บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบฐานข้อมูล (Database Systems)

เนื้อหาของ ระบบฐานข้อมูล (Database Systems) ดังต่อไปนี้อ้างอิงมาจากหนังสือ: โอภาส เอี่ยมสิริวงค์. ระบบฐานข้อมูล Database Systems. ปีที่พิมพ์ พ.ศ. 2551, 14-53.

2.1.1 ข้อมูลและสารสนเทศ (Data Versus Information)

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า การทำความเข้าใจเบื้องต้นเกี่ยวกับฐานข้อมูลนั้น จำเป็นต้องรับรู้ถึงความแตกต่างระหว่างข้อมูลกับสารสนเทศเสียก่อน สำหรับข้อมูลในยุคปัจจุบันอาจเป็นชนิดข้อมูลแบบมีโครงสร้าง (Structured Data Types) หรือไม่มีโครงสร้าง (Unstructured Data Types) ก็ได้ โดยข้อมูลทั้งชนิดแบบมีโครงสร้างและไม่มีโครงสร้างบ่อยครั้งมักถูกนำมาใช้ร่วมกันบนฐานข้อมูลเดียวกัน ตัวอย่างเช่น อู่ซ่อมรถยนต์ที่สามารถนำข้อมูลทั้งแบบมีโครงสร้าง (ข้อมูลลูกค้า ข้อมูลรถ) นำมาใช้ร่วมกันกับข้อมูลแบบมัลติมีเดีย (ภาพถ่ายดิจิตอล ภาพสแกน) ด้วยการนำภาพถ่ายรถยนต์ที่เกิดอุบัติเหตุหรือภาพสแกนข้อมูลสำคัญต่างๆ นำส่งพร้อมกับแบบฟอร์มข้อมูลเพื่อส่งเคลมกับบริษัทประกันภัย เป็นต้น

ความจริงแล้ว คำว่าข้อมูล (Data) และสารสนเทศ (Information) นั้นมีความสัมพันธ์อย่างไกล้ชิดทำให้มีการนำไปใช้งานสลับกันอยู่บ่อยครั้ง แต่ทั้งสองคำมีความหมายที่แตกต่างกัน โดยข้อมูลหมายถึงข้อเท็จจริง วัตถุ หรือเหตุการณ์ที่เกิดขึ้น กล่าวคือเป็นข้อมูลดิบ (Raw Data) ที่ยังไม่ได้ผ่านการประมวลผลในขณะที่สารสนเทศหมายถึง ข้อมูลที่ผ่านการประมวลผลแล้ว และสามารถนำไปใช้ให้เกิดประโยชน์ตามแต่ละบุคคลที่ต้องการได้ โดยพิจารณาจากรูป (โอภาส เอี่ยมสิริวงค์, 2551:15)

2.1.2 การจัดการข้อมูล (Data Management)

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า แนวคิดการจัดการข้อมูลได้เกิดขึ้นตั้งแต่อดีตกาล ซึ่งเป็นไปตามเทคโนโลยีในแต่ละยุคสมัยการจัดการข้อมูลได้ริเริ่มจากการบันทึกข้อมูล โดยวิธีพื้นฐานที่สุดคือการบันทึกข้อมูลลงในสมุด ถ้าหากต้องการค้นหาข้อมูลที่เคยบันทึกไว้ ก็พลิกหน้าหนังสือไปยังเลขหน้าที่ต้องการเพื่อดูรายละเอียดในแต่ละหัวข้อของข้อมูลนั้นๆ โดยไม่มีรูปแบบการจดบันทึกข้อมูลที่แน่นอน

ต่อมาเมื่อเริ่มมีปริมาณข้อมูลเพิ่มขึ้นอย่างต่อเนื่อง จึงได้มีการพัฒนารูปแบบการจัดเก็บข้อมูลให้มีระเบียบ มีมาตรฐานยิ่งขึ้น ด้วยการบันทึกข้อมูลลงในเอกสารและจัดเก็บไว้ในแฟ้มเอกสารอย่างเป็นหมวดหมู่ มีการจัดทำสารบัญ การทำดัชนีเพื่อให้การค้นหาข้อมูลมีความเร็วขึ้น นอกจากนั้นยังจำเป็นต้องมีอุปกรณ์หรือเครื่องมือที่ใช้จัดเก็บแฟ้มเอกสารเหล่านั้น เพื่อให้เกิดความปลอดภัยยิ่งขึ้นด้วย

Computer Science Department Faculty of Informatics Mahasarakham University

การมีตู้เก็บเอกสาร ซึ่งก็มีทั้งขนาดเล็กและขนาดใหญ่ให้เลือกใช้งานตามความเหมาะสมเพื่อจัดเก็บแฟ้มเอกสารเหล่านั้น

ต่อมาเมื่อมีการนำเทคโนโลยีคอมพิวเตอร์มาใช้เพื่อการจัดเก็บข้อมูล จึงทำให้ช่วยลดจำนวนเอกสารลงได้มาก เนื่องจากสื่อบันทึกข้อมูลในปัจจุบันสามารถจัดเก็บข้อมูลได้อย่างมหาศาล โดยเปรียบเทียบให้เห็นภาพได้จากจำนวนเอกสารมากมายที่จัดเก็บอยู่ในตู้เอกสารหลายๆ ตู้ที่สามารถจัดเก็บลงในฮาร์ดิสก์หรือในซีกีรอมเพียงไม่กี่แผ่น อีกทั้งยังช่วยค้นหาข้อมูลเป็นไปด้วยความสดวกและรวดเร็ว นอกจากนี้ ข้อมูลที่จัดเก็บยังไม่ถูกลบเลือนไปตามกาลเวลา เหมือนกับข้อมูลที่บันทึกลงในเอกสารที่อาจถูกลบเลือนได้ตามกาลเวลาที่ผ่านไป

อย่างไรก็ตาม เป็นที่เข้าใจแล้วว่าในการนำคอมพิวเตอร์มาเป็นเครื่องมือในการประมวลผลข้อมูลนั้นจะไม่สามารถดำเนินการใดๆ ได้เลยหากปราศจากข้อมูล โดยข้อมูลที่จัดเก็บจะถูกจัดเก็บในรูปแบบของแฟ้มเพื่อให้คอมพิวเตอร์สารมารถอ่านข้อมูลจากแฟ้มข้อมูลเหล่านั้นได้ ดังนั้นเพื่อให้เกิดความเข้าใจยิ่งขึ้น จึงจำเป็นต้องเข้าใจถึงพื้นฐานโครงสร้างแฟ้มข้อมูลเสียก่อน (โอภาส เอี่ยมสิริวงค์, 2551:19)

2.1.3 โครงสร้างแฟ้มข้อมูล

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า โครงสร้างแฟ้มข้อมูลประกอบไปด้วยโครงสร้างพื้นฐานที่ลำดับจากหน่วยเล็กที่สุดไปยังหน่วยที่ใหญ่ขึ้นตามลำดับต่อไปนี้คือ บิต (Bit) ไบต์ (Byte) ฟิวด์ (Field) เรคอร์ด (Record) และไฟล์ (File)

2.1.3.1 บิต (Bit) บิตประกอบด้วยเลขฐานสอง (Binary Digit) ถือเป็นหน่วยที่เล็กที่สุดของข้อมูลในคอมพิวเตอร์ โดยบิตจะมีเพียงหนึ่งใน 2 สถานะเท่านั้นคือ 0 หรือ 1 เพื่อแทนสัญญาณไฟฟ้า Off หรือ On ดังนั้นบิตจึงไม่สามารถแทนค่าข้อมูลในปริมาณมากๆ ได้ (น.20)

2.1.3.2 ไบต์ (Byte) เมื่อบิตไม่สามารถแทนค่าข้อมูลในปริมาณมากๆ ได้ เนื่องมาจากมีเพียง 2 สถานะเท่านั้น จึงได้มีการนำบิคหลายๆ บิตมารวมกันเป็นไบต์ ซึ่งโดยปกติแล้ว 1 ไบต์จะประกอบไปด้วย 8 บิต ดังนั้นจึงทำให้หนึ่งไบต์สามารถสร้างรหัสแทนข้อมูลขึ้นมาใช้แทนตัวอักขระให้แตกต่างกันได้ถึง 28หรือ 256 อักขระด้วยกัน

