

OOAD & UML

วิชาวิเคราะห์และออกแบบระบบเชิงวัตถุ
Object-Oriented Analysis and Design

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส (Requirement Analysis and Use case Model)



โดย ผู้ช่วยศาสตราจารย์ ดร. บัณฑิต ส่องเนียม
เว็บไซต์ : <http://www.siam2dev.com>
อีเมล : siam2dev@hotmail.com

สาขาวิชา วิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏพระนคร



บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส (Requirement Analysis and Use case Model)



สัญลักษณ์ที่ผ่านมาสำหรับการเรียน ต่อไปนี้จะใช้
UML เป็นมาตรฐาน

บทที่ 5 การวิเคราะห์ความต้องการและแบบจำลองยูสเคส

Agenda

5.1 บทนำ

5.1.1 ความสำคัญของ Requirement Analysis

5.1.2 เหตุผลที่โครงการซอฟต์แวร์ล้มเหลวจาก Requirement ที่ไม่ชัดเจน

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis แบบสั้น

5.2 ความต้องการ (Requirement)

5.2.1 Requirement คืออะไร

5.2.2 User Requirement

5.2.3 System Requirement

5.2.4 Requirement Specification / SRS

5.3 แหล่งที่มาของความต้องการ

- System Owner / Sponsor
- End User
- Power User
- Administrator
- Executive User
- External User
- เหตุผลที่ต้องแยกกลุ่มผู้ใช้

บทที่ 5 การวิเคราะห์ความต้องการและแบบจำลองยูสเคส

5.4 กระบวนการวิเคราะห์ความต้องการ

- Information Gathering
- Requirement Identification
- Requirement Selection / Scope / Problem Domain
- Requirement Classification and Structuring
- Prioritization and Negotiation
- Requirement Validation
- Requirement Specification

5.5 เทคนิคการเก็บรวบรวมข้อมูล

- Observation
- Interview
- Questionnaire
- Document Review
- Workshop / JAD
- Prototyping
- Agile / Scrum

5.6 ประเภทของความต้องการ

- Functional Requirement
- Non-functional Requirement
- Product / Organizational / External Requirement
- ตัวอย่างการแยกประเภท

บทที่ 5 การวิเคราะห์ความต้องการและแบบจำลองยูสเคส

5.7 การเขียน Requirement Specification

- รูปแบบ ReqID
- ชื่อ Requirement
- Actor / ผู้เกี่ยวข้อง
- รายละเอียด
- Priority
- Dependency
- Test Criteria
- ตัวอย่างระบบโรงแรม / คาร์แคร์ / โรงพยาบาล

5.8 Requirement Validation

- Validity
- Consistency
- Completeness
- Realism
- Verifiability

5.9 การวิเคราะห์เชิงวัตถุด้วย Use Case Model

- Actor
- Use Case
- System Boundary
- Association
- Include
- Extend
- Generalization
- วิธีแปล Requirement เป็น Use Case

บทที่ 5 การวิเคราะห์ความต้องการและแบบจำลองยูสเคส

5.10 Use Case Description

- Use Case Name
- Actor
- Preconditions
- Main Flow
- Alternative Flow
- Exception Flow
- Postconditions
- Business Rules

5.11 ตัวอย่าง Use Case Diagram

- ระบบซื้อขายสินค้า
- ระบบลงทะเบียน
- ระบบจองห้องพัก
- ระบบร้านค้าออนไลน์

5.12 แบบฝึกหัด / งานกลุ่ม

- วิเคราะห์ผู้ใช้
- เขียน Requirement
- แยก Functional/NFR
- จัด Priority
- เขียน Use Case Diagram
- เขียน Use Case Description

Project งานกลุ่ม

- กลุ่มละ ... คน
- วิเคราะห์และออกแบบระบบ ใช้หลักการ OOAD ..
- **รูปเล่มรายงาน**
 - ปกหน้า หลัง
 - คำนำ
 - สารบัญ
 - **บทที่ 1**
 - หลักการและเหตุผล /ที่มาและความสำคัญของปัญหา
 - วัตถุประสงค์
 - ขอบเขต
 - Glossary คำศัพท์
 - แผนการดำเนินงาน Gantt Chart
 - เครื่องมือที่ใช้
 - ประโยชน์ที่คาดว่าจะได้รับ
 - **บทที่ 2 Unified Process**
 - กระบวนการ
 - ตำแหน่ง ความรับผิดชอบ
 - **บทที่ 3 การวิเคราะห์ระบบ**
 - **Business Rule**
 - การวิเคราะห์ความต้องการ SRS
 - Use case Diagram
 - Class Diagram
 - Sequence Diagram / Collaboration Diagram
 - Activity Diagram
 - **บทที่ 4 การออกแบบระบบ**
 - การออกแบบหน้าจอโปรแกรม Layout / Wire Frame , Mockup
 - System Architecture
 - Component Diagram
 - Deployment Diagram
 - **บทที่ 5 สรุป**
 - **บรรณานุกรม**

1.

รูปเล่ม และ
Google
Drive

ส่งวันสุดท้าย
วันสอบปลายภาค

2.

ทำเป็น Clip Video ลง Youtube

- **แนะนำงาน**
 - ชื่อเรื่อง
 - ส่งงาน อ. ดร. นัฐพงศ์ ส่งเนียม
 - รูป
 - เป็นส่วนหนึ่งของราย วิชา ..รหัสวิชา
 - แนะนำสาขา ..คณะ .. ม.
 - แนะนำสมาชิกในกลุ่ม
- **อธิบายงาน**
 - หลักการและเหตุผล
 - อธิบายการวิเคราะห์ระบบ
 - อธิบายการออกแบบระบบ
- **สรุป**

10 -20 นาที

ส่งเข้า Google drive งานกลุ่มของรายวิชา

บทที่ 5

5.1 บทนำ

การวิเคราะห์ความต้องการ

การวิเคราะห์ความต้องการ หรือ Requirement Analysis เป็นขั้นตอนสำคัญในกระบวนการวิเคราะห์และออกแบบระบบเชิงวัตถุ เพราะเป็นขั้นตอนที่ช่วยให้ทีมพัฒนาระบบเข้าใจว่า ผู้ใช้ต้องการอะไร ระบบควรทำอะไร และขอบเขตของระบบอยู่ตรงไหน ก่อนเข้าสู่การออกแบบและพัฒนาระบบจริง หากวิเคราะห์ความต้องการไม่ครบถ้วนหรือไม่ชัดเจน อาจทำให้ระบบที่พัฒนาขึ้นไม่ตรงกับความต้องการของผู้ใช้ ส่งผลให้ต้องแก้ไขระบบหลายครั้ง เสียเวลา งบประมาณ และอาจทำให้โครงการล้มเหลวได้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.1 ความสำคัญของ Requirement Analysis

Requirement Analysis มีความสำคัญเนื่องจากเป็นพื้นฐานของการพัฒนาระบบทั้งหมด โดยทำหน้าที่เชื่อมโยงระหว่างผู้ใช้ เจ้าของระบบ นักวิเคราะห์ระบบ และทีมพัฒนา เพื่อให้ทุกฝ่ายเข้าใจตรงกันเกี่ยวกับสิ่งที่ระบบต้องทำ ความสำคัญหลัก ได้แก่

1. ช่วยให้เข้าใจปัญหาที่แท้จริงของผู้ใช้
นักวิเคราะห์ระบบต้องค้นหาคำว่าปัญหาเดิมคืออะไร ผู้ใช้ทำงานอย่างไร และต้องการให้ระบบใหม่ช่วยแก้ปัญหาคือใด
2. ช่วยกำหนดขอบเขตของระบบ
ทำให้ทราบว่าสิ่งใดอยู่ในระบบ และสิ่งใดอยู่นอกขอบเขตของระบบ เช่น ระบบจองห้องพักอาจครอบคลุมการจอง การชำระเงิน และการยกเลิกการจอง แต่ไม่รวมระบบบัญชีเติมรูปแบบ
3. ช่วยลดความผิดพลาดในการพัฒนาระบบ
หากระบุความต้องการได้ชัดเจนตั้งแต่ต้น ทีมพัฒนาจะออกแบบและเขียนโปรแกรมได้ตรงตามเป้าหมาย
4. ช่วยให้สามารถตรวจสอบและทดสอบระบบได้
Requirement ที่ดีต้องสามารถนำไปใช้ตรวจสอบได้ว่าระบบที่พัฒนาขึ้นทำงานได้ตรงตามที่กำหนดหรือไม่
5. ช่วยลดต้นทุนและเวลาในการแก้ไขภายหลัง
ความผิดพลาดที่เกิดจาก Requirement หากพบช้าในช่วงพัฒนา ทดสอบ หรือติดตั้งจริง จะมีค่าใช้จ่ายในการแก้ไขสูงกว่าการแก้ไขตั้งแต่ช่วงวิเคราะห์

