

บทที่ 6

USE CASE DIAGRAM

อ.สกรณ บุษบง
สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยราชภัฏบุรีรัมย์

OVERVIEW

- ส่วนประกอบของ **use case diagram**
- **Use case** กับ **Scenario**
- ตัวอย่างการเขียน **use case diagram**
- ข้อควรระวังในการเขียน **use case diagram**
- **Use case description**

การวิเคราะห์ความต้องการของระบบ โดยใช้ Use case Diagram

- มีวัตถุประสงค์เพื่ออธิบายเรื่องราวของ **Problem Domain** ทั้งหมด
- บอกความสัมพันธ์ของส่วนต่าง ๆ ในระบบ
- เป็นการเริ่มต้นการวิเคราะห์ระบบงานว่าสามารถทำอะไรให้กับผู้ใช้ได้บ้าง
- ช่วยให้ผู้พัฒนาระบบสามารถแยกแยะกิจกรรมที่อาจเกิดขึ้นในระบบ
- แต่ละสถานการณ์ที่ระบบสามารถบริการให้ผู้ใช้สำเร็จลุล่วงตามที่ต้องการได้ เรียกว่า **Use case**
- แต่ละ **Use case** จะแสดงสถานการณ์(**Scenario**) ที่สามารถบรรยายได้ว่าผู้ใช้ (**Actor**) ได้ต่อบกับระบบอย่างไร

Use Case Diagram

■ Use Case

- ความสามารถ/หน้าที่ของระบบ
- ใน 1 Use Case Diagram มักมีหลาย Use Case

■ Actor

- ผู้กระทำ/ผู้ใช้งาน Use Case นั้นๆ
- Actor ไม่จำเป็นต้องเป็นคนเสมอไป อาจจะเป็นสิ่งอื่นหรือระบบอื่นๆ ก็ได้ เช่น ระบบบัญชี ระบบโทรศัพท์ เป็นต้น

Use Case Diagram


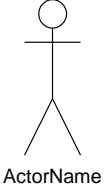

- Relationship

- เส้นแสดงความสัมพันธ์ระหว่าง Use Case กับ Actor




- System

- ระบบที่กำลังพัฒนา

Use Case Modeling : Core Elements

Construct	Description	Syntax
use case	A sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system.	
actor	A coherent set of roles that users of use cases play when interacting with these use cases.	
system boundary	Represents the boundary between the physical system and the actors who interact with the physical system.	

Use Case Modeling : Core Relationships

Construct	Description	Syntax
association	The participation of an actor in a use case. i.e., instance of an actor and instances of a use case communicate with each other.	
generalization	A taxonomic relationship between a more general use case and a more specific use case.	
extend	A relationship from an <i>extension</i> use case to a <i>base</i> use case, specifying how the behavior for the extension use case can be inserted into the behavior defined for the base use case.	

Use Case Modeling : Core Relationships

Construct	Description	Syntax
include	An relationship from a <i>base</i> use case to an <i>inclusion</i> use case, specifying how the behavior for the inclusion use case is inserted into the behavior defined for the base use case.	<pre><<include>> -----></pre>

Use Cases V.S. Scenario

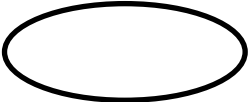
- Use Case (Class)

- ความสามารถ หรือ หน้าี่การทำงานองระบบ
- แต่ละ Use Case แทนชุดของ transactions ที่ระบบทำงานโต้ตอบกับ ผู้ใช้งาน หรือระบบอื่นๆ ภายนอก

- Scenario (Object)

- สถานการณ์ หรือตัวอย่างเรื่องราวการใช้งานระบบ
- Scenario จัดเป็น instance ของ use case

- เช่น


withdrawal cash



a user withdrawals
\$200

Use Cases V.S. Scenario

- **Use Case** จะอธิบายทุกกรณีที่เกิดขึ้นได้ทั้งหมด
- **Scenario** จะแสดงถึงเหตุการณ์จริงที่เกิดขึ้นภายใต้เงื่อนไขต่าง ๆ ของ **Use Case** ซึ่งอาจจะไม่ได้เกิดทุกกรณีที่ระบุไว้ใน **Use Case** ก็ได้
- จุดประสงค์ของการเขียน **Use Case Diagram** เพื่อเล่าเรื่องราวของ **Problem Domain** ทั้งหมดว่ามีส่วนประกอบอะไรบ้าง และเกี่ยวข้องกันเป็นระบบได้อย่างไร

