประกอบย่อยนี้จะไม่สามารถนำออกจากส่วนประกอบหลักได้ เมื่อนำส่วนประกอบย่อยออกไปส่วนประกอบหลักก็จะสูญเสียคุณสมบัติของความเป็นคลาสนั้น ๆ จัดเป็นองค์ประกอบที่ไม่อาจขาดได้ดังภาพที่ 2-8



ภาพที่ 2-8 แสดงสัญลักษณ์การเป็นองค์ประกอบ (Composition)

(ที่มา : <http://projectumobjectdiagram.blogspot.com/p/class-diagram-2-class-relationship.html>.)

4. ความเกี่ยวข้องกัน (Association) เป็นความสัมพันธ์ระหว่างคลาสในระดับเดียวกัน คือ ไม่มีคลาสใดสำคัญมากกว่าคลาสใดจัดการความสัมพันธ์ขั้นต้น ใช้เส้นตรงเชื่อมระหว่างคลาส พร้อมกับชื่อความสัมพันธ์หรืออาจจะใช้เส้นตรงมีหัวลูกศรแบบก้างปลา เรียกว่าความสัมพันธ์แบบบอกทิศทาง (Navigation) โดยที่ชื่อความสัมพันธ์จะมีเครื่องหมายแสดงความชัดเจน Visibility ของคลาสด้วยดังภาพที่ 2-9



ภาพที่ 2-9 แสดงสัญลักษณ์การเป็นองค์ประกอบ (Association)

(ที่มา : <http://agilemodeling.com/artifacts/classDiagram.htm#Associations>.)

ซีควเอน ไดอะแกรม (Sequence Diagram) เป็นแผนภาพแสดงลำดับการโต้ตอบระหว่างอ็อบเจกต์ เพื่อตอบสนองต่อการสั่งงานจากผู้ใช้ระบบ โดยมีอ็อบเจกต์และเวลาเป็นตัวกำหนดลำดับของงาน และเน้นไปที่อินแทน (Instant) ของ Object Sequence Diagram เป็นไดอะแกรมซึ่งแสดงปฏิสัมพันธ์ร่วมกัน ระหว่างอ็อบเจกต์ตามลำดับของเหตุการณ์ที่เกิดขึ้น ณ เวลาที่กำหนดเมสเสจ (Message) ที่เกิดขึ้นระหว่าง คลาสจะสามารถนำไปสู่การสร้าง Method ในคลาสที่เกี่ยวข้องได้ Sequence Diagram จะประกอบด้วยสัญลักษณ์ต่างๆ ดังที่แสดงในภาพที่ 2-10

สัญลักษณ์	ชื่อ	ความหมาย
	Actor	ผู้ที่เกี่ยวข้องกับระบบ
	Object	ออบเจกต์ที่ต้องทำหน้าที่ตอบสนองต่อ Actor
	Lifeline	เส้นแสดงชีวิตของออบเจกต์หรือคลาส
	Focus of Control / Activation	จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่
	Message	คำสั่งหรือฟังก์ชันที่ออบเจกต์หนึ่งส่งให้อีกออบเจกต์หนึ่ง ซึ่งสามารถส่งกลับได้
	Callback / Self Delegation	การประมวลผลและคืนค่าที่ได้ภายในออบเจกต์เดียวกัน

ภาพที่ 2-10 แสดงสัญลักษณ์ของ Sequence Diagram

(ที่มา: <http://sequencediagram.circlecamp.com/?page=Mean&language=th>)

ผู้พัฒนาระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ ได้นำทฤษฎีที่เกี่ยวข้องกับการวิเคราะห์และออกแบบเชิงวัตถุด้วย UML มาประยุกต์ใช้ในการวิเคราะห์และออกแบบระบบ ในการเขียนแผนภาพแบบจำลอง Use Case Diagram , Class Diagram และ Sequence Diagram แสดงขั้นตอนการทำงานของระบบงาน

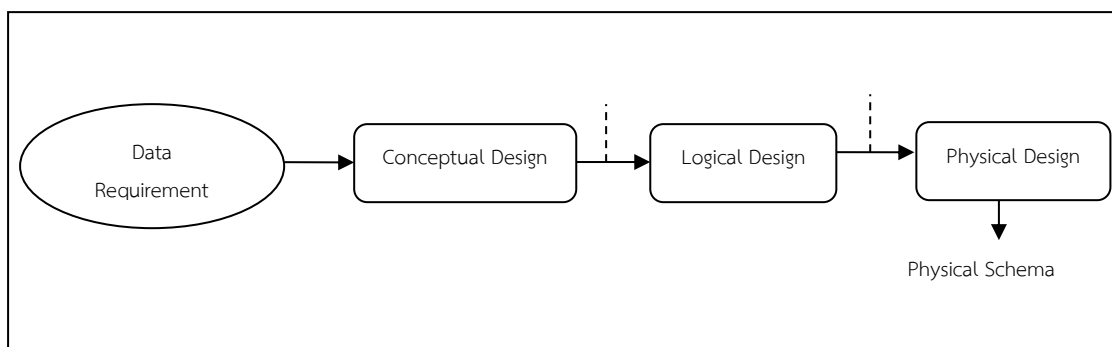
2.4 ทฤษฎีที่เกี่ยวข้องกับการออกแบบฐานข้อมูล