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.2 เหตุผลที่โครงการซอฟต์แวร์ล้มเหลวจาก Requirement ที่ไม่ชัดเจน

โครงการซอฟต์แวร์จำนวนมากล้มเหลวหรือไม่ประสบความสำเร็จ เพราะไม่ได้วิเคราะห์ความต้องการอย่างละเอียดตั้งแต่ต้น ปัญหาที่พบได้บ่อย เช่น

- ผู้ใช้ไม่สามารถอธิบายความต้องการของตนเองได้ชัดเจน
- นักวิเคราะห์ระบบเข้าใจความต้องการของผู้ใช้คลาดเคลื่อน
- ความต้องการเปลี่ยนแปลงบ่อยระหว่างการพัฒนา
- ขอบเขตของระบบไม่ชัดเจน ทำให้งานขยายเกินแผน
- ขาดการมีส่วนร่วมของผู้ใช้จริง
- ไม่มีการจัดลำดับความสำคัญของความต้องการ
- Requirement เขียนกำกวม ไม่สามารถทดสอบได้
- ทีมพัฒนาเริ่มออกแบบหรือเขียนโปรแกรมก่อนเข้าใจปัญหาอย่างแท้จริง

ตัวอย่างเช่น ถ้าผู้ใช้บอกเพียงว่า “ต้องการระบบขายสินค้าที่ใช้งานง่าย” ข้อความนี้ยังไม่เพียงพอ เพราะไม่สามารถบอกได้ว่าระบบต้องมีฟังก์ชันอะไรบ้าง ใช้งานง่ายในระดับใด ใครเป็นผู้ใช้งาน และจะทดสอบอย่างไรว่าใช้งานง่ายจริง ดังนั้นจึงต้องแปลงความต้องการดังกล่าวให้เป็นข้อกำหนดที่ชัดเจน เช่น “ระบบต้องสามารถบันทึกการขายสินค้าได้ภายในไม่เกิน 3 ขั้นตอน” หรือ “ระบบต้องแสดงรายการสินค้าและคำนวณราคารวมอัตโนมัติ”

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

ในกระบวนการ Unified Process หรือ UP การวิเคราะห์ความต้องการไม่ได้ทำเพียงครั้งเดียวแล้วจบ แต่จะเกิดขึ้นอย่างต่อเนื่องในหลายเฟส โดยเฉพาะในช่วง Inception และ Elaboration ในเฟส Inception จะเน้นการทำความเข้าใจภาพรวมของโครงการ เช่น ปัญหา วัตถุประสงค์ ขอบเขตของระบบ ผู้ใช้หลัก และ Use Case สำคัญเบื้องต้น เพื่อประเมินว่าโครงการควรดำเนินการต่อหรือไม่ ในเฟส Elaboration จะวิเคราะห์ความต้องการให้ละเอียดขึ้น โดยระบุ Use Case หลัก รายละเอียดการทำงาน ความเสี่ยง และสถาปัตยกรรมของระบบ เพื่อเตรียมเข้าสู่การออกแบบและพัฒนาจริง กล่าวโดยสรุป Requirement Analysis ใน UP มีบทบาทสำคัญในการเปลี่ยน “ความต้องการของผู้ใช้” ให้กลายเป็น “แบบจำลองและข้อกำหนดของระบบ” เช่น SRS, Use Case Diagram, Use Case Description และแบบจำลองอื่น ๆ ที่ใช้ในการออกแบบระบบเชิงวัตถุต่อไป

Requirement Analysis คือกระบวนการศึกษาปัญหา ความต้องการ และขอบเขตของระบบจากผู้ใช้และผู้เกี่ยวข้อง เพื่อนำมาจัดทำเป็นข้อกำหนดความต้องการของระบบอย่างชัดเจน หากขั้นตอนนี้ไม่สมบูรณ์ จะส่งผลให้ระบบที่พัฒนาขึ้นไม่ตรงกับความต้องการจริง เกิดการแก้ไขซ้ำ ใช้งบประมาณสูง และอาจทำให้โครงการล้มเหลวได้ ดังนั้น Requirement Analysis จึงเป็นพื้นฐานสำคัญของการวิเคราะห์และออกแบบระบบเชิงวัตถุ และเป็นจุดเริ่มต้นของการสร้าง Use Case Model ในขั้นนี้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

UP: Unified Process

Inception

Elaboration

Construction

Transition

time

Requirement Analysis

ออกเป็น 4 เฟส (phases)

- **Inception** :: กำหนดโครงการ การวางแผนโครงการ วิเคราะห์ความต้องการ ซึ่งเป็นระยะแรกของการดำเนินโครงการ การพัฒนาระบบสารสนเทศตามหลักการเชิงวัตถุ

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

โครงสร้างกรรมวิธี - Lifecycle Phases



Requirement Analysis

หรือออกเป็น 4 เฟส (phases)

ใช้ทรัพยากร วางแผนกำหนดหน้าที่ ใครทำอะไร

- **เตรียมงาน (Inception)** - นิยามขอบเขตของโครงการ , ขอบเขตของระบบที่จะพัฒนา
- **ทำรายละเอียด (Elaboration)** - วางแผนโครงการ จัดทำรายละเอียดความต้องการ จัดสร้างสถาปัตยกรรมระบบ
- **จัดสร้าง (Construction)** - สร้างและทดสอบโปรแกรม
- **ถ่ายโอน (Transition)** - ติดตั้งถ่ายโอนระบบให้กับผู้ใช้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

UP : Phase 1. Inception

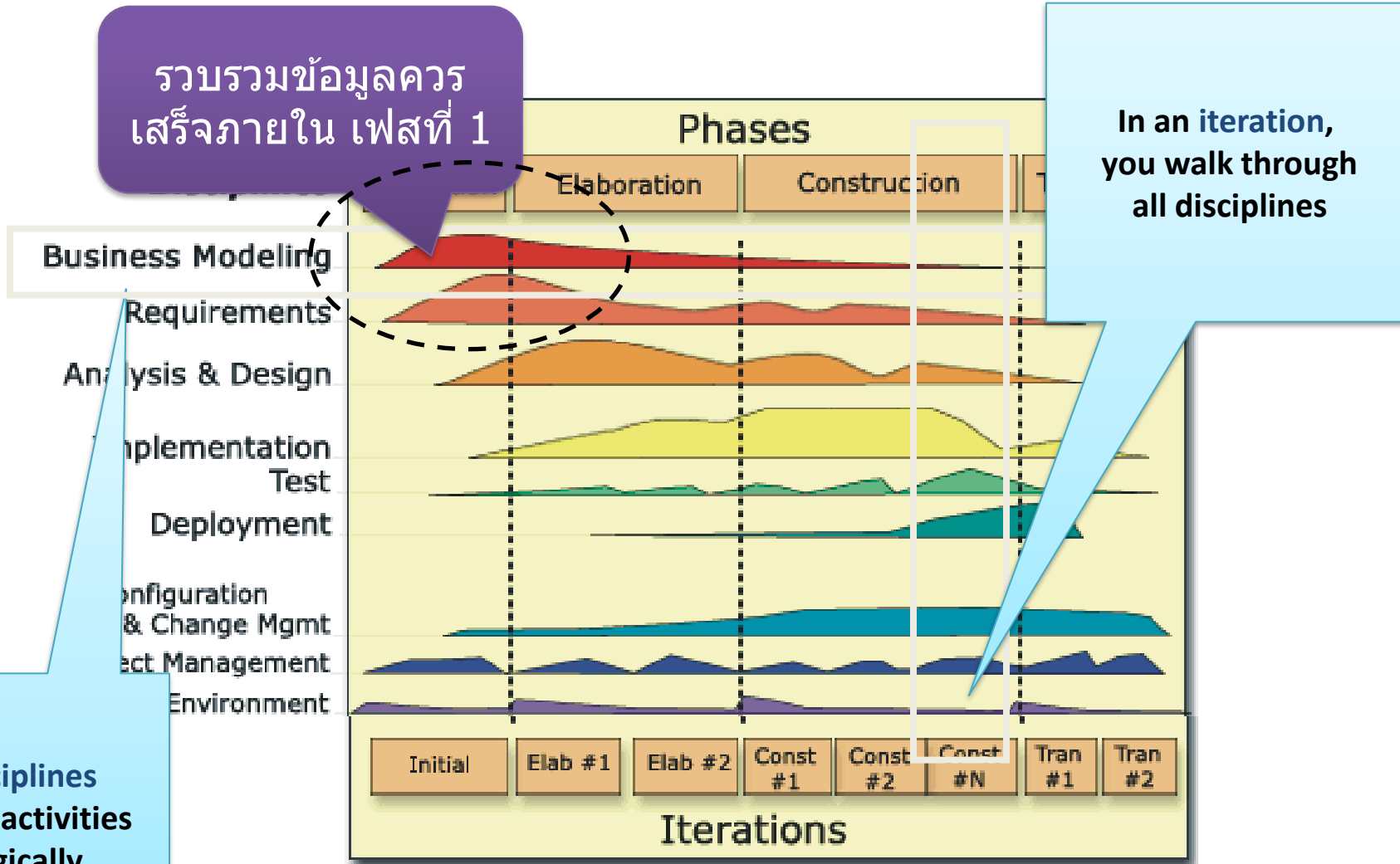
- ช่วงเริ่มต้นของโครงการ
 - ได้รับมอบหมายจาก เจ้าของกิจการ / ลูกค้า / หัวหน้า ให้รับผิดชอบโครงการเราจึงเรียก ว่า เป็น **PM : Project Manager**
 - ในขั้นต้นสิ่งที่คุณต้องทำ ก็คือ จัดหาทีม ร่วมพัฒนา และทำ Proposal >> เค้าโครงโครงการ/แบบเสนอโครงการ/ต่อผู้บริหาร
 - หัวข้อหลักๆ
 - ชื่อโครงการ เช่น การพัฒนาระบบจองห้องพักโรงแรม การพัฒนาระบบ ฯลฯ
 - ที่มา ความสำคัญของปัญหา 1-2 หน้ากระดาษ A4
 - วัตถุประสงค์
 - ขอบเขต
 - » ด้านเนื้อหา
 - » ด้านประชากรและกลุ่มตัวอย่าง
 - เทคโนโลยี / นวัตกรรม / กระบวนการที่ใช้ / วิธีการที่ใช้
 - แผนการดำเนินงาน ระยะเวลา / Gantt Chart
 - งบประมาณ PERT CHART/ อุปกรณ์ / เครื่องมือ
 - ผล/ประโยชน์ที่คาดว่าจะได้รับ