2.1.3.3 ฟิลด์ (Field) ฟิลด์คือการนำอักขระหรือไบต์ตั้งแต่ 1 ไบต์ขึ้นไปมารวมกันเพื่อให้เกิดความหมายขึ้นมา เช่น ฟิลด์ name เป็นฟิลด์ที่ใช้แทนชื่อของพนักงาน หรือฟิลด์ address ที่ใช้เก็บที่อยู่ของพนักงาน เป็นต้น

2.1.3.4 เรคอร์ด (Record) เรคอร์ดคือกลุ่มของฟิลด์ที่มีความสัมพันธ์กัน กล่าวคือ ใน 1 เรคอร์ดจะประกอบไปด้วยฟิลด์ต่างๆ ที่เกี่ยวข้องรวมกันเป็นชุด ตัวอย่างเช่น เรคอร์ดพนักงาน ประกอบ

Computer Science Department
Faculty of Informatics, Mahasarakham University

ไปด้วยฟิลด์รหัสพนักงาน ชื่อ นามสกุล ที่อยู่ เบอร์โทรศัพท์ ตำแหน่ง เพศ วันเกิด และเงินเดือน เป็นต้น ดังนั้นภายใน 1 เรคอร์ดจึงจำเป็นต้องมีอย่างน้อย 1 ฟิลด์เพื่อใช้สำหรับอ้างอิงข้อมูลในเรคอร์ดนั้นๆ

2.1.3.5 ไฟล์ (File) ไฟล์คือกลุ่มของเรคอร์ดที่สัมพันธ์กัน ตัวอย่างเช่น ในแฟ้มพนักงานจะประกอบไปด้วยเรคอร์ดต่างๆ ของพนักงานทั้งหมดที่อยู่ในบริษัติ ดังนั้นไฟล์ 1 ไฟล์จึงจำเป็นต้องมีอย่างน้อย 1 เรคอร์ดเพื่อใช้สำหรับอ่านข้อมูลขึ้นมาใช้งาน

2.1.4 ระบบแฟ้มข้อมูล (Files-Based System)

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า วิธีดังเดิมของการจัดการกับเอกสารตามหน่วยงานหรือบริษัตต่างๆ ก็คือ การจัดเก็บเอกสารไว้ในแฟ้มเอกสาร ซึ่งแฟ้มเอกสารต่างๆ เหล่านี้ แต่ละแผนกก็จะมีการจัดเก็บเป็นของตนเองที่เป็นไปตามความต้องการของแต่ละแผนก ครั้นเมื่อเวาลาผ่านไป บริษัตเติมโตยิ่งขึ้น จึงทำให้เอกสารมีจำนวนมากขึ้นตามลำดับ ดังนั้นการค้นหาเอกสารในแฟ้มเอกสารจึงเป็นไปด้วยความลำบาก และต้องใช้เวลานานมากขึ้น

ต่อมาเมื่อมีการนำเทคโนโลยีคอมพิวเตอร์มาใช้จัดเก็บข้อมูล ซึ่งในยุคแรกนั้น วิธีการจัดเก็บข้อมูลด้วยคอมพิวเตอร์ยังคงมีรูปแบบคล้ายคลึงกับการเก็บแฟ้มเอกสารด้วยมือ เพียงแต่ต่างกันตรงที่ข้อมูลที่จัดเก็บนั้นถูกจัดเก็บลงในคอมพิวเตอร์ที่เรียกว่า แฟ้มข้อมูล หรือที่มักเรียกกันว่าไฟล์ (File) แต่ละแผนกต่างก็มีโปรแกรมที่จัดทำขึ้นเพื่อใช้งานกัลแฟ้มข้อมูลเฉพาะส่วนงานของตน ต่อมาในปี ค.ศ. 1960 ที่อยู่ในยุคการประมวลผลข้อมูลด้วยคอมพิวเตอร์ ได้กำเนิดภาษาโคบอลขึ้นมา โดยจัดเป็นภาษารุ่นที่ 3 (Third-Generation Language: 3GL) ซึ่งรุ่นที่ 3 นั้นจำเป็นต้องพึ่งพาโปรแกรมเมอร์ในการเขียนโปรแกรมเพื่อจัดการกับแฟ้มข้อมูลในระบบ

และด้วยข้อจำกัดของระบบแฟ้มข้อมูล ที่แต่แผนกต่างก็มีโปรแกรมเพื่อจัดการกับแฟ้มข้อมูลของตนเอง กล่าวคือ แผนกบุคลากรมีโปรแกรมใช้งานของตน ในขณะที่แผนกขายก็มีโปรแกรมใช้งานบนแฟ้มข้อมูลของตนเช่นกัน ด้วยเหตุนี้จะเห็นได้ว่าได้เกิดความซ้ำซ้อนในข้อมูลขึ้นแล้ว คือข้อมูลพนักงาน (Employees) ของแผนกบุคลากร และข้อมูลพนักงานขาย (Salesman) ของแผนกขาย ซึ่งทั้งนี้พนักงานขายก็คือพนักงานคนหนึ่งนั่นเอง และนอกจากปัญหาความซ้ำซ้อนที่เกิดขึ้นแล้ว ยังมีปัญหาเกี่ยวกับ ข้อมูลที่ผูกติดกับแอปพริเคชั่นโปรแกรมที่ส่งผลให้เกิดปัญหาการปรับปรุงโครงสร้างข้อมูลในอนาคต กล่าวคือหากมีความต้องการเพิ่มฟิลด์หรือปรับปรุงความกว้างของข้อมูล รวมถึงการปรับเปลี่ยนรูปแบบการเข้าถึงไฟล์ข้อมูลใหม่ ก็จำเป็นต้องแก้ไขโปรแกรมใหม่เสมอ เนื่องจากรูปแบบโครงสร้างข้อมูลได้ผูกติดกับภาษาโปรแกรม ซึ่งผู้ใช้งานทั่วไปจะไม่มีความรู้ความสามารถในการเข้าไปแก้ไขชุดคำสั่งใดๆ ได้ นอกจากผู้ที่มีความเชี่ยวชาญโดยตรง ซึ่งก็คือโปรแกรมเมอร์เท่านั้น (โอภาส เอี่ยมสิริวงค์, 2551:25)

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.1.5 ข้อจำกัดของวิธีแฟ้มข้อมูล (Limitations of the File-Based Approach)

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า ข้อจำกัดของวิธีแฟ้มข้อมูล (Limitations of the File-Based Approach) มีรายละเอียดดังนี้

2.1.5.1 ข้อมูลมีความซ้ำซ้อน (Duplication of Data) สืบเนื่องมาจากข้อมูลที่มีการจัดเก็บแยกออกจากกัน ทั้งแฟ้มผนักงานและพนักงานขายต่างถูกจัดเก็บไว้ในแต่ละแผนก จึงไม่สามารถควบคุมข้อมูลไม่ให้เกิดความซ้ำซ้อนได้ ความซ้ำซ้อนของข้อมูลก่อให้เกิดค่าใช้จ่ายตามมา กล่าวคือ จะต้องเสียค่าใช้จ่ายเกี่ยวกับเงินและเวลาในการจัดเก็บข้อมูล จะต้องสูญเสียเนื้อที่ในการจัดเก็บข้อมูลจากทั้งสองแหล่ง และที่สำคัญ ความซ้ำซ้อนของข้อมูล จะก่อให้เกิดความผิดพลาดในข้อมูล (Data Anomalies) 3 ประการด้วยกัน (1) ข้อผิดพลาดจากการเพิ่มข้อมูล, (2) ข้อผิดพลาดจากการลบข้อมูล และ (3) ข้อผิดพลาดจากการเปลี่ยนแปลงข้อมูล

2.1.5.2 มีรูปแบบไม่ตรงกัน (Incompatible File Formats) เมื่อโครงสร้างของไฟล์ข้อมูลถูกตรึงหรือผูกติดด้วยภาษาโปรแกรม ตัวโครงสร้างก็จะขึ้นอยู่กับภาษาโปรแกรมที่ใช้เขียน เช่น โครงสร้างไฟล์ข้อมูลที่ถูกสร้างขึ้นด้วยภาษาโคบอล อาจมีความแตกต่างจาดโครงสร้างไฟล์ข้อมูลที่สร้างขึ้นด้วยภาษาซี และด้วยรูปแบบข้อมูลที่แตกต่างกัน จึงไม่สามารถนำมาใช้งานร่วมกันได้ (Incompatible) ซึ่งเป็นการยากสำหรับการนำไฟล์ทั้งสองประเภทมาประมวลผลร่วมกัน (น.35)