2.4.1 การออกแบบฐานข้อมูล

ฐานข้อมูลมีความสำคัญอย่างมากต่อระบบสารสนเทศที่สนับสนุนการทำงานของหน่วยงานหรือองค์กรต่าง ๆ ในการจัดเก็บข้อมูล ให้ปลอดภัยและมีความน่าเชื่อถือซึ่งสามารถช่วยให้หน่วยหรือผู้ใช้สามารถนำข้อไปใช้ในการวางแผนการการตัดสินใจได้สะดวก การออกแบบฐานข้อมูลที่ดี ควรจะมีคุณสมบัติที่เข้าใจง่าย อธิบายได้ชัดเจน และเชื่อถือได้ตลอดจนสามารถรองรับการขยายตัวของข้อมูลในระบบได้

การออกแบบฐานข้อมูล เป็นการกำหนดโครงสร้างของฐานข้อมูล ก่อนที่จะนำโครงสร้างที่ได้ไปพัฒนาให้เป็นฐานข้อมูลที่จัดเก็บอยู่จริงบนหน่วยความจำสำรอง โดยการออกแบบฐานข้อมูลมีหลายแนวทาง

ในที่นี้ผู้พัฒนาขอกล่าวถึง Data-driven Approach ซึ่งเป็นแนวทางที่ให้ความสำคัญกับตัวข้อมูลมากกว่าโปรแกรม คือ จะออกแบบตัวข้อมูลจนมีความสมบูรณ์ ก่อนที่จะออกแบบโปรแกรมเป็นลำดับต่อไป ได้แบ่งการออกแบบฐานข้อมูลออกเป็น 3 ระดับ ดังนี้คือ ระดับแนวคิด ระดับตรรกะ และระดับกายภาพ ดังภาพที่ 2-11



ภาพที่ 2-11 แสดงขั้นตอนการออกแบบฐานข้อมูล

(ที่มา : กิตติ ภัคดีวัฒนกุล, 2551)

1) การออกแบบฐานข้อมูลระดับแนวคิด (Conceptual Database Design) เป็นขั้นตอนในการสร้างแบบจำลองข้อมูลในระดับแนวคิด เพื่อแสดงให้เห็นเพียงข้อมูลที่เกิดขึ้นในระบบว่าประกอบไปด้วย แอททริบิวต์ (Attribute) อะไร และข้อมูลเหล่านี้มีความสัมพันธ์กันอย่างไร

2) การออกแบบฐานข้อมูลระดับตรรกะ (Logical Database Design) เป็นการนำ

โครงร่างที่ได้จากการออกแบบระดับแนวคิด มาแปลงให้เป็นโครงร่างระดับตรรกะ ตามชนิดของฐานข้อมูลที่ใช้ ซึ่งผู้พัฒนาได้เลือกใช้ฐานข้อมูลเชิงสัมพันธ์ ดังนั้นโครงร่างระดับตรรกะที่ได้จากขั้นตอนนี้จึงเป็น “Relation” แต่ยังไม่สามารถนำไปใช้ในการออกแบบฐานข้อมูลในระดับกายภาพได้ เนื่องจากโครงร่างอาจยังมีความซ้ำซ้อนของข้อมูลอยู่ ดังนั้นจึงต้องมีการปรับปรุงด้วยกระบวนการ “Normalizaton” เพื่อให้ได้โครงร่างที่เหมาะสมต่อการจัดเก็บข้อมูล

3) การออกแบบฐานข้อมูลฐานระดับกายภาพ (Physical Database Design) เป็นการนำความสัมพันธ์ ที่ถูกแปลงมาจาก อีอาร์ ไดอะแกรม (E-R Diagram) มาแปลงให้อยู่ในรูปของตารางของฐานข้อมูล พร้อมทั้งกำหนดโครงสร้างทางกายภาพให้กับฐานข้อมูล ได้แก่ กำหนดชนิดของข้อมูล ขนาดและขอบเขตของ Attribute รวมถึงประเภทของคีย์ (Key) ของข้อมูล ตลอดจนการกำหนดวิธีการความปลอดภัยให้กับฐานข้อมูลด้วย