ต้องได้รับอนุมัติ
เมื่อเสนอแล้ว และได้รับอนุมัติจึง
สามารถทำเฟส 2 ต่อได้

สิ่งที่สำคัญ คือ ความน่าเชื่อถือ
- ดูจากอะไร ?

*** อยู่ในบทที่ 1 ของรูปเล่มรายงานของงานกลุ่ม

The Iterative Approach

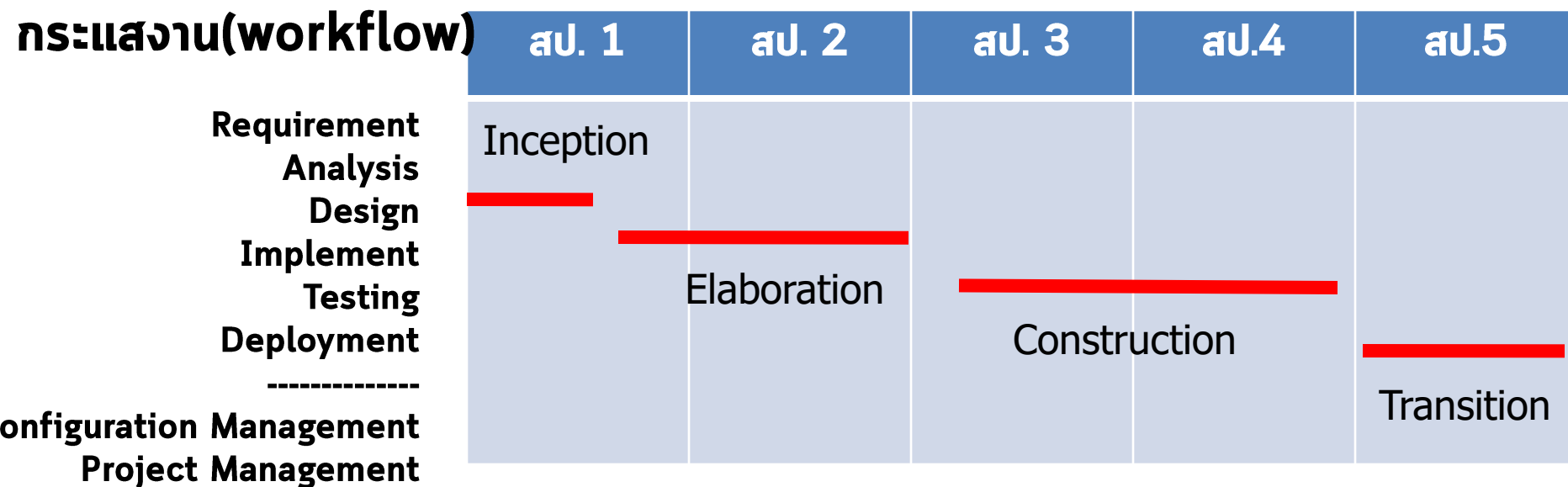


5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

รอบผลงาน

Gantt Chart

Project ใหญ่ หน่วยนับเวลา เป็นเดือน หรือ ปี
Project เล็ก ก็นับเป็น สป.



*****การเขียนแผนในโครงการกลุ่ม นั้นให้ยึดตามความจริง ไม่ใช่เวลาในการ**

ให้กำหนดหรือ สมมติ ว่างานนั้น ใช้เวลาทั้งระบบ ประมาณ ที่เดือน ที่ปี

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

1 ปี 12 เดือน

รายการ	ม.ย.				ก.ค.				ส.ค.				ก.ย.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
กระบวนการ(workflow)																
Requirement Inception																
Analysis																
Design																
Implement																
Testing																
Deployment																

Configuration Management																
Project Management																

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

เมื่อตกลงได้รับอนุมัติให้เริ่มจัดทำโครงการ

สิ่งที่จะทำลำดับต้น ๆ คือการรวบรวมข้อมูล จากการ
วิเคราะห์ความต้องการของผู้ใช้หรือลูกค้า นั้นเอง
Requirement Analysis

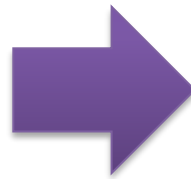
สิ่งที่ต้องรู้ คือ User มีใครบ้าง

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.1.3 ความสัมพันธ์ระหว่าง UP กับ Requirement Analysis

Software Requirement Specification
ข้อกำหนดความต้องการของซอฟต์แวร์ → เล่ม
รายงาน SRS นั้นเอง

Requirement Analysis



Requirement Specification

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2 ความต้องการ (Requirement)

5.2 ความต้องการ (Requirement)

Requirement หรือ ความต้องการของระบบ หมายถึง สิ่งที่ผู้ใช้ เจ้าของระบบ หรือผู้มีส่วนเกี่ยวข้องคาดหวังให้ระบบสารสนเทศสามารถทำได้ หรือเป็นเงื่อนไขที่ระบบต้องปฏิบัติตาม เพื่อให้ระบบที่พัฒนาขึ้นสามารถแก้ปัญหา สนับสนุนการทำงาน และตอบสนองเป้าหมายขององค์กรได้อย่างถูกต้อง กล่าวอย่างง่ายคือ Requirement เป็นคำตอบของคำถามว่า

“ระบบที่กำลังจะพัฒนาต้องทำอะไรได้บ้าง และต้องมีเงื่อนไขอะไรบ้างในการทำงาน”

ตัวอย่างเช่น หากต้องการพัฒนาระบบจองห้องพักโรงแรม ความต้องการของระบบอาจประกอบด้วย ระบบต้องสามารถค้นหาห้องว่าง จองห้องพัก บันทึกข้อมูลลูกค้า ดำเนินค่าห้อง รับชำระเงิน และออกใบยืนยันการจองได้

5.2.1 Requirement คืออะไร

Requirement คือ ข้อกำหนดหรือคำอธิบายเกี่ยวกับความสามารถ คุณสมบัติ เงื่อนไข หรือข้อจำกัดของระบบที่ต้องมี เพื่อให้ระบบทำงานได้ตรงตามความต้องการของผู้ใช้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2 ความต้องการ (Requirement)

- Requirement ที่ดีควรมีลักษณะดังนี้
 - ชัดเจน
 - อ่านแล้วเข้าใจตรงกัน ไม่กำกวม
 - ครบถ้วน
 - ระบุรายละเอียดสำคัญที่จำเป็นต่อการพัฒนาระบบ
 - ตรวจสอบได้
 - สามารถนำไปทดสอบได้ว่าระบบทำงานตรงตามที่กำหนดหรือไม่
 - สอดคล้องกัน
 - ไม่มีข้อกำหนดที่ขัดแย้งกันเอง
 - เป็นไปได้จริง
 - สามารถพัฒนาได้ภายในเวลา งบประมาณ และเทคโนโลยีที่มี