2.1.5.3 รายงานต่างๆ ถูกสร้างขึ้นแบบคงที่ตายตัว (Fixed Queries) เนื่องจากระบบแฟ้มข้อมูลขึ้นอยู่กับโปรแกรมที่ใช้เขียนเป็นอย่างมาก ดังนั้นรายงาน หรือการสืบค้นต่างๆ ผู้ใช้งานทั่วไปจะไม่สามารถกระทำได้ด้วยตนเอง เนื่องจากจำเป็นต้องเข้าใจภาษาโปรแกรมที่ใช้เขียน ซึ่งเป็นเรื่องยากเกินความสามารถสำหรับผู้ใช้งานทั่วไป ดังนั้นรายงานเหล่านี้จะถูกพัฒนาโดยโปรแกรมเมอร์แบบเสร็จสรรพ กล่าวคือ ผู้ใช้จะไม่สามารถโต้ตอบกับระบบเพื่อทำการสอบถามรายงานเพิ่มเติมได้ด้วยตนเอง ทั้งนี้หากต้องการรายงานใหม่เพิ่มเติม ก็ต้องว่าจ้างโปรแกรมเมอร์เขียนโปรแกรมให้ นั่นหมายถึงต้องเสียค่าใช้จ่ายเพิ่ม (โอภาส เอี่ยมสิริวงค์, 2551:35)

2.1.6 ระบบฐานข้อมูล (Database Systems)

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า จากราบระเอียดของระบบแฟ้มข้อมูลที่กล่าวไว้ในข้างต้น ทำให้ได้ทราบถึงปัญหาต่างๆ ที่เกิดจากการประมวลผลด้วยวิธีแฟ้มข้อมูล ไม่ว่าจะเป็นด้านความซ้ำซ้อนในข้อมูล ความไม่ยืดยุ่น และความไม่คล่องตัวในหลายๆ ด้าน ต่อมาวิวัฒนาการด้านเทคโนโลยีการจัดระบบข้อมูลได้เกิดขึ้นใหม่ที่เรียกว่าระบบฐานข้อมูลโดยแนวคิดดังกล่าวสามารถนำมาแก้ไขปัญหาที่เกิดขึ้นจากวิธีแฟ้มข้อมูลได้เป็นอย่างดี

ปกติในระบบแฟ้มข้อมูล ข้อมูลจะถูกจัดเก็บแบบกระจายไปตามหน่วยงานหรือแผนกต่างๆ ซึ่งแต่ละแผนกต่างก็มีกระบวนการจัดเก็บแฟ้มข้อมูลเป็นของตนเอง โดยหากมีผู้ใดผู้หนึ่งในแผนกได้มีการเข้าถึงข้อมูลเพื่อใช้งานในขณะนั้น บุคคลอื่นในแผนกจะไม่สามารถเข้าถึงข้อมูลชุดเดียวกันในช่วงเวลา

ดังกล่าวได้ แต่สำหรับแนวคิดของฐานข้อมูลจะตรงกันข้ามกับระบบแฟ้มข้อมูลโดยสิ้นเชิง กล่าวคือ ฐานข้อมูลคือศูนย์รวมข้อมูลต่างๆ ที่มีความสัมพันธ์กัน (Relationship) โดยจะมีกระบวนการจัดหมวดหมู่ข้อมูลอย่างมีระบบเบียบแบบแผน ก่อให้เกิดฐานข้อมูลที่เป็นแหล่งรวมของข้อมูลจากแผนกต่างๆ ซึ่งถูกจัดเก็บไว้อย่างมีระบบภายในฐานข้อมูลชุดเดียว โดยผู้ใช้งานแต่ละแผนกสามารถเข้าถึงข้อมูลส่วนกลางนี้เพื่อนำไปประมวลผลร่วมกันได้ และการที่มีศูนย์กลางข้อมูลเพียงแหล่งเดียว รวมถึงความสามารถในการเข้าถึงข้อมูลเพื่อให้งานร่วมกันได้จะช่วยแก้ปัญหาความซ้ำซ้อนของข้อมูล และที่สำคัญ ข้อมูลในฐานข้อมูลจะไม่ผูกติดกับโปรแกรม กล่าวคือ จะมีความอิสระในข้อมูล (Program-Data Independence)

และด้วยแนวคิดของฐานข้อมูลนี้ จึงทำให้สามารถแก้ไขปัญหาต่างๆ ที่เกิดขึ้นจากการประมวลผลด้วยวิธีแฟ้มข้อมูลได้เป็นอย่างดี แต่อย่างไรก็ตาม แนวคิดของระบบฐานข้อมูลนี้ ผู้ใช้จำเป็นต้องเรียนรู้ถึงกระบวนการจัดการและต้องพึ่งพาผู้เชี่ยวชาญ หรือผู้ที่มีประสบการณ์ด้านฐานข้อมูลเป็นอย่างดี (โอภาส เอี่ยมสิริวงค์, 2551:35)

2.1.7 บทบาทหน้าที่ของบุคลากรในระบบฐานข้อมูล (Roles in the Database Systems)

โอภาส เอี่ยมสิริวงค์ (2551) กล่าวว่า บทบาทหน้าที่ของบุคลากรต่างๆ ในระบบฐานข้อมูลสามารถแบ่งกลุ่มผู้ใช้ได้ดังต่อไปนี้

2.1.7.1 ผู้บริหารข้อมูลและผู้บริการฐานข้อมูล (Data and Database Administrators) ฐานข้อมูลและ DBMS จัดเป็นทรัพยากรที่ต้องนำมาใช้งานร่วมภายในองค์กร นั่นหมายความว่าจะต้องได้รับการจัดสรรให้กับผู้ใช้งานอย่างมีประสิทธิภาพ ดังนั้น ผู้บริหารข้อมูล (Data Administrators: DA) และผู้บริหารฐานข้อมูล (Database Administrators: DBA) จึงเป็นบุคคลหนึ่งที่มีหน้าที่รับผิดชอบเกี่ยวกับการจัดการทรัพยากรข้อมูล ซึ่งประกอบด้วย การว่างแผนฐานข้อมูล การพัฒนาและการบำรุงรักษาการกำหนดนโยบายและขั้นตอนการปฏิบัติงาน และรวมถึงการออกแบบฐานข้อมูลในระดับแนวคิด

2.1.7.2 นักวิเคราะห์และโปรแกรเมอร์ (Systems Analysis and Programmers) โปรแกรมเมอร์จะเขียนโปรแกรมตามข้อกำหนดที่นักวิเคราะห์ระบบได้ออกแบบไว้ โดยแต่ละโปรแกรมจะบรรจุด้วยชุดคำสั่งต่างๆ ที่จัดการกับ DBMS เพื่อปฏิบัติการกับข้อมูลในฐานข้อมูล เช่น การเรียกดูข้อมูล การเพิ่ม การปรับปรุง และการลบข้อมูล เป็นต้น ซึ่งโปรแกรมที่เขียนขึ้นอาจเขียนด้วยภาษาระดับสูง เป็นภาษาแบบรุ่นที่ 3 (3GL) หรือภาษายุคที่ 4 (4GL) ก็ได้ เมื่อโปรแกรมเมอร์เขียนโปรแกรมและทำการทดสอบเป็นที่เรียบร้อย ก็จะนำโปรแกรมไปติดตั้งเพื่อให้ยูสเซอร์ใช้งานต่อไป (น.44)

2.1.7.3 ผู้ใช้ปลายทาง (End-Users) ผู้ใช้ปลายทาง คือผู้ปฏิบัติงานกับโปรแกรมเพื่อใช้งานประจำวัน ซึ่งผู้ใช้ปลายทางอาจเป็นผู้ใช้ที่ปฏิบัติงานบนโปรแกรมที่ถูกพัฒนาเรียบร้อยแล้วโดย

Computer Science Department
Faculty of Informatics, Mahasarakham University

โปรแกรมเมอร์ หรืออาจเป็นผู้ใช้ที่มีความสามารถจัดการกับข้อมูลได้บางอย่าง เช่น สามารถสอบถามข้อมูลเพิ่มเติมจากฐานข้อมูลได้

2.1.8 การประยุกต์ใช้งานฐานข้อมูล (The Range of Database Applications)

