บทที่ 2

ทฤษฏีที่เกี่ยวข้อง

**ทฤษฏีที่เกี่ยวข้องจะประกอบไปด้วย**

**2.1 ทฤษฎีและโปรแกรมที่เกี่ยวข้อง**

2.1.1 ทฤษฎีเกี่ยวกับ PHP (Personal Home Page Tool)

2.1.2 ทฤษฎีที่เกี่ยวกับ phpMyAdmin

2.1.3 ทฤษฎีที่เกี่ยวกับ ภาษา HTML (Hyper Text Markup Language)

**2.2 ทฤษฎีที่เกี่ยวข้องกับการวิเคราะห์และออกแบบระบบเชิงวัตถุ**

2.2.1 วงจรการพัฒนาระบบ (System Development Life Cycle: SDLC)

2.2.2 โมเดล prototype model

**2.3 ทฤษฎีฐานข้อมูล**

2.3.1 ความหมายของระบบฐานข้อมูล

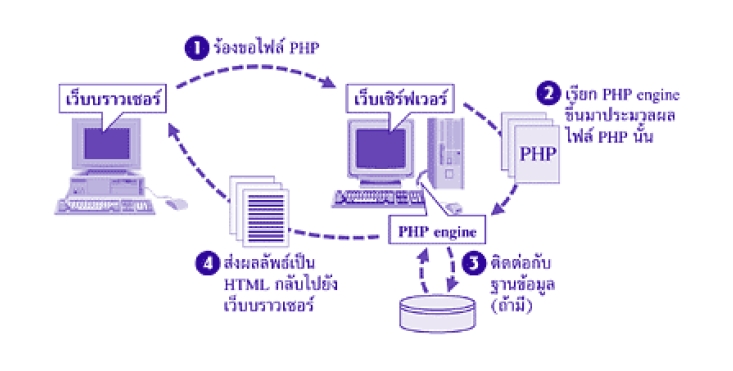
2.3.2 การจัดการฐานข้อมูล

2.3.3 ส่วนประกอบระบบการจัดการฐานข้อมูล

**2.1 ทฤษฎีและโปรแกรมที่เกี่ยวข้อง**

**2.1.1 ทฤษฎีเกี่ยวกับ PHP (Personal Home Page Tool)**

PHP ย่อมาจากคำ ว่า "Personal Home Page Tool" เป็น Server side script ที่มีการทำงานที่ฝั่งของเครื่องคอมพิวเตอร์ Server ซึ่งรูปแบบในการเขียนคำสั่งการทำงานนั้นจะมีลักษณะ คล้ายกับภาษา Perl หรือภาษา C และสามารถที่จะใช้ร่วมกับภาษา HTML ได้อย่างมีประสิทธิภาพ ซึ่งจะทำ ให้การเขียนโปรแกรมบนเว็บไซต์ทำได้ง่ายยิ่งขึ้น



ภาพที่ 2.1 แผนภาพ server-side script

จากรูปเป็นการทำงานของเว็บเพจ (เรียกว่า ไฟล์ PHP) เมื่อเว็บบราวเซอร์ร้องขอไฟล์ PHP ไฟล์ใด เว็บเซิร์ฟเวอร์จะเรียก PHP engine ขึ้นมาแปล (interpret) และประมวลผลคำสั่งที่อยู่ในไฟล์ PHP นั้น โดยอาจมีการดึงข้อมูลจากฐานข้อมูล หรือ เขียนข้อมูลลงไปยังฐานข้อมูลด้วย หลังจากนั้นผลลัพธ์ในรูปแบบ HTML (และสคริปต์ที่ทำงาน ทางฝั่งบราวเซอร์ เช่น client-side JavaScript) จะถูกส่งกลับไปยังบราวเซอร์ บราวเซอร์ก็จะ แสดงผลตามคำสั่ง HTML ที่ได้รับมาซึ่งย่อมไม่มีคำสั่ง PHP ใดๆ หลงเหลืออยู่ เนื่องจากถูกแปล และประมวลผลโดย PHP engine ที่ฝั่งเซิร์ฟเวอร์ไปหมดแล้ว ให้สังเกตว่าการทำงานของบราวเซอร์ ในกรณีนี้ไม่แตกต่างจากกรณีของเว็บเพจธรรมดาเพราะสิ่งที่บราวเซอร์ต้องกระทำก็คือการร้องขอไฟล์จากเว็บเซิร์ฟเวอร์จากนั้นก็รอรับผลลัพธ์กลับมาแล้วแสดงผล ความแตกต่างจริงๆอยู่ที่การทำงานทางฝั่งเซิร์ฟเวอร์ ซึ่งกรณีหลังนี้ เว็บเพจ (ไฟล์ PHP) จะผ่านการประมวลผลก่อน แทนที่จะถูกส่งไปยังบราวเซอร์เลยทันที การฝังสคริปต์ PHP ไว้ในเว็บเพจ ช่วยให้เราสร้างเว็บเพจแบบ dynamic ได้ ซึ่งหมายถึงเว็บเพจที่มีเนื้อหาสาระหรือหน้าตาเปลี่ยนแปลงไปได้ในแต่ละครั้งที่ผู้ใช้เปิดดู โดยขึ้นอยู่กับเงื่อนไขต่างๆ เช่น ข้อมูลที่ผู้ใช้ส่งมาให้ (ผ่านมาทางฟอร์มของ HTML) ข้อมูลในฐานข้อมูล ฯลฯ PHP เป็นภาษาจาพวก scripting language คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า สคริปต์ (script) และเวลาใช้งานต้องอาศัยตัวแปลชุดคำสั่ง ตัวอย่างของภาษาสคริป ก็ เช่น JavaScript, Perl เป็นต้น

ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ HTML โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้างเอกสารแบบ Dynamic HTML ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น

เนื่องจากว่า PHP ไม่ได้เป็นส่วนหนึ่งของตัว Web Server ดังนั้นถ้าจะใช้ PHP ก็จะต้องดูก่อนว่า Web server นั้นสามารถใช้สคริปต์ PHP ได้หรือไม่ ยกตัวอย่างเช่น PHP สามารถใช้ได้กับApache Web Server และPersonal Web Server (PWP) สำหรับระบบปฏิบัติการ Windows 95/98/NT ในกรณีของ Apache เราสามารถใช้ PHP ได้สองรูปแบบคือ ในลักษณะของ CGI และApache Module ความแตกต่างอยู่ตรงที่ว่า ถ้าใช้ PHP เป็นแบบโมดูล PHP จะเป็นส่วนหนึ่งของApache หรือเป็นส่วนขยายในการทำงานนั่นเอง ซึ่งจะทำงานได้เร็วกว่าแบบที่เป็น CGI เพราะว่า ถ้าเป็น CGI แล้ว ตัวแปลชุดคำสั่งของ PHP ถือว่าเป็นแค่โปรแกรมภายนอก ซึ่ง Apache จะต้องเรียกขึ้นมาทางานทุกครั้ง ที่ต้องการใช้ PHP ดังนั้น ถ้ามองในเรื่องของประสิทธิภาพในการทำงาน การใช้PHP แบบที่เป็นโมดูลหนึ่งของ Apache จะทำงานได้มีประสิทธิภาพมากกว่า