ตัวอย่าง Requirement ที่ยังไม่ดี:

ระบบต้องใช้งานง่าย

ข้อความนี้ยังไม่ชัดเจน เพราะไม่สามารถวัดหรือทดสอบได้ว่า “ง่าย” หมายถึงอะไร

ตัวอย่าง Requirement ที่ดีขึ้น:

ระบบต้องสามารถบันทึกข้อมูลลูกค้าได้ภายในไม่เกิน 3 ขั้นตอน และต้องแสดงข้อความแจ้งเตือนเมื่อกรอกข้อมูลไม่ครบ
ข้อความนี้ชัดเจนกว่า เพราะระบุสิ่งที่ระบบต้องทำและสามารถนำไปทดสอบได้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.2 User Requirement

User Requirement คือ ความต้องการของผู้ใช้ที่เขียนด้วยภาษาที่เข้าใจง่าย เป็นภาษาธรรมชาติ ไม่เน้นรายละเอียดทางเทคนิคมากเกินไป เพื่อให้ผู้ใช้ เจ้าของระบบ หรือผู้เกี่ยวข้องสามารถอ่านแล้วเข้าใจได้ User Requirement มักตอบคำถามว่า

“ผู้ใช้ต้องการให้ระบบช่วยทำอะไร”

ตัวอย่าง User Requirement ของระบบร้านค้าออนไลน์ เช่น

- ลูกค้าสามารถสมัครสมาชิกได้
- ลูกค้าสามารถดูรายการสินค้าได้
- ลูกค้าสามารถสั่งซื้อสินค้าได้
- พนักงานสามารถตรวจสอบรายการสั่งซื้อได้
- ผู้ดูแลระบบสามารถเพิ่ม แก้ไข และลบข้อมูลสินค้าได้
- เจ้าของร้านสามารถดูรายงานยอดขายประจำเดือนได้

User Requirement จึงเป็นความต้องการในมุมมองของผู้ใช้ เน้นอธิบายภาพรวมของสิ่งที่ระบบควรทำ โดยยังไม่ลงลึกถึงวิธีการออกแบบฐานข้อมูล โครงสร้างโปรแกรม หรือเทคโนโลยีที่ใช้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.3 System Requirement

System Requirement คือ ความต้องการของระบบที่แปลงมาจาก User Requirement ให้มีรายละเอียดมากขึ้น ชัดเจนขึ้น และสามารถนำไปใช้เป็นแนวทางในการออกแบบ พัฒนา และทดสอบระบบได้ System Requirement มักตอบคำถามว่า

“ระบบต้องทำงานอย่างไร เพื่อให้ตอบสนองความต้องการของผู้ใช้”

ตัวอย่างการแปลงจาก User Requirement เป็น System Requirement

User Requirement: **ลูกค้าสามารถสมัครสมาชิกได้**

System Requirement:

ระบบต้องมีหน้าจอสสมัครสมาชิก โดยให้ลูกค้ากรอกชื่อ นามสกุล อีเมล เบอร์โทรศัพท์ ชื่อผู้ใช้ และรหัสผ่าน ระบบต้องตรวจสอบว่าอีเมลไม่ซ้ำกับข้อมูลเดิม และรหัสผ่านต้องมีความยาวไม่น้อยกว่า 8 ตัวอักษร จะเห็นว่า System Requirement มีรายละเอียดมากกว่า User Requirement และสามารถนำไปใช้ในการออกแบบหน้าจอ ฐานข้อมูล และกรณีทดสอบระบบได้

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.4 Requirement Specification / SRS

Requirement Specification หรือ Software Requirement Specification: SRS คือ เอกสารข้อกำหนดความต้องการของซอฟต์แวร์ ที่รวบรวมรายละเอียดความต้องการของระบบไว้อย่างเป็นระบบ เพื่อใช้เป็นข้อตกลงร่วมกันระหว่างผู้ใช้ เจ้าของระบบ นักวิเคราะห์ระบบ และทีมพัฒนา

SRS ทำหน้าที่เป็นเอกสารอ้างอิงว่า

“ระบบที่ต้องการพัฒนาจะต้องมีความสามารถอะไรบ้าง มีขอบเขตอย่างไร และต้องทำงานภายใต้เงื่อนไขใด”

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.4 Requirement Specification / SRS

เนื้อหาใน SRS มักประกอบด้วย

- วัตถุประสงค์ของระบบ
- ขอบเขตของระบบ
- กลุ่มผู้ใช้ระบบ
- Functional Requirements
- Non-functional Requirements
- Business Rules
- ข้อจำกัดของระบบ
- Use Case Diagram
- Use Case Description
- Requirement Priority
- Requirement Dependency
- Acceptance Criteria หรือเงื่อนไขการยอมรับระบบ

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.4 Requirement Specification / SRS

ตัวอย่าง SRS แบบย่อ

ReqID: **REQ-001**

ชื่อ Requirement	: บันทึกข้อมูลลูกค้า
ผู้ใช้ที่เกี่ยวข้อง	: พนักงานขาย
รายละเอียดและที่อยู่	: ระบบต้องสามารถบันทึกข้อมูลลูกค้า ได้แก่ รหัสลูกค้า ชื่อ นามสกุล เบอร์โทรศัพท์ อีเมล
เงื่อนไข	: เบอร์โทรศัพท์ต้องเป็นตัวเลข 10 หลัก และอีเมลต้องมีรูปแบบถูกต้อง
ระดับความสำคัญ	: สูง
การทดสอบ	: หากกรอกข้อมูลไม่ครบ ระบบต้องแสดงข้อความแจ้งเตือนและไม่บันทึกข้อมูล

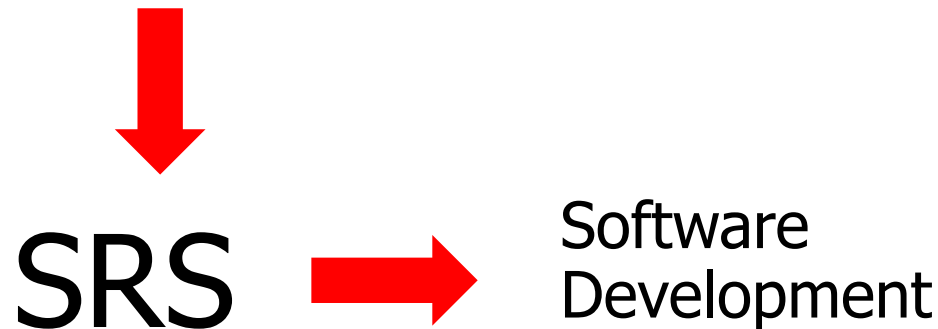
Requirement คือข้อกำหนดที่อธิบายว่าระบบต้องทำอะไรได้บ้าง และต้องมีเงื่อนไขใดในการทำงาน โดยแบ่งได้เป็น User Requirement ซึ่งเป็นความต้องการในมุมมองของผู้ใช้ และ System Requirement ซึ่งเป็นรายละเอียดเชิงระบบที่นำไปใช้ในการออกแบบและพัฒนา ส่วน SRS คือเอกสารที่รวบรวมข้อกำหนดทั้งหมดไว้อย่างเป็นระบบ เพื่อใช้เป็นแนวทางร่วมกันระหว่างผู้ใช้นักวิเคราะห์ระบบ และทีมพัฒนาในการสร้างระบบให้ตรงตามความต้องการจริงครับ.