โอภาส เอี่ยมสิริวงศ์ (2551) กล่าวว่า การประยุกต์ใช้งานฐานข้อมูลในที่นี้หมายถึง การนำฐานข้อมูลมาประยุกต์ใช้งานในระดับต่างๆ ซึ่งตั้งอยู่บนพื้นฐานการออกแบบเพื่อใช้งานตามความเหมาะสมหรือบนขนาดของธุรกิจเป็นสำคัญ โดยสามารถแบ่งประเภทการประยุกต์ใช้งานฐานข้อมูลได้ดังต่อไปนี้

2.1.8.1 ฐานข้อมูลส่วนบุคคล (Personal Database) ฐานข้อมูลส่วนบุคคลถูกออกแบบมาเพื่อสนับสนุนการใช้งานของผู้ใช้งานเพียงหนึ่งคน ซึ่งสามารถพบได้บนการใช้งานของเครื่องพีซีคอมพิวเตอร์ โน้ตบุคคอมพิวเตอร์ และล่าสุดก็คือ คอมพิวเตอร์แบบพกพาหรือที่เรียกว่าพีดีเอ (Personal Digital Assistants: PDA) และใช่ว่าจะสามารถนำคอมพิวเตอร์มาใช้งานได้เท่านั้น แต่ด้วยเทคโนโลยีโทรศัพท์มือถือที่ใช้งานปัจจุบัน ที่มีฟังก์ชั่นการใช้งานใกล้เคียงคอมพิวเตอร์เข้าไปทุกวัน สามารถนำมาใช้เพื่อการจัดเก็บข้อมูลและสื่อสารผ่านเว็บ สำหรับตัวอย่างการประยุกต์ใช้ฐานข้อมูลส่วนบุคคลอย่างง่าย เช่น พนักงานขายได้มีการจัดเก็บข้อมูลลูกค้า และบันทึกข้อมูลรายละเอียดการเข้าพบลูกค้าในแต่ละวันลงในคอมพิวเตอร์โน้ตบุ๊คหรือเครื่องพีดีเอ เป็นต้น (น.45)

2.1.8.2 ฐานข้อมูลระดับแผนก (Department Databases) แผนกคือหน่วยงานที่มีหน้ารับผิดชอบภายในองค์กร เช่น แผนกบุคลากร แผนกขาย แผนกผลิต และแผนกบัญชี เป็นต้น ปกติแล้วแผนกจะมีขนาดใหญ่กว่าเวิร์กกรุ๊ป ซึ่งจะมีจำนวนคนตั้งแต่ 25-100 คนโดยแต่ละแผนกมีหน้าที่รับผิดชอบที่แตกต่างกัน

2.1.8.3 ฐานข้อมูลอินเตอร์เน็ต (Internet Databases) ผลกระทบต่อสภาแวดล้อมของฐานข้อมูลที่เกิดขึ้นไม่นานมานี้ก็คือระบบอินเตอร์เน็ต ซึ่งเป็นเครือข่ายที่มีการโยงใยกันทั่วโลก ดังนั้นฐานข้อมูลบนเว็บ (Web-Enabled Database) จึงเป็นสิ่งจำเป็น และเป็นที่มาของรูปธุรกิจแบบพาณิชย์อิเล็กทรอนิกส์ (E-Commerce) ไม่ว่าจะเป็น B-to-C (Business to Consumer) วึ่งเป็นรูปแบบธุรกิจค้าขายสินค้าและบริการจากภาคธุรกิจไปยังกลุ่มผู้บริโภคโดยตรง รวมถึงรูปแบบ B-to-B (Business To Business) ที่ภาคธุรกิจสามารถติดต่อกับภาคธุรกิจอื่นๆ ที่เป็นเครือข่ายพันธมิตรทางธุรกิจระหว่างกัน ทำให้ประหยัดต้นทุนลงได้มาก ดังนั้นจะเห็นได้ว่า ในปัจจุบันมีหลายบริษัทได้มีการนำเทคโนโลยีอินเทอร์เน็ตมาเชื่อมโยงกับฐานข้อมูลภายในองค์กร ทั้งนี้อาจมีการจำกัดการเข้าถึงสำหรับบุคคลภายนอก ที่เรียกว่าเครือข่ายอินที่เน็ต (Intranet) ที่อนุญาตให้เฉพาะบุคคลภายในใช้งานเท่านั้น

Computer Science Department
Faculty of Informatics, Mahasarakham University

โดยมีไฟร์วอล์ (Firewall) ช่วยในการกลั่นกรองผู้บุกรุก กรณีมีผู้ไม่หวังดีพยายามเข้าฐานข้อมูลภายในองค์กรโดยมิชอบ ทั้งนี้ก็เพื่อความปลอดภัยในข้อมูล

2.1.9 ข้อดีของวิธีฐานข้อมูล (Advantages of Database Approach)

โอภาส เอี่ยมสิริวงค์ (2551) ได้กล่าวถึงข้อดีของวิธีฐานข้อมูล (Advantages of Database Approach) ดังต่อไปนี้

2.1.9.1 ลดความซ้ำซ้อนในข้อมูล (Minimal Data Redundancy) ต้องเป็นที่เข้าใจว่าวัตถุประสงค์ของระบบฐานข้อมูลก็คือ การขจัดความซ้ำซ้อนในข้อมูลที่มีการเก็บแยกต่างหากออกจากกัน โดยให้ไฟล์เหล่านั้นรวมกันอยู่แหล่งเดียว และด้วยข้อมูลที่จัดเก็บไว้ในศูนย์กลางเพียงแหล่งเดียวนี้เอง จึงทำให้ควบคุมข้อมูลได้ง่าย ไม่ว่าจะเป็นการปรับปรุงหรือแก้ไขข้อมูล ก็จะดำเนินการแก้ไขที่แหล่งเดียว

2.1.9.2 การใช้ข้อมูลร่วมกัน (Improved Data Sharing) ฐานข้อมูลถูกออกแบบมาเพื่อแบ่งปั่นการใช้งานทรัพยากรร่วมกัน แม้ว่าข้อมูลจะถูกจัดเก็บไว้ที่ศูนย์กลางเพียงแหล่งเดียว แต่ในมุมมองการใช้งานของผู้ใช้งาน จะเสมือนว่าข้อมูลเหล่านั้นอยู่ที่แผนกของตนถึงแม้จำเป็นต้องใช้ข้อมูลจากแผนกอื่นก็ตาม แต่ในความเป็นจริงข้อมูลนั้นมีเพียงชุดเดียว แต่สามารถแชร์การใช้งานร่วมกันได้ต่างหาก ดังนั้นไม่ว่าผู้ใช้งานจะอยู่แผนกใดก็ตาม ก็สามารถเข้าถึงข้อมูลทั้งที่เป็นส่วนของตน รวมถึงการเชื่อมโยงข้อมูลส่วนงานตนไปยังข้อมูลของแผนกอื่นๆ (น.49)

2.1.9.3 เพิ่มคุณประโยชน์ในการพัฒนาโปรแกรมประยุกต์ (Increased Productivity of Application Development) ประการหนึ่งที่สำคัญของการประมวลผลข้อมูลด้วยวิธีฐานข้อมูลก็คือการลดต้นทุนด้านเวลาในการพัฒนาโปรแกรมประยุกต์ต่างๆ ในธุรกิจ โดยมีผลสำคัญ 2 ประการที่ทำให้การพัฒนาโปรแกรมประยุกต์สามารถเสร็จสิ้นได้ด้วยระยะเวลาอันสั้น ซึ่งประกอบด้วย

2.1.10 ข้อเสียของวิธีฐานข้อมูล (Disadvantages of Database Approach)

โอภาส เอี่ยมสิริวงค์ (2551) ได้กล่าวถึงข้อเสียของวิธีฐานข้อมูล (Disadvantages of Database Approach) ดังต่อไปนี้

2.1.10.1 ความซับซ้อนที่เพิ่มขึ้น (Complexity) ต้องยอมรับว่าเทคโนโลยีฐานข้อมูลนั้นมีความสลับซับซ้อนมากกว่าเทคโนโลยีแบบไฟล์ โดบเฉพาะ DBMS มีประสิทธิภาพสูง มักจะมีความซับซ้อนสูงตามมาเช่นกัน ดังนั้นในด้านของทีมงาน ไม่ว่าจะเป็นนักออกแบบฐานข้อมูลโปรแกรมเมอร์ ผู้บริหารฐานข้อมูล และรวมถึงผู้ใช้จะต้องทำความเข้าใจกับบทบาทหน้าที่ที่ตนรับผิดชอบ ต้องเรียนรู้ขั้นตอนปฏิบัติ การใช้ชุดคำสั่งในการจัดการอย่างลึกซึ่งเพื่อให้การใช้งาน DBMS ก่อประโยชน์สูงสุด