**2.1.2 ทฤษฎีที่เกี่ยวกับ phpMyAdmin**

PhpMyAdmin เป็นโปรแกรมประเภท mySQL Client ตัวหนึ่งที่ใช้ในการจัดการฐานข้อมูลmySQL ผ่านบราวเซอร์ได้โดยตรง โปรแกรมนี้เขียนขึ้นโดยนาย Tobias Retschiller และเป็นโปรแกรมแบบ Open Source ด้วย เพราะว่าเขียนด้วย PHP ทั้งหมด

PhpMyAdmin ตัวนี้จะทำงานบน Web Server เป็น PHP Application ที่ใช้ควบคุมจัดการกับ mySQL Server ความสามารถของ phpMyAdmin คือ

2.1.2.1 สร้างและลบ Database

2.1.2.2 สร้างและจัดการ Table เช่น แทรก record, ลบ record, แก้ไข record หรือลบ

2.1.2.3 Table, เพิ่มหรือแก้ไข field ในตาราง

2.1.2.4 โหลดเท็กซ์ไฟล์เข้าไปเก็บเป็นข้อมูลในตารางได้

หาผลสรุป (Query) ด้วยคำสั่ง SQL และอีกหลายๆ ความสามารถที่ phpMyAdmin ทำได้การเข้าใช้งานโปรแกรม PhpMyAdmin จะใช้งานผ่านโปรแกรม Internet Explorer และพิมพ์ URLไปยังไดเรกทอรีของ phpMyAdmin เช่น http://localhost/phpMyAdmin เป็นต้น

(ชื่อไดเรกทอรีอาจจะเป็นชื่ออื่น ขึ้นอยู่กับขั้นตอนการเปลี่ยนชื่อไดเรกทอรี) และต้องแน่ใจว่าได้ติดตั้งโปรแกรม Web Server เช่น Apache เป็นต้น ไว้ภายในเครื่องเป็นที่เรียบร้อยแล้ว

ก่อนที่จะเก็บข้อมูลลงใน MySQL จะต้องเลือกเสียก่อนว่า จะสร้างฐานข้อมูลชื่ออะไรจากนั้นจึงทำการสร้างตาราง และฟิลด์ เพื่อบันทึกข้อมูล

**2.1.3 ทฤษฎีที่เกี่ยวกับ ภาษา HTML (Hyper Text Markup Language)**

ภาษา HTML (Hyper Text Markup Language) เป็นภาษาหลักที่ใช้ในการสร้าง Web Pageเป็นภาษาประเภท Markup Language เกิดขึ้นจากการพัฒนาระบบ World Wide Web ในเดือนมีนาคม 1989 โดยนักวิจัยจากสถาบัน CERN (Conseil European Pour La Recherche Nucleaire) ซึ่งเป็นห้องทดลองในเมืองเจนีวา ประเทศสวิสเซอร์แลนด์ ชื่อ ทิม เบอร์เนอร์ - ลี (Tim Berners - Lee) ซึ่ง ทิม เบอร์เนอร์ - ลี ได้นาแนวความคิดในเรื่อง Hypertext ของ Vannevar Bush และ Ted Nelson มาใช้เพื่อกระจายข้อมูลในองค์กร ต่อมามีการพัฒนาและกำหนดมาตรฐานโดยองค์กรที่ชื่อว่า W3C (World Wide Web Consortium) ภาษา HTML เป็นภาษาที่มีลักษณะของข้อมูลที่เป็นตัวอักษรในมาตรฐานของรหัสแอสกี (ASCII Code) โดยเขียนอยู่ในรูปของเอกสารข้อความ (Text Document) จึงกำหนดรูปแบบและโครงสร้างได้ง่าย ภาษา HTML ได้ถูกพัฒนาขึ้นอย่างต่อเนื่องตั้งแต่ HTML Level 1 (รุ่นดั้งเดิม), HTML 2.0, HTML 3.0, HTML 3.2 และ HTML 4.0 ซึ่งเป็นรุ่นที่นิยมเขียนกันในปัจจุบัน (ขณะนี้ W3C ได้พัฒนา HTML 4.01 ออกมาแล้ว เพื่อลองรับมาตรฐานภาษา XML) จึงทำให้ภาษา HTML ในปัจจุบันสามารถแสดงภาพทาง กราฟฟิก และระบบเสียงได้ เพื่อตอบสนองในการทำงานในปัจจุบัน ภาษา HTML สามารถสร้างขึ้นได้จากโปรแกรมสร้างไฟล์ข้อความ (Text Editor) ทั่ว ๆ ไป เช่น Notepad หรือ Word Processing ได้ อีกทั้งง่ายต่อการเรียนรู้เพราะภาษา HTML ไม่มีโครงสร้างความเป็น Programming เลยแม้แต่น้อย และไฟล์ที่ได้จากการสร้างเอกสาร HTML ยังมีขนาดเล็กอีกด้วย นามสกุลของไฟล์ HTML จะเป็นไฟล์นามสกุล .htm หรือ .html ซึ่งใช้ในทั้งระบบปฏิบัติการยูนิกซ์ (UNIX) และระบบปฏิบัติการ Windows และเรียกใช้งานได้จากเว็บบราวเซอร์ (Web Browser) เช่น Internet Explorer หรือ NetScape - Tag Tag เป็นลักษณะเฉพาะของภาษา HTML ใช้ในการระบุรูปแบบคำสั่ง หรือการลงรหัสคำสั่ง HTML ภายในเครื่องหมาย less-than bracket ( < ) และ greater-than bracket ( > ) โดยที่ Tag HTML

แบ่งได้ 2 ลักษณะ คือ Tag เดี่ยว เป็น Tag ที่ไม่ต้องมีการปิดรหัส เช่น <P>, <BR> <img> <HR>เป็นต้น Tag เปิด/ปิด เป็น Tag ที่ประกอบด้วย Tag เปิด และ Tag ปิด โดย Tag ปิด จะมีเครื่องหมาย slash ( / ) นาหน้าคาสั่งใน Tag นั้นๆ เช่น <B>…</B>, <BLINK>…</BLINK> เป็นต้น - Attributes Attributes เป็นส่วนขยายความสามารถของ Tag จะต้องใส่ภายในเครื่องหมาย < > ในส่วน Tag เปิดเท่านั้น Tag คาสั่ง HTML แต่ละคาสั่ง จะมี Attribute แตกต่างกันไป และมีจำนวนไม่เท่ากัน การระบุ Attribute มากกว่า 1 Attribute ให้ใช้ช่องว่างเป็นตัวคั่น อ้างอิงจาก: (http://www.pyayam.com/html/mean.php, 2556)