ผู้พัฒนาได้นำทฤษฎีที่เกี่ยวข้องกับการออกแบบฐานข้อมูลมาประยุกต์ใช้ในการออกแบบและจัดการฐานข้อมูลของระบบงาน เพื่อให้การจัดเก็บฐานข้อมูลเป็นไปอย่างถูกต้องตรงตามความเป็นจริง

2.4.2 ฐานข้อมูลเชิงสัมพันธ์

ดร.ศิริลักษณ์ โรจนกิจอำวย. (2552 : 13 – 27) ได้กล่าวไว้ว่า ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นรูปแบบฐานข้อมูลที่เข้าใจง่ายสำหรับผู้ใช้งาน ไม่ซ้ำซ้อนและยังรวมถึงรูปแบบฐานข้อมูลที่มีระบบจัดการฐานข้อมูล (Database Management Systems: DBMS) และยังช่วยการรักษาความอิสระของข้อมูล (Data Independence) และความเป็นอิสระของโครงสร้างของข้อมูลในแต่ละระดับ (Structural Independence) อีกทั้งยังมีคุณลักษณะที่ช่วยลดความซ้ำซ้อน ตลอดจนปัญหาที่เกิดจากการปรับปรุงหรือการเพิ่มลบข้อมูลด้วยการนำแนวคิดการทำให้เป็นบรรทัดฐาน (Normalization) มาใช้ในการออกแบบ

การทำให้เป็นบรรทัดฐาน เป็นกระบวนการที่ใช้ในการทดสอบการออกแบบบรีเลชันตามเกณฑ์ของขั้นตอนต่างๆ เป็นการวิเคราะห์การออกแบบในลักษณะ Bottom – Up ซึ่งเป็นการพิจารณาว่า คีย์หลักหรือคีย์คู่แข่งสามารถระบุค่าของแอททริบิวต์อื่นๆ ของทูเพิลหนึ่งในบรีเลชันได้ ซึ่งจะช่วยลดความซ้ำซ้อนในฐานข้อมูล

โครงสร้างข้อมูลของฐานข้อมูล จำเป็นที่จะต้องใช้แบบจำลองข้อมูลในการนำเสนอ ซึ่งเรียกว่า แบบจำลองฐานข้อมูล (Database Model)

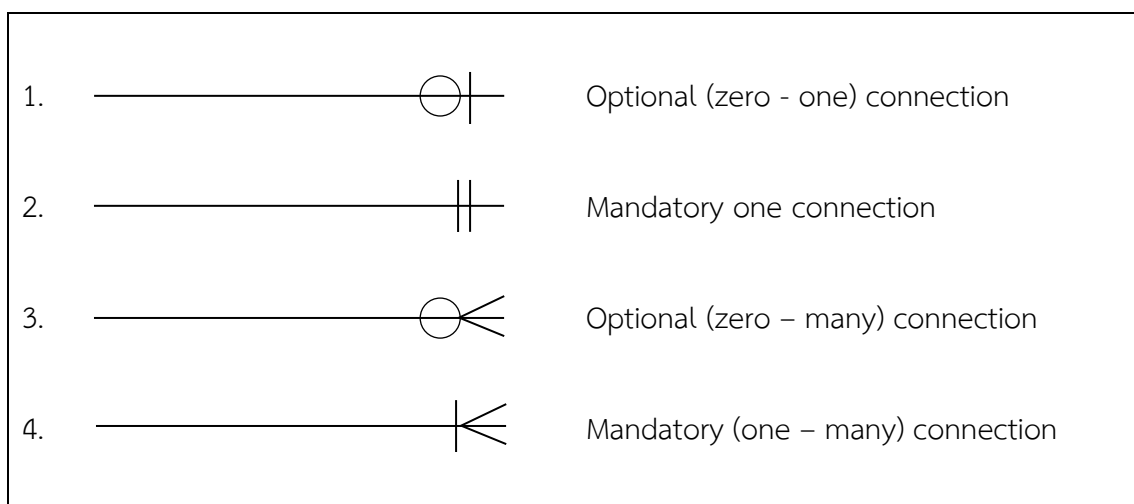
แบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Relational Database Model) เป็นแบบจำลองที่นำเสนอโครงสร้างของข้อมูลที่อยู่ในฐานข้อมูลเชิงสัมพันธ์ โดยนำเสนอในรูปแบบของตาราง หรือ Relation