2.1.10.2 มีขนาดความจุเพิ่มขึ้น (Size) เมื่อความซับซ้อนในซอฟต์แวร์ DBMS มีมากขึ้นรวมทั้งประสิทธิภาพของ DBMS สูงขึ้น ย่อมทำให้ขนาดของ DBMS ใหญ่โตขึ้นโดยปริยาย ดังนั้นจึง

จำเป็นต้องใช้ความจุพื้นที่ในดิสก์หลายเมกะไบต์หรือหลายกิกะไบต์ รวมถึงขนาดของหน่วยความจำหลักที่ต้องมีมากขึ้น เพื่อให้ DBMS สามารถรันงานได้อย่างเต็มประสิทธิภาพ

2.2 ภาษา Dart (Dart Language)

เนื้อหาของ ภาษา Dart **(Dart Language)** ดังต่อไปนี้อ้างอิงมาจากเว็บไซต์: Tanapoj Chaivanichanan. 2020. ทำความรู้จักภาษา Dart ฉบับโปรแกรมเมอร์. https://www.centrilliontech.co.th/blog/2570/dart-101-introduction-for-programmers/

ปี 2011 กูเกิลได้เปิดตัวภาษาโปรแกรมตัวใหม่ชื่อว่าภาษา Dart โดยโครงสร้างของภาษา Dart คล้ายกับ C/C++ และ Java โดยที่จะมีความเป็นภาษาแบบ Structure Programming แต่ก็ยังมีความสามารถแบบภาษาประเภท Object Oriented Programming ด้วย นั่นคือมี Class และ Inheritance ให้ใช้งาน เป้าหมายของการสร้างภาษา Dart ขึ้นมา กูเกิลบอกได้อธิบายว่าอยากสร้างภาษาเชิงโครงสร้างที่ยืดหยุ่นมากพอ (Structured Yet Flexible Language) และเป็นการออกแบบตัวภาษาไปพร้อมกับตัว Engine สำหรับรันภาษาเพื่อแก้ปัญหาโปรแกรมทำงานช้าและกินหน่วยความจำ ซึ่งเป้าหมายของภาษา Dart คือเป็นภาษาที่เรียนรู้ง่าย และทำงานได้บนอุปกรณ์พกพาขนาดเล็กมือถือไปจนถึงเซิฟเวอร์ซึ่งสิ่งที่เด่นที่สุดสำหรับภาษา Dart ในตอนนี้คือเป็นภาษาที่ใช้ในการสร้าง Application ด้วยเฟรมเวิร์ก Flutter

2.2.1 เริ่มต้นเขียนโปรแกรม (Start with ‘Hello World!’)

ตัวอย่างโปรแกรมของ Dart นั้นหน้าตาคล้ายๆ กับภาษา C หากผู้อ่านได้เคยเขียนภาษา C หรือภาษาตระกูล C มาก่อน (เช่น C++, C#, Java) จะคุ้นกับ Syntax พวกนี้ทำให้เรียนรู้ได้ไม่ยาก Dart เป็นภาษากลุ่ม Compiler นั่นคือจำต้อง Compile ก่อนเอาโปรแกรมไปรัน ไม่เหมือนภาษากลุ่ม Script ที่ใช้ interpreter ในการรันตัว Source Code โดยตรง

ตัวโปรแกรมจะเริ่มทำงานที่ฟังก์ชัน main เป็นหลัก โดยไม่สามารถเขียน statement นอกฟังก์ชันได้ การแสดงผลมาตราฐานจะใช้คำสั่ง print (คำสั่งนี้จะ auto-newline เสมอ) เรื่องหนึ่งที่ควรจำคือภาษา Dart นั้นการเขียน ; (semi-colon) ไม่ใช่ optional คือจำเป็นต้องใส่ ; ทุกครั้งหลังจบ statement ไม่สามารถละ ; ได้แบบภาษาตระกูล C ยุคใหม่ๆ เช่น JavaScript หรือ Kotlin

2.2.2 คอมเมนท์ (Comment) การใส่คอมเมนท์ทำได้เหมือนภาษา C ทุกประการคือ

```
* สำหรับ Inline Comment
เปิดด้วย /* และปิดด้วย */ สำหรับ Multi-line Comment


int x; //ตั้งแต่ตรงนี้ไป เป็นส่วนของคอมเมนท์
/* ในนี้ทั้งหมดเป็นคอมเมนท์ */
```

ภาพประกอบที่ 2.1 การใส่ Comment ในภาษา Dart

Computer Science Department Faculty of Informatics Mahasarakham University

2.2.3 ตัวแปรและชนิดข้อมูล (Variable and Data Type)

ตารางที่ 2.1 ตัวแปรและชนิดข้อมูล

type	**คำอธิบาย**	**ตัวอย่าง**
int	เลขจำนวนเต็ม	0, 1, -5, 86400
double	เลขทศนิยม	0.0, 0.1, 0.14, -12.34
num	เลขทศนิยม หรือ เลขทศนิยม	123, 0.123
bool	ค่าทางตรรกศาสตร์	true, false
string	สายอักขระ (ประโยค)	'hello world!', "This is a book" ในภาษา Dart สามารถใช้ได้ทั้ง " (double quote) และ ' (single quote)
dynamic	ตัวแปรชนิดเปลี่ยนแปลงได้	1, 0.14, true, 'Hi!'

```
int x;
double d;
bool isDone;
String name;

//ตัวแปรทั้งหมดมีค่าเป็น null เพราะยังไม่ได้กำหนดค่า
```

ภาพประกอบที่ 2.2 ตัวแปรและชนิดข้อมูลของ Dart

ตารางที่ 2.2 ตัวอย่างตัวแปรและชนิดข้อมูล

Type	**ตัวอย่าง**
var	เป็นการละ type เอาไว้ให้โปรแกรมกำหนดให้ (ตาม value)
final	เหมือน var แต่ไม่สามารถเปลี่ยนแปลงค่าได้
const	ค่าคงที่

2.2.4 เครื่องหมายดำเนินการคณิตศาสตร์ (Math Operation)

การใช้ +, -, *, / และ % เหมือนกับภาษาอื่นๆ แต่มีข้อควรระวังที่ตัว / สำหรับภาษาอื่นถ้าเรานำ int / int ผลที่ออกมาจะได้เป็น int แน่นอน แต่สำหรับ Dart นั้นการหารจะได้ค่าออกมาเป็น double เสมอ

Computer Science Department Faculty of Informatics, Mahasarakham University

```
int x = 4 / 2;
// Error: A value of type 'double' can't be assigned to a variable of type
'int'.

int y = (int)(4 / 2);
// Error: case แบบภาษา C ไม่ได้ด้วยนะ
```

ภาพประกอบที่ 2.3 เครื่องหมายดำเนินการคณิตศาสตร์ในภาษา Dart

วิธีการแก้คือใช้ operation ~/ คือการหารแล้วปัดจุดทิ้ง หรือใช้คำสั่ง as หรือจะใช้คำสั่ง toInt() ก็ได้

```
int x = 4 ~/ 2;          // Ok! แบบนี้ได้
int x = 4 / 2 as int;    // Ok! แบบนี้ได้
int x = (4 / 2).toInt();       // Ok! แบบนี้ได้
```

ภาพประกอบที่ 2.4 การแก้เครื่องหมายดำเนินการคณิตศาสตร์ในภาษา Dart

2.2.5 การต่อสตริง (String Concatenate)

การต่อสตริงใช้เครื่องหมาย + เหมือนภาษาทั่วๆ ไป แต่ก็มีข้อควรระวัง คือไม่สามารถต่อสตริงกับตัวแปรที่ไม่ใช่สตริงได้!

```
int x = 100;
print('x is ' + x);
// Error: A value of type 'int' can't be assigned to a variable of type 'String'.
```

ภาพประกอบที่ 2.5 การต่อสตริงในภาษา Dart

เราจะต้องแปลงตัวแปรที่ต้องการจะต่อสตริงให้เป็น String ก่อน หรือทำ String Interpolation ก่อน โดยใช้ตัว **$** เพื่อระบุว่าตรงนี้เป็นตัวแปร (ถ้ามี Expression ให้ครอบด้วย ${})

```
int x = 100, y = 200;
print('x is $x');        // x is 100
print('x is ${x}');          // x is 100
print('x + y is ${x + y}');   // x + y is 300
```