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.1

ความหมายของความต้องการ (Requirements)

- ความต้องการ (Requirements) ในที่นี้หมายถึงคุณลักษณะในด้านต่าง ๆ ของระบบสารสนเทศที่กำลังจะทำการพัฒนาขึ้น เพื่อให้ระบบสามารถทำงานตอบสนองต่อผู้ใช้ได้อย่างแท้จริง
- แหล่งของความต้องการนั้นมาจาก**ผู้ใช้ (USER) ไม่ใช่มาจาก SA**
- นักวิเคราะห์ระบบจะต้องเป็นผู้สังเคราะห์ความต้องการนั้นจากข้อมูลต่างๆ ที่ได้รับมาจากผู้ใช้ โดยทำให้เป็นข้อกำหนดของความต้องการ (**Requirement specifications**) เพื่อใช้เป็นเป้าหมายและขอบเขตของการพัฒนาระบบต่อไป

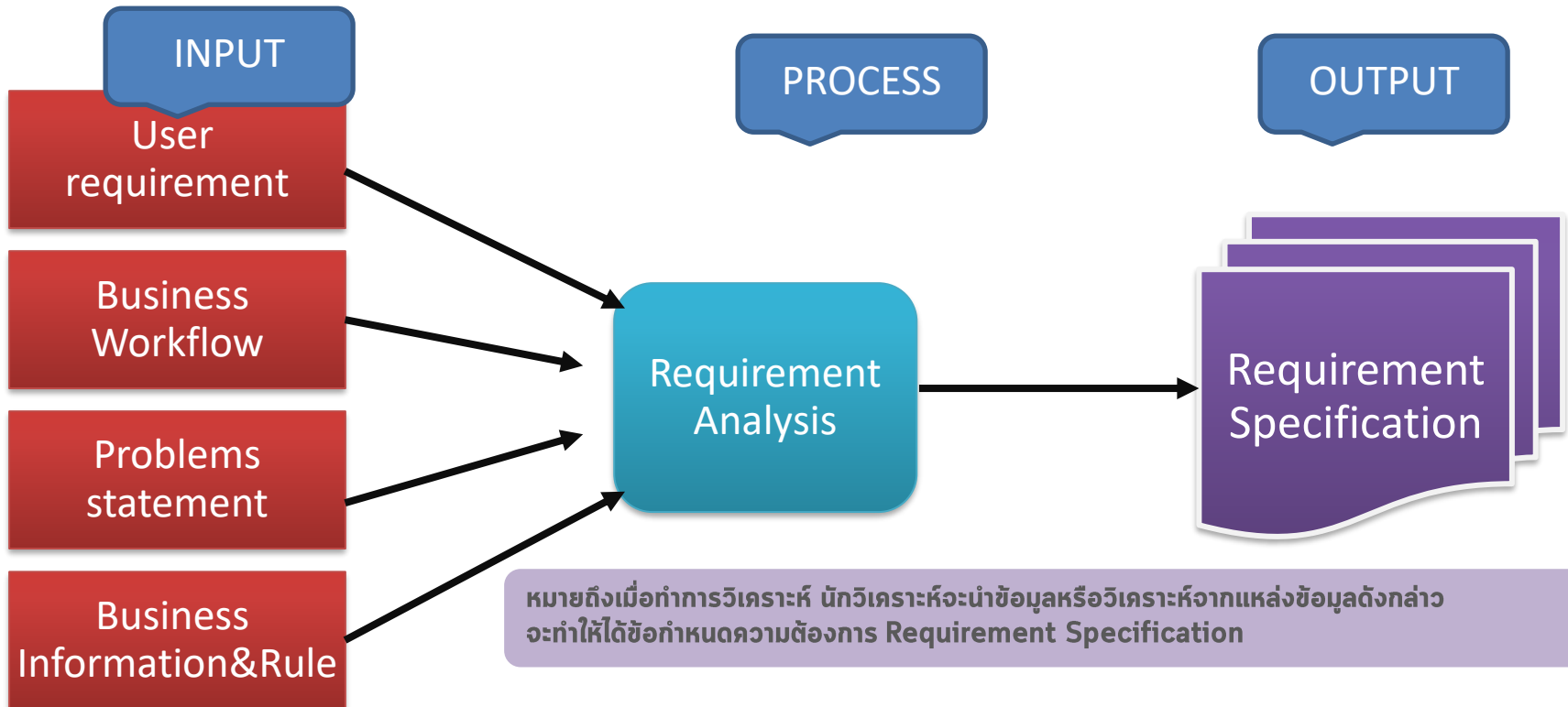


บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.2

ความหมายของการวิเคราะห์ความต้องการ (Requirement Analysis)

- การวิเคราะห์ความต้องการ คือกระบวนการวิเคราะห์เพื่อหาข้อกำหนดความต้องการของผู้ใช้ โดยจะต้องอาศัยข้อมูลในด้านต่าง ๆ ที่ได้รับมาจากผู้ใช้และองค์กรของผู้ใช้เพื่อทำการวิเคราะห์



บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.2.3

ตัวอย่าง การวิเคราะห์ความต้องการ

เพื่อให้ตอบคำถามสิ่งเหล่านี้ได้ ต้องทำอย่างไร นั่นก็คือต้องไปเก็บรวบรวมข้อมูลและวิเคราะห์ความต้องการจากผู้ใช้

- กระบวนการตั้งแต่ คนไข้ เข้า โรงพยาบาล จนกระทั่งรักษาเสร็จ/หายป่วย ทำอะไรบ้าง
- กระบวนการตั้งแต่ นักเรียนมาสมัครเป็น นศ. และเข้าเรียนได้ทำอะไรบ้าง

1.
2.
3.
4.
5.

1.
2.
3.
4.
5.

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.3.1 ผู้ใช้ระบบสารสนเทศ: แหล่งของความต้องการ

- **เจ้าของระบบ (System owners/Sponsors)** คุณจะได้ทำงานก็ต่อเมื่อนำเสนอ Proposal ผ่าน
 - มีส่วนได้ส่วนเสียจากการลงทุนสร้างระบบสารสนเทศ เช่น เจ้าของผู้บริหาร ผู้จัดการ ผุ่ตอ หุ่น ผู้ที่มีสิทธิ์การอนุมัติให้จัดทำโครงการได้
- **ผู้ใช้ภายใน (Internal users)**
 - End-users คือผู้ใช้ที่ป้อนข้อมูลเข้าสู่ระบบโดยตรง ไม่จำเป็นต้องมีทักษะหรือความรู้มาก เน้นความถูกต้องและรวดเร็วของการป้อนข้อมูลเข้าสู่ระบบ เช่น จนท. ฝ่ายทะเบียน จนท. ธนาคาร
 - Power-users คือผู้ใช้ที่มีความรู้ความชำนาญเฉพาะด้าน ที่สูงกว่าผู้ใช้ทั่วไป และสามารถใช้งานฟังก์ชันของระบบในส่วนที่มีความซับซ้อนได้ เช่น หัวหน้า พนง. ธนาคาร หัวหน้าฝ่ายบัญชี หัวหน้าฝ่ายบุคคล
 - Administrators คือผู้ที่ดูแลและควบคุมให้ระบบสามารถดำเนินการได้อย่างราบรื่นตามวัตถุประสงค์ที่ตั้งไว้ คอยความคุมดูแล ติดตาม และประเมินผล กำหนดสิทธิ์ในการเข้าถึงระบบของผู้ใช้ได้
 - Executive users คือผู้ใช้ที่ต้องการสารสนเทศมาเพื่อการตัดสินใจและบริหารองค์กร ให้มีประสิทธิภาพ CIO , CEO (EIS/MIS/DS) เป็นผู้ใช้งาน ระบบทางอ้อม
- **ผู้ใช้ภายนอก (External users)**
 - อาจจะเป็นคน หรือ กลุ่มบุคคล เป็นผู้ใช้ซึ่งเป็นบุคคลภายนอกองค์กร แต่สามารถเข้าถึงบริการของระบบในองค์กรได้ เช่น ลูกค้า , คนใช้ในรพ.

5.3.1 ผู้ใช้ระบบสารสนเทศ: แหล่งของความต้องการ

จงตอบคำถามเหล่านี้

- โรงพยาบาล ใครคือ System Owner / Sponsor
-
- โรงเรียนมัธยม ใครคือ System Owner / Sponsor
-
- โรงพยาบาล ใครคือ System Owner / Sponsor
-
- โรงแรมใครคือ System Owner / Sponsor
-
- มหาวิทยาลัย Power User
- ในมหาวิทยาลัย End User
- ในธนาคาร executive user

ทำไมต้องรู้จัก User หรือ ทำไมต้องแบ่ง User ออกเป็นกลุ่มๆ

เนื่องจากความต้องการใช้ระบบสารสนเทศ ไม่เหมือนกัน เพื่อไปเก็บข้อมูลตรงตามความต้องการของผู้ใช้

5.3.2 ทำไมต้องรู้จัก User หรือ ทำไมต้องแบ่ง User ออกเป็นกลุ่มๆ

ทำไมต้องรู้จัก User หรือ ทำไมต้องแบ่ง User ออกเป็นกลุ่มๆ

- **เหตุผล คือ แต่ละคนต้องการใช้งาน ระบบ ไม่เหมือนกัน ดังนั้น แต่ละคน ก็จะทำให้ข้อมูลไม่เหมือนกัน ดังนั้นเราต้องรู้ว่า จะเก็บข้อมูลนี้จากใคร ?**

ถ้าคุณ จะพัฒนาระบบ ของ รพ. และอยากรู้ว่า ย่าแบ่งออกเป็นกี่ประเภท ไปตาม ผอ. ได้ หรือไม่ ไปตาม แม่บ้านได้หรือไม่