ตัวอย่าง Use Case

- ผู้ใช้งานสอดบัตร **ATM** เข้าสู่เครื่องรับบัตร หากบัตรใช้งานได้จึงเข้าสู่หน้าจอ **Main Menu** หากใช้งานไม่ได้บัตร **ATM** จะถูกปล่อยคืน (**Reject**) ออกมา หากบัตรใช้ได้ ผู้ใช้งานต้องระบุประเภทบัญชีและจำนวนเงินที่ต้องการถอน หากมีเงินในบัญชีมากกว่าหรือเท่ากับจำนวนที่ระบุ ผู้ใช้งานสามารถนำเงินออกจากเครื่อง **ATM** ได้

ตัวอย่าง Scenario

Scenario ที่ 1

- นายสมชายสอดบัตร **ATM** ของ ธ.กรุงเทพ สาขาหาดใหญ่ แต่บัตรเสีย บัตรจึงถูก **reject** ออกมา

ตัวอย่าง Scenario

Scenario ที่ 2

- นางสนใจสอดบัตร **ATM** ของ ธ.ทหารไทย สาขาบางเขน บัตรสามารถใช้การได้ แต่เงินในบัญชีไม่พอจ่าย จึงไม่สามารถนำเงินไปใช้ได้

ตัวอย่าง Scenario

Scenario ที่ 3

- นายสมบัติสอดบัตร **ATM** ของ ธ.ทหารไทย สาขาบางเขน บัตรสามารถใช้งานได้ และมีเงินในบัญชีเพียงพอ เขาต้องการถอน 100 บาท และในบัญชีมีเงินจำนวน 250 บาท ดังนั้นนายสมบัติจึงสามารถนำเงินออกจากเครื่อง **ATM** ไปใช้ได้

Actors

- **Actor** หมายถึง **someone** หรือ **some thing** ที่มีการปฏิสัมพันธ์โต้ตอบกับระบบ
 - สิ่งใดก็ตามที่มีความต้องการในการแลกเปลี่ยน **information** กับระบบ หรือ สิ่งใดก็ตามที่อยู่ภายนอกระบบ และมีการใช้งาน **Use Case** ของระบบ
- ตัวอย่างของ **Actors**
 - **Customer -- maintain their account**
 - **Cashier -- verify withdrawal amount**



