

ภาคผนวก ซ

ส่วนประกอบ
ด้านเกม

บทที่ 3 การวิเคราะห์และออกแบบระบบงาน

3.1 องค์ประกอบของเกม

3.1.1 โครงเรื่องของเกม

โครงเรื่องของเกม คือ ส่วนประกอบที่สำคัญที่สุด กล่าวคือ เนื้อเรื่องเป็นการกำหนดขององค์ประกอบทั้งหมดในเกมเข้าด้วยกัน ไม่ว่าจะเป็นอาร์ตเวิร์ก อินเตอร์เฟซ วิธีการเล่น ฯลฯ ดังนั้นเนื้อเรื่องที่ดีควรทำให้ผู้เล่นเกิดความรู้สึกร่วมกับตัวละคร โดยเราสามารถจินตนาการเนื้อเรื่องเสมือนว่าเรากำลังเขียนบทภาพยนตร์ ซึ่งเนื้อเรื่องที่ดีจะทำให้เกมน่าสนใจมากยิ่งขึ้น

3.1.2 ลักษณะของเกมและวิธีการเล่น

วิธีการเล่นเกมเป็นปัจจัยที่สำคัญในขั้นตอนการออกแบบซึ่งมักจะเป็นการนำเสนอความคิดของวิธีเล่น เพื่อให้เกิดเกมแนวใหม่ๆ ที่มีความแตกต่าง หรือแม้เป็นการพัฒนาสิ่งใหม่ๆ ขึ้นมาทั้งหมดให้ต่างออกไปจากเกมที่เคยมีมาก่อน เช่น เกม Angry bird เป็นเกมที่น่าเสนอวิธีการเล่นโดยการยิงตัวละคร ตามวิถีแบบโปรเจกไทล์ (Projectile) ตามหลักการฟิสิกส์

3.1.3 สตอรี่บอร์ด

ใช้สัญลักษณ์ด้วยคอมพิวเตอร์ หรือวาดด้วยมือ



คำอธิบาย

ลิงโยนเถาวัลย์จากฝั่งซ้ายไปฝั่งขวา โดยไม่ให้ตกน้ำ และ เถาวัลย์จะแกว่งไปมาตลอดเวลา

ภาพที่ แสดงการเขียนสตอรี่บอร์ด

3.1.4 ออกแบบหน้าจอ

เกมดิจิทัลทั้งหมดนั้นล้วนใช้หน้าจอเพื่อให้ผู้เล่นใช้งาน โดยทั่วไปการเริ่มต้นเล่นเกมผู้เล่นจะทำการโหลดเกม และจะต้องหาจุดเริ่มต้นของเกมว่าเริ่มจากตรงไหน แล้วมีฟังก์ชันอะไรให้ใช้ได้บ้าง การออกแบบหน้าจอที่ดีนั้นจะต้องนำเสนอข้อมูลให้กับผู้เล่นได้ แม้ในขณะที่เล่นเกมอยู่ เกมส่วนมากจะออกแบบส่วนแสดงผลตรงตำแหน่งด้านบนของจอ

เลือกอาวุธ	ส่วนแสดงตัวละคร	แสดงระดับการเล่นของตัวละคร
<ul style="list-style-type: none"> ■ ปืน ■ เสือเกราะ ■ ระเบิด ■ กระเป๋าเป้ ■ หมวก 		แสดงสถานะออนไลน์
		เมนู
		<ul style="list-style-type: none"> ■ บันทึกข้อมูล ■ ออกจากเกม ■ เริ่มเล่นใหม่

ภาพที่ แสดงการออกแบบหน้าจอ

3.1.5 ข้อกำหนดของเกม

การสร้างแนวคิดหรือรูปแบบ เป็นการออกแบบตัวละคร ฉากหลังในแต่ละด่านแต่ละระดับ ระดับความยากง่ายของแต่ละด่าน ซึ่งทั้งหมดนี้เป็นหน้าที่ของผู้ออกแบบเกม เพื่อพัฒนาภาพรวมหรือความรู้สึกของเกม เช่น

3.1.5.1 การออกแบบตัวละคร เช่น ตัวละครผู้เล่น ตัวละครศัตรู ท่าทางของตัวละคร

3.1.5.2 การออกแบบเกม เช่น พื้นหลัง การออกแบบฉาก การออกแบบภาพในเกม เสียงประกอบเกม

3.1.5.3 การโจมตี เช่น มีวิธีในการโจมตีศัตรูอย่างไร

3.1.5.4 เส้นทางการเล่นของเกม เช่น ลำดับการเล่น การออกแบบระดับความยากง่ายของเกมหรือด่านต่างๆ

3.2 ยูสเคส ไดอะแกรม (Use Case Diagram)

ยูสเคส ไดอะแกรม เป็นไดอะแกรมที่ช่วยให้ผู้พัฒนาทราบถึงความสามารถของระบบว่าสามารถทำอะไรได้บ้าง ทราบถึงผู้ใช้งานในแต่ละส่วนของระบบ และเกิดความสะดวกในการสื่อสารระหว่างผู้พัฒนากับผู้ใช้ระบบ

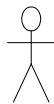
3.2.1 สัญลักษณ์ที่ใช้ในยูสเคส ไดอะแกรม

3.2.1.1 ยูสเคส (Use Case) คือความสามารถหรือฟังก์ชันของระบบซอฟต์แวร์ที่จะพัฒนา โดยการเขียนยูสเคสจะใช้วงรีและคำอธิบายฟังก์ชันการทำงานอยู่ในวงรีนั้น



ภาพที่ 3.1 แสดงสัญลักษณ์ยูสเคส

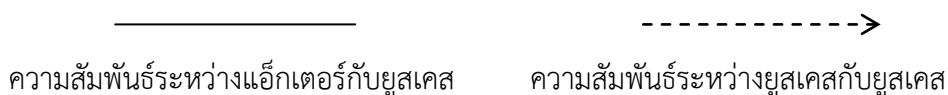
3.2.1.2 แอ็กเตอร์ (Actor) คือ ผู้ที่กระทำกับระบบ หรือผู้ที่เกี่ยวข้อง โดยเป็นผู้ที่แลกเปลี่ยนข้อมูลข่าวสารกับระบบที่จะพัฒนา เราจะใช้สัญลักษณ์รูปคนแทนแอ็กเตอร์



Actor Name

ภาพที่ 3.2 แสดงสัญลักษณ์แอ็กเตอร์

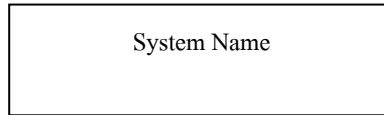
3.2.1.3 เส้นแสดงความสัมพันธ์ (Relationship) คือ เส้นที่ใช้ในการแสดงความสัมพันธ์ระหว่างแอ็กเตอร์ กับ ยูสเคส หรือ ยูสเคส กับ ยูสเคส