ตาราง (Relation) เป็นตาราง 2 มิติที่ใช้ในการเก็บข้อมูล โดยแต่ละตารางจะประกอบด้วยชุดแถวเรียกว่า “Tuple” และคอลัมน์โดยที่ คอลัมน์จะใช้แสดง Attribute ส่วนแถวจะแทนค่าข้อมูล

ของ Attribute โดยข้อมูลที่จัดเก็บอยู่ในแต่ละความสัมพันธ์จะเป็นข้อมูลที่แยกเป็นเอกเทศ แต่สามารถนำมาสร้างความสัมพันธ์ร่วมกันได้ คุณลักษณะของข้อมูลที่จัดเก็บของรีเลชันมีดังนี้ คือ

- 1) ข้อมูลในแต่ละแถวจะไม่ซ้ำกัน
- 2) การเรียงลำดับของข้อมูลในแต่ละแถวไม่เป็นสาระสำคัญ
- 3) การเรียงลำดับของแอททริบิวต์จะเรียงลำดับก่อนหลังอย่างไรก็ได้
- 4) ค่าของข้อมูลในแต่ละแอททริบิวต์ของทุเพิลหนึ่ง ๆ จะเก็บข้อมูลได้เพียงค่าเดียว
- 5) ค่าของข้อมูลในแต่ละแอททริบิวต์จะเก็บค่าของข้อมูลประเภทเดียวกัน

ฐานข้อมูลเชิงสัมพันธ์ทุกรีเลชันจะต้องมีแอททริบิวต์ที่เป็นคีย์หลัก ซึ่งมีคุณสมบัติเป็นค่าไม่ซ้ำ และต้องไม่มีค่าว่าง และการเชื่อมโยงระหว่างรีเลชันจะใช้แอททริบิวต์เป็นคีย์นอกเชื่อมโยงกับแอททริบิวต์ที่เป็นคีย์หลักของอีกตารางหนึ่งเพื่อให้สามารถเรียกข้อมูลที่ผู้ใช้งานต้องการได้และยังเน้นที่ความสัมพันธ์ของเอนทิตี ซึ่งสัญลักษณ์ความสัมพันธ์ของเอนทิตีที่มีดังภาพที่ 2-12



ภาพที่ 2-12 แสดงสัญลักษณ์ที่ใช้ในการเชื่อมโยงความสัมพันธ์ระหว่างเอนทิตี

(ที่มา : สุนทริน วงศ์ศิริกุล, 2550)

รายละเอียดสัญลักษณ์ที่ใช้ในการเชื่อมโยงความสัมพันธ์ระหว่างเอนทิตีมีดังนี้

1. การเชื่อมแบบมีหนึ่งทางเลือก (Optional Connection:Zero-One) คือวัตถุ A เชื่อมโยงไปสู่วัตถุ B ได้เพียงหนึ่งรายการหรือไม่เชื่อมโยง
2. การเชื่อมแบบบังคับเพียงหนึ่ง (Mandatory One Connection) คือวัตถุ A จะต้องทำการเชื่อมโยงไปสู่วัตถุ B เพียงหนึ่งความสัมพันธ์เท่านั้น
3. การเชื่อมแบบมีหลายทางเลือก (Optional Connection:Zero-Many) คือวัตถุ A อาจจะเชื่อมโยงไปสู่วัตถุ B ได้หลายรายการหรือไม่เชื่อมโยงเลย

4. การเชื่อมแบบบังคับหลายรายการ (Mandatory Many Connection) คือวัตถุ A ต้องเชื่อมโยงไปสู่วัตถุ B หนึ่งรายการหรือมากกว่าหนึ่งรายการ

ผู้พัฒนาระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพได้นำทฤษฎีที่เกี่ยวข้องฐานข้อมูลเชิงสัมพันธ์มาประยุกต์ในการจัดการฐานข้อมูล เพื่อลดความซ้ำซ้อนของข้อมูลและการจัดการข้อมูลให้มีความถูกต้อง