แต่ละ User จะให้ข้อมูลเฉพาะ ฟังก์ชันงานตัวเอง

ผู้อำนวยการ รพ. จะให้ข้อมูลอะไรบ้างแก่ระบบ หรือ ต้องการข้อมูลอะไรบ้างจากระบบสารสนเทศ

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

ถ้าคุณ จะพัฒนาระบบเวชระเบียนของ รพ. แห่งหนึ่ง และอยากรู้ว่ายาแบ่งออกเป็นกี่ประเภท

- **จะต้องถาม user ใด**
 - ก) **System owner**
 - ข) **End user**
 - ค) **Power user**
 - ง) **Executive User**
 - จ) **ไม่มีข้อใดถูกต้อง**

5.1 การวิเคราะห์ความต้องการ

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

ถ้าคุณ จะพัฒนาระบบร้านค้าเร่ และอยากรู้ว่าบริการของร้านมีกี่แบบ แต่ละแบบราคาเท่าไร

- **ควรจะต้องถาม user ใด**
 - ก) **System owner**
 - ข) **End userพนักงานทั่วไป , ช่าง , พนักงานขาย, พนักงานบัญชี**
 - ค) **Power user หัวหน้าแผนก หัวหน้าช่าง ..หัวหน้าพนักงานขาย**
 - ง) **Executive User**
 - จ) **ไม่มีข้อใดถูกต้อง**

5.1 การวิเคราะห์ความต้องการ

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

ผู้อำนวยการ รพ. จะให้ข้อมูลอะไรบ้างแก่ระบบ หรือ ต้องการข้อมูลอะไรบ้างจากระบบสารสนเทศ

1. การวางแผนขยายธุรกิจ (เปิดสาขา)
2. การปรับกลยุทธ์ สถานการณ์โควิด (การร่วมประกันโควิด)
3. .เชิงบริหารจัดการ เชิงนโยบาย เชิงกลยุทธ์
4.
5.

5.1 การวิเคราะห์ความต้องการ

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

ผู้บริหารโรงแรม(CEO) จะให้ข้อมูลอะไรบ้าง หรือต้องการข้อมูลอะไรบ้างจากระบบบ้าง

1. ตัวอย่าง การจะปิด หรือเปิดขยายสาขา

2.

3.

4.

5.

*** สิ่งเหล่านี้จะถูกนำไปเขียนเป็น DFD สำหรับระบบแบบดั้งเดิม สำหรับเชิงวัตถุ จะถูกนำไปวิเคราะห์ด้วยแบบจำลองยูสเคส

5.1 การวิเคราะห์ความต้องการ

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

ในขั้นตอนการเก็บรวบรวมความต้องการระบบ แบบเชิงวัตถุ ผู้บริหารโรงแรม(CEO) จะให้ข้อมูลอะไรบ้าง หรือต้องการข้อมูลอะไรบ้างจากระบบบ้าง

- ข้อมูลเกี่ยวกับวัตถุ
 - ข้อมูลเกี่ยวกับแนวโน้มตลาด เช่น จำนวนนักท่องเที่ยว ประเภทห้องพักที่ได้รับความนิยม
 - ข้อมูลเกี่ยวกับคู่แข่ง เช่น กลยุทธ์ทางการตลาดของคู่แข่ง บริการที่คู่แข่งนำเสนอ
 - ข้อมูลเกี่ยวกับความต้องการของลูกค้า เช่น ความคิดเห็นของลูกค้าเกี่ยวกับโรงแรม ความต้องการของลูกค้าในอนาคต
- ข้อมูลเกี่ยวกับความสัมพันธ์ระหว่างวัตถุ
 - ความสัมพันธ์ระหว่างแนวโน้มตลาดกับโรงแรม เช่น แนวโน้มตลาดใดบ้างที่ส่งผลกระทบต่อโรงแรม
 - ความสัมพันธ์ระหว่างคู่แข่งกับโรงแรม เช่น คู่แข่งใดบ้างที่มีกลยุทธ์ทางการตลาดที่คล้ายคลึงกับโรงแรม
 - ความสัมพันธ์ระหว่างความต้องการของลูกค้ากับโรงแรม เช่น ความต้องการลูกค้าใดบ้างที่โรงแรมสามารถตอบสนองได้

5.1 การวิเคราะห์ความต้องการ

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

โรงพยาบาล

- System Owner ::ผู้อำนวยการโรงพยาบาล..
- End User :พยาบาล. เจ้าหน้าที่
- Power User :**IT Support** หมอ
- Administrator :**ผู้ดูแลระบบ**
- Executive User :ผู้บริหาร / ผอ. /กรรมการ
- External User :ผู้ป่วย / **บุคคลภายนอก ? / ญาติผู้ป่วย /**
แขกผู้ป่วย / การไฟฟ้า / เจ้าหน้าที่ส่งของ

5.1 การวิเคราะห์ความต้องการ

5.3.3 ตัวอย่าง การวิเคราะห์ความต้องการ

โรงพยาบาลนคร

- System Owner ::เจ้าของโรงพยาบาล ?
- End User :พนักงานโรงพยาบาล ? / พนง. ชายตัว / พนง. ชายป๊อบคอน
- Power User :
- Administrator :
- Executive User :
- External User :ลูกค้า หรือผู้ชมภาพยนต์

Agenda(Cont.)

5.4 กระบวนการวิเคราะห์ความต้องการ

- **Information Gathering**
- **Requirement Identification**
- **Requirement Selection / Scope / Problem Domain**
- **Requirement Classification and Structuring**
- **Prioritization and Negotiation**
- **Requirement Validation**
- **Requirement Specification**

5.4 กระบวนการวิเคราะห์ความต้องการ

5.4.1 ขั้นตอนกระบวนการวิเคราะห์ความต้องการ

- การบวนการวิเคราะห์ความต้องการมีขั้นตอนคร่าว ๆ ดังต่อไปนี้
 1. เก็บรวบรวมข้อมูลที่เป็นข้อเท็จจริงต่าง ๆ (Data/Information gathering)
 1. เก็บข้อมูลลูกค้า เก็บข้อมูลสินค้า เก็บข้อมูลพนักงานบริการ ฯลฯ
 2. วิเคราะห์เพื่อระบุถึงความต้องการต่าง ๆ (Requirement Identification)
 3. ตัดเลือกส่วนที่เป็นสาระสำคัญและอยู่ในขอบเขตการพัฒนา (Requirement selection/ Problem Domain)
 4. จัดจำแนกและจัดโครงสร้างของความต้องการ (Requirement classification and structuring) Prioritize Use case architecture
 5. จัดลำดับความสำคัญและตกลงเจรจา (Prioritization and negotiation)
 6. ตรวจสอบความถูกต้อง (Requirement validation)
 7. จัดทำรายงานข้อกำหนดความต้องการ Requirement Specification

5.2 กระบวนการวิเคราะห์ความต้องการ

5.4.2

วิธีการเก็บรวบรวมข้อมูล (Data/Information gathering)

- หมายถึง จะทำอะไร กับข้อมูลนี้ มีความต้องการอะไรบ้าง
 - เช่น ในการจัดเก็บหรือบันทึกข้อมูลลูกค้า มีเงื่อนไข หรือข้อกำหนดต่าง ๆ ดังนี้
 - การเก็บข้อมูล จะต้องเก็บให้ครบได้ รหัส เลขที่บัตรประชาชน ชื่อ ที่อยู่ เบอร์โทร เพศ สถานะ อีเมล
-
1. จะขาดสิ่งใดสิ่งหนึ่งไม่ได้ ระบบจะต้องไม่ยอมให้บันทึก เลขที่บัตรประชาชน
 2. เบอร์โทร จะเก็บ 10 หลัก ห้ามเกิน ห้ามขาด
 3. เบอร์โทรจะต้องไม่เป็นตัวอักษร 1222333333333333
 4. เลขที่บัตรประชาชน จะต้องเป็น 13 หลัก และเป็นตัวเลขเท่านั้น
 5. สถานะ จะต้องประกอบไปด้วย โสด สมรส หย่าร้าง
 6. อีเมล จะต้องถูกต้องตามหลัก โดยมี @

5.2 กระบวนการวิเคราะห์ความต้องการ

5.4.2

1. เก็บรวบรวมข้อมูลที่เป็นข้อเท็จจริงต่าง ๆ (Information Gathering)

- สังเกต (Observation) ไปดู เช่น สังเกตการทำงานของพนักงานในโรงพยาบาล
- สัมภาษณ์ (Interview) จัดทำแบบสัมภาษณ์ตามรายบุคคล หรือกลุ่มบุคคล
- แบบสอบถาม(Questionnaire)
- ทบทวนเอกสาร (Document reviews)
- ลงมือทำ (Workshop)