ภาพที่ 3.3 แสดงสัญลักษณ์เส้นแสดงความสัมพันธ์

- ความสัมพันธ์แบบขยาย (Extend Relationship) ใช้เพื่อบอกว่ายูสเคสหนึ่ง ถูกช่วยเหลือโดยการทำงานยูสเคสอื่น โดยจะใช้ <<extend>> เป็นเครื่องหมายอ้างอิง
- ความสัมพันธ์แบบรวม (Include Relationship) ใช้เพื่อบอกว่ายูสเคสหนึ่งถูกอาศัยการทำงานของยูสเคสอื่น ๆ โดยจะใช้ <<include>> เป็นเครื่องหมายอ้างอิง

3.2.1.4 ขอบเขต (System Boundary) คือ เส้นแบ่งขอบเขตระหว่างระบบกับแก็กเตอร์ จะใช้สี่เหลี่ยมเป็นสัญลักษณ์



ภาพที่ 3.4 แสดงสัญลักษณ์ขอบเขต

3.2.2 การเขียนคำอธิบายยูสเคส

เนื่องจากแต่ละยูสเคสประกอบไปด้วยการทำงานหลาย ๆ อย่าง ดังนั้นเพื่อให้ทราบรายละเอียดปลีกย่อยของขั้นตอนการทำงานในแต่ละยูสเคส จึงต้องมีการเขียนคำอธิบายสำหรับยูสเคสควบคู่กันไปเราเรียกคำอธิบายยูสเคส ดังกล่าวว่า กระแสของเหตุการณ์ (Flow of Event)

ในการเขียนคำบรรยายกระแสของเหตุการณ์ในปัจจุบันนั้นยังมีรูปแบบที่แตกต่างกันออกไป แต่โดยส่วนใหญ่คำบรรยายกระแสเหตุการณ์นี้จะประกอบไปด้วยสองส่วนสำคัญได้แก่ เหตุการณ์หลัก (Main Flow) และ เหตุการณ์พิเศษ (Exceptional Flow) โดยเหตุการณ์ทั้งสองส่วนจะต้องระบุถึงสาเหตุของการเริ่มต้นและสิ้นสุดกิจกรรมด้วยเสมอ

3.2.2.1 เหตุการณ์หลัก(Main Flow) คือลำดับกิจกรรมเมื่อ ยูสเคสดำเนินกิจกรรมตามปกติ โดยการเขียนคำอธิบายในลักษณะเป็นย่อหน้าและเหตุการณ์หลักจะต้องมีเพียงหนึ่งเดียว

3.2.2.2 เหตุการณ์พิเศษ(Exceptional Flow) คือลำดับกิจกรรมเมื่อ ยูสเคสดำเนินกิจกรรมผิดจากปกติโดยสามารถมีได้มากกว่า 1 เหตุการณ์

3.2.3 ข้อเสนอแนะในการเขียนยูสเคส ไตอะแกรม

3.2.3.1 ใช้เพื่อแสดงข้อมูลความต้องการของผู้ใช้ระบบเท่านั้น หรืออาจกล่าวได้ว่าเราใช้ยูสเคส ไตอะแกรม เพื่อเก็บรวบรวมข้อมูลความต้องการจากผู้ใช้ระบบเท่านั้น ดังนั้นเป็นไปได้ว่าในการเก็บข้อมูลในรอบแรกอาจยังไม่ครอบคลุมความต้องการของผู้ใช้ จึงสามารถนำกลับมาปรับปรุงแก้ไขเพิ่มเติมได้จนกว่าจะได้ข้อมูลครบถ้วน

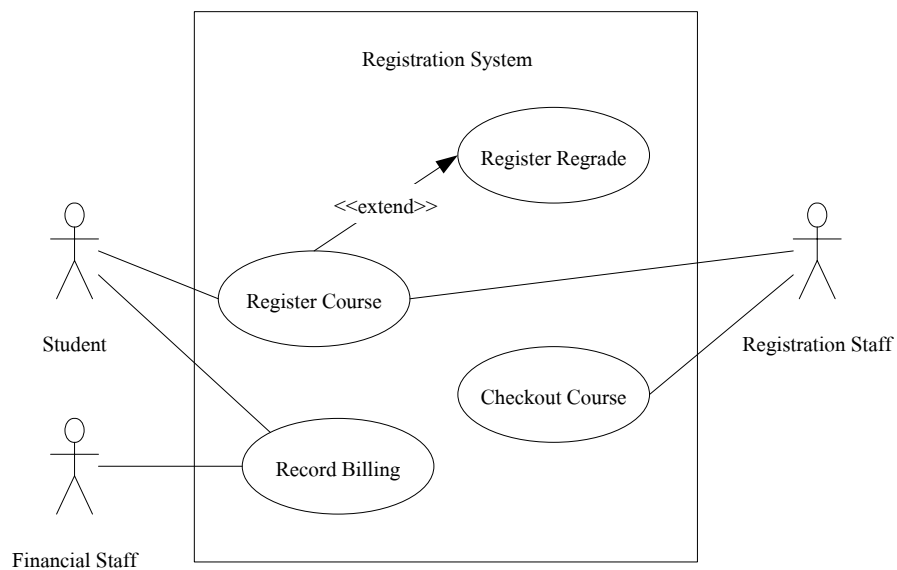
3.2.3.2 ยูสเคส ไตอะแกรม ใช้เพื่อสื่อสารระหว่างนักวิเคราะห์ระบบกับผู้ใช้ ไม่ได้ใช้ในการสื่อสารระหว่างนักวิเคราะห์ระบบกับโปรแกรมเมอร์ ดังนั้นยูสเคส ไตอะแกรม จะแสดงเพียงหน้าที่ที่ระบบต้องทำตามความต้องการของผู้ใช้ ทำให้มองเห็นภาพกว้าง ๆ ของระบบมากกว่าภาพเชิงลึก โดยส่วนใหญ่ ยูสเคส ไตอะแกรม จะไม่แสดงให้เห็นถึงระดับการจัดการข้อมูลในฐานข้อมูล เช่น เพิ่ม ลบ แก้ไข หรือปรับปรุงข้อมูล เป็นต้น

3.2.3.3 ยูสเคส ไตอะแกรม อาจจะมีรายละเอียดมากหรือน้อยก็ได้ ขึ้นอยู่กับมุมมอง เทคนิค และประสบการณ์ของทีมงานหรือนักวิเคราะห์ระบบ จึงไม่มีข้อสรุปได้ว่าลักษณะของยูสเคส ไตอะแกรมแบบใดถูกหรือผิด

3.2.3.4 แก็กเตอร์ทุกตัวต้องมีปฏิสัมพันธ์กับยูสเคสอย่างน้อยหนึ่งตัว และยูสเคสทุกตัวต้องมีปฏิสัมพันธ์อย่างใดอย่างหนึ่งกับ แก็กเตอร์ หรือ ยูสเคสตัวอื่น ๆ เสมอ