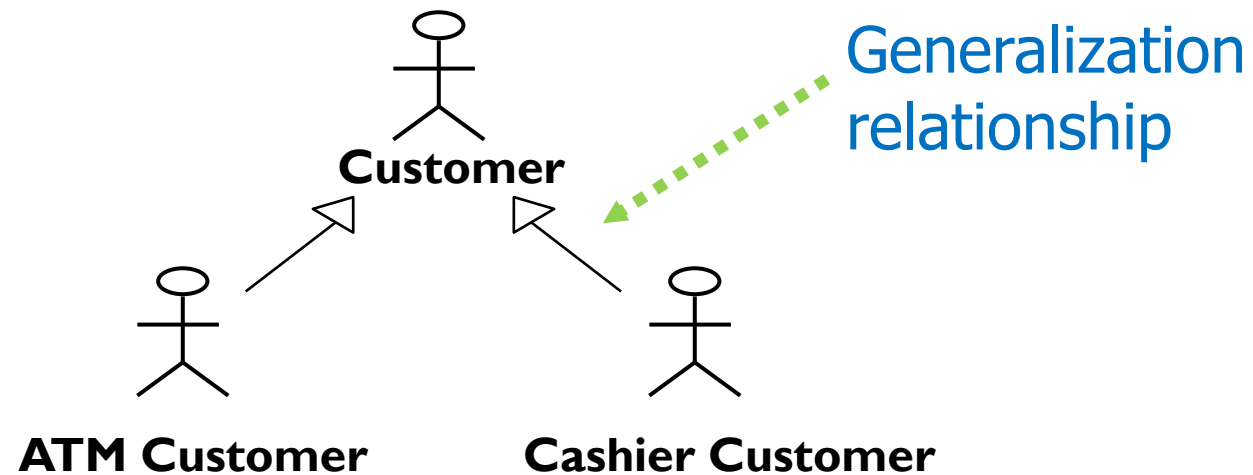
Customer



Cashier

Actors

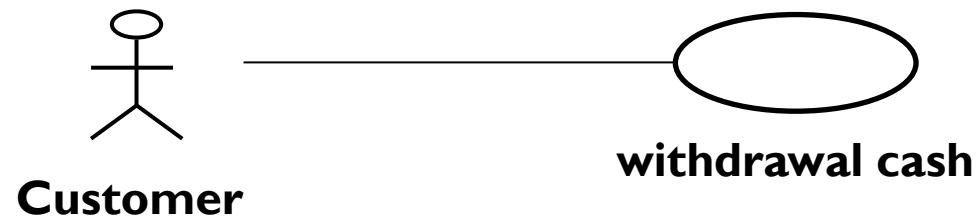
- Actors สามารถอธิบายโดยใช้ Generalization/Specialization Relationship



- อาจพิจารณา Actors เป็นคลาส ใน UML เนื่องจากมี relationships เช่นเดียวกับที่คลาสมี

Actors

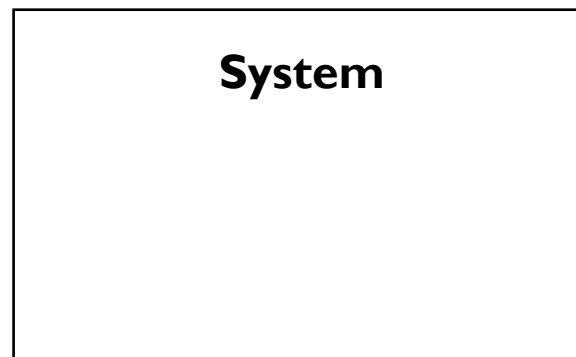
- เชื่อมต่อกับ **use cases** โดยใช้เส้นแสดงความเกี่ยวข้อง ปฏิสัมพันธ์ (**association**)
- **Association** = ความสัมพันธ์ที่มีการติดต่อสื่อสารกัน (ทั้งการรับ และส่ง **messages** ให้แก่กันและกัน)



- ไม่จำเป็นต้องอธิบายรายละเอียดของ **Association** เนื่องจากไม่มีการ **Implement** ส่วนของ **Actor** ในระบบ

System

- วัตถุประสงค์ใน **use-case modeling** คือ เพื่อบอกขอบเขตของระบบที่กำลังพัฒนา (**system boundary**) ว่าจะต้องประกอบด้วย **Use case** อะไรบ้าง
- ใช้สัญลักษณ์เป็นรูปสี่เหลี่ยมที่ล้อมรอบ **Use case** ไว้



ความสัมพันธ์ของ Use Case

■ Extend

- Use Case หนึ่งไปมีผลต่อการทำงานตามปกติของอีก Use Case หนึ่ง
- Use Case ที่ Extend จากอีก Use case หนึ่ง จะเป็น Use case ที่เกิดขึ้นในกรณีเดียวกันแต่เป็นกรณีที่พิเศษมากกว่า
- Use Case ที่มา Extend นั้นจะมีผลทำให้การดำเนินการของ Use Case ที่ถูก Extend ถูกรบกวนหรือมีการสะดุด หรือมีการเปลี่ยนกิจกรรมไป

ตย. Use Case Diagram ที่มี Extend

- **Use case diagram** ที่แสดงการรับโทรศัพท์ ซึ่งขณะที่รับโทรศัพท์ปกติ หากมีสายเรียกซ้อนเข้ามา อาจทำให้ต้องมีการรับสายเรียกซ้อนก่อน ซึ่งทำให้การรับสายโทรศัพท์ตามปกติต้องชะงักชั่วคราว