5.2 กระบวนการวิเคราะห์ความต้องการ

5.4.3

2. การระบุความต้องการ (Requirement Identification)

จากข้อมูลที่ได้มา นำมาสู่การระบุความต้องการโดย

- แต่ละความต้องการควรเป็นอิสระในตัวเอง ไม่ผสมปะปนกับความต้องการอื่นๆ
- ความต้องการจะต้องสามารถทำการทดสอบได้ (Testable) ในภายหลัง
- ความต้องการจะต้องสามารถจัดการดังต่อไปนี้ได้
 - จัดกลุ่มของความต้องการ (Grouped) เช่น จัดตาม Viewpoint ของผู้ใช้ เป็นต้น
 - จัดลำดับความสำคัญ (Prioritized)
 - กำหนดในระดับของนามธรรมและรายละเอียด (Level of Abstraction)
 - กำหนดความสัมพันธ์ในเชิงการขึ้นอยู่กับกัน (Dependency)
- ความต้องการแต่ละความต้องการควรมีรหัสที่เป็นเอกเทศใช้ในการอ้างอิงได้ (ID)
- ควรแสดงด้วยประโยคพรรณนา (Descriptive) ที่เรียบง่ายตรงไปตรงมา ไม่ควรอธิบายด้วยประโยคที่ซับซ้อนจนไม่สามารถถอดใจความได้

5.2 กระบวนการวิเคราะห์ความต้องการ

5.4.3

2. การระบุความต้องการ (Requirement Identification)

กำหนดความสัมพันธ์ในเชิงการขึ้นอยู่กับกัน (Dependency)

การขึ้นอยู่กับกัน

- ถ้าไม่มีการเก็บข้อมูลค่า
 - จะไม่สามารถค้นหา ข้อมูลผู้เข้าพัก
- ถ้าไม่มีการเก็บข้อมูลห้องพัก
 - ก็จะไม่สามารถจองห้องพักได้

5.2 กระบวนการวิเคราะห์ความต้องการ

5.2.3

2. การระบุความต้องการ (Requirement Identification)

ความต้องการนั้นจะต้อง Testable ได้

- เช่น เลขบัตรประจำตัวประชาชน ต้องกำหนดเป็นตัวเลข 13 หลัก test ได้
- *** ระบบจะต้องมีประสิทธิภาพ Test อย่างไร (“ระบบจะต้องมีประสิทธิภาพ” เป็น NFR ที่ยังไม่ทดสอบได้ ควรปรับเป็น “ระบบต้องตอบสนองภายใน 3 วินาที เมื่อมีผู้ใช้พร้อมกันไม่เกิน 100 คน”)
- จะต้องป้อนข้อมูลให้ครบ ถ้าไม่ครบ ระบบจะไม่ยอมบันทึกให้

Form Register:

Username: *

Password: *

Password confirm: *

Emailaddress: *

Submit

5.2 กระบวนการวิเคราะห์ความต้องการ

5.4.3

2. การระบุความต้องการ (Requirement Identification)

Req. 01 ระบบจะต้องบันทึกข้อมูลสินค้า

- ข้อมูลสินค้า จะต้องจัดเก็บให้ครบถ้วนสมบูรณ์ เช่น
 - การเก็บรหัสสินค้า
 - รหัส สินค้า ต้องเก็บ 6 หลัก โดย 3 ตัวแรก เป็น proxxx
 - Pro001 - Pro999
 - จำนวนสินค้า จะเก็บได้ไม่เกิน 1000 รายการ
 - จะต้องเก็บราคาเป็นทศนิยม
 - ราคา 1987.45
 - สกุลเงิน เป็น บาท
 - จำนวนสินค้าต้องเป็น จำนวนเต็มเท่านั้น ใส่เป็นจุดทศนิยมไม่ได้
 - จำนวนสินค้าแต่ละครั้งที่สั่งซื้อ ไม่เกิน 1000 ชิ้น
 - ราคาสินค้าจะต้องไม่ติดลบ -5.00 บาท ?
 - ราคา เป็นตัวอักษรไม่ได้ ราคา-ก.-บาท
- *** requirement เหล่านี้จะต้อง ทดสอบได้ในภายหลัง เมื่อถึงขั้นตอนทดสอบระบบ**

บทที่ 5 การวิเคราะห์ความต้องการ และแบบจำลองยูสเคส

ตัวอย่าง SRS แบบย่อ

ReqID: **REQ-001**

ชื่อ Requirement	: บันทึกข้อมูลลูกค้า
ผู้ใช้ที่เกี่ยวข้อง	: พนักงานขาย
รายละเอียดและที่อยู่	: ระบบต้องสามารถบันทึกข้อมูลลูกค้า ได้แก่ รหัสลูกค้า ชื่อ นามสกุล เบอร์โทรศัพท์ อีเมล
เงื่อนไข	: เบอร์โทรศัพท์ต้องเป็นตัวเลข 10 หลัก และอีเมลต้องมีรูปแบบถูกต้อง
ระดับความสำคัญ	: สูง
การทดสอบ	: หากกรอกข้อมูลไม่ครบ ระบบต้องแสดงข้อความแจ้งเตือนและไม่บันทึกข้อมูล

Requirement คือข้อกำหนดที่อธิบายว่าระบบต้องทำอะไรได้บ้าง และต้องมีเงื่อนไขใดในการทำงาน โดยแบ่งได้เป็น User Requirement ซึ่งเป็นความต้องการในมุมมองของผู้ใช้ และ System Requirement ซึ่งเป็นรายละเอียดเชิงระบบที่นำไปใช้ในการออกแบบและพัฒนา ส่วน SRS คือเอกสารที่รวบรวมข้อกำหนดทั้งหมดไว้อย่างเป็นระบบ เพื่อใช้เป็นแนวทางร่วมกันระหว่างผู้ใช้นักวิเคราะห์ระบบ และทีมพัฒนาในการสร้างระบบให้ตรงตามความต้องการจริงครับ.

5.2 กระบวนการวิเคราะห์ความต้องการ

5.4.3

2. การระบุความต้องการ (Requirement Identification)

Req. 01 ระบบจะต้องบันทึกข้อมูลสินค้า

ข้อกำหนดความต้องการ "Req. 01: ระบบจะต้องบันทึกข้อมูลสินค้า" เป็นความต้องการที่ระบบต้องปฏิบัติตามเพื่อให้ระบบทำงานได้ตามวัตถุประสงค์ของโครงการหรือระบบที่กำลังพัฒนา ดังนั้นเราสามารถแยกข้อกำหนดความต้องการนี้เพื่อเข้าใจและทำความเข้าใจเพิ่มเติมได้ดังนี้:

1. บันทึกข้อมูลสินค้า: ระบบจะต้องสามารถรับข้อมูลที่เกี่ยวข้องกับสินค้า และบันทึกข้อมูลนั้นลงในฐานข้อมูลหรือระบบจัดเก็บข้อมูลอย่างเหมาะสม โดยข้อมูลสินค้าอาจรวมถึงรายละเอียดเกี่ยวกับชื่อสินค้า ราคา จำนวนคงเหลือ รหัสสินค้า เป็นต้น
2. การเพิ่มข้อมูลสินค้า ระบบจะต้องมีการรองรับการเพิ่มข้อมูลสินค้าใหม่ลงในระบบ ซึ่งอาจเป็นการป้อนข้อมูลโดยผู้ใช้หรือการนำเข้าข้อมูลจากแหล่งอื่น เช่น ไฟล์ CSV หรือระบบอื่น ๆ
3. การอัปเดตข้อมูลสินค้า ระบบจะต้องสามารถอัปเดตข้อมูลสินค้าที่มีอยู่ในระบบ อาจเป็นการแก้ไขราคา จำนวนคงเหลือ หรือข้อมูลอื่น ๆ ที่เกี่ยวข้อง
4. การลบข้อมูลสินค้า ระบบจะต้องสามารถลบข้อมูลสินค้าที่ไม่ได้ใช้งานออกจากระบบ เช่น สินค้าที่ถูกเพิกถอนหรือไม่มีการจำหน่ายแล้ว
5. การค้นหาและแสดงข้อมูล ระบบจะต้องมีการค้นหาและแสดงข้อมูลสินค้าให้ผู้ใช้งาน ซึ่งสามารถค้นหาด้วยเงื่อนไขต่าง ๆ เช่น ชื่อสินค้า ราคา หรือรหัสสินค้า
6. การรายงาน ระบบควรสามารถสร้างรายงานเกี่ยวกับข้อมูลสินค้า เช่น รายงานการขายสินค้า รายงานสินค้าที่หมดคลัง หรือรายงานเกี่ยวกับสถิติต่าง ๆ
7. ความปลอดภัยของข้อมูล ระบบจะต้องรักษาความปลอดภัยของข้อมูลสินค้าในระบบ และมีการควบคุมการเข้าถึงข้อมูลเพื่อป้องกันการแอบแฝงข้อมูลหรือการเข้าถึงข้อมูลที่ไม่เป็นอำนาจ
8. การบันทึกประวัติ ระบบควรบันทึกประวัติการเปลี่ยนแปลงข้อมูลสินค้า เพื่อทราบถึงการแก้ไขหรือการเปลี่ยนแปลงข้อมูลในอดีต
9. การสนับสนุน ระบบควรมีการสนับสนุนและการบริการหลังการขายสำหรับผู้ใช้งาน เช่น การตอบคำถาม แก้ไขปัญหา หรือการฝึกอบรมในการใช้ระบบ