**2.2 ทฤษฎีที่เกี่ยวข้องกับการวิเคราะห์และออกแบบระบบเชิงวัตถุ**

วิกิพีเดีย สารานุกรมเสรี (2554: ออนไลน์) การวิเคราะห์และออกแบบระบบ (อังกฤษ: Systems analysis) เป็นการศึกษาถึงปัญหาที่เกิดขึ้นในระบบงานปัจจุบัน (Current System) เพื่อออกแบบระบบการทำงานใหม่ (New System) นอกจากออกแบบสร้างระบบงานใหม่แล้ว เป้าหมายในการวิเคราะห์ระบบต้องการปรับปรุงและแก้ไขระบบงานเดิมให้มีทิศทางที่ดีขึ้น โดยก่อนที่ระบบงานใหม่ยังไม่นำมาใช้งาน ระบบงานที่ดำเนินการอยู่ในปัจจุบันเรียกว่า ระบบปัจจุบัน แต่ถ้าต่อมามีการพัฒนาระบบใหม่และนำมาใช้งาน เราะจะเรียกระบบปัจจุบันที่เคยใช้นั้นว่า ระบบเก่า (Old system)วิทวัฒน์ พัฒนา (2553: 21) การวิเคราะห์และออกแบบระบบ คือ วิธีการที่ใช้ในการสร้างระบบสารสนเทศขึ้นมาใหม่ในธุรกิจใดธุรกิจหนึ่งหรือระบบย่อยของธุรกิจ นอกจากการสร้างระบบสารสนเทศใหม่แล้ว การวิเคราะห์ระบบ ช่วยในการแก้ไขระบบสารสนเทศเดิมที่มีอยู่แล้วให้ดีขึ้นด้วยก็ได้

**2.2.1 วงจรการพัฒนาระบบ (System Development Life Cycle: SDLC)**

คือกระบวนการทางความคิด (Logical Process) พัฒนาระบบสารสนเทศเพื่อแก้ปัญหาทางธุรกิจและตอบสนองความต้องการของผู้ใช้ วงจรพัฒนาระบบมีทั้งหมด 7 ขั้นตอนคือ

**2.2.1.1 เข้าใจปัญหา (Problem Recognition)**

การที่จะแก้ไขระบบเดิมที่มีอยู่แล้วไม่ใช่เรื่องทีง่ายนัก หรือแม้แต่การสร้างระบบใหม่ ดั้งนั้นควรจะมีการศึกษาเสียก่อนว่า ความต้องการของเราเพียงพอที่เป็นไปได้หรือไม่ ได้แก่ "การศึกษาความเป็นไปได้ "(Feasibility Study)

**2.2.1.2 ศึกษาความเป็นไปได้ (Feasibility Study)**

จุดประสงค์ของการศึกษาความเป็นไปได้ก็คือ การกำหนดว่าปัญหาคืออะไรและตัดสินใจว่าการพัฒนาสร้างระบบสารสนเทศ หรือการแก้ไขระบบสารสนเทศเดิมมีความเป็นไปได้หรือไม่โดยเสียค่าใช้จ่ายและเวลาน้อยที่สุด และได้ผลเป็นที่น่าพอใจปัญหาต่อไปคือ นักวิเคราะห์ระบบจะต้องกำหนดให้ได้วาการแก้ไขปัญหาดังกล่าวมีความเป็นไปได้ทางเทคนิคและบุคลากร ปัญหาทางเทคนิคก็จะเกี่ยวข้องกับเรื่องคอมพิวเตอร์ และเครื่องมือเก่าๆถ้ามี รวมทั้งเครื่องคอมพิวเตอร์ ซอฟต์แวร์ด้วย ตัวอย่างคือ คอมพิวเตอร์ ที่ใช้อยู่ในบริษัทเพียงพอหรือไม่ คอมพิวเตอร์ อาจจะมีเนื้อที่ของฮาร์ดดิสก์ไม่เพียงพอ รวมทั้งซอฟต์แวร์ ว่าอาจจะต้องซื้อใหม่ หรือพัฒนาขึ้นใหม่ เป็นต้น ความเป็นไปได้ทางด้านบุคลากร คือ บริษัทมีบุคคลที่เหมาะสมที่จะพัฒนาและติดตั้งระบบเพียงพอหรือไม่ ถ้าไม่มีจะหาได้หรือไม่ จากที่ใด เป็นต้น นอกจากนั้นควรจะให้ความสนใจว่าผู้ใช้ระบบมีความคิดเห็นอย่างไรกับการเปลี่ยนแปลง รวมทั้งความเห็นของผู้บริหารด้วย

**2.2.1.3 วิเคราะห์ (Analysis)**

เริ่มเข้าสู่ การวิเคราะห์ระบบ การวิเคราะห์ระบบเริ่มตังแต่การศึกษาระบบการทำงานของธุรกิจนั้นในกรณี ที่ระบบเราศึกษานั้นเป็นระบบสารสนเทศอยู่แล้วจะต้องศึกษาว่าทำงานอย่างไรเพราะเป็นการยากที่จะออกแบบระบบใหม่โดยที่ไม่ทราบว่าระบบเดิมทำงานอย่างไร หรือธุรกิจดำเนินการอย่างไร หลังจากนั้นกำหนดความต้องการของระบบใหม่ ซึ่งนักวิเคราะห์ระบบจะต้องใช้เทคนิคในการเก็บข้อมูล (Fact-Gathering Techniques) ได้แก่ ศึกษาเอกสารที่มีอยู่ ตรวจสอบวิธีการทำงานในปัจจุบัน สัมภาษณ์ ผู้ใช้และผู้จัดการที่มีส่วนเกี่ยวข้องกับระบบ เอกสารที่มีอยู่ ได้แก่คู่มือการใช้งาน แผนผังใช้งานขององค์กรรายงานต่างๆที่หมุนเวียนในระบบการศึกษาวิธีการทำงานในปัจจุบันจะทำให้นักวิเคราะห์ระบบรู้ว่าระบบจริงๆทำงานอย่างไร ซึ่งบางครั้งค้นพบข้อผิดพลาดได้ตัวอย่าง เช่น เมื่อบริษัทได้รับใบเรียกเก็บเงินจะมีขั้นตอนอย่างไรในการจ่ายเงิน ขั้นตอนที่เสมียนป้อนใบเรียกเก็บเงินอย่างไร เฝ้าสังเกตการทำงานของผู้เกี่ยวข้อง เพื่อให้เข้าใจและเห็นจริงๆ ว่าขั้นตอนการทำงานเป็นอย่างไร ซึ่งจะทำให้นักวิเคราะห์ระบบค้นพบจุดสำคัญของระบบว่าอยู่ที่ใด