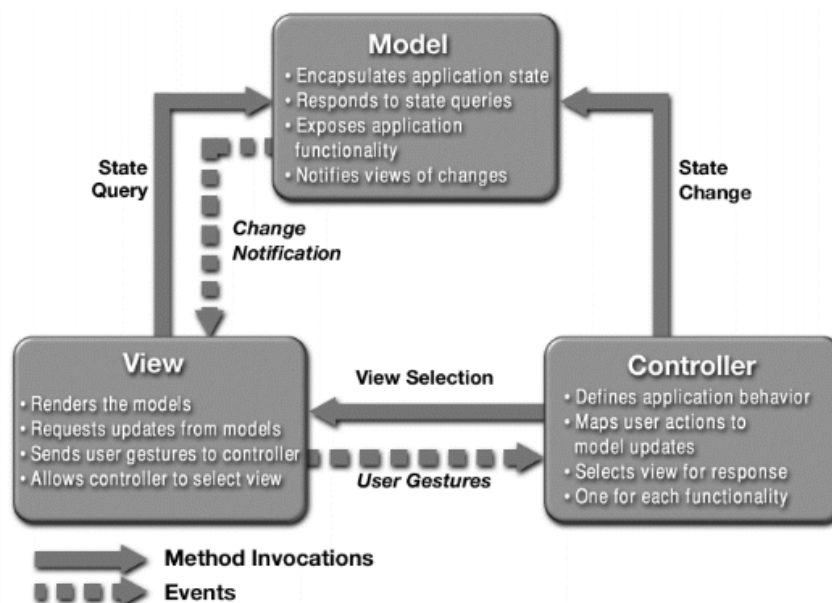
2.5 ทฤษฎีที่เกี่ยวข้องกับเครื่องมือที่ใช้ในการพัฒนาระบบ

2.5.1 ทฤษฎีแนวคิด เอ็มวีซี (MVC: Model View and Controller)

เอ็มวีซีได้มีการคิดค้นและพัฒนาร่วมกับภาษาสมอทอก-80 (Smalltalk-80) โดยการแยกส่วน อ็อบเจกต์ที่เก็บข้อมูล ส่วนอ็อบเจกต์ที่แสดงผลข้อมูลและส่วนอ็อบเจกต์ที่ติดต่อกับผู้ใช้ออกจากกันอย่างชัดเจน โดยการสื่อสารกันระหว่างอ็อบเจกต์แต่ละส่วนจะใช้รูปแบบอ็อบเซิร์ฟเวอร์แพตเทิร์น (Observer Pattern) ซึ่งการแจ้งเหตุการณ์หรือข้อมูลจากคลาสต้นกำเนิดไปยังคลาสที่รับข้อมูลนั้นซึ่งการแยกอ็อบเจกต์ออกจากกัน จะทำให้ง่ายต่อการแก้ไขโปรแกรม สามารถเปลี่ยนแปลงส่วนที่จะแสดงผลจากเอชทีเอ็มแอลเป็นดับเบิลยูเอ็มแอล (WML: Wireless Markup Language) หรือสามารถเปลี่ยนรูปแบบการแสดงผลแบบอื่นได้โดยไม่ต้องทำการแก้ส่วนที่เก็บข้อมูล เนื่องจากเอ็มวีซีสามารถสนับสนุนหลาย ๆ รูปแบบการแสดงผลพร้อม ๆ กันซึ่งมีผู้ดัดแปลงเอ็มวีซีไปใช้หลายรูปแบบและเรียกชื่อที่ต่างกันไป แต่หลักการก็ยังคงคล้าย ๆ เดิม เพียงแต่แตกต่างที่การติดต่อกันระหว่างส่วนการทำงานทั้ง 3 ส่วนว่าส่วนไหนเป็นส่วนที่จะแจ้งหรือตอบรับการเปลี่ยนแปลงเหตุการณ์ (Event notification) เอ็มวีซีจะทำการจัดแบ่งส่วนการทำงานออกเป็นกลุ่ม ๆ ตามลักษณะงานที่ต้องทำโดยแบ่งเป็น 3 กลุ่มดังนี้

- 1) แบบจำลอง (Model) คืออ็อบเจกต์ที่ทำหน้าที่เป็นตัวแทนของข้อมูลหรือติดต่อกับฐานข้อมูลไม่ว่าข้อมูลจะถูกจัดเก็บในรูปแบบใดในระบบฐานข้อมูลหรือในไฟล์ เมื่อข้อมูลนั้นถูกโหลดเข้ามาในแอปพลิเคชัน สามารถเปลี่ยนมันให้อยู่ในรูปแบบของอ็อบเจกต์
- 2) มุมมอง (View) เป็นส่วนการสร้างรูปแบบจากแบบจำลองมาแปลงเป็นข้อมูลให้ผู้ใช้ระบบสามารถเข้าใจหรือการนำข้อมูลที่ได้จากแบบจำลองมาแสดงผล
- 3) ตัวควบคุม (Controller) เป็นส่วนที่ทำหน้าที่จัดการคำขอและทำการเลือกข้อมูลจากแบบจำลองและเลือกหน้ามุมมองที่จะใช้แสดงผลให้แก่ผู้ใช้ เมื่อผู้ใช้ตัดสินใจส่งคำขอมายังตัวควบคุมตัวควบคุมจะทำการเลือกข้อมูลหรือปรับปรุงแบบจำลองตามคำขอและจะทำการเลือกมุมมองเพื่อส่งกลับไปให้กับผู้ใช้ ซึ่งตัวควบคุมนี้จะช่วยให้ง่ายต่อการพัฒนาเนื่องจากผู้พัฒนาสามารถเรียกใช้ฟังก์ชันที่มีอยู่หรือที่มีการเขียนไว้ก่อนหน้านี้