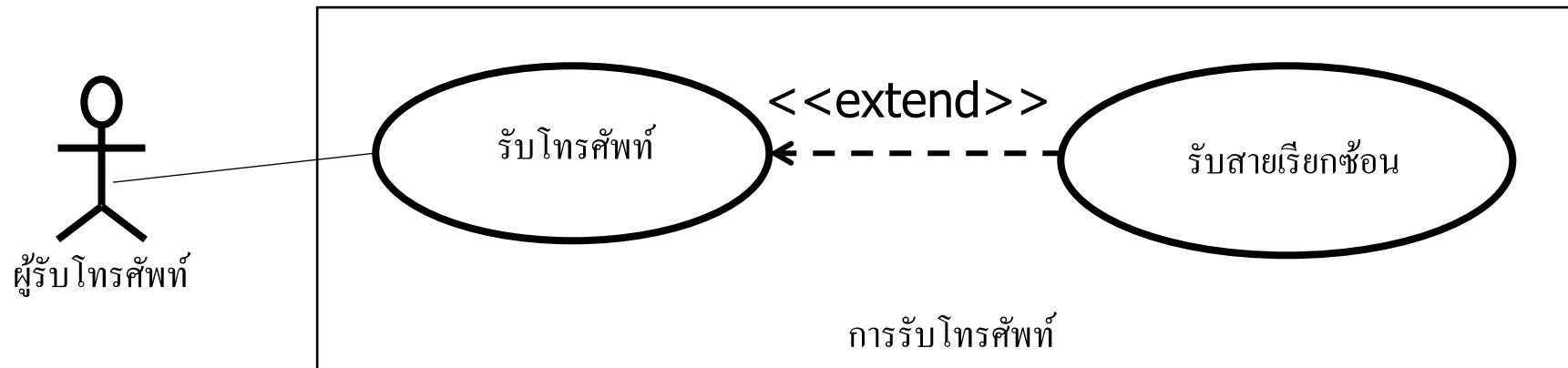
ขั้นตอนที่ 1 : หา Use Case และ Actor ของระบบ

- Use case ของระบบคือ
 - การรับโทรศัพท์
 - การรับสายเรียกซ้อน
- actor ของระบบคือ
 - ผู้รับโทรศัพท์

ขั้นตอนที่ 2 : เขียน Scenario ของระบบ

- **scenario ที่ 1 : เกิดสายเรียกซ้อน**
 - เมื่อเกิดสายเรียกซ้อน ทำให้ **use case** การรับโทรศัพท์ เกิดการชะงักงัน ซึ่งผู้รับอาจหยุดการสนทนาชั่วคราว
 - หรือผู้รับเปลี่ยนไปรับสายที่เรียกซ้อนแทน
- **scenario ที่ 2 : ไม่เกิดสายเรียกซ้อน**

ขั้นตอนที่ 3 : เขียน Use Case Diagram



ความสัมพันธ์ของ Use Case (ต่อ)

- Include/Use
- Use case ที่ถูก **includes** จะถูกรวมการทำงานเข้าไว้กับ Use case ที่เป็นตัวหลักด้วยเสมอ
- คล้ายกับการเรียกใช้งานโปรแกรมย่อยโดยโปรแกรมหลัก

ตย. Use Case Diagram ที่มี Include/Use

- **use case diagram** เพื่ออธิบายการตรวจสอบ **user** ที่เข้ามาในระบบคอมพิวเตอร์ขององค์กรต่าง ๆ ต้องมีการตรวจสอบรหัสผ่านรวมอยู่ด้วย โดย **actor** ของระบบนี้คือผู้จัดการระบบ

ขั้นตอนที่ 1 : หา Use Case และ Actor ของระบบ

- Use Case ของระบบคือ
 - การตรวจสอบ User (Validate User)
 - การตรวจสอบรหัสผ่าน (Check Password)
- Actor ของระบบคือ
 - ผู้จัดการระบบ (System Administrator)