ตัวอย่าง

ตัวอย่างต่อไปนี้เป็นระบบลงทะเบียนเรียน (Registration System) ประกอบด้วย 3 แอ็กเตอร์ ได้แก่ นักศึกษา(Student) เจ้าหน้าที่การเงิน(Financial Staff) และ เจ้าหน้าที่ฝ่ายทะเบียน (Registration Staff) และประกอบด้วยยูสเคส ซึ่งสามารถทำรายการลงทะเบียน (Register Course) ตรวจสอบวิชาที่ลงทะเบียน (Checkout Course) และบันทึกรายการชำระเงินค่าลงทะเบียน (Record Billing) นอกจากนี้อาจจะมีเหตุการณ์หรือเงื่อนไขพิเศษเช่น มีนักศึกษาลงทะเบียนรายวิชาซ้ำเพื่อทำการปรับเกรด (Register Regrade) ด้วยจึงเพิ่ม Extending Use Case เพื่อรองรับหน้าที่ดังกล่าว



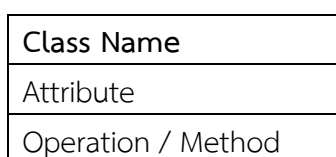
ภาพที่ 3.5 ตัวอย่างยูสเคส ไดอะแกรมระบบลงทะเบียน

คำอธิบายยูสเคส

Use Case Title : Register Course
Main Flow : ระบบลงทะเบียนมีกลุ่มบุคคลที่เกี่ยวข้อง 2 กลุ่มได้แก่ นักศึกษา และพนักงานของมหาวิทยาลัย (เจ้าหน้าที่ฝ่ายลงทะเบียนและเจ้าหน้าที่ฝ่ายการเงิน) ในแต่ละเทอมจะต้องมีนักศึกษามาลงทะเบียนเรียนของภาคเรียนปกติโดยนักศึกษาจะต้องกรอกรายวิชาเรียนในแบบฟอร์มลงทะเบียนไม่เกิน 21 หน่วยกิต แล้วนำไปยื่นกับเจ้าหน้าที่เพื่อทำการตรวจสอบวิชาที่นักศึกษาได้ลงทะเบียนในแบบฟอร์มกับประวัติการลงทะเบียนว่าถูกต้องหรือไม่ เนื่องจากบางรายวิชาของแต่ละเทอมมีเงื่อนไขว่าจะลงทะเบียนรายวิชาได้ก็ต่อเมื่อสอบผ่านรายวิชาหนึ่งมาก่อน เมื่อตรวจสอบว่าถูกต้องแล้ว เจ้าหน้าที่ฝ่ายทะเบียนจะคำนวณเงินค่าลงทะเบียนเรียน จากนั้นบันทึกลงในฐานข้อมูล ระบบแสดงข้อความบันทึกข้อมูลเรียบร้อยแล้ว ส่งพิมพ์ใบรับลงทะเบียนโดยแบ่งออกเป็น 2 ส่วน ส่วนที่ 1 นักศึกษาเก็บไว้เอง ส่วนที่ 2 นำไปชำระเงินโดยโอนฝ่ายทางธนาคาร
Exceptional Flow ที่ 1 : กรณีที่เจ้าหน้าที่ฝ่ายทะเบียนป้อนรหัสนักศึกษาผิดพลาด ระบบจะแสดงข้อความแจ้งเตือนว่า “ไม่พบข้อมูลนักศึกษา” เพื่อแสดงให้เจ้าหน้าที่ทราบว่ามีการผิดพลาดเกิดขึ้น และให้เจ้าหน้าที่ป้อนรหัสนักศึกษาใหม่อีกครั้ง
Exceptional Flow ที่ 2 : กรณีที่เจ้าหน้าที่ป้อนข้อมูลสำคัญไม่ครบถ้วน ระบบจะไม่สามารถบันทึกการลงทะเบียนเรียนได้ ดังนั้นระบบจะมีข้อความแจ้งเตือน “ป้อนข้อมูลสำคัญไม่ครบถ้วน กรุณากลับไปป้อนข้อมูลให้ครบ” ให้เจ้าหน้าที่ป้อนข้อมูลสำคัญให้ครบถ้วนระบบจึงจะสามารถรับข้อมูลต่อได้

3.3 คลาส ไดอะแกรม (Class Diagram)

เป็นแผนภาพที่ใช้ในการแสดงกลุ่มของคลาส โครงสร้างของคลาส อินเตอร์เฟซ (Interface) และแสดงความสัมพันธ์ (Relationship) ระหว่างคลาส ซึ่งแผนภาพนี้เป็นแผนภาพที่จะพบมากที่สุดในการออกแบบเชิงวัตถุ โดยสัญลักษณ์ของคลาสจะอธิบายถึงคุณสมบัติ (Attribute) และ การดำเนินการ (Operation/Method) ดังภาพที่ 3.6



ภาพที่ 3.6 แสดงสัญลักษณ์ของคลาส

3.3.1 ส่วนประกอบของคลาส

3.3.1.1 Class Name เป็นคำนาม ขึ้นต้นด้วยตัวอักษรตัวใหญ่ ไม่มีช่องว่าง หากเป็นชื่อคลาสจะเป็นตัวหนา แต่ถ้าเป็นชื่อของ Abstract Class จะเป็นตัวเอียง

3.3.1.2 Attribute ประกอบด้วย

- Visibility เป็นเครื่องหมายแสดงสิทธิ์การเข้าถึง Attribute ของคลาสนั้น ได้แก่ Public(+) Private(-) Protected(#)

- ชื่อ Attribute

- เครื่องหมาย :

- ประเภทของ Attribute แบ่งได้ 2 ประเภทคือ Primitive Type และ Class Type

3.3.1.3 Operation / Method ประกอบด้วย

- Visibility เป็นเครื่องหมายแสดงสิทธิ์การเข้าถึง Operation / Method ของคลาสนั้น ได้แก่ Public(+) Private(-) Protected(#)

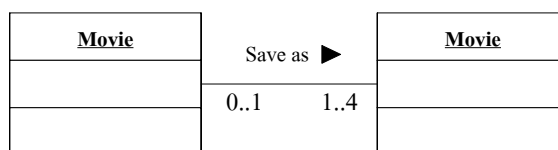
- ชื่อของ Operation / Method

- พารามิเตอร์ อยู่ภายในเครื่องหมายวงเล็บ เป็นตัวแปรหรือ object ที่ถูกส่งเข้าไปใน Operation / Method

- Return Type อยู่ต่อจากเครื่องหมาย : เป็นชนิดของผลลัพธ์ที่ได้จากการดำเนินงานซึ่งจะถูกส่งออกมาสู่ภายนอก

