SUT
SURANAREE UNIVERSITY OF TECHNOLOGY

INSTITUTE OF
ENGINEERING

ENG23 3031 & 523331

System Analysis and Design

การวิเคราะห์และออกแบบระบบ

1 / 2567

Lecture 3: Use Case Analysis

Goal

1. สามารถนำ User Requirement และ User Story จากที่ได้เก็บมา มาเขียนให้อยู่ในรูปแบบของ Business Use Case Diagram ได้
2. สามารถนำ Business Use Case ของตนเองมาวิเคราะห์เพื่อหา System Use Case ที่ต้องทำต่อได้
3. สามารถเขียน System Use Case Diagram ที่อธิบายการทำงานของระบบของตนเองได้

Use Case ช่วยย่อย requirements ให้เป็น story

User Story

ในบทบาทของ

ฉันต้องการ

เพื่อ

Business Use Case Diagram

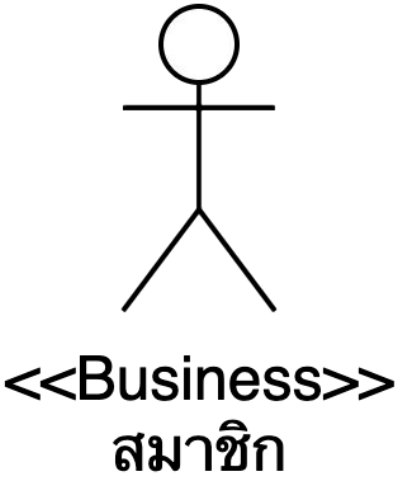
จับกระบวนการ manual

จินตนาการว่าเป็นโลกที่ยังไม่มี computer ใช้

รวม business use case ทั้งหมดเข้าด้วยกันเป็นระบบหลักที่สมบูรณ์ขึ้น

Business Actor

ผู้ใช้งาน หรือบุคคลที่เกี่ยวข้องกับระบบ โดย actor จะเป็นบุคคลหรือระบบภายนอกอื่นๆก็ได้



Use Case

งานหรือเป้าหมายเฉพาะที่ระบบต้องบรรลุ โดยผู้ที่กำหนดหรือต้องการบรรลุเป้าหมายนั้นคือผู้ใช้งานหรือ Actor

Relationship

ความสัมพันธ์ระหว่างผู้ใช้ (actor) และระบบ (Use Case) แสดงด้วยเส้นตรง เรียกว่า association line

Bussiness Use Case

ขึ้นต้นด้วยคำ **กริยา**

แทนการกระทำทาง business ที่จบในตัว

ได้ output ตาม requiremments ที่ต้องการตามระบบย่อยๆนั้น ๆ

Recap: OneSound

ระบบที่เกิดขึ้นจาก Requirement

1. ระบบบันทึกการฟัง
2. ระบบให้คะแนนไฟล์เสียง
3. ระบบจัดการ Playlist -> ระบบสร้าง Playlist
4. ระบบจัดการไฟล์เสียง