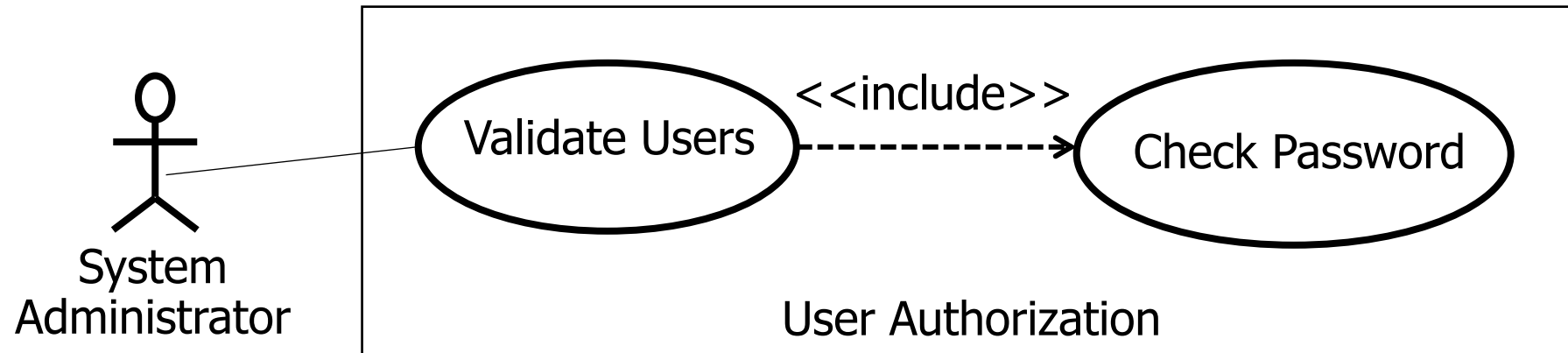
ขั้นตอนที่ 2 : เขียน Scenario ของระบบ

- scenario ที่ 1 : user ป้อน password ที่ถูกต้อง
 - การตรวจสอบ password ใน use case ชื่อ check password ตรวจสอบได้ถูกต้อง ทำให้กิจกรรมใน validate user ดำเนินต่อไปได้

ขั้นตอนที่ 2 : เขียน Scenario ของระบบ

- scenario ที่ 2 : user ป้อน password ที่ไม่ถูกต้อง
- ทำให้ use case ชื่อ check password ถูกเรียกใช้อีกหลายครั้ง จนกว่าจะถูก หรือจนกว่าจะครบ 3 ครั้ง จึงตัด user คนนั้นออกจากระบบ

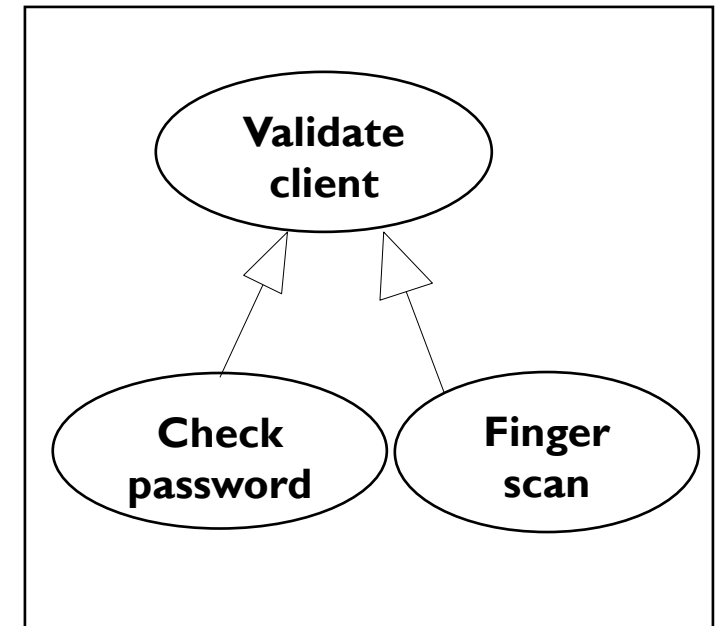
ขั้นตอนที่ 3 : เขียน Use Case Diagram



ความสัมพันธ์ของ Use Case (ต่อ)

- **Generalization**

- **Child Use case** รับถ่ายทอดคุณสมบัติมาจาก **Parent Use Case**
- **Child** สามารถเปลี่ยนแปลงพฤติกรรมที่รับจาก **Parent** หรือเพิ่มเติมพฤติกรรม
- **Child** อาจนำไปแทนที่ ในที่ๆ **Parent** ปรากฏ



ตัวอย่าง การเขียน Use Case Diagram

- จงสร้าง **use case diagram** เพื่ออธิบายการลงทะเบียนของนักเรียน ซึ่งเกิดจากผลของการวิเคราะห์ความต้องการเบื้องต้น สามารถเขียนเป็นรายการได้ดังนี้

ความต้องการ

- ในแต่ละภาคการศึกษานักศึกษาจะมีการลงทะเบียน
- การลงทะเบียนในแต่ละครั้งจะมีการเก็บหลักฐานและชำระค่าลงทะเบียนเรียน
- ซึ่งการลงทะเบียนเรียนจะเสร็จสิ้นได้ก็ต่อเมื่อหลักฐานที่ได้รับมาครบถ้วนถูกต้อง
- และในขณะเดียวกันเงินค่าลงทะเบียนเรียนที่เรียกเก็บได้ก็ต้องมีจำนวนครบถ้วนด้วย
- เจ้าหน้าที่ของสถาบันการศึกษาจะเป็นจัดเก็บหลักฐาน

ความต้องการ...