ภาพประกอบที่ 2.6 การแปลงตัวแปรที่ต้องการจะต่อสตริงให้เป็น String ในภาษา Dart

2.2.6 การจัดการกับค่าว่าง (Null Handling)

ตัวแปรใน Dart เป็นแบบ reference ดังนั้นเลยสามารถเป็นค่า null ได้ทุกตัวเลย ภาษาDart เลยมี operation สำหรับจัดการค่า null พวกนี้มาให้เราใช้งานด้วย

2.2.6.1 กำหนดค่าว่างโดยใช้การรวมตัวกัน (?? Null Coalescing) เป็นการเช็กว่าตัวแปรตัวนี้ ถ้ามีค่าเป็น null ให้ใช้ค่า default ที่กำหนดให้แทน

```
output = input ?? defaultValue;

// เป็น short-hand ของ...
if (input != null) {
      output = input;
} else {
      output = defaultValue;
}
```

ภาพประกอบที่ 2.7 การกำหนดค่าว่างในภาษา Darrt

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.2.6.2 กำหนดค่าว่างโดยใช้เงื่อนไข (?. Null Conditional) ซึ่งสามารถเรียกใช้งาน method ต่างๆ ได้ ... แต่ถ้า object ตัวนั้นเป็น null ก็จะเกิดปัญหา Null Pointer Exception ได้

```
object?.action();

// เป็น short-hand ของ...
if (object != null) {
      object.action();
}
```

ภาพประกอบที่ 2.8 การกำหนดค่าว่างโดยใช้เงื่อนไขในภาษา Dart

2.2.6.3 กำหนดค่าว่างโดยใช้การมอบหมายค่ารวมตัวกัน (??= Null Coalescing Assignment) หากไม่แน่ใจว่าตัวแปรตัวนั้นเป็น null รึเปล่า สามารถใช้ ??= กำหนดค่า default

```
variable ??= defaultValue

// เป็น short-hand ของ...
variable = variable ?? defaultValue;

// หรือใช้ Ternary Operator
variable = variable != null ? variable : defaultValue;

// หรือเขียนแบบ if-else
if (input == null) {
      output = defaultValue;
}
```

ภาพประกอบที่ 2.9 การกำหนดค่าว่างโดยใช้การมอบหมายค่ารวมตัวกันในภาษา Dart

2.2.7 การควบคุมการไหล (Flow Control)

2.2.7.1 เงื่อนไขแบบ if-else

```
if (condition) {
      ….
} else {
      ….
}
```

ภาพประกอบที่ 2.10 เงื่อนไขแบบ if-else ในภาษา Dart

2.2.7.2 เงื่อนไขแบบ switch case

```
switch (command) {
      case 'PENDING':
            executePending();
            break;
      case 'APPROVED':
            executeApproved();
            break;
      case 'DENIED':
            executeDenied();
            break;
      default:
            executeUnknown();
}
```

ภาพประกอบที่ 2.11 เงื่อนไขแบบ switch case ในภาษา Dart

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.2.7.3 การวนซ้ำแบบ while, do-while

```
while (!isDone()) {
      doSomething();
}

do {
      printLine();
} while (!atEndOfPage());
```

ภาพประกอบที่ 2.12 การวนซ้ำแบบ while, do-while ในภาษา Dart

2.2.7.4 การวนซ้ำแบบ for

```
for (var i = 0; i < 5; i++) {
      print(i);
} // output: 0 1 2 3 4
```

ภาพประกอบที่ 2.13 การวนซ้ำแบบ for ในภาษา Dart

2.2.8 ฟังก์ชัน (Function)

*** การสร้างฟังก์ชันในภาษา Dart มี syntax เหมือนภาษา C แต่สามารถละ type ทิ้งไปได้ เช่น**

```
int add(int x, int y) {
      return x + y;
}

// สามารถเขียนย่อได้ว่า
add(x, y) {
      return x + y;
}
```

ภาพประกอบที่ 2.14 การสร้างฟังก์ชันในภาษา Dart

2.2.8.1 ฟังก์ชันลูกศร (Arrow Function) และหากเคยเขียนภาษา JavaScript มา มีหลายครั้งที่สร้างฟังก์ชันที่มี return statement เดียวเท่านั้น ก็สามารถเขียนย่อโดยใช้ Arrow Function ได้ ... และแน่นอน Dart ก็ทำได้เหมือนกัน โดยใช้ =>

```
int add(int x, int y) {
      return x + y;
}

// สามารถเขียนย่อได้ว่า
add(x, y) => x + y;
```

ภาพประกอบที่ 2.15 การสร้างฟังก์ชันลูกศรในภาษา Dart

Computer Science Department
Faculty of Informatics Mahasarakham University

2.2.8.2 ฟังก์ชันแบบมีพารามิเตอร์เสริม (Optional Parameter Function) โดยสามารถกำหนดค่าเริ่มต้นให้ parameter ได้โดยใช้ [] ครอบ parameter ที่อยากประกาศให้เป็น optional

```
int add(int x, [int y = 1]) {
      return x + y;
}
add(10, 20);       // result: 30
add(10);           // ไม่เซ็ตค่า y, ดังนั้น y = 1 result: 11
```

ภาพประกอบที่ 2.16 การสร้างฟังก์ชันแบบมีพารามิเตอร์เสริมในภาษา Dart

2.2.8.3 ฟังก์ชันแบบกำหนดพารามิเตอร์ (Named Parameter) บางกรณีการสร้างฟังก์ชันที่มี parameter เยอะมาก ตอนที่เรียกใช้ฟังก์ชันอาจจะสับสนเรื่องลำดับตัวแปรก็เป็นได้

```
int setConfig(
      String basePath,
      String appPath,
      int retry,
      int maxThread,
      String defaultController
) {
      // TODO
}

setConfig("/", "/app", 10, 4, "Main");
```

ภาพประกอบที่ 2.17 การสร้างฟังก์ชันแบบกำหนดพารามิเตอร์ในภาษา Dart

2.2.9 คราสโดยฟังก์ชัน (First Class Function)

```
* ตามสไตล์ภาษาสมัยใหม่ เราสามารถจับฟังก์ชันใส่ตัวแปรได้
int getNumber() => 123;

void main(){
      var func = getNumber;          // ไม่ใช่ getNumber() นะ,ไม่มี ()
      print(func());                 // output: 123
}
```

ภาพประกอบที่ 2.18 การสร้างคราสโดยฟังก์ชันในภาษา Dart

หรือจะกำหนดว่าตัวแปรฟังก์ชันจะเป็น Type อะไรและมี Parameter อะไรบ้างก็ได้ โดยใช้รูปแบบการกำหนด Type ดังนี้ *return-type* Function(params-type)

```
void func1(){ ... }
int func2(){ ... }
String func3(int x){ ... }

void main(){
      void Function() f1 = func1;
      int Function() f2 = func2;
      String Function(int) f3 = func3;
}
```

ภาพประกอบที่ 2.19 การสร้างคราสโดยตัวแปรในภาษา Dart

Computer Science Department
Faculty of Informatics, Mahasarakham University

และสามารถใช้ได้กับ method ได้อีกด้วย เช่น

```
class People{
      String sayHi() => "Hi!";
}

void main(){
      People p = People();
      String Function() f = p.sayHi;
      print(f());     // output: Hi!
}
```

ภาพประกอบที่ 2.20 การสร้างคราสโดยตัวแปรกับ Method ในภาษา Dart

2.2.9.1 สรุป (Conclude) เนื้อหานี้ได้แนะนำภาษา Dart ซึ่งจะเห็นว่าตัวภาษานั้นให้อารมณ์เหมือนภาษาตระกูล C ที่มีการปรับอะไรให้เป็นภาษาสมัยใหม่มากขึ้น แต่ก็ยังไม่ทิ้งความเป็น Structure Language อยู่หากผู้อ่านเขียน C หรือหรือแม้แต่ภาษาที่เคยโมเดิร์นอย่าง Java มาก่อน จะพบว่าตัวภาษามีความคล่องตัวในการเขียนมากขึ้น แต่ถ้าเอาไปเทียบกับภาษายุคใหม่อย่างเช่น Kotlin, Swift ก็ยังเรียกว่าสู้ไม่ได้ อาจจะมีอะไรขัดใจอยู่หลายอย่างในขณะที่เขียน แต่ในการพัฒนาแอปพริเคชันบนมือถือโดยใช้เฟรมเวิร์คอย่าง Flutter ความรู้พื้นฐานของภาษา Dart นั้นจะมีบทบาทในการพัฒนามากเลยทีเดียว