User Story: ระบบจัดการ Playlist

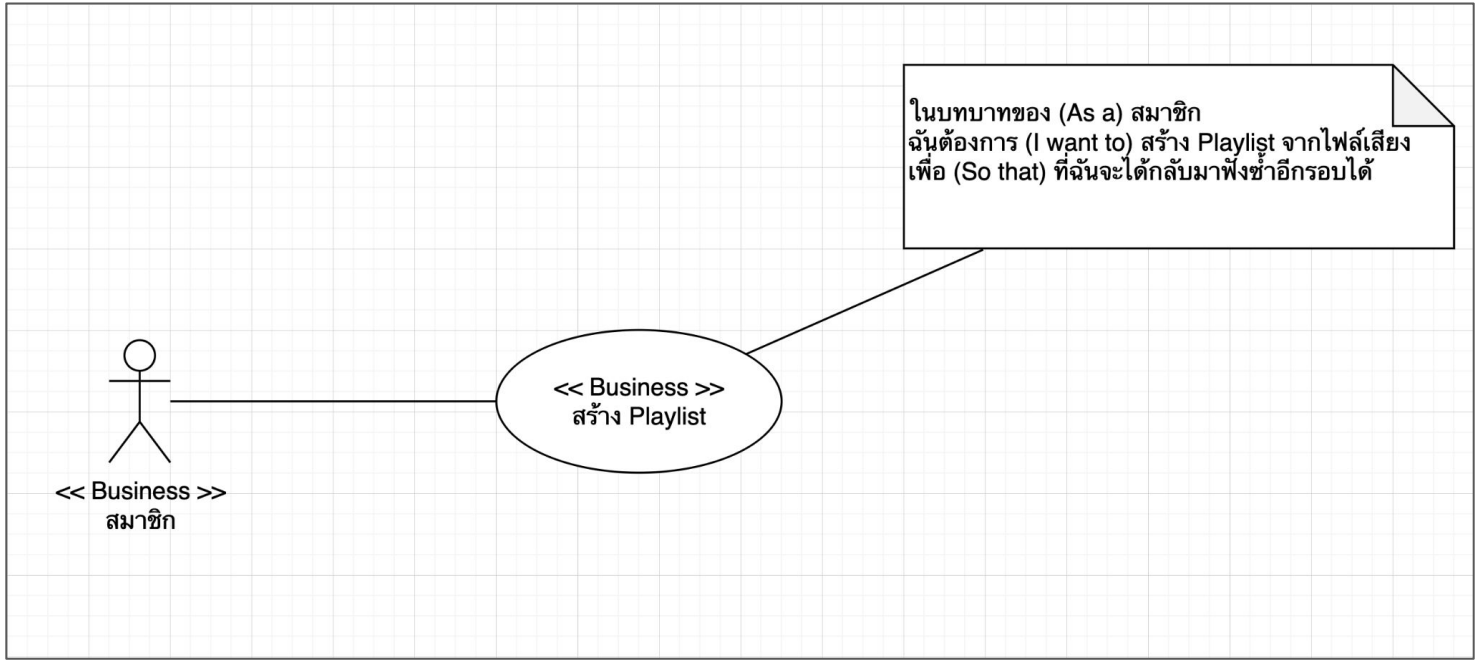
ในบทบาทของ (As a) สมาชิก

ฉันต้องการ (I want to) สร้าง Playlist จากไฟล์เสียง

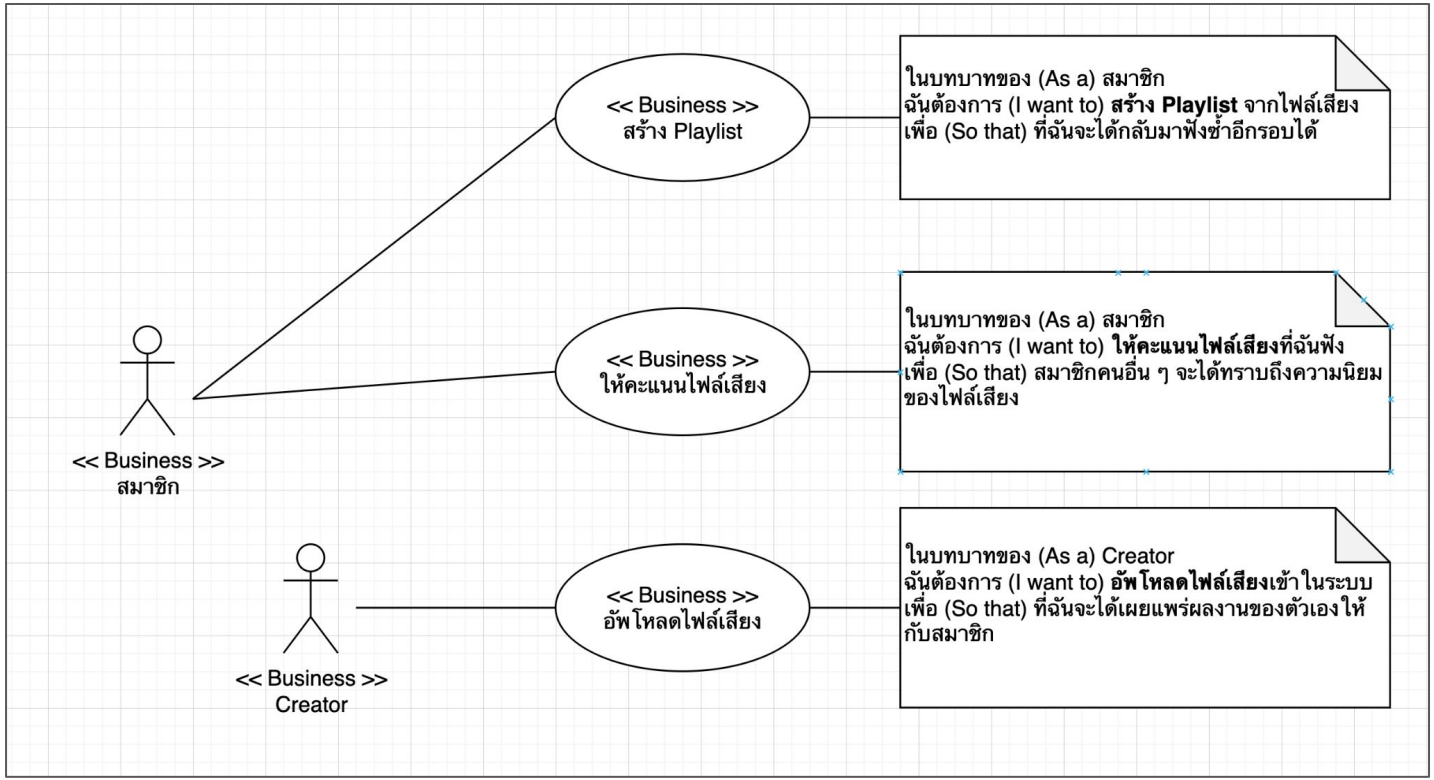
เพื่อ (So that) ที่ฉันจะได้กลับมาฟังซ้ำอีกรอบได้