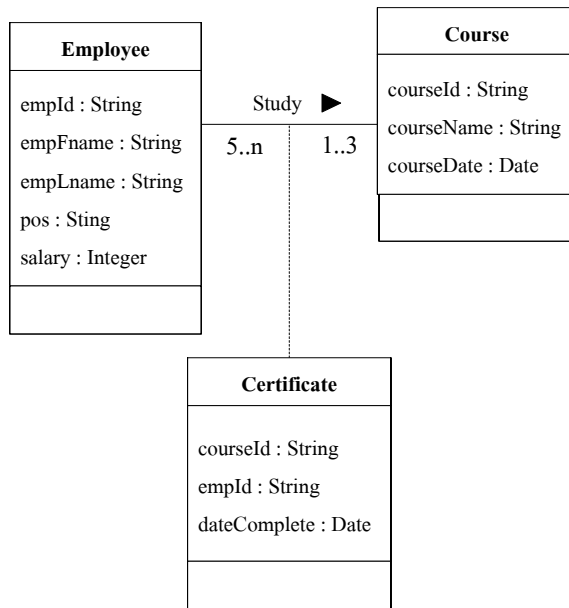
3.3.2 ความสัมพันธ์ระหว่างคลาส

3.3.2.1 Association เป็นความสัมพันธ์ในระดับเดียวกัน ไม่มีคลาสใดคลาสหนึ่งมีความสำคัญมากกว่าคลาสอื่น



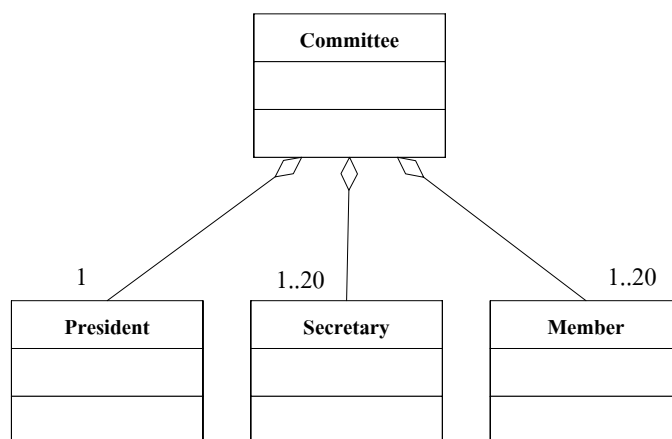
ภาพที่ 3.7 แสดงตัวอย่างความสัมพันธ์แบบ Association

Associative Class สำหรับการวิเคราะห์และออกแบบระบบเชิงวัตถุความสัมพันธ์ของคลาสแบบ Association มี 2 แบบคือ One-to-One หรือ One-to-Many ซึ่งถือว่าเป็นความสัมพันธ์แบบธรรมดา แต่หากความสัมพันธ์เป็นแบบ Many-to-Many แล้วจะต้องแตกคลาสเพิ่มอีก 1 คลาส และเรียกคลาสที่เพิ่มขึ้นมาว่า Associative Class



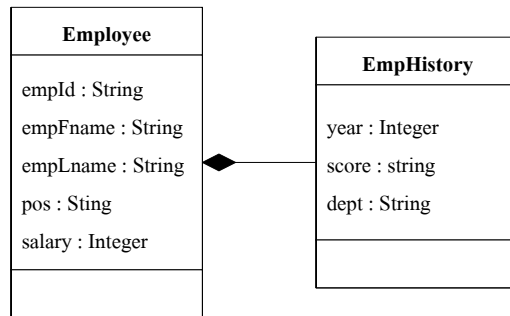
ภาพที่ 3.8 แสดงตัวอย่าง Associative Class

3.3.2.2 Aggregation เป็นความสัมพันธ์ระหว่างคลาสแบบต่างระดับ ในลักษณะของการเป็นองค์ประกอบ โดยคลาสที่เป็นองค์ประกอบเรียกว่า “Part Class” ส่วนคลาสที่เกิดจากการรวมกันขององค์ประกอบต่าง ๆ เรียกว่า “Whole Class” โดยใช้สัญลักษณ์เส้นตรงหัวข่าว หลามตัดไปร่งลากจาก Part Class ไปยัง Whole Class ลักษณะสำคัญของความสัมพันธ์แบบ Aggregation คือ เมื่อลบ Whole Class ที่ไป คลาสที่เป็น Part Class จะไม่ถูกลบไปด้วย จะยังคงอยู่ต่อไปโดยไม่จำเป็นต้องพึ่งพา Whole Class



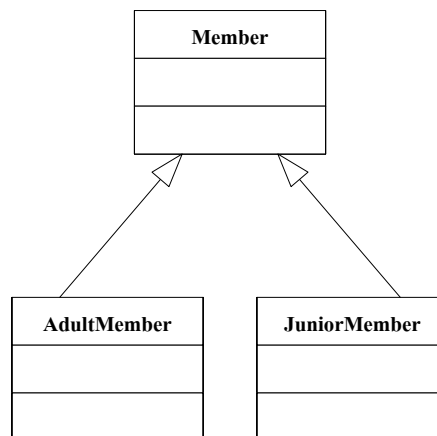
ภาพที่ 3.9 แสดงตัวอย่างความสัมพันธ์แบบ Aggregation

3.3.2.3 Composition เป็นความสัมพันธ์แบบ Aggregation ชนิดพิเศษที่ Whole Class มีผลต่อการดำรงอยู่ของ Part Class กล่าวคือ หาก Whole Class ถูกทำลายหรือลบทิ้งไป Part Class ก็จะถูกทำลายหรือลบทิ้งไปด้วย หรือหากกล่าวอีกนัยหนึ่ง Part Class จะต้องทำงานร่วมกับ Whole Class ไปจนตลอดอายุ Whole Class สัญลักษณ์ที่ใช้คือเส้นตรงหัวข้าวหลามตัดที่ปลายจาก Part Class ไปยัง Whole Class



ภาพที่ 3.10 แสดงตัวอย่างความสัมพันธ์แบบ Composition

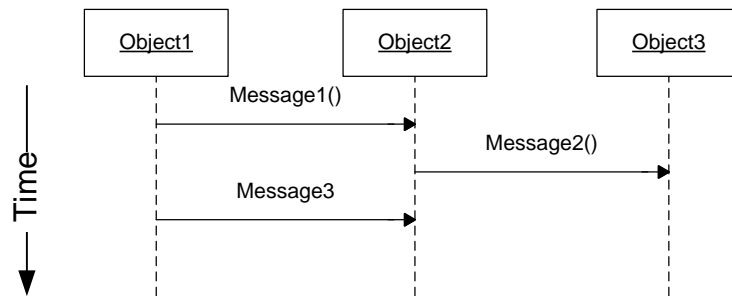
3.3.2.4 Generalization / Specialization เป็นการอธิบายความสัมพันธ์ระหว่างคลาสในลักษณะจำแนกชนิด การจำเพาะเจาะจงรายละเอียด หรือการหาลักษณะร่วมกันของคลาสต่างชนิดกัน เพื่อสร้างคลาสที่เป็นตัวแทนของกลุ่มคลาสเหล่านั้น