2.2.1.4 ออกแบบ (Design)

ในระยะแรกของการออกแบบ นักวิเคราะห์ระบบจะนำการตัดสินใจ ของฝ่ายบริหารที่ได้จากขั้นตอนการวิเคราะห์การเลือกซื้อคอมพิวเตอร์ ฮาร์ดแวร์ และซอฟต์แวร์ด้วย (ถ้ามีหรือเป็นไปได้)หลังจากนั้นนักวิเคราะห์ระบบจะนำแผนภาพต่างๆ ที่เขียนขึ้นในขั้นตอนการวิเคราะห์มาแปลงเป็นแผนภาพลำดับขั้น (แบบต้นไม้) เพื่อให้มองเห็นภาพลักษณ์ที่แน่นอนของโปรแกรมว่ามีความสัมพันธ์กัน อย่างไร และโปรแกรมอะไรบ้างที่ จะต้องเขียนในระบบ หลัง จากนั้นก็เริ่มตัดสินใจว่า ควรจะจัดโครงสร้างจากโปรแกรมอย่างไร การเชื่อมระหว่างโปรแกรมควรจะทำอย่างไร ในขั้นตอนการวิเคราะห์ นักวิเคราะห์ระบบต้องหาว่า "จะต้องทำอะไร (What)" แต่ในขั้นตอนการออกแบบต้องรู้ว่า“จะต้องทำอย่างไร (How)” ในการออกแบบโปรแกรมต้องคำนึงถึงความปลอดภัย (Security) ของระบบด้วยเพื่อป้องกันการผิดพลาดที่อาจจะเกิดขึ้น เช่น "รหัส" สำหรับผู้ใช้ที่มีสิทธิ์สำรองไฟล์ขอมูลทั้งหมดเป็นต้น

ถัดมาระบบจะต้องออกแบบวิธีการใช้งาน เช่น กำหนดว่าการป้อนข้อมูลจะต้องทำอย่างไรจำนวนบุคลากรที่ ต้องการในหน้าที่ ต่างๆ แต่ถ้านักวิเคราะห์ระบบตัดสินใจว่าการซื้อซอฟต์แวร์ ดีกว่าการเขียนโปรแกรม ขั้นตอนการออกแบบก็ไม่จำเป็นเลย เพราะสามารถนำซอฟต์แวร์ สำเร็จรูปมาใช้งานได้ทันที

สิ่งที่นักวิเคราะห์ระบบออกแบบมาทั้งหมดในขั้นตอนที่กล่าวมาทั้งหมดจะนำมาเขียนรวมเป็นเอกสารชุดหนึ่งเรียกว่า "ข้อมูลเฉพาะของการออกแบบระบบ" (System Design Specification) เมื่อสำเร็จแล้วโปรแกรมเมอร์ สามารถใช้เป็นแบบในการเขียนโปรแกรมได้ทันที สิ่งสำคัญก่อนทีจะส่งถึงมือโปรแกรมเมอร์เราควรจะตรวจสอบกับผู้ใช้ว่าพอใจหรือไม่ และตรวจสอบกับทุกคนในทีมว่าถูกต้องสมบูรณ์หรือไม่ และแน่นอนที่สุดต้องส่งให้ฝ่ายบริหารเพื่อตัดสินใจว่าจะดำเนินการ ต่อไปหรือไม่ ถ้าอนุมัติก็ผ่านเข้าสู่ขั้นตอนการสร้างหรือพัฒนาระบบ (Construction)

**2.2.1.5 สร้างหรือพัฒนาระบบ (Construction)**

ในขั้นตอนนี้โปรแกรมเมอร์ จะเริ่มเขียนและทดสอบโปรแกรมว่า ทำงานถูกต้องหรือไม่ต้องมีการทดสอบกับข้อมูลจริงที่เลือกแล้ว ถ้าทุกอย่างเรียบร้อย เราจะได้โปรแกรมที่พร้อมที่จะนำไปใช้งานจริงต่อไป หลังจากนั้นต้องเตรียมคู่มือการใช้และการฝึกอบรมผู้ใช้งานจริงของระบบโปรแกรมเมอร์เขียนโปรแกรมตามข้อมูลที่ได้จากเอกสารข้อมูลเฉพาะของการออกแบบ(Design Specification) ปกติแล้วนักวิเคราะห์ระบบไม่มีหน้าที่เกี่ยวข้องในการเขียนโปรแกรม แต่ถ้าโปรแกรมเมอร์คิดว่าการเขียนอย่างอื่นดีกว่าจะต้องปรึกษานักวิเคราะห์ระบบเสียก่อน เพื่อที่ว่านักวิเคราะห์จะบอกได้ว่าโปรแกรมที่จะแก้ไขนั้นมีผลกระทบกับระบบทั้งหมดหรือไม่ โปรแกรมเมอร์เขียนเสร็จแล้วต้องมีการทบทวนกับนักวิเคราะห์ระบบและผู้ใช้งาน เพื่อค้นหาข้อผิดพลาด วิธีการนี้เรียกว่า "Structure Walkthrough" การทดสอบโปรแกรมจะต้องทดสอบกับข้อมูลที่เลือกแล้วชุดหนึ่ง ซึ่งอาจจะเลือกโดยผู้ใช้ การทดสอบเป็นหน้าที่ของโปรแกรมเมอร์ แต่นักวิเคราะห์ระบบต้องแน่ใจว่า โปรแกรมทั้งหมดจะต้องไม่มีข้อผิดพลาด

**2.2.1.6 การปรับเปลี่ยน (Conversion)**

ขั้นตอนนี้บริษัทนำระบบใหม่มาใช้แทนของเก่าภายใต้การดูแลของนักวิเคราะห์ระบบ การป้อนข้อมูลต้องทำให้เรียบร้อย และในที่สุดบริษัทเริ่มต้นใช้งานระบบใหม่นี้ได้ การนำระบบเข้ามาควรจะทำอย่างค่อยเป็นค่อยไปทีละน้อย ที่ดีที่สุดคือ ใช้ระบบใหม่ควบคู่ไปกับระบบเก่าไปสักระยะหนึ่ง โดยใช้ข้อมูลชุดเดียวกันแล้วเปรียบเทียบผลลัพธ์ว่าตรงกันหรือไม่ ถ้าเรียบร้อยก็เอาระบบเก่าออกได้ แล้วใช้ระบบใหม่ต่อไป