Business Use Case Diagram (แบบเดี่ยว)

Turn story into UML for details



Business Use Case Diagram (แบบรวม)

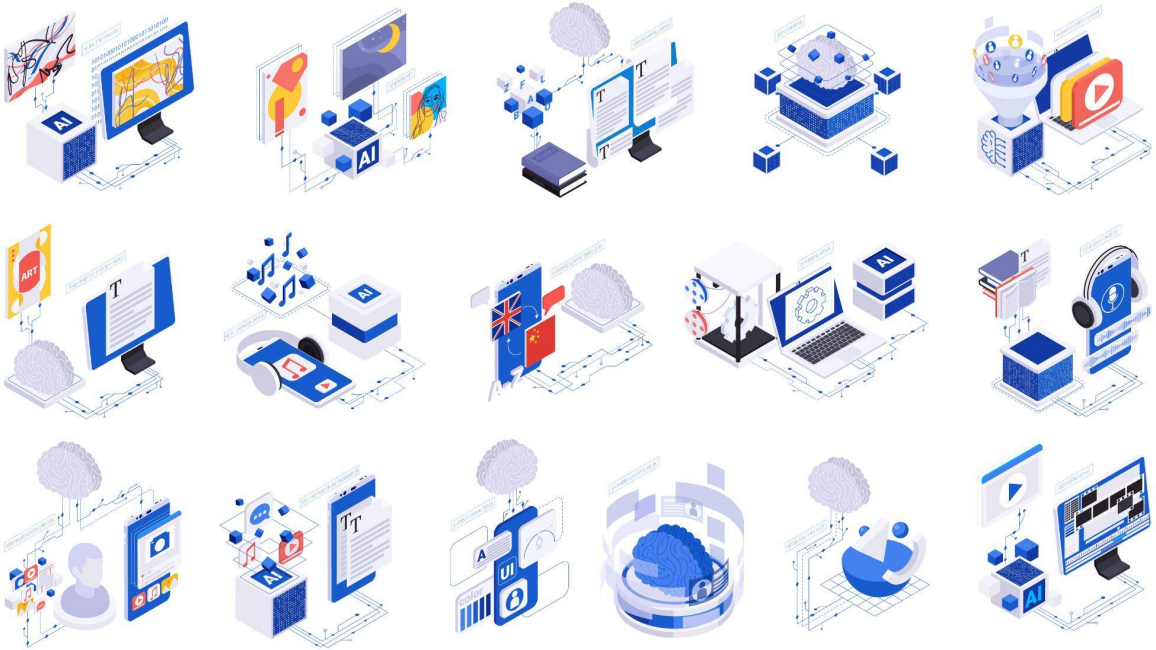


Checklist: Business Use Case Diagram

1. Business Actor มี <<Business>> กำกับ
2. Business Actor เป็นชื่อบทบาทเท่านั้น
3. Business Use Case มี <<Business>> กำกับ
4. Business Use Case ขึ้นต้นด้วยคำที่แสดงพฤติกรรม
5. พฤติกรรมของ Business Use Case ต้องมาจาก User Story
6. มี Note แสดง User Story

System Use Case

The use case that from requirement that specific for the system on the things need to deve



ความสัมพันธ์ระหว่าง Use Case ด้วยกันเอง

เมื่อ Use Case โต้ตอบหรือมีความสัมพันธ์ระหว่างกันเอง จะใช้สัญลักษณ์เป็นเส้นประแบบมีหัวลูกศร พร้อมทั้งเขียนชื่อรูปแบบความสัมพันธ์ไว้กลางเส้นประนั้น คลุมด้วยเครื่องหมาย