ภาพที่ 3.11 แสดงตัวอย่างความสัมพันธ์แบบ Generalization / Specialization

3.4 Sequence Diagram

เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจ็กต์โดยเฉพาะ การส่ง Message ระหว่างอ็อบเจ็กต์ตามลำดับของเวลา (Sequence) ที่เกิดเหตุการณ์ขึ้น จากนั้นน้อยไปมาก โดยจะมีสัญลักษณ์แสดงให้เห็นลำดับของการส่ง Message ตามเวลาส่งอย่างชัดเจน



ภาพที่ 3.12 แสดงลักษณะของ Sequence Diagram

3.4.1 ส่วนประกอบของ Sequence Diagram

3.4.1.1 Actor คือ ผู้กระทำต่อระบบ

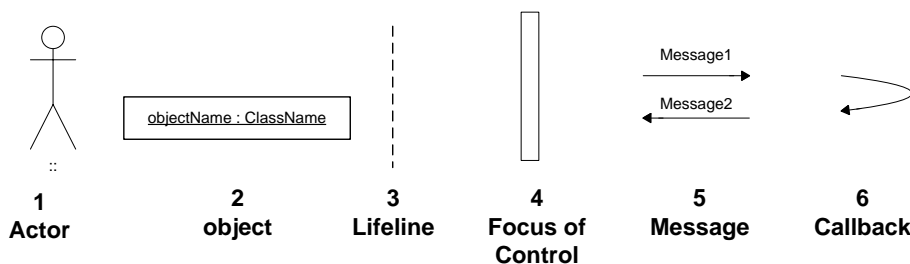
3.4.1.2 Object คือ อ็อบเจกต์ที่ต้องทำหน้าที่

3.4.1.3 Lifeline คือ เส้นแสดงชีวิตของอ็อบเจกต์หรือคลาส

3.4.1.4 Focus of Control / Activation จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่

3.4.1.5 A Message คำสั่งหรือฟังก์ชันที่คลาสหนึ่งส่งให้อีกคลาสหนึ่ง ซึ่งสามารถส่งกลับได้ด้วย

3.4.1.6 Callback / Self Delegation คือ การประมวลผลและคือค่าที่ได้ภายในอ็อบเจกต์เดียวกัน



ภาพที่ 3.13 แสดงสัญลักษณ์ภายใน Sequence Diagram

3.4.2 Stereotype

เป็นเทคนิคที่ใช้ในการเพิ่มชนิดของสัญลักษณ์พิเศษในภาษา UML จากสัญลักษณ์เดิมที่มีอยู่ แล้ว ให้กลายเป็นสัญลักษณ์ชนิดใหม่ สำหรับคลาสที่นำมาใช้ในกระบวนการสร้าง Sequence Diagram ได้แก่ Entity Class Boundary Class และ Control Class

3.4.2.1 Entity Class คือคลาสที่เก็บข้อมูลสำคัญของระบบไว้ เป็นคลาสที่ใช้เป็นตัวแทนของฐานข้อมูลระบบ ลักษณะสำคัญของ Entity Class คือข้อมูลที่ถูกเก็บอยู่ใน Entity Class จะเป็นข้อมูลที่คงที่อยู่ตลอดเวลาแม้ว่าเครื่องคอมพิวเตอร์หรือระบบจะถูกปิดลงก็ตาม



Entity Class Name

ภาพที่ 3.14 แสดงสัญลักษณ์ Entity Class

3.4.2.2 Boundary Class คือ คลาสที่ถูกใช้โดยแอดแดกเตอร์ เพื่อปฏิสัมพันธ์กับระบบเป็นตัวกลางระหว่างแอดแดกเตอร์ กับระบบ เช่น คลาสของฟอร์มใบสั่งซื้อ หน้าจอแสดงผล กล่องข้อความ เป็นต้น



Boundary Class Name

ภาพที่ 3.15 แสดงสัญลักษณ์ของ Boundary Class



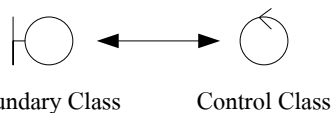
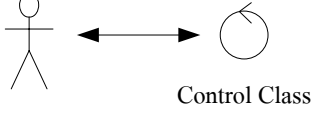
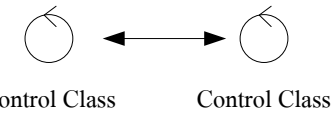
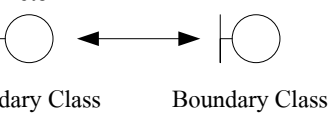
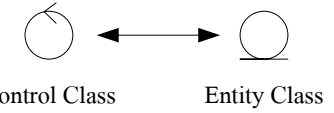
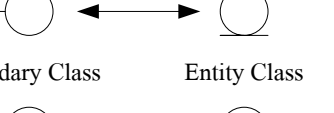
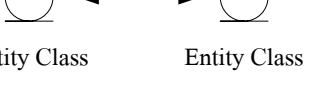
3.4.2.3 Control Class คือ คลาสที่คอยกำหนดกฎเกณฑ์และควบคุมการทำงานของระบบในแต่ละ Use case เมื่อมีการทำงานใด ๆ ก็ตามที่เกี่ยวข้องกับข้อมูลสำคัญของระบบ เช่น การเพิ่ม ลบ หรือ แก้ไขข้อมูลในฐานข้อมูล เป็นต้น จะต้องอยู่ภายใต้การควบคุมของ Control Class เพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับข้อมูล

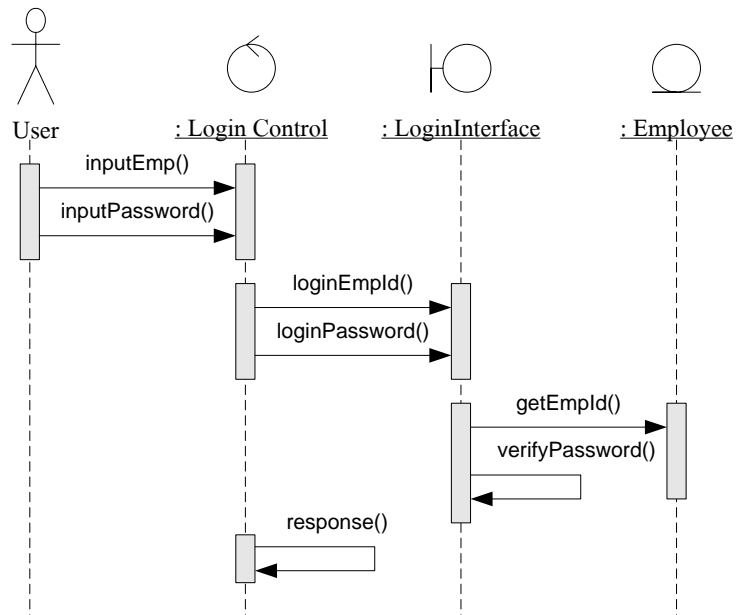


Control Class Name

ภาพที่ 3.16 แสดงสัญลักษณ์ของ Control Class

ในปี ค.ศ. 1999 Rosenberg และ Scott ได้กำหนดเงื่อนไขบางประการในการวางตำแหน่งของ Actor และคลาสชนิดพิเศษทั้ง 3 ชนิดใน Sequence Diagram ขึ้นเพื่อเพิ่มความถูกต้อง