- สำหรับนักศึกษาบางคนที่ได้รับสิทธิพิเศษเช่น
 - ได้รับทุนเรียนฟรี
 - เป็นนักกีฬาของสถาบัน
 - หรือเป็นผู้ทำชื่อเสียงให้สถาบัน
- จะมีสิทธิได้รับยกเว้นค่าเล่าเรียนในบางภาคการศึกษา

หา Use Case ของระบบ

- **use case** ของระบบคือ
 - การลงทะเบียนเรียน
 - การเก็บหลักฐาน
 - การชำระค่าเล่าเรียน

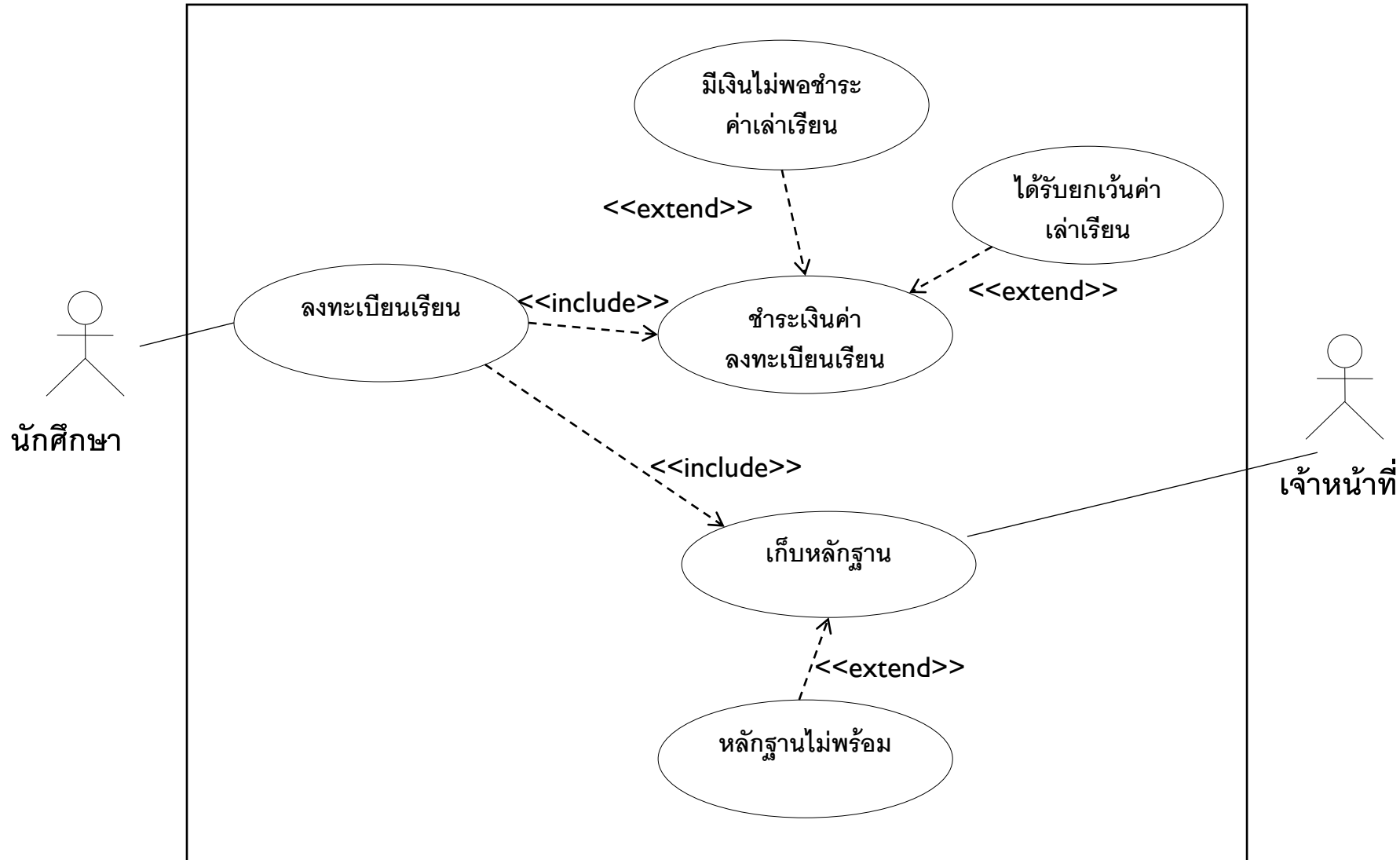
หา Use Case อื่นที่เกี่ยวข้อง

- หา **use case** อื่นที่เกี่ยวข้องคือ
 - การเก็บหลักฐาน
 - หลักฐานไม่พร้อม
 - การชำระค่าเล่าเรียน
 - มีเงินไม่พอชำระค่าเล่าเรียน
 - ได้รับการยกเว้นค่าเล่าเรียน

หา Actor ของระบบ

- Actor ของระบบคือ
 - เจ้าหน้าที่
 - นักศึกษา

เขียน Use Case Diagram การลงทะเบียนเรียนของนักเรียน



Use Case Diagram ทำเมื่อไรและทำอย่างไร

- Requirements capture
 - ใช้ในการกำหนด Requirement ของระบบ
 - สร้างแบบจำลอง (Model) ของ User requirements ด้วย Use Case
- Test Scenarios
 - สร้างแบบจำลอง (Model) ของสถานการณ์การทดสอบระบบ (test scenarios) ด้วย Use Case
- Use Case: ระบุสิ่งที่ต้องการให้มีในระบบ
 - ตั้งชื่อให้ Use Case
 - เขียนคำอธิบายสั้นๆ
 - เพิ่มรายละเอียดในภายหลัง

สรุปการสร้าง Use Case Diagram

- ระบุ **actors** ที่มีปฏิสัมพันธ์กับระบบ
- พิจารณาแนวทางของระบบ ในการปฏิสัมพันธ์กับ **actors**
- จำแนกพฤติกรรมของระบบใน การปฏิสัมพันธ์กับ **actors** ให้เป็น **use cases** โดยกำหนดความสัมพันธ์ระหว่าง **Use Case**

ข้อควรระวังในการเขียน Use Case Diagram

- **Use Case** ใช้เป็นเครื่องมือสื่อสารกับผู้ใช้ ดังนั้นต้องให้ง่ายที่สุดเท่าที่จะทำได้