<< >> (นอกจากนั้นจะเห็นว่ามีการเขียนชื่อความสัมพันธ์บนเส้นด้วยบ่อยครั้ง)

<<include>>

ความสัมพันธ์แบบ <<include>> คือ ความสัมพันธ์ที่ Use Case หนึ่งต้องพึ่งพาความสามารถหรือฟังก์ชั่นอื่นๆของ Use Case อื่น กล่าวได้ว่าการทำงานของ Use Case หนึ่งเป็นส่วนหนึ่งของการทำงานของอีก Use Case หนึ่ง โดยเรียก Use Case ที่เป็นส่วนหนึ่งของ Use Case อื่นว่า Base Use Case และ Use Case ที่ถูกดึงมาใช้ว่า Include Use Case การวาดความสัมพันธ์ คือ ลากเส้นประจาก Base Use Case หันลูกศรชี้ไปที่ Include Use Case และเขียนชื่อความสัมพันธ์ไว้ตรงกลาง

- - - - - - <<Include>> - - - - ->

<<extend>>

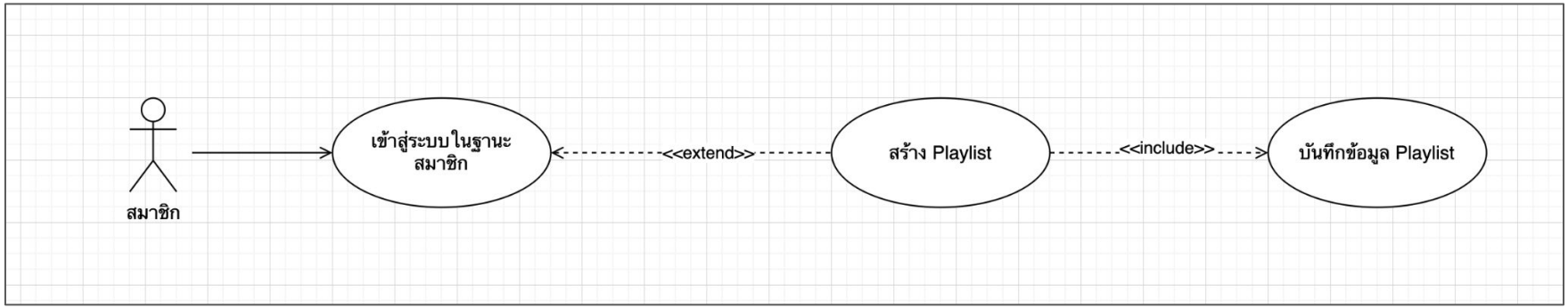
ความสัมพันธ์แบบ <<extend>> คือ ความสัมพันธ์ที่ Use Case หนึ่งถูกกระตุ้นหรือรบกวนด้วยบางเหตุการณ์ จนอาจทำให้การทำงานของ Use Case นั้นเปลี่ยนไป หรือกรณีที่ Use Case หนึ่งต้องพึ่งพาคุณสมบัติหรือความสามารถของ Use Case ในการบรรลุการทำงานของตัวเอง เราสามารถเขียนสิ่งที่กระตุ้นหรือสิ่งที่ต้องพึ่งพาคุณสมบัติของ Use Case อื่นได้ในรูปแบบของ Use Case อีกหนึ่งอัน เรียกว่า Extending Use Case ส่วน Use Case ที่ถูกกระตุ้นหรือถูกรบกวนเรียกว่า Base Use Case