**2.2.1.7 บำรุงรักษา (Maintenance)**

การบำรุงรักษาได้แก่ การแก้ไขโปรแกรมหลังจากการใช้งานแล้ว สาเหตุที่ต้องแก้ไขโปรแกรมหลังจากใช้งานแล้ว สาเหตุที่ต้องแก้ไขระบบส่วนใหญ่มี 2 ข้อ คือ

2.2.1.7.1. มีปัญหาในโปรแกรม (Bug)

2.2.1.7.2. การดำเนินงานในองค์กรหรือธุรกิจเปลี่ยนไป

จากสถิติของระบบที่พัฒนาแล้วทั้งหมดประมาณ 40%ของค่าใช้จ่ายในการแก้ไขโปรแกรม เนื่องจากมี "Bug" ดังนั้นนักวิเคราะห์ระบบควรให้ความสำคัญกับการบำรุงรักษา ซึ่งปกติจะคิดว่าไม่มีความสำคัญมากนักเมื่อธุรกิจขยายตัวมากขึ้น ความต้องการของระบบอาจจะเพิ่มมากขึ้นเช่น ต้องการรายงานเพิ่มขึ้น ระบบที่ดีควรจะแก้ไขเพิ่มเติมสิ่งที่ต้องการได้การบำรุงรักษา

ระบบควรจะอยู่ภายใต้การดูแลของนักวิเคราะห์ระบบ เมื่อผู้บริหารต้องการแก้ไขส่วนใดนักวิเคราะห์ระบบต้องเตรียมแผนภาพต่าง ๆ และศึกษาผลกระทบต่อระบบ และให้ผู้บริหารตัดสินใจต่อไปว่าควรจะแก้ไขหรือไม่

**2.2.2 โมเดล Prototype Model**

Prototype Model คือกระบวนการพัฒนาระบบซอฟต์แวร์โดยสร้างตัวต้นแบบขึ้นมาเพื่อที่จะให้ผู้ใช้งานระบบได้เห็นถึงรูปร่างลักษณะของระบบที่ตนเองต้องการ ทำให้ความต้องการของผู้ใช้งานชัดเจน ลดข้อผิดพลาดที่อาจจะเกิดจากความต้องการไม่ชัดเจนได้

Prototype model แบ่งออกเป็นสองชนิดย่อยๆคือ

2.2.2.1 Throw Away Prototype - การสร้างตัวต้นแบบ แบบใช้งานเสร็จแล้วทิ้ง สร้างมาเพื่อให้เห็นรูปแบบของ interface ตัวโปรแกรมต่างๆ ส่วนมากเป็นการเขียนในกระดาษ

2.2.2.2 Evolutionary Prototype - การสร้างตัวต้นแบบที่สามารถใช้งานได้จริงๆ อาจจะค่อยๆทำในทีละส่วน ปรับปรุงแก้ไขโปรแกรมจนกระทั่งเป็นโปรแกรมจริงที่สามารถใช้งานได้ตามความต้องการ



ภาพที่ 2.2 แผนภาพ Prototyping Model

**2.2.2.1 ซอฟแวร์ต้นแบบ**

2.2.2.1.1 ต้นแบบเป็นรูปแบบการทำงานของซอฟต์แวร์ที่มีฟังก์ชันการทำงานบางอย่างที่ จำกัด

2.2.2.1.2 ต้นแบบไม่เคยถือตรรกะที่แน่นอนที่ใช้ในการประยุกต์ใช้ซอฟแวร์ที่เกิดขึ้นจริงและเป็นความพิเศษที่จะได้รับการพิจารณาภายใต้การประเมินความพยายาม

2.2.2.1.3 การสร้างต้นแบบที่ใช้ในการช่วยให้ผู้ใช้ประเมินข้อเสนอการพัฒนาและลองพวกเขาออกก่อนที่จะดำเนินการ

2.2.2.1.4 นอกจากนี้ยังจะช่วยให้เข้าใจความต้องการของผู้ใช้ที่มีความเฉพาะ เจาะจงและอาจไม่ได้รับการพิจารณาโดยนักพัฒนาในการออกแบบผลิตภัณฑ์

**2.2.2.2 ประเภทซอฟแวร์ต้นแบบ**

มีหลายประเภทแตกต่างกันของต้นแบบซอฟต์แวร์ที่ใช้ในอุตสาหกรรมที่มี ต่อไปนี้เป็นประเภทต้นแบบซอฟต์แวร์หลักที่ใช้กันอย่างแพร่หลาย

2.2.2.2.1ใบปลิว / สร้างต้นแบบอย่างรวดเร็ว: ต้นแบบใบปลิวจะเรียกว่าเป็นการสร้างต้นแบบอย่างรวดเร็วหรือใกล้สิ้นสุดชนิดของการสร้างต้นแบบนี้ใช้ความพยายามน้อยมากกับการวิเคราะห์ความต้องการขั้นต่ำในการสร้างต้นแบบ เมื่อความต้องการที่เกิดขึ้นจริงมีความเข้าใจต้นแบบถูกยกเลิกและระบบที่เกิดขึ้นจริงได้รับการพัฒนาด้วยความเข้าใจที่ชัดเจนมากของความต้องการของผู้ใช้

2.2.2.2.2ต้นแบบวิวัฒนาการ: ต้นแบบวิวัฒนาการเรียกว่าเป็นเขียงหั่นขนมต้นแบบจะขึ้นอยู่กับการสร้างต้นแบบการทำงานที่เกิดขึ้นจริงกับการทำงานน้อยที่สุดในการเริ่มต้น ต้นแบบการพัฒนารูปแบบการเต้นของหัวใจของต้นแบบในอนาคตที่ด้านบนของซึ่งระบบทั้งหมดถูกสร้างขึ้น โดยใช้ต้นแบบวิวัฒนาการเพียงต้องการเข้าใจดีจะรวมอยู่ในต้นแบบและความต้องการจะเพิ่มเป็นและเมื่อพวกเขามีความเข้าใจ

2.2.2.2.3ต้นแบบที่เพิ่มขึ้น: การสร้างต้นแบบที่เพิ่มขึ้นหมายถึงการสร้างต้นแบบทำงานได้หลายระบบย่อยต่างๆแล้วรวมทั้งหมดต้นแบบที่มีอยู่ในรูปแบบระบบที่สมบูรณ์