- ไม่ควรใช้ `<<include>>`, `<<extend>>` และ **inheritance** โดยที่ไม่จำเป็น
- จำนวน **Use Case** ไม่ควรเกิน 20 Use Case โดยนับจาก Use Case ที่ไม่มีความสัมพันธ์

Use case Description

- เนื่องจากรายละเอียดที่ปรากฏใน **Use case Diagram** เป็นเพียงภาพรวมของระบบ แต่ยังไม่มีความอธิบายการทำงานของแต่ละ **Use case**
- การเขียน **Requirement Specification** ให้สมบูรณ์จะเพิ่มเติมส่วนที่เป็น **Use case Description** เข้าไปเพื่อให้ทีมพัฒนาเข้าใจ **Use case** ได้ดียิ่งขึ้น
- **Use case Description** ไม่ได้เป็นมาตรฐานของ **UML**

Use case Description

- ชื่อของ Use Case
- ภาพรวมของการทำงาน (Overview)
- Actor หลัก (Primary Actor)
- Actor รอง (Secondary Actor)
- จุดเริ่มต้น (Starting Point)
- จุดสิ้นสุด (End point)
- การทำงานของ Use Case (Flow of Events)
- การทำงานของ Use Case เมื่อมีปัญหาเกิดขึ้น (Alternative flow of Events)
- ผลของการทำงานของ Use Case (Measurable Result)

Use Case : Create Order

- ภาพรวมของการทำงาน (Overview)
 - จุดประสงค์หลักของ Use Case นี้ เพื่อทำการลงข้อมูลในใบสั่งซื้อสินค้าจากลูกค้า
- Actor หลัก (Primary Actor)
 - ตัวแทนฝ่ายขายสินค้า
- Actor รอง (Secondary Actor)
 - ไม่มี
- จุดเริ่มต้น (Starting Point)
 - Use Case ตัวนี้เริ่มต้นเมื่อ Actor ตัวแทนฝ่ายขายสินค้าขอให้ระบบทำการลงข้อมูลการสั่งซื้อสินค้า
- จุดสิ้นสุด (End point)
 - คำขอเพื่อทำการลงข้อมูลการสั่งซื้อสินค้าเสร็จสิ้นสมบูรณ์หรือไม่ก็ถูกยกเลิก