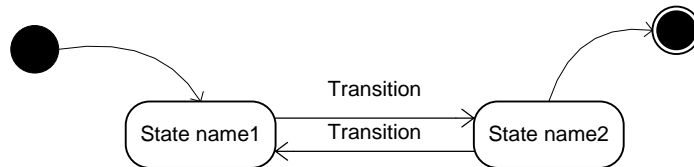
ถูกต้อง	ไม่ถูกต้อง
 <p>Actor</p> <p>Boundary Class</p>	 <p>Actor</p> <p>Entity Class</p>
 <p>Boundary Class</p> <p>Control Class</p>	 <p>Actor</p> <p>Control Class</p>
 <p>Control Class</p> <p>Control Class</p>	 <p>Actor</p> <p>Boundary Class</p> <p>Boundary Class</p>
 <p>Control Class</p> <p>Entity Class</p>	 <p>Boundary Class</p> <p>Entity Class</p>
	 <p>Entity Class</p> <p>Entity Class</p>



ภาพ 3.17 แสดงตัวอย่างการใช้งาน Sequence Diagram

3.5 Statechart Diagram (ถ้ามี)

Statechart Diagram เป็นแผนภาพที่แสดงให้เห็นพฤติกรรมของอ็อบเจ็กต์เช่นเดียวกับแผนภาพในกลุ่ม Behavioral Diagram อื่น ๆ แต่ Statechart Diagram จะเน้นที่การแสดงให้เห็นถึงสถานะ (State) การเปลี่ยนสถานะ (Transition) ที่มีเหตุการณ์ (Event) ที่เกิดขึ้นในช่วงชีวิตของอ็อบเจ็กต์ 1 ช่วง (1 Sequence) แสดงลักษณะของ Statechart Diagram ดังภาพที่ 3.18



ภาพที่ 3.18 แสดงลักษณะของ Statechart Diagram

3.5.1 สัญลักษณ์ของ Statechart Diagram

3.5.1.1 Initial State คือ จุดเริ่มต้นการเปลี่ยนสถานะ



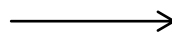
ภาพที่ 3.19 แสดงสัญลักษณ์ของ Initial Stage

3.5.1.2 Final State คือ จุดสิ้นสุดของการเปลี่ยนสถานะ



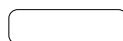
ภาพที่ 3.20 แสดงสัญลักษณ์ของ Final Stage

3.5.1.3 Transition คือ เส้นกระตุ้นให้เปลี่ยนสถานะซึ่งจะมีชื่อของเหตุการณ์ที่ทำให้เกิดการเปลี่ยนสถานะกำกับอยู่



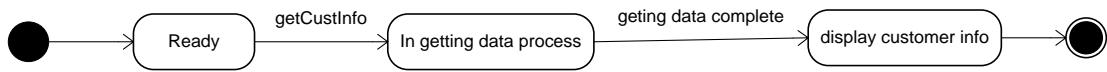
ภาพที่ 3.21 แสดงสัญลักษณ์ของ Transition

3.5.1.4 State คือ สถานะของอ็อบเจ็กต์ ภายในเป็นชื่อของสถานะซึ่งมักเป็นวลีหรือคำที่แสดงกริยาของสถานะ



ภาพที่ 3.22 แสดงสัญลักษณ์ของ State

ตัวอย่าง

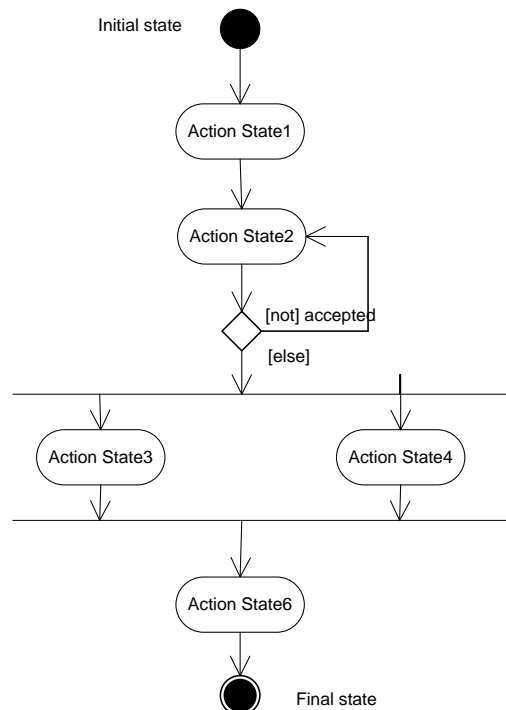


ภาพที่ 3.23 แสดงตัวอย่าง Statechart Diagram

แสดงให้เห็นสถานะที่เปลี่ยนแปลงไปของอ็อบเจกต์ “Customer” โดยเริ่มต้นที่สถานะ “เตรียมพร้อม (Ready)” จากนั้นมีอ็อบเจกต์อื่นส่ง Message “getCustInfo” มากระตุ้นทำให้ “Customer” เกิดการเปลี่ยนสถานะไปเป็น “อยู่ในกระบวนการดึงข้อมูล (in getting data process)” จากนั้นเมื่อกระบวนการดึงข้อมูลเสร็จสิ้น (getting data complete) ทำให้อ็อบเจกต์ “Customer” เปลี่ยนสถานะไปเป็น “แสดงสินค้า (display customer info)” และสิ้นสุดการเปลี่ยนแปลงสถานะ

3.6 Activity Diagram (ถ้ามี)

เป็นแผนภาพที่แสดงให้เห็นลำดับการดำเนินกิจกรรม (Activity) จากกิจกรรมหนึ่งไปยังกิจกรรมหนึ่งภายในระบบที่เกิดจากการทำงานของอ็อบเจกต์ ลักษณะของแผนภาพจะคล้ายกับ Flow Chart ดังภาพที่ 3.24