2.3 หลักการออกแบบ UX และ UI (UX and UI Design Principles)

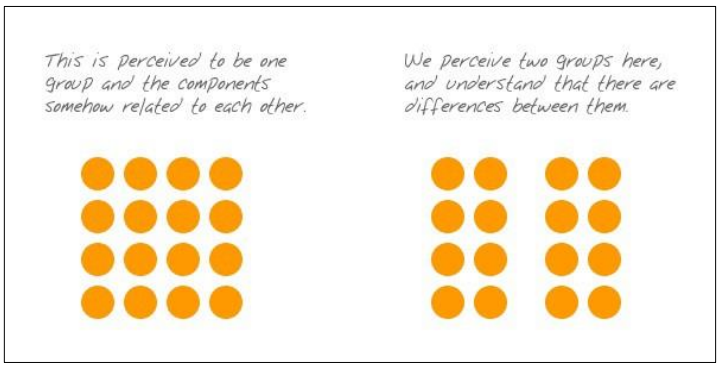
เนื้อหาของ **หลักการออกแบบ UX และ UI (UX and UI Design Principles)** ดังต่อไปนี้ อ้างอิงมาจากเว็บไซต์: Srikate. 2020. Design Principles — เข้าใจการออกแบบ UI ฉบับ developer รัก designer นะ. https://medium.com/@srikatekuersirikul/design-principle—เข้าใจการออกแบบ-ui-ฉบับ-developer-รัก- designer-นะ-dc172521824a

ในปัจจุบันอุตสาหกรรมการผลิตซอฟต์แวร์เริ่มให้ความสำคัญกับการทำงานเป็นทีม และการสื่อสารกันระหว่างคนในโปรเจคมากขึ้น คงหลีกเลี่ยงไม่ได้ที่นักพัฒนาจะได้คุยกับ นักออกแบบ UX/UI โดยตรงหรือแม้กระทั่งบางครั้งก็อาจจะจำเป็นต้องกลายเป็นนักออกแบบไปเสียเอง เมื่อมีการคุยกันมากขึ้น แน่นอนว่าการเข้าใจงานของคนที่เรากำลังคุยด้วยเป็นสิ่งสำคัญมากที่จะช่วยให้งานสำเร็จตามนิยามของคำว่า “ทีม” ได้อย่างแท้จริง

2.3.1 หลักการออกแบบ (Design Principles)

ถ้าพูดถึงหลักการที่ช่วยลดการทำงานของสมองในการตัดสินใจของมนุษย์ เช่น การจัดกลุ่ม Content ต่างๆ ใน UI เพื่อให้คนที่ใช้งานรู้ว่าส่วนไหนสำคัญ ส่วนไหนที่เกี่ยวข้องกัน โดยใช้เวลาให้น้อยที่สุด ประกอบไปด้วย 4 หัวข้อหลักๆ ได้แก่ Repetition , Proximity, Contrast และ Alignment ดังต่อไปนี้

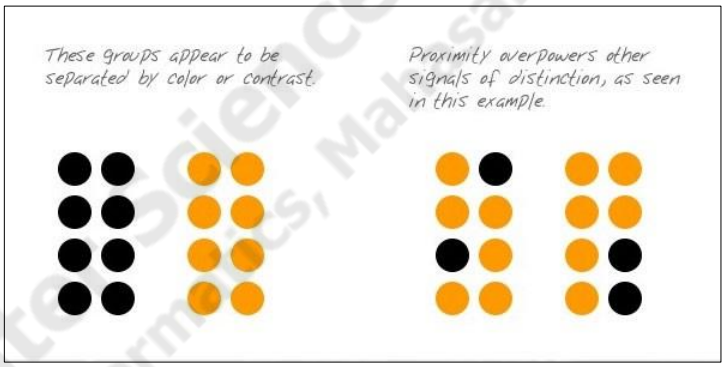
Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 2.21 Repetition & Proximity

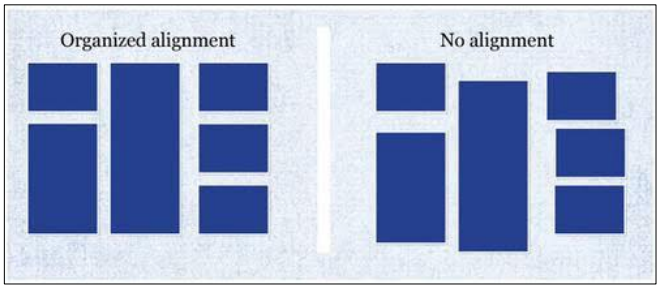
2.3.1.1 การแยกด้วยความเหมือน (Repetition) จากรูปด้านซ้าย เราจะมองเห็นว่าวัตถุวงกลมนั้นมันเป็นกลุ่มก้อนเดียวกัน

2.3.1.2 การแยกด้วยช่องว่าง (Proximity) จากรูปด้านขวา สามารถทำเพียงแค่เว้นช่องว่างตรงกลางออกเป็นสองก้อน จากวงกลมที่เคยเป็นกลุ่มเดียวกันหรือพวกเดียวกัน ก็ถือว่าถูกแยกออกมาเป็นคนละก้อนแล้ว



ภาพประกอบที่ 2.22 Contrast & Proximity

2.3.1.3 การแยกด้วยความต่าง (Contrast) จากรูปด้านบน เมื่อใส่สีที่แตกต่างกันระหว่างสองกลุ่ม ก็จะยิ่งเห็นชัดเจนว่าทั้งสองนั้นอยู่คนละกลุ่มกัน หรือถ้าจะนำช่องว่างออก การแบ่งกลุ่มโดยใช้สี ก็ยังคงทำให้เราสามารถรับรู้ได้ว่าในรูปภาพด้านซ้ายนั้นถูกแบ่งออกเป็น 2 กลุ่มอยู่ดี นี่คือพลังของ Contrast นั่นเอง



ภาพประกอบที่ 2.23 Alignment

Computer Science Department
Faculty of Informatics, Mahasarakham University

2.3.1.4 การแยกกันด้วยการวางให้เป็นแนวเดียวกัน (Alignment)

จากรูปด้านบนทางขวามือจะเห็นได้ว่าถ้าเราไม่ได้วางกล่องให้เรียงเป็นแนวเดียวกัน เราจะไม่รู้เลยว่าส่วนไหนที่เป็นกลุ่มก้อนเดียวกัน การมองให้สบายตา Alignment เป็นสิ่งสำคัญมากจริงๆ หากนักออกแบบหลงลืม การใช้ Alignment ออกแบบ UI มาให้เรา หลังจากรู้หลักการนี้นักพัฒนาก็คงแก้ UI เองได้อย่างง่ายดาย

2.4 การเปรียบเทียบระบบที่ใกล้เคียง

2.4.1 ลาซาด้า (Lazada)

สำหรับการออกแบบหน้าจอของแอปหน้าแรกจะเป็นหน้าไฮไลท์ที่แสดงข้อมูลต่าง ๆ ของสินค้าที่จัดกิจกรรมอยู่ สามารถเลื่อนหน้าจอซ้าย-ขวา เพื่อเลือกดูหมวดสินค้าได้ ซึ่งในส่วนนี้ทาง Lazada ได้ปรับปรุงระบบขึ้นใหม่ ทั้งทางเว็บและแอปฯ เพื่อให้การค้นหาสะดวกและมีหน้าตาสวยงามยิ่งขึ้น

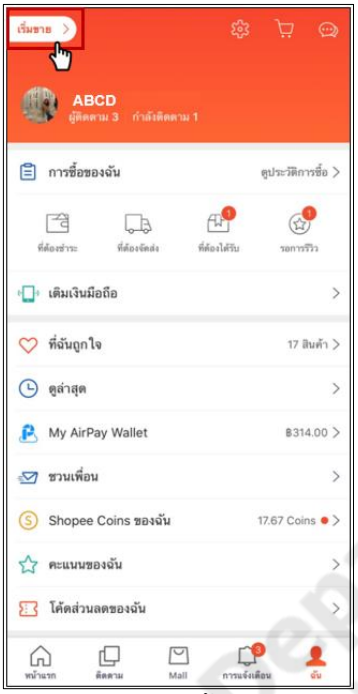
การทำงานของแอปนอกจากจะสามารถสั่งซื้อสินค้าออนไลน์ได้ง่ายแล้ว ยังมีทางเลือกในการชำระเงินที่หลากหลายและง่ายสำหรับผู้ใช้งาน และนอกจากสินค้าจะหลากหลายแล้วยังคุ้มครองผู้ซื้อถ้าไม่ได้รับสินค้าหรือสินค้ามีปัญหาก็สามารถขอรับเงินคืนได้ และอีกเหตุผลในการซื้อของออนไลน์ที่นอกจากจะประหยัดค่ารถและเวลาในการเดินทางไปซื้อของแล้ว ยังมีส่วนลดจากโปรโมชั่นต่างๆ

ภาพประกอบที่ 2.24 หน้าไฮไลท์ที่แสดงข้อมูลลาซาด้า (Lazada)

2.4.2 ช้อปปี้ (Shopee) ช้อปปิ้งออนไลน์แพลตฟอร์มที่จะเปิดโอกาสให้คุณเพิ่มพื้นที่ในการขายและเพิ่มรายได้ให้กับคุณ!