เอ็มวีซีเป็นรูปแบบการออกแบบที่แบ่งส่วนของการประมวลผลระหว่างข้อมูลออกจากการแสดงผลเพื่อลดความซ้ำซ้อนในการพัฒนาและง่ายต่อการดูแลรักษาและลดความซ้ำซ้อนในส่วนการอ่านข้อมูล เนื่องจากข้อมูลถูกสร้างแบบจำลอง ซึ่งสามารถส่งไปยังมุมมองต่าง ๆ ได้ลักษณะการทำงานของเอ็มวีซีดังภาพที่ 2-13



ภาพที่ 2-13 สถาปัตยกรรมเอ็มวีซี

(ที่มา <http://www.pisit.in.th/php/mvc>)

ผู้พัฒนาระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ ได้นำทฤษฎีแนวคิด เอ็มวีซี (MVC: Model View and Controller) มาประยุกต์ใช้ในการเขียนโปรแกรม เพื่อลดความซ้ำซ้อนในการพัฒนาและง่ายต่อการดูแลรักษาและลดความซ้ำซ้อนในส่วนการอ่านข้อมูล

2.5.2 พีเอชพี (Hypertext Preprocessor : PHP)

ฉันทพัฒน์ วงศ์รัตน์. (2556 : 1-3) ได้กล่าวไว้ว่า ภาษาพีเอชพี เป็นภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ เป็นภาษาที่ใช้ในการพัฒนาเว็บไซต์ ซึ่งสามารถสร้างระบบงานหรือเว็บแอปพลิเคชันได้มากมาย ภาษาพีเอชพีสามารถทำงานร่วมกับโค้ดภาษาเอชทีเอ็มแอล

2.5.3 จาวาสคริปต์ (JavaScript)

ฉันทพัฒน์ วงศ์รัตน์. (2556 : 231-233) ได้กล่าวไว้ว่า จาวาสคริปต์ (JavaScript) เป็นภาษาโปรแกรมประเภทหนึ่งซึ่งเรียกว่า “สคริปต์ (Script)” มีลักษณะการทำงานแบบ แปลความและ

ดำเนินงานไปที่ละคำสั่ง ซึ่ง จาวาสคริปต์ ได้ถูกพัฒนาขึ้นมาเพื่อช่วยให้เว็บเพจสามารถแสดงเนื้อหาที่มีการเปลี่ยนแปลง เคลื่อนไหว หรือสามารถโต้ตอบกับผู้ใช้ได้

จาวาสคริปต์ (JavaScript) เป็นภาษายุคใหม่สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ตที่กำลังได้รับความนิยมอย่างสูง เพื่อใช้ประโยชน์สำหรับงานด้านต่างๆ ทั้งการคำนวณ การแสดงผล การรับ-ส่งข้อมูล และสามารถโต้ตอบกับผู้ใช้ได้อย่างทันทีทันใด นอกจากนี้ยังมีความสามารถด้านอื่นๆ อีกหลายประการที่ช่วยสร้างความน่าสนใจให้กับเว็บเพจของเราได้อย่างมาก ภาษาจาวาสคริปต์ถูกพัฒนาโดยเน็ตสเคปคอมมูนิตีส์ โดยใช้ชื่อว่าไลฟ์สคริปต์ (Live Script) ออกมาพร้อมกับเน็ตสเคป เนวิเกเตอร์ 2.0 (Netscape Navigator 2.0) เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบไลฟ์ไวร์ (Live Wire) ต่อมาเน็ตสเคปจึงได้ร่วมมือกับบริษัทซันไมโครซิสเต็มส์ปรับปรุงระบบของเบราร์เซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุงไลฟ์สคริปต์ใหม่เมื่อ ปี พ.ศ. 2538 แล้วตั้งชื่อใหม่ว่า จาวาสคริปต์ (JavaScript)

2.5.4 ภาษา SQL (Standard Query Language)

ภาษาเอสคิวแอล เป็นภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ ซึ่งเป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) ซึ่งสามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ และยังใช้คำสั่งเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกัน แต่จะได้ผลลัพธ์เหมือนกัน ซึ่งเราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ยึดติดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้ภาษาเอสคิวแอลยังมีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ ซึ่งแบ่งการทำงานได้เป็น 4 ประเภท ดังนี้