2.2.2.2.4มาก Prototyping: การสร้างต้นแบบที่มากที่สุดที่ใช้ในการพัฒนาเว็บโดเมน มันประกอบด้วยสามขั้นตอนต่อเนื่องครั้งแรกต้นแบบขั้นพื้นฐานที่มีทุกหน้ามีอยู่จะถูกนำเสนอในรูปแบบ html ที่ แล้วการประมวลผลข้อมูลที่มีการจำลองการใช้บริการที่ชั้นต้นแบบ ในที่สุดการบริการที่มีการดำเนินการและบูรณาการที่จะต้นแบบสุดท้ายกระบวนการนี้เรียกว่าที่สุดต้นแบบที่ใช้ในการดึงความสนใจไปขั้นตอนที่สองของกระบวนการที่ UI ทำงานได้อย่างเต็มที่คือการพัฒนาโดยไม่คำนึงถึงน้อยมากที่จะให้บริการที่เกิดขึ้นจริง

**2.2.2.3 ซอฟแวร์ต้นแบบการประยุกต์ใช้**

ซอฟแวร์ต้นแบบมีประโยชน์มากที่สุดในการพัฒนาระบบที่มีระดับสูงของการปฏิสัมพันธ์ของผู้ใช้เช่นระบบออนไลน์ ระบบที่ต้องการให้ผู้ใช้สามารถกรอกแบบฟอร์มหรือผ่านหน้าจอต่างๆก่อนที่จะมีการประมวลผลข้อมูลที่สามารถใช้งานได้อย่างมีประสิทธิภาพการสร้างต้นแบบมากที่จะให้ดูแน่นอนและความรู้สึกแม้กระทั่งก่อนที่ซอฟแวร์ที่เกิดขึ้นจริงได้รับการพัฒนา

ซอฟแวร์ที่เกี่ยวข้องกับการมากเกินไปของการประมวลผลข้อมูลและส่วนใหญ่ของการทำงานภายในที่มีส่วนติดต่อผู้ใช้น้อยมากมักจะไม่ได้รับประโยชน์จากการสร้างต้นแบบ การพัฒนาต้นแบบอาจจะมีค่าใช้จ่ายเพิ่มเติมในโครงการดังกล่าวและอาจต้องใช้ความพยายามมากเป็นพิเศษ

**2.2.2.4 ข้อดีของการทำ Prototypes**

2.2.2.4.1เรียนรู้จาก Software จริง

2.2.2.4.2ช่วยสื่อสารระหว่าง User กับ Developer เพื่อทำให้ทราบว่างานก้าวหน้าไปถึงไหน

2.2.2.4.3User มีส่วนร่วมในการทำ

2.2.2.4.4ทำให้ Requirement ชัดเจน (ใช่หรือไม่ ที่ต้องการ)

2.2.2.4.5การ Test Prototypes ทำให้ Project สมบูรณ์

2.2.2.4.6ลดความต้องการ Document-user เห็นแล้วเข้าใจ

2.2.2.4.7ลดต้นทุนในการ Maintenance-user เคยเห็นแล้วจะทำให้ลดต้นทุน, ค่าใช้จ่าย ในการMaintenance ลงเหลือ 10-20 % ส่วนใหญ่ Maintenance ในส่วนของ Error

2.2.2.4.8ลดข้อจำกัดของ Feature ต่างๆ ในการทำงาน

2.2.2.4.9ทำให้เห็นผลลัพธ์ว่าเป็นอย่างไร

**2.2.2.5 ข้อเสียของ Prototypes**

2.2.2.5.1User เห็น Prototypes เข้าใจว่าเสร็จแล้ว มักต้องการรู้ว่าจะใช้ได้เมื่อไหร่

2.2.2.5.2ไม่มี Security, Response Time

2.2.2.5.3ควบคุมยาก เมื่อเห็นแล้ว User มักจะมีการเปลี่ยนแปลง Requirement

2.2.2.5.4เสียเวลาและค่าใช้จ่ายในการทำ Prototypes

2.2.2.5.5ประสิทธิภาพของจริงกับ Prototype ต่างกันได้ เช่น Prototypes เร็ว ของจริง ช้า, Prototypes ช้าของจริงเร็ว

2.2.2.5.6User กับ Developer ไม่ได้อยู่ที่เดียวกันทำให้ไม่สะดวกในการทำPrototype

**2.2.2.6 รูปแบบของการทำ Prototypes**

2.2.2.6.1Mock–ups ทำหน้าจอให้เห็น แต่ใช้จริงไม่ได้ (ใช้อธิบาย)

2.2.2.6.2Simulated Interaction จำลองการใช้งานจริง เช่น การกรอกข้อมูล การเรียกใช้ข้อมูล แต่ใช้งานจริงไม่ได้ (วิธีนี้นิยมมาก)

2.2.2.6.3Partial working Model แบ่งเป็น

2.2.2.6.3.1Vertical (ลึก) สามารถใช้งาน Feature บางส่วนได้จริง (ไม่ทั้งหมด) ใช้กับ Simulate

2.2.2.6.3.2Horizontal (กว้าง) มี Feature ครบ ทำให้ทราบว่ามี Feature อะไรบ้าง แต่ใช้งานไม่ได้ทั้งหมด (ไม่ลึก) ใช้กับ Mock–up

**2.2.2.7 ตัวต้นแบบ (Prototype)**

มีหลายแบบสามารถแบ่งอธิบายได้ดังนี้

2.2.2.7.1การตอบสนองของผู้ใช้ระบบที่มีต่อระบบงาน (initial user reactions) นักวิเคราะห์ระบบสมารถรวบรวมปฏิกิริยาของผู้ใช้ จากการสังเกต การสัมภาษณ์ หรือการพูดคุยและจากแบบสอบถามที่นักวิเคราะห์ระบบได้จัดทำเอาไว้ ซึ่งจะพบความหลากหลายของปฏิกิริยาของผู้ใช้ที่มีต่อระบบ ทั้งนี้เพราะต่างคนต่างมีความคิดเห็นที่แตกต่างกันออกไป อาจมีทั้งความเต็มใจและต่อต้านระบบที่ติดตั้งใหม่เพราะผู้ใช้ระบบคิดว่าระบบงานใหม่นั้นจะไปแทนตัวเอง

2.2.2.7.2 ข้อเสนอแนะต่าง ๆของผู้ใช้ระบบ (user suggestions) ข้อมูลที่สำคัญอีกสิ่งหนึ่งที่นักวิเคราะห์ระบบได้ในระหว่างการนำเสนอระบบ โดยการทำโปรโตไทป์ คือ ข้อคิดเห็นและข้อเสนอแนะจากผู้ใช้ทั้งนี้เพราะผู้ใช้ต่างคนต่างมีความคิดเห็นที่แตกต่างกัน ข้อเสนอแนะเหล่านี้จะเป็นตัวชี้ช่องให้นักวิเคราห์ระบบเห็นว่าควรจะจัดการอย่างไรในการพัฒนาระบบ เช่น การแก้ไข ปรับปรุง หรืออาจจะต้องโละทิ้ง แล้วทำใหม่ ที่มีความเหมาะสมและสามารถรองรับความต้องการของผู้ใช้ได้ดีที่สุด