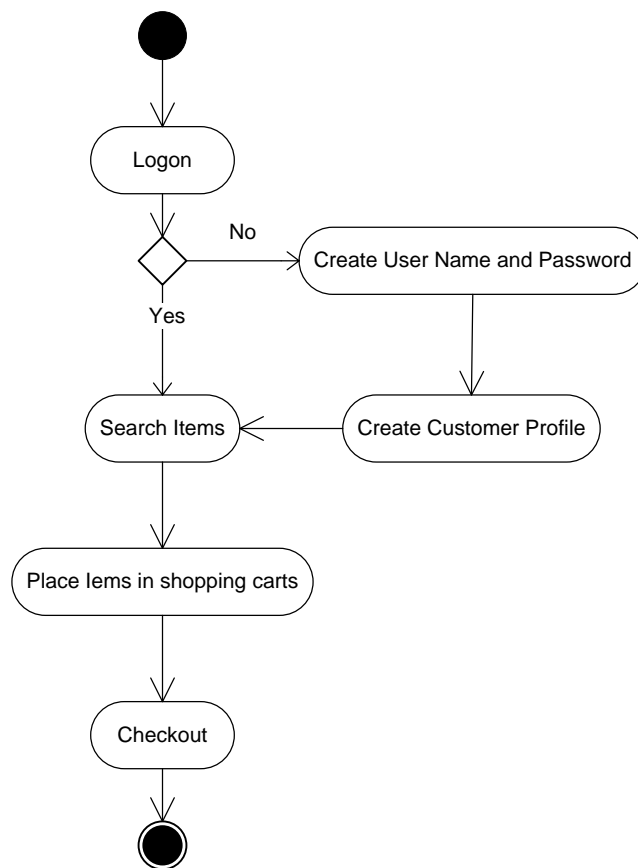


ภาพที่ 3.24 แสดงลักษณะของ Activity Diagram

สำหรับสัญลักษณ์ของ Activity Diagram จะคล้ายๆ กับ Statechart Diagram ได้แก่ จุดเริ่มต้น จุดสิ้นสุด และกิจกรรม แต่สำหรับ Activity Diagram จะไม่แสดงให้เห็นการเปลี่ยนแปลงสถานะแต่จะแสดงให้เห็นลำดับของกิจกรรมต่างๆ ซึ่งสามารถเขียนได้หลายรูปแบบดังนี้

3.6.1 แบบทางเลือกตัดสั้นใจ

การเขียน Activity Diagram แบบทางเลือกตัดสั้นใจ สามารถเขียนได้โดยลากลูกศรผ่านสัญลักษณ์ข้าวหลามตัดก่อนแล้วจึงลากไปยังแต่ละทางเลือก

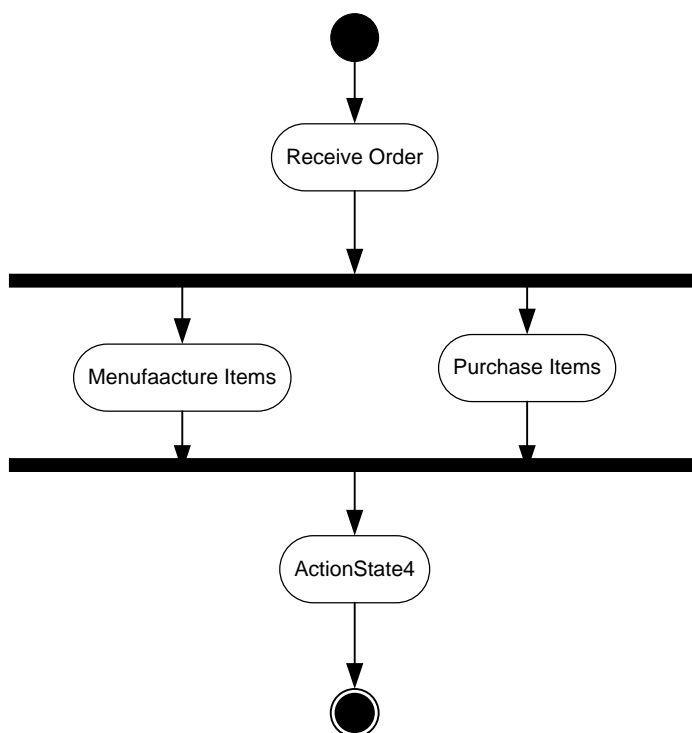


ภาพที่ 3.25 Activity Diagram แบบมีทางเลือกตัดสั้นใจ

จากภาพที่ 3.25 แสดงลำดับขั้นตอนการสั่งซื้อสินค้าทางอินเทอร์เน็ต เริ่มต้นเมื่อลูกค้าเข้ามาที่ เว็บไซต์ (Log on) โดยมีทางเลือกให้ดำเนินการต่อไปหากลูกค้ามีชื่อและรหัสสมาชิกแล้ว แต่หากยังไม่มีให้ไปลงทะเบียนเพื่อกำหนดชื่อและรหัสสมาชิกใหม่ก่อน จึงจะสามารถเลือกรายการสินค้าที่ต้องการได้ แล้วทำการวางสินค้าที่เลือกไว้ในตะกร้า เมื่อเลือกสินค้าครบแล้ว ระบบจะทำการคำนวณราคาสินค้า พร้อมกับรายงานผล จากนั้นลูกค้าก็สามารถออกจากระบบได้

3.6.2 แบบมีการทำงานพร้อมกัน

สำหรับลักษณะการทำงานพร้อมกันจะเรียกว่า “Transition Fork” ซึ่งหมายถึง จุดเปลี่ยนแยก โดยจะต้องลากเส้นตรงขวางแนวนอนก่อนแยกสู่กิจกรรม ที่จะต้องกระทำพร้อมกัน เมื่อเสร็จสิ้นแล้วก็ลากเส้นตรงขวางแนวนอนก่อนแล้วจึงลากลูกศร รวมมายัง กิจกรรมอื่นต่อไป ลักษณะรวมนี้เรียกว่า “Transition Join” ซึ่งหมายถึงจุดเปลี่ยนรวม ดังภาพที่ 3.24 เป็นลำดับขั้นตอน “การผลิตสินค้าตามสั่ง” Activity Diagram ในลักษณะนี้สามารถมี Decision ได้ด้วย ดังตัวอย่างในภาพที่ 3.26