- 1) คำสั่งที่ใช้สำหรับดึงข้อมูลที่ต้องการ (Select Query)
- 2) คำสั่งที่ใช้สำหรับแก้ไขข้อมูล (Update Query)
- 3) คำสั่งที่ใช้สำหรับเพิ่มข้อมูล (Insert Query)
- 4) คำสั่งที่ใช้สำหรับลบข้อมูล (Delete Query)

ประเภทของคำสั่งภาษา SQL มีดังนี้

1) ภาษานิยามข้อมูล (Data Definition Language : DDL) เป็นคำสั่งที่ใช้ในการสร้างฐานข้อมูล กำหนดโครงสร้างข้อมูลว่ามี (Attribute) ใดบ้าง ชนิดของข้อมูล รวมทั้งการเปลี่ยนแปลงตาราง และการสร้างดัชนี คำสั่ง : CREATE , DROP , ALTER

2) ภาษาจัดการข้อมูล (Data Manipulation Language :DML) เป็นคำสั่งที่ใช้ในการเรียกใช้ เพิ่ม ลบ และเปลี่ยนแปลงข้อมูลในตาราง คำสั่ง : SELECT , INSERT , UPDATE , DELETE

3) ภาษาควบคุมข้อมูล (Data Control Language : DCL) เป็นคำสั่งที่ใช้ในการกำหนดสิทธิการอนุญาต หรือ ยกเลิก การเข้าถึงฐานข้อมูล เพื่อป้องกันความปลอดภัยของฐานข้อมูล คำสั่ง : GRANT, REVOKE

ผู้พัฒนาระบบจัดการข้อมูลการฝึกประสบการณ์วิชาชีพ ได้นำทฤษฎีทางด้านภาษาหรือเครื่องมือไม่ว่าจะเป็นพีเอชพี จาวาสคริปต์ และ ภาษา SQL มาประยุกต์ใช้ในการพัฒนาระบบงานเพื่อให้ได้ระบบงานที่ดีและมีประสิทธิภาพ

2.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.6.1 ระบบนักศึกษาฝึกงานวิทยาลัยเทคนิคบ้านค่าย

ระบบนักศึกษาฝึกงานวิทยาลัยเทคนิคบ้านค่าย พัฒนาขึ้นจะอำนวยความสะดวกให้แก่เจ้าหน้าที่ นักศึกษาอาจารย์ที่ปรึกษา และสถานประกอบการ นักศึกษาสามารถสืบค้นสถานประกอบการที่นักศึกษาต้องการเข้าฝึกงานตามสาขาที่ต้องการ โดยการวิจัยนี้ได้มีการออกแบบระบบโดยใช้ทั้ง (Data Flow Diagram) และ (ER-Diagram) จากนั้นจึงนำความสัมพันธ์ของข้อมูลต่างๆ มาออกแบบตัวอย่างข้อมูลและทำต้นแบบ หน้าจอโปรแกรม นอกจากนี้ระบบดังกล่าวยังช่วยให้การทำงานของเจ้าหน้าที่ที่มีความถูกต้องแม่นยำในด้านการจัดการข้อมูลและช่วยลดขั้นตอนและความซ้ำซ้อนในการดำเนินงานที่เกี่ยวข้องกับการฝึกงานในสถานประกอบการทั้งหมด

2.6.2 ระบบการจัดการการฝึกงาน

ระบบการจัดการการฝึกงานพัฒนาขึ้นโดยการนำเทคโนโลยีสารสนเทศมาประยุกต์ใช้งานร่วมกับเทคโนโลยีเครือข่าย ภายใต้การทำงานของเว็บแอปพลิเคชัน ASP.Net ซึ่งทำหน้าที่จัดการทั้งระบบ ทั้งนี้ระบบที่พัฒนาขึ้นนี้จะอำนวยความสะดวกให้แก่เจ้าหน้าที่ นักศึกษา อาจารย์ที่ปรึกษา และสถานประกอบการ โดยนักศึกษาสามารถสืบค้นสถานประกอบการที่เคยมีการฝึกงานของนักศึกษามหาวิทยาลัยเทคโนโลยีมหานครและสถานประกอบการที่ต้องการรับนักศึกษาเข้าฝึกงานตามสาขาวิชาที่ต้องการ รวมไปถึงการยื่นเรื่องฝึกงานและติดตามสถานะ การดำเนินงานผ่านทางเว็บไซต์ของระบบได้ นอกจากนี้ระบบดังกล่าวยังช่วยให้การทำงานของเจ้าหน้าที่ที่มีความถูกต้องแม่นยำในด้านการจัดการข้อมูลและช่วยลดขั้นตอนและความซ้ำซ้อนในการดำเนินงานที่เกี่ยวข้องกับการฝึกงานทั้งหมด