2.2.2.7.3 ระบบใหม่ที่เกิดขึ้นภายหลัง (innovations) ระบบใหม่ที่อาจเกิดขึ้นในระหว่างการพัฒนาระบบโดยส่วนมากจะยังไม่มีใครพูดถึงหรือคิดถึงว่าควรจะมีระบบนี้เกิดขึ้น แต่เมื่อผู้ใช้และนักวิเคราะห์ระบบได้เข้าไปสัมผัสแล้ว ก็อาจเห็นว่าเป็นระบบที่มีความจำเป็นที่จะต้องทำการพัฒนาขึ้น เพื่อเพิ่มเติมความสมบูรณ์ของระบบที่กำลังพัฒนาอยู่

2.2.2.7.4 การทบทวนแผนงานและการพัฒนาระบบ (revision plans) การทำโปรโตไทป์จะทำให้นักวิเคราะห์ระบบมองเห็นถึงระบบงานที่ควรจะต้องพัฒนาต่อไปในอนาคต ทำให้เกิดการทบทวนแผนงานต่าง ๆ อยู่ตลอดเวลาซึ่งจะช่วยให้นักวิเคราะห์ระบบสามารถจัดลำดับความสำคัญของระบบว่าควรจะทำโปรโตไทป์ระบบใดก่อนหลัง

**2.2.2.8 เมื่อใช้แบบจำลองต้นแบบ**

2.2.2.8.1รุ่นต้นแบบที่ควรจะใช้เมื่อระบบที่ต้องการความต้องการที่จะมีจำนวนมากของการมีปฏิสัมพันธ์กับผู้ใช้

2.2.2.8.2โดยปกติระบบออนไลน์เชื่อมต่อเว็บที่มีจำนวนที่สูงมากของการมีปฏิสัมพันธ์กับผู้ใช้จะเหมาะที่สุดสำหรับรุ่นต้นแบบ มันอาจจะใช้เวลาสักครู่เพื่อให้ระบบที่จะสร้างที่ช่วยให้ความสะดวกในการใช้งานและความต้องการฝึกอบรมน้อยที่สุดสำหรับผู้ใช้

2.2.2.8.3การสร้างต้นแบบเพื่อให้แน่ใจว่าผู้ใช้อย่างต่อเนื่องการทำงานกับระบบและให้ข้อเสนอแนะที่เป็นนิติบุคคลที่จัดตั้งในต้นแบบที่จะส่งผลในระบบพร้อมใช้ พวกเขาเป็นที่ดีสำหรับการออกแบบระบบอินเตอร์เฟซคอมพิวเตอร์ของมนุษย์ที่ดี

**2.3 ทฤษฎีฐานข้อมูล**

**2.3.1 ความหมายของระบบฐานข้อมูล**

ฐานข้อมูลประกอบไปด้วยรายละเอียดของข้อมูลที่เกี่ยวข้องกันซึ่งถูกนำมาใช้ในงานด้านต่างๆเช่นด้านการเก็บข้อมูลการจองสนามฟุตบอล ด้านข้อมูลของลูกค้า จะมีฐานข้อมูลเกี่ยวกับลูกค้าบันทึกรายรับจากการจองสนามฟุบอลหญ้าเทียม ฯลฯ เป็นต้น ข้อมูลเหล่านี้จะถูกเก็บอย่างมีระบบเพื่อประโยชน์ในการจัดการและเรียบเรียงใช้ข้อมูลได้อย่างมีประสิทธิภาพ

**2.3.2 องค์ประกอบของระบบฐานข้อมูล**

ระบบฐานข้อมูลส่วนใหญ่เป็นระบบที่มีการนำระบบคอมพิวเตอร์เข้ามาช่วยในการจัดเก็บโดยมีโปรแกรม Software ช่วยในการจัดการข้อมูลเหล่านี้ เพื่อให้ได้ข้อมูลตามที่ผู้ใช้ต้องการองค์ประกอบของฐานข้อมูลแบ่งออกเป็น 4 ประเภท คือ

**2.3.2.1. ฮาร์ดแวร์ (Hardware)**

ในระบบฐานข้อมูลที่มีประสิทธิภาพควรมีฮาร์ดแวร์ต่างๆที่พร้อมจะอำนวยความสะดวกในการบริหารข้อมูลอย่างมีประสิทธิภาพไม่ว่าจะเป็นความเร็วของหน่อยประมวลผลกลางขนาดของหน่วยความจำหลัก อุปกรณ์นำเข้าและออกข้อมูล รายงานหน่วยความจำสำรองที่จะรองรับการประมวลผลข้อมูลในระบบได้อย่างมีประสิทธิภาพ

**2.3.2.2. ซอฟต์แวร์ (Software)**

ในการประมวลผลข้อมูลอาจจะใช้ซอฟต์แวร์หรือโปรแกรมที่แตกต่างกัน ทั้งนี้ขึ้นอยู่กับคอมพิวเตอร์ที่นำมาใช้ว่าเป็นแบบใด โปรแกรมจะทำหน้าที่ดูแลการสร้าง การเรียกใช้ข้อมูล การจัดทำรายงาน การปรับเปลี่ยน แก้ไข โครงสร้างกาวควบคุม หรืออาจกล่าวได้อีกอย่างหนึ่งว่า ระบบจัดการฐานข้อมูล (Database Management System: DBMS) คือโปรแกรมที่ทำหน้าที่ในการจัดการฐานข้อมูลโดยเป็นสื่อกลางระหว่างผู้ใช้และโปรแกรมประยุกต์ต่างๆ ที่มีอยู่ในฐานข้อมูลตัวอย่างของโปรแกรมระบบจัดการฐานข้อมูลได้แก่ DBASE, VEXCEL, ACCESS, INFORMIX, ORACLE เป็นต้น

**2.3.2.3. ข้อมูล (Data)**

ฐานข้อมูลเป็นการจัดเก็บรวบรวมข้อมูลให้เป็นศูนย์กลางข้อมูลอย่างมีระบบ ซึ่งข้อมูลเหล่านี้สามารถเรียกใช้ร่วมกันได้ ผู้ใช้ข้อมูลในระบบฐานข้อมูล จะมองภาพข้อมูลในลักษณะที่แตกต่างกัน เช่น การใช้บางคนมองภาพของข้อมูลที่จัดเก็บไว้ในสื่อข้อมูล ผู้ใช้บางคนมองภาพข้อมูลจากการใช้งาน เป็นต้น