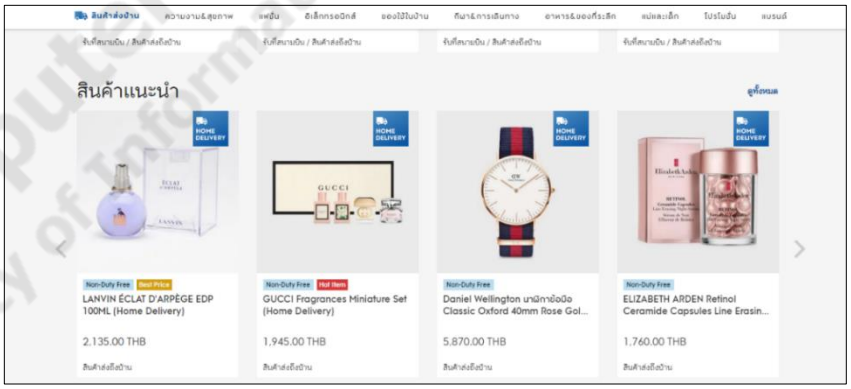
ก่อนอื่น คุณต้องมั่นใจว่าสินค้าของคุณจะต้องไม่อยู่ใน รายการสินค้าที่ห้ามขาย ใน Shopee และ คุณต้องทำการยืนยันด้วยหมายเลขโทรศัพท์ของคุณ ก่อนที่จะสามารถซื้อขายสินค้าใน Shopee ได้

Computer Science Department Faculty of Informatics Mahasarakham University



ภาพประกอบที่ 2.25 หน้าการซื้อของฉันช้อปปี้ (Shopee)

2.4.3 คิง เพาเวอร์ (King Power) กลุ่มบริษัท คิง เพาเวอร์ อินเตอร์เนชันแนล (King Power International Group) เป็นบริษัทด้านธุรกิจค้าปลีกสินค้าปลอดอากรของไทย ก่อตั้งในปี พ.ศ. 2532 โดยวิชัย ศรีวัฒนประภา ใช้ชื่อเดิมว่า บริษัท ดาวน์ทาวน์ ดี.เอฟ.เอส (ไทยแลนด์) จำกัด ได้ร่วมทุนกับ ททท. เปิดดำเนินกิจการร้านค้าปลอดอากรในเมืองเป็นรายแรกในประเทศไทย

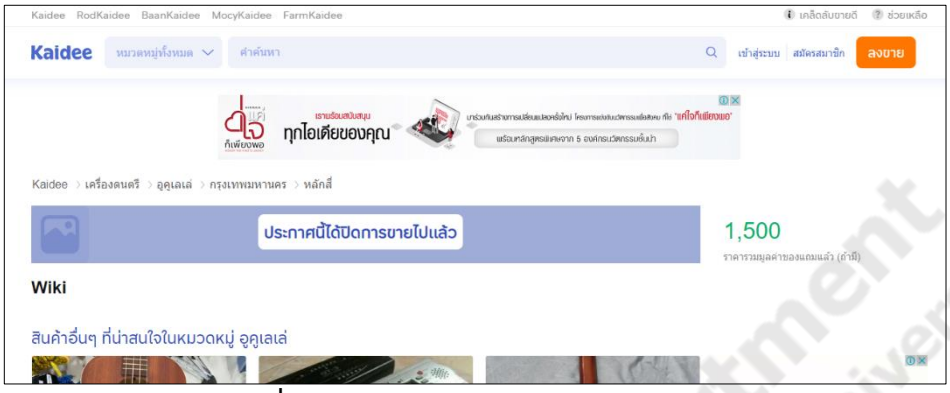


ภาพประกอบที่ 2.26 หน้าแนะนำสินค้า คิง เพาเวอร์ (King Power)

2.4.4 ขายดีดอทคอม (kaidee.com) หรือเดิมเป็นที่รู้จักกันในนาม OLX.co.th เป็นตลาดซื้อ-ขายของมือสองออนไลน์ โดยเราที่มุ่งมั่นพัฒนาแพลตฟอร์มที่ดีที่สุดสำหรับคนไทยสำหรับการซื้อ-ขายของมือสอง เพื่อให้ได้ใช้งานง่าย สะดวก รวดเร็ว ผ่านจำนวนผู้ซื้อและผู้ขายที่มีอยู่ทั่วประเทศ เป้าหมาย

Computer Science Department
Faculty of Information Technology Mahasarakham University

ของเราคือการเป็นส่วนหนึ่งในการช่วยยกระดับคุณภาพชีวิต ด้วยการช่วยให้ผู้ซื้อและผู้ขายมาพบปะกัน เพื่อแลกเปลี่ยนสินค้าในราคาที่ทั้งผู้ซื้อและผู้ขายพึงพอใจที่สุด



ภาพประกอบที่ 2.27 หน้าแรกของขายดีดอทคอม (kaidee.com)

2.4.5 Line Shop (ไลน์ช้อป) คือเครื่องมือที่ช่วยเพิ่มประสิทธิภาพทางด้านการขายสินค้าบน LINE OA ให้ง่ายและมีประสิทธิภาพมากขึ้นกว่าเดิม LINE My Shop มีฟีเจอร์ต่างๆ มากมายที่สร้างออกมาเพื่อแก้ปัญหาต่างๆ ของงานขาย ตัวอย่างเช่น การออกใบสั่งซื้อ การมีระบบแคตตาล็อกสินค้าให้ลูกค้าเลือก รวมไปถึงระบบชำระเงินที่ลูกค้าสามารถชำระเงินผ่าน LINE ได้โดยตรง การเริ่มต้นใชLINE My Shop ก็ง่ายและฟรี เพียงแค่เชื่อม LINE OA ของร้านค้า เข้ากับ LINE My Shop เท่านั้น การขายสินค้าผ่าน LINE ก็จะเป็นเรื่องง่ายทันที

ภาพประกอบที่ 2.28 หน้าแรกสินค้าของ Line Shop (ไลน์ช้อป)

Computer Science Department
Faculty of Informatics Mahasarakham University

2.4.6 การเปรียบเทียบของแอปตัวอย่าง

ตารางที่ 2.3 ตารางเปรียบเทียบของแอปตัวอย่าง

ฟังก์ชันการทำงาน	Lazada	Shopee	KingPower	Kaidee	Lineshop	Mini Plant
สมัครเป็นสมาชิก	✓	✓	✓	✓	✓	✓
ล็อกอินและล็อกเอาต์ระบบ	✓	✓	✓	✓	✓	✓
ยกเลิกการเป็นสมาชิก	✓	✓	✓	✓	✓	✓
แก้ไขข้อมูลของตนเอง	✓	✓	✓	✓	✓	✓
ค้นหาสินค้า	✓	✓	✓	✓	✓	✓
จัดโปรโมชัน	✓	✓	✓	✓	✓	✓
ดำเนินรายการสั่งซื้อสินค้า	✓	✓	✓	✓	✓	✓
คะแนนความพึงพอใจต่อร้านค้า				✓		✓
การตอบความคิดเห็น	✓	✓		✓		✓
ประวัติการสั่งซื้อสินค้า	✓	✓	✓	✓	✓	✓
สมัครเป็นผู้ขาย	✓	✓		✓		
เก็บเงินปลายทาง	✓			✓		

Computer Science Department
Faculty of Informatics, Mahasarakham University