2.7 สถิติที่ใช้ในการประเมินผล

สถิติที่ใช้กันอยู่ในทางวิจัย แบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือ สถิติเชิงบรรยายหรือสถิติเชิงพรรณนา (Descriptive Statistics) และสถิติเชิงอ้างอิงหรือสถิติอนุมาน (Inferential Statistics)

สถิติใช้ในการอธิบายคุณลักษณะ หรือรายละเอียดของกลุ่มที่ศึกษา ได้แก่

1) ร้อยละ (Percentage) เป็นค่าสถิติที่นิยมใช้กันมากโดยเป็นการเปรียบเทียบความถี่หรือจำนวนที่ต้องการกับความถี่หรือจำนวนทั้งหมดที่เทียบเป็น 100

2) การวัดแนวโน้มเข้าสู่ส่วนกลาง (Measures of Central Tendency) เป็นสถิติที่ใช้เป็นตัวแทนของข้อมูล โดยที่นิยมใช้มี 3 ประเภท ได้แก่ ค่าเฉลี่ย มัธยฐาน ฐานนิยม

3) การวัดการกระจาย (Measures of Variability) เป็นสถิติที่ช่วยให้ทราบถึงความแตกต่างหรือการแปรผันของคะแนนในชุดนั้นหรือกลุ่มนั้นถ้าค่าที่ได้มามีค่าสูง หมายถึงคะแนนมีความแตกต่างกันมาก ถ้าค่าที่ได้มีค่าต่ำหมายถึงคะแนนไม่แตกต่างกันมากนักหรือใกล้เคียงกัน

สถิติที่ใช้วิเคราะห์ข้อมูลในครั้งนี้ ใช้สถิติเชิงอ้างอิงหรือสถิติอนุมาน (Inferential Statistics) ในการวัดค่าร้อยละ และการวัดแนวโน้มเข้าสู่ส่วนกลางจะใช้การวัดค่าเฉลี่ยและฐานนิยม ดังนี้

2.7.1 ร้อยละ (Percentage)

$$\text{จากสูตร } p = \frac{f}{N} \times 100$$

เมื่อ p แทน ค่าร้อยละ

f แทน ความถี่ที่ต้องการเปลี่ยนแปลงให้เป็นค่าร้อยละ

N แทน จำนวนความถี่ทั้งหมด

2.7.2 ค่าเฉลี่ยเลขคณิต (Arithmetic Mean)

$$\bar{x} = \frac{\sum x}{N}$$

จากสูตร

เมื่อ \bar{x} แทน ค่าเฉลี่ย

$\sum x$ แทน ผลรวมทั้งหมดของข้อมูล

N แทน จำนวนข้อมูลทั้งหมด

2.7.3 ค่าเบี่ยงเบนมาตรฐาน (Standard Deviation)

$$\text{จากสูตร } s.d. = \sqrt{\frac{\sum (x - \bar{x})^2}{(N - 1)}}$$

เมื่อ $s.d.$ แทน ค่าเบี่ยงเบนมาตรฐาน

\bar{x} แทน ค่าเฉลี่ย

x แทน ข้อมูล

N แทน จำนวนข้อมูลทั้งหมด

2.7.4 เกณฑ์การยอมรับประสิทธิภาพของการทำงานของโปรแกรม

พิจารณาจากค่าเฉลี่ยเลขคณิตและค่าเบี่ยงเบนมาตรฐานของผู้ใช้และผู้เชี่ยวชาญที่ได้ทำการทดสอบโปรแกรม โดยต้องมีค่าเฉลี่ยเลขคณิตมากกว่า 2.6 ขึ้นไปจึงถือว่าโปรแกรมมีประสิทธิภาพ

ในระดับที่ใช้งานได้จริง ซึ่งช่วงคะแนนค่าเฉลี่ยเลขคณิตมีเกณฑ์การประเมินที่สามารถแบ่งออกเป็น 5 ระดับ ดังนี้

- ช่วงคะแนน 4.60 ถึง 5.00 จัดว่าอยู่ในระดับดีมาก
- ช่วงคะแนน 3.60 ถึง 4.59 จัดว่าอยู่ในระดับดี
- ช่วงคะแนน 2.60 ถึง 3.59 จัดว่าอยู่ในระดับพอใช้
- ช่วงคะแนน 1.60 ถึง 2.59 จัดว่าอยู่ในระดับปรับปรุง
- ช่วงคะแนน 1.00 ถึง 1.59 จัดว่าอยู่ในระดับไม่เหมาะสม