**2.3.1.4. เขตข้อมูล (Field)**

ขนาดความยาวของเขตข้อมูล(Field) ที่สามารถกำหนดได้ เช่น เขตข้อมูล “ชื่อ” จะใช้ความยาวถึงขนาด 30 ตัวอักษร ส่วนเขตข้อมูล “อายุ” อาจใช้เพียง 2 ตัวอักษร เป็นต้น แฟ้มข้อมูล(Field) แต่ละแฟ้มจะต้องกำหนดเขตข้อมูลที่ใช้ชื่อเดียวกันให้ยาวเท่ากัน เช่น ถ้าเป็นเขตข้อมูล “ชื่อ”ไม่ว่าจะชื่อสั้นหรือยาวเพียงใด ก็ต้องใช้เนื้อที่ 30 ตัว อักขระหมดทั้งแฟ้ม

**2.3.1.4.1.ชื่อเขตข้อมูล (Field Name)**

ตั้งชื่อฟิลด์ได้ยาว ไม่เกิน 64 ตัวอักขระ มีช่องว่าง เว้นวรรคได้ แต่ยกเว้นเครื่องหมายต่อไปนี้ . ! [] สามารถตั้งชื่อเป็นภาษาไทยได้ / ภาษาอังกฤษ

**2.3.1.4.2.ชนิดของเขตข้อมูล (Data Type)**

Data Type อธิบาย

CHAR (M) จะเอาไว้เก็บข้อมูลที่เป็น string (สายอักษร) โดยที่ขนาดของการเก็บมีความคงที่ first name CHAR (25);

VARCHAR (M) ข้อมูลทีเป็น String (สายอักษร) โดยที่ขนาดของการเก็บมีความคงที่ first name CHAR (25);

INT (M) Unsigned INT เก็บค่าจำนวนเต็ม มีค่าตั้งแต่ -2147483648 ถึง +2147483647 แต่ถ้าใส่ Unsigned จะมีค่าตั้งแต่ 0

**2.3.1.4.3.ขนาดหรือความยาวของเขตข้อมูล (Field size)**

ในกรณีที่มีการกำหนด Field ใดๆ เป็นตัวอักษร (Number) จะต้องกำหนด Field Size ของตัวเลขให้ตรงตามการด้วย ซึ่งแบบของ Field Size แต่ละชนิดจะแตกต่างกันที่ ขอบเขตตัวเลขที่สามารถเก็บได้ ในแต่ละ Field

|  |  |
| --- | --- |
| **ชนิดขนาดเขตข้อมูล** | **คำอธิบาย** |
| Byte | ตัวเลขจำนวนเต็มมีค่าระหว่าง 0 ถึง 255 |
| Integer | ข้อมูลตัวเลขจำนวนเต็มทีค่าระหว่าง -32,768 ถึง 32,767 |
| Long Integer | ข้อมูลตัวเลขจำนวนเต็มมีค่าระหว่าง -2,147,483,648 ถึง 2, 147,483,647 |
| Single | ข้อมูลตัวเลขทศนิยมตั้งแต่ -3.4 x 1038 ถึง 3.4 x 1038 |
| Double | ข้อมูลตัวเลขทศนิยมตั้งแต่ -1.797 x 10308 ถึง 1.797 x 10308 |
| Replication ID | ข้อมูลสำหรับ Replication |
| Decimal | ข้อมูลตัวเลขเป็นทศนิยม 1038 |

ตารางที่ 2.1 ขนาดหรือความยาวของเขตข้อมูล

**2.3.2 การจัดการฐานข้อมูล**

การจัดการฐานข้อมูล(Database Management) คือการบริหารแหล่งข้อมูลที่ถูกเก็บรวบรวมไว้ที่ศูนย์กลางเพื่อตอบสนองต่อการใช้โปรแกรมประยุกต์อย่างมีประสิทธิภาพและลดการซ้ำซ้อนของข้อมูลรวมทั้งความขัดแย้งของข้อมูลที่เกิดขึ้นภายในองค์การในอดีตการเก็บข้อมูลมักจะเป็นอิสระต่อกันไม่มีการเชื่อมโยงของข้อมูลเกิดการสิ้นเปลืองพื้นที่ในการเก็บข้อมูล

**2.3.3 ส่วนประกอบระบบการจัดการฐานข้อมูล**

**2.3.3.1 ภาษาคำนิยามของข้อมูล [Data Definition Language (DDL)]**

ในส่วนนี้จะกล่าวถึงส่วนประกอบของระบบการจัดการฐานข้อมูลว่าข้อมูลแต่ละส่วนประกอบด้วยอะไรบ้าง [Data element] ในฐานข้อมูลซึ่งเป็นภาษาทางการที่นักเขียนโปรแกรมใช้ในการสร้างเนื้อหาข้อมูลและโครงสร้างข้อมูลก่อนที่ข้อมูลดังกล่าวจะถูกแปลงเป็นแบบฟอร์มที่ต้องการของโปรแกรมประยุกต์หรือในส่วนของ DDL จะประกอบด้วยคำสั่งที่ใช้ในการกำหนดดัชนี เป็นต้น

**2.3.3.2 ภาษาการจัดการฐานข้อมูล [Data Manipulation Language (DML)]**

เป็นภาษาเฉพาะที่ใช่ในการจัดการระบบฐานข้อมูลซึ่งอาจจะเป็นการเชื่อมโปรแกรมภาษาในยุคที่สามและยุคที่สี่เข้าด้วยกัน เพื่อจัดการข้อมูลในฐานข้อมูลภาษานี้มักจะประกอบด้วยคำสั่งที่อนุญาตให้ผู้ใช้สามารถสร้างโปรแกรมพิเศษขึ้นมารวมถึงข้อมูลต่างๆ ในปัจจุบันที่นิยมใช้ ได้แก่ ภาษาSQL (Structure Query Language) แต่ถ้าหากเป็นเครื่องคอมพิวเตอร์ขนาดใหญ่ DBMS มักจะสร้างด้วยภาษาโคนบอล (COBOL language) ภาษาฟอร์แทน (FORTRAN) และ ภาษาอื่นในยุคที่สาม

**2.3.3.3 พจนานุกรมข้อมูล (Data Dictionary)**

เป็นเครื่องมือสำหรับการเก็บและการจัดการข้อมูลสำหรับการบำรุงรักษาในฐานข้อมูล โดยพจนานุกรมจะมีการกำหนดชื่อของสิ่งต่างๆ (Entity) และระบุไว้ในโปรแกรมฐานข้อมูล เช่น ชื่อของฟิลด์ ชื่อของโปรแกรมที่ใช้รายละเอียดของข้อมูลผู้มีสิทธิ์ใช้และผู้ที่รับผิดชอบแสดงส่วนประกอบของระบบการจัดการฐานข้อมูล