Use Case : Create Order

- การทำงานของ Use Case (Flow of Events)
 - จาก User Interface บนจอเพื่อทำการลงข้อมูลการสั่งซื้อ Actor จะต้องทำการใส่ข้อมูลเกี่ยวกับการสั่งซื้อ เป็นต้นว่า วันที่ลูกค้าต้องการให้สินค้าส่งมอบถึงมือ (Required Date) ปริมาณที่ต้องการสั่งซื้อ (Quantity) ต้องการให้ส่งมอบสินค้าโดยบริษัทส่งสินค้าไหน (ShipVia) ต้องการให้ให้ส่งมอบสินค้าที่ไหน (ShipAddress) หลังจากนั้นแล้ว Actor สามารถเลือกที่จะทำการบันทึกข้อมูลลงไปในฐานข้อมูล หรือยกเลิกการทำงานทั้งหมด ถ้า Actor เลือกทำการบันทึก ใบสั่งซื้อใบใหม่ก็จะถูกสร้างขึ้นมาในฐานข้อมูล

Use Case : Create Order

- การทำงานของ Use Case เมื่อมีปัญหาเกิดขึ้น (Alternative Flow of Events)
 - ถ้าไม่มีสินค้าที่ต้องการอยู่ในคลังสินค้า ระบบจะต้องแจ้งให้ Actor ทราบพร้อมกันนั้นก็ต้องยกเลิกการทำงานที่เหลือของ Use Case นี้
- ผลของการทำงานของ Use Case (Measurable Result)
 - จะมีใบสั่งซื้อสินค้าใหม่ 1 ใบขึ้นมาในระบบ

Use Case	Create Order
Overview	จุดประสงค์หลักของ Use Case นี้ เพื่อทำการลงข้อมูลในใบสั่งซื้อสินค้าจากลูกค้า
Primary Actor	ตัวแทนฝ่ายขายสินค้า
Secondary Actor	ไม่มี
Starting Point	Use Case ตัวนี้เริ่มต้นเมื่อ Actor ตัวแทนฝ่ายขายสินค้าขอให้ระบบทำการลงข้อมูลการสั่งซื้อสินค้า
End point	คำขอเพื่อทำการลงข้อมูลการสั่งซื้อสินค้าเสร็จสิ้นสมบูรณ์หรือไม่ก็ถูกยกเลิก
Flow of Events	จาก User Interface บนจอเพื่อทำการลงข้อมูลการสั่งซื้อ Actor จะต้องทำการใส่ข้อมูลเกี่ยวกับการสั่งซื้อ เป็นต้นว่า วันที่ลูกค้าต้องการให้สินค้าส่งมอบถึงมือ (Required Date) ปริมาณที่ต้องการสั่งซื้อ (Quantity) ต้องการให้ส่งมอบสินค้าโดยบริษัทส่งสินค้าไหน (ShipVia) ต้องการให้ให้ส่งมอบสินค้าที่ไหน (ShipAddress) หลังจากนั้นแล้ว Actor สามารถเลือกที่จะทำการบันทึกข้อมูลลงไปไว้ในฐานข้อมูล หรือยกเลิกการทำงานทั้งหมด ถ้า Actor เลือกทำการบันทึก ใบสั่งซื้อใบใหม่ก็จะถูกสร้างขึ้นมาจากในฐานข้อมูล
Alternative Flow of Events	ถ้าไม่มีสินค้าที่ต้องการอยู่ในคลังสินค้า ระบบจะต้องแจ้งให้ Actor ทราบพร้อมกันนั้นก็ต้องยกเลิกการทำงานที่เหลือของ Use Case นี้
Measurable Result	จะมีใบสั่งซื้อสินค้าใหม่ 1 ใบขึ้นมาในระบบ

การบ้าน

- จงเขียน **use case diagram** ของการซื้อสินค้าจากร้านค้าแห่งหนึ่ง โดยสั่งได้ตามจำนวนสินค้าคงเหลือที่มีอยู่ในคลังเท่านั้น การซื้อสินค้านี้สามารถชำระเงินด้วยบัตรเครดิตหรือเงินสดก็ได้

THANK