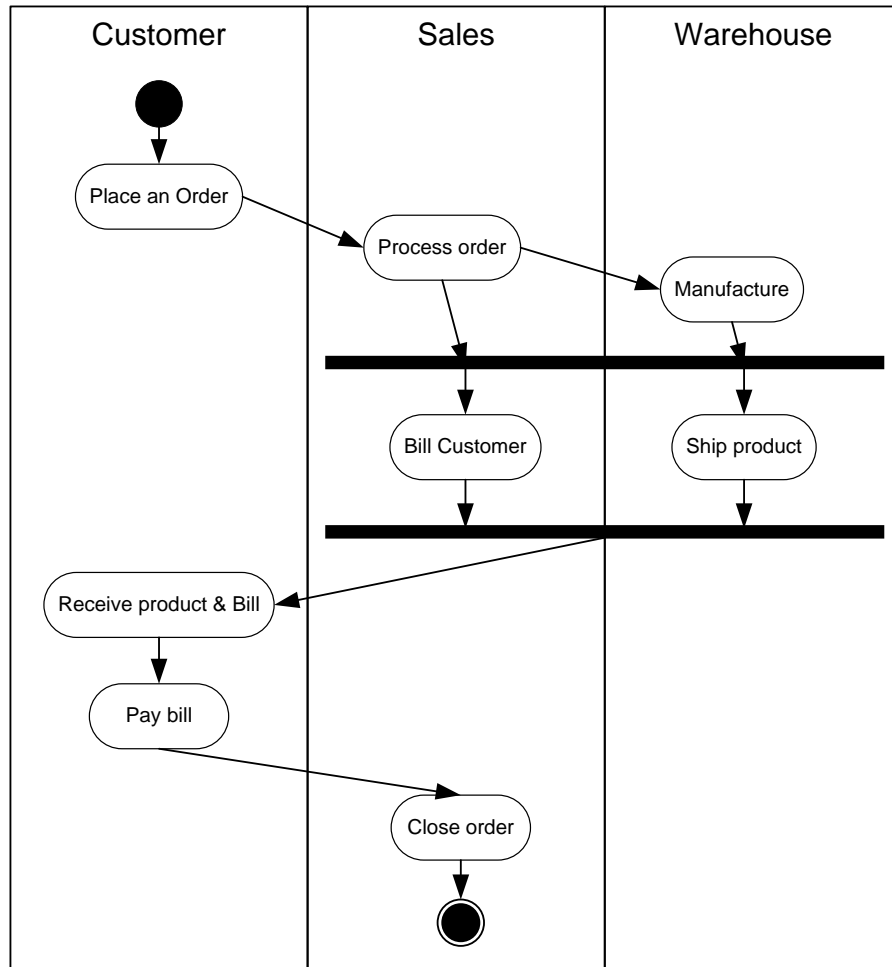
ภาพที่ 3.26 Receive Order

ลำดับต่อไปจะเป็นการผลิตสินค้าซึ่งบางส่วนอาจผลิตตามใบสั่งซื้อ (Manufacture Item) หรือบางส่วนอาจสั่งซื้อจากที่อื่น (Purchase Item) ก็ได้ หลังจากนั้นจึงจบกระบวนการรับใบสั่งซื้อ (Complete Order)

3.6.3 แบบแบ่งส่วนด้วย Swimlanes

เป็นการแบ่งส่วนการทำงานออกเป็นหลายๆ ส่วนที่เกี่ยวข้องกันใน 1 กิจกรรมโดยจะแบ่งเป็นแนวตั้งคล้ายกับลูในสระว่ายน้ำ แต่ละลูคือแต่ละส่วน (ซึ่งอาจเป็นได้ทั้งบุคคล หน่วยงาน แผนก หรือสถานที่) ดังภาพที่ เป็นการแบ่งส่วนการทำงาน 3 ส่วน ได้แก่ ลูกค้า (Customer) งานขาย (Sales) และคลังสินค้า (Warehouse) ที่มีการทำงานเกี่ยวข้องกันในกิจกรรม “การขายสินค้า” โดยจากภาพกิจกรรมเริ่มต้นที่ “การสั่งซื้อสินค้า (Place an order)” ซึ่งเกิดขึ้นที่ส่วนของลูกค้า (Customer) จากนั้น เมื่อฝ่ายขาย (Sales) ได้รับคำสั่งซื้อจึงดำเนินการงานขาย (Process Order) ส่งเรื่องต่อไปยัง

ฝ่ายงานคลัง (Warehouse) เพื่อจัดทำสินค้าที่ลูกค้าต้องการ แล้วส่งสินค้าไปยังลูกค้า (Ship product) พร้อมทั้งที่ฝ่ายขายแนบเอกสารการเก็บเงิน (Bill Customer) ไปด้วย เมื่อลูกค้าได้รับสินค้า (Receive product) แล้ว ก็ทำการชำระเงินตามระยะเวลาที่ตกลงกันไว้ (Pay Bill) ฝ่ายขายก็สามารถปิดใบสั่งซื้อรายการนี้ได้ (Close Order) สำหรับ Activity Diagram ในรูปแบบนี้สามารถมี Decision ได้



ภาพที่ 3.27 Activity Diagram แบบแบ่งส่วนการทำงานด้วย Swimlanes

**** หมายเหตุ ****

Statechart Diagram และ Activity Diagram อาจจะไม่จำเป็นต้องมีอยู่ในการวิเคราะห์ระบบงานบางระบบก็ได้ แต่ในบางระบบงาน เช่น การสร้างเกมส์ ระบบงานด้านฮาร์ดแวร์ เป็นต้น อาจจะมีมีความจำเป็นต้องมี Statechart Diagram และ Activity Diagram เข้ามาช่วยเพื่อให้สามารถมองเห็นวิธีการทำงานได้ชัดเจนมากยิ่งขึ้น

บทที่ 4 ผลการทำงานของระบบ

นำเสนอผลการศึกษาข้อมูลหรือการทดลองการทำงานของเกม และการประเมินผลเกม

4.1 วิธีการทดสอบการเล่นเกม

ออกแบบวิธีการทดสอบการทำงานของเกม เพื่อตรวจสอบประสิทธิภาพและความสามารถในการทำงานของเกม ว่าบรรลุวัตถุประสงค์

4.2 ผลการทดสอบการทำงานของระบบ

อธิบายผลการทดสอบการทำงานของเกมตามวัตถุประสงค์ที่ตั้งไว้ โดยอาจมีตารางเปรียบเทียบการทำงาน ภาพประกอบ หรือวิธีการทางสถิติประกอบเพื่อให้การตีความข้อมูลชัดเจนขึ้น

4.3 อื่น ๆ

นอกจากนี้อาจใช้รูปแบบอื่นๆ เพื่ออธิบายผลการทำงานของเกม นอกเหนือจากที่ยกตัวอย่างได้ตามความเหมาะสม

บทที่ 5 สรุปและข้อเสนอแนะ

นำเสนอผลการค้นคว้าวิจัย โดยสรุปประเด็นสำคัญสาระสำคัญ หรือ ประจักษ์พยานผลของการศึกษาวิจัย ประกอบด้วย

5.1 การอภิปรายผลการศึกษา

วิจารณ์ผลการทดลอง หรือผลการทำงานของระบบ โดยมีจุดมุ่งหมายให้ผู้อ่านคล้อยตาม หลักการที่แสดงออกมาจากผลการทดลอง โดยอาจเปรียบเทียบกับผลการทดลองที่มีผู้เสนอมาก่อน

5.2 ปัญหาและอุปสรรค

อธิบายถึงปัญหาและอุปสรรคในการทำงานของระบบที่ทำการออกแบบ รวมทั้งข้อจำกัดของระบบที่ต้องคำนึงถึงในการใช้งาน

5.3 ข้อเสนอแนะ

นำเสนอแนวทางในการพัฒนาระบบที่ทำการออกแบบ สำหรับการศึกษาและค้นคว้าต่อไป

5.4 สรุปผลการศึกษา

สรุปผลการทดลองหรือผลการทำงานของระบบที่ทำการออกแบบ รวมทั้งประสิทธิภาพและความสามารถในการทำงานของระบบ