<- - - - <<Extend>> - - - - -

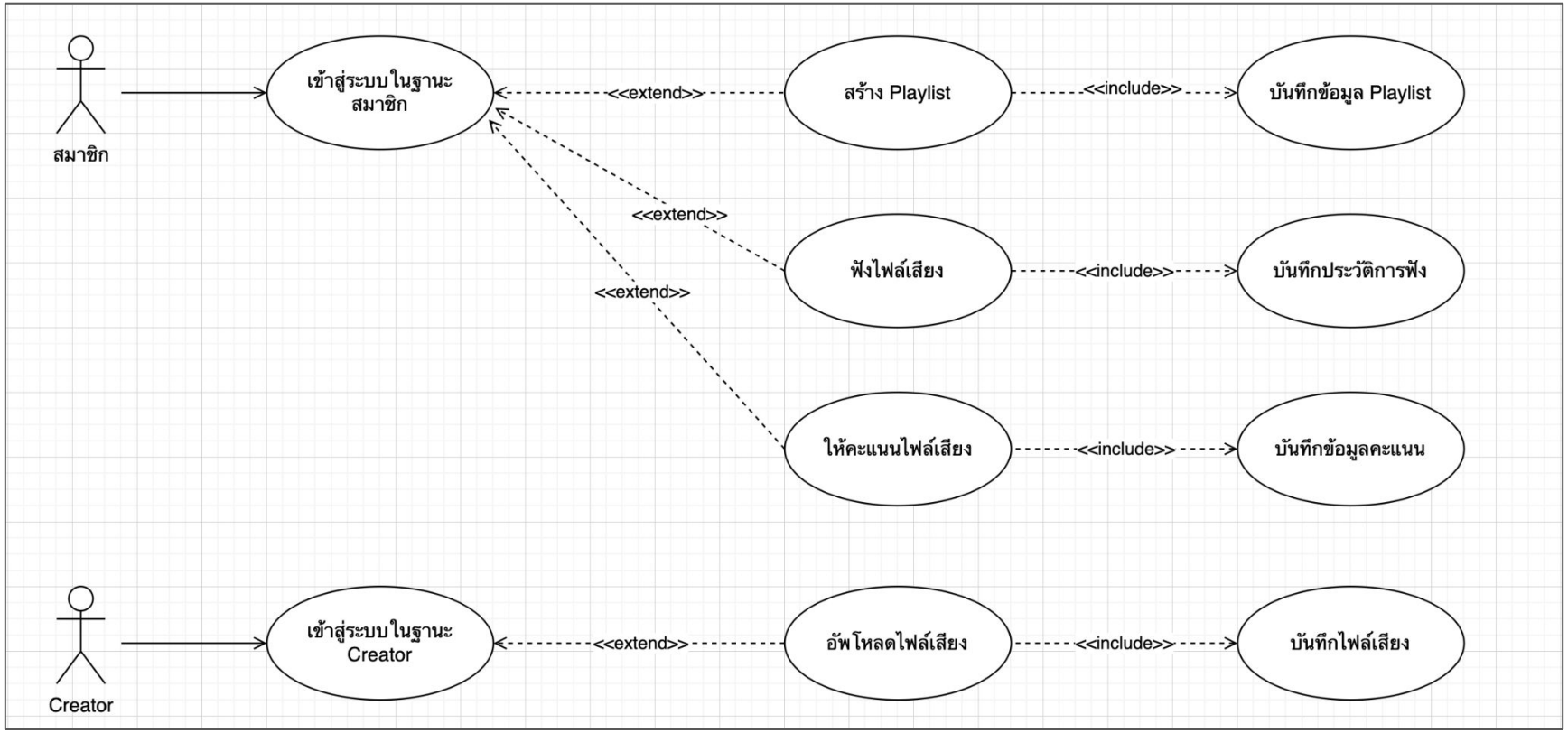
System Use Case Diagram

From Business into System

- ในบทบาทของ (As a) สมาชิก
  ฉันต้องการ (I want to) สร้าง Playlist จากไฟล์เสียง
  เพื่อ (So that) ที่ฉันจะได้กลับมาฟังซ้ำอีกรอบได้



System Use Case Diagram



System Use Case Diagram

Consider Keys

1. Can Business Actor be System Actor?
2. Can Business Use Case can turn into Activities inside the system?
3. To perform this use case, Do we need any security process?
4. Do we need additional use case to make Business Use Case complete?

Checklist: System Use Case Diagram

1. System Actor และ System Use Case ต้องไม่มีอะไรกำกับ
2. เส้นโยงจาก System Actor ไปหา System Use Case จะต้องเป็นเส้นทึบ หัวลูกศรปลายเปิดชี้ไปหา System Use Case
3. System Actor ต้องเป็นบทบาทของคนที่ใช้ระบบจริง ๆ
4. System Use Case ต้องเป็นคำที่แสดงพฤติกรรม
5. ถ้ามีกลไลทาง Security มาเกี่ยว ชื่อ System Use Case จะอยู่ในรูปแบบ **“เข้าระบบในฐานะ <Actor>”** เท่านั้น
6. การใช้ **<---<<extend>>--->** ต้องเป็นเส้นประ หัวลูกศรปลายเปิด ชี้จาก System Use Case ทางเลือกไปยัง System Use Case หลัก
7. การใช้ **<---<<include>>--->** ต้องเป็นเส้นประ หัวลูกศรปลายเปิด ชี้จาก System Use Case ที่ใหญ่กว่า ไปยัง System Use Case ที่จะถูกรวมเข้ามา