ข้อกำหนดความต้องการนี้จะช่วยในการกำหนดขอบเขตของระบบและคุณสมบัติที่จำเป็น โดยชัดเจนและทำให้ทุกคนในทีมพัฒนาและผู้ใช้งานเข้าใจถึงความต้องการของระบบที่กำลังพัฒนา ทำให้กระบวนการพัฒนาระบบเป็นไปอย่างราบรื่น

Req. 01 ระบบจะต้องบันทึกข้อมูลสินค้า

รับสินค้าเข้า

รหัสผู้ขาย: ก-0001 บริษัท แกรนด์ อินเทอร์เน็ต จำกัด

รายการเอกสาร: R01 ชื่อสินค้า: **Goods Receive (WH)**

เลขที่เอกสาร: WHD05311-00001 วันที่เอกสาร: 10/02/2553 เลขที่ใบ DO: WHD05311-00001

เลขที่ Approve PO: / / แผนก: 004 แผนกการขาย

ผู้บันทึก: EMP-00009 นางสาว อิมพวรรณ ปอแก้ว ผู้รับสินค้า: EMP-00007 นาย สมศักดิ์ สุระชัย

หมายเหตุ: บันทึกการซื้อสินค้าวันที่ 05/02/2253


No.	รหัสสินค้า	ชื่อสินค้า	คลัง	ที่เก็บ	หน่วยนับ	จำนวน
1	IC-0001	คอมพิวเตอร์ รุ่น SAMSUNG	01	01	เครื่อง	250.00
2	IC-0003	คีย์บอร์ด	01	01	อัน	400.00
3	IC-0004	เครื่อง SCAN ลายนิ้วมือ	01	01	ตัว	150.00
4	IC-0005	เครื่อง BIO SCAN	01	01	ตัว	150.00
5						.00
6						.00
7						.00
8						.00
9						.00
10						.00
11						.00
รวม						950.00

Detail Description History

New Save Delete Find Copy Ref.PO Print Close

อ้างอิง <https://www.prosoftwinspeed.com/Article/Detail/99824>

หน้า 1 / 1



บริษัท ตัวอย่าง จำกัด



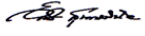
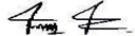
2571/1 ซ.รามคำแหง 75 ถ.รามคำแหง แขวงหัวหมาก เขตบางกะปิ กรุงเทพฯ 10240
 โทร. 0-2739-5900 โทรสาร 0-2739-5910 เลขประจำตัวผู้เสียภาษี 31523611000

ใบรับสินค้าเข้า

รหัสเข้าหนี้ ก-0001 ชื่อเจ้าหนี้ บริษัท แกรนด์ อินเทอร์เน็ต จำกัด ที่อยู่ 259/83 ซ.พิบูลสงคราม ถ.สุขุมวิท71 แขวงพระโขนงเหนือ เขตวัฒนา กทม. 10110 โทร. 0-2391-3030 โทรสาร. 0-2382-1687	เลขที่เอกสาร WHDO5311-00001 วันที่เอกสาร 10/02/2553 เลขที่ใบ PO 5302-0001 วันที่ใบ PO 10/02/2553 เลขที่ Approve PO วันที่ Approve PO เลขที่ใบ DO WHDO5311-00001
---	--

ลำดับ	รหัสสินค้า	รายการ	หน่วย	ที่เก็บ	คลัง	จำนวน
1	IC-0001	คอมพิวเตอร์ รุ่น SAMSUNG	เครื่อง	01	01	250.00
2	IC-0003	ที่ยับอร์ด	อัน	01	01	400.00
3	IC-0004	เครื่อง SCAN สายนามือ	ตัว	01	01	150.00
4	IC-0005	เครื่อง BIO SCAN	ตัว	01	01	150.00
รวม						950.00

หมายเหตุ

 พนักงานคลังสินค้า วันที่ 10/02/2553	 ผู้ตรวจสอบ วันที่ 10/02/2553	 ผู้อนุมัติ วันที่ 10/02/2553	 ผู้ส่งของ วันที่ 10/02/2553
---	--	---	---

จัดกลุ่มความต้องการ ตามผู้ใช้

- ดูรายการสินค้า
- เปลี่ยนชื่อลูกค้า
- เปลี่ยนที่อยู่
- เปลี่ยนที่อยู่ในการจัดส่ง
- แก้ไขผู้บันทึกข้อมูลสินค้า
- แก้ไขราคาสินค้า
- แก้วันที่จัดส่งสินค้า
- แก้ไขจำนวนสินค้า
- แก้ไขรายการสินค้า
- แก้ไขข้อมูลลูกค้า
- แก้ไขข้อมูลพนักงาน
- แก้ไขข้อมูลสินค้า
- ยกเลิกการสั่งซื้อ
- ยกเลิกการจองห้องพัก
- ยกเลิกการขายสินค้า

User ...ลูกค้า
ความต้องการ

User ...พนักงาน
ความต้องการ

User ...ผู้ดูแลระบบ Data/Information
gathering
ความต้องการ

แยกความต้องการตาม user

ลูกค้า	พนักงาน	ผู้ดูแลระบบ	เจ้าของ (System Owner)
ดูรายการสินค้า	ดูรายการสินค้า	ดูรายการสินค้า	
เปลี่ยนชื่อลูกค้า		เปลี่ยนชื่อลูกค้า	

จัดกลุ่มความต้องการ ตามผู้ใช้

- ดูรายการสินค้า
- เปลี่ยนชื่อ
- เปลี่ยนที่อยู่
- เปลี่ยนที่อยู่ในการจัดส่ง
- แก้ไขผู้บันทึกข้อมูลสินค้า
- แก้ไขราคาสินค้า
- แก้วันที่จัดส่งสินค้า
- แก้ไขจำนวนสินค้า
- แก้ไขรายการสินค้า
- แก้ไขข้อมูลลูกค้า
- แก้ไขข้อมูลพนักงาน
- แก้ไขข้อมูลสินค้า
- ยกเลิกการสั่งซื้อ
- ยกเลิกการจองห้องพัก
- ยกเลิกการขายสินค้า

User ...ลูกค้า
ความต้องการ

User ...พนักงาน
ความต้องการ

Userผู้ดูแลระบบ Data/Information
gathering
ความต้องการ

จัดกลุ่มของความต้องการ (Grouped) เช่น จัดตาม Viewpoint ของผู้ใช้ เป็นต้น

ผู้ใช้ end user สามารถ

1. ป้อนข้อมูลพนักงานได้
2.บันทึกได้
3. บันทึกข้อมูลสินค้าได้
4. ตรวจสอบรายการสินค้า
5. ขายสินค้าได้
6. แก้ไขข้อมูลรายสั่งซื้อได้

ผู้ใช้ Admin สามารถ

1. ค้นหาข้อมูลสมาชิกได้
2. แก้ไขข้อมูลลูกค้า
3. ลบข้อมูลลูกค้า
4. บันทึกได้
5. แก้ไขสิทธิ์ ของผู้ใช้ได้
6. ลบผู้ใช้ระบบได้
- 7....
- 8.....

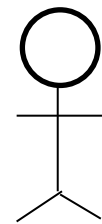
ความต้องการจะต้องสามารถจัดการดังต่อไปนี้ได้
จัดกลุ่มของความต้องการ (Grouped) เช่น จัดตาม Viewpoint ของผู้ใช้ เป็นต้น

- **ลูกค้า**

- ดูรายสินค้าได้
- สั่งซื้อสินค้าได้
- ยกเลิกการสั่งซื้อได้
- แก้ไขข้อมูลการสั่งซื้อได้
- แก้ไขข้อมูลลูกค้า

- **เจ้าหน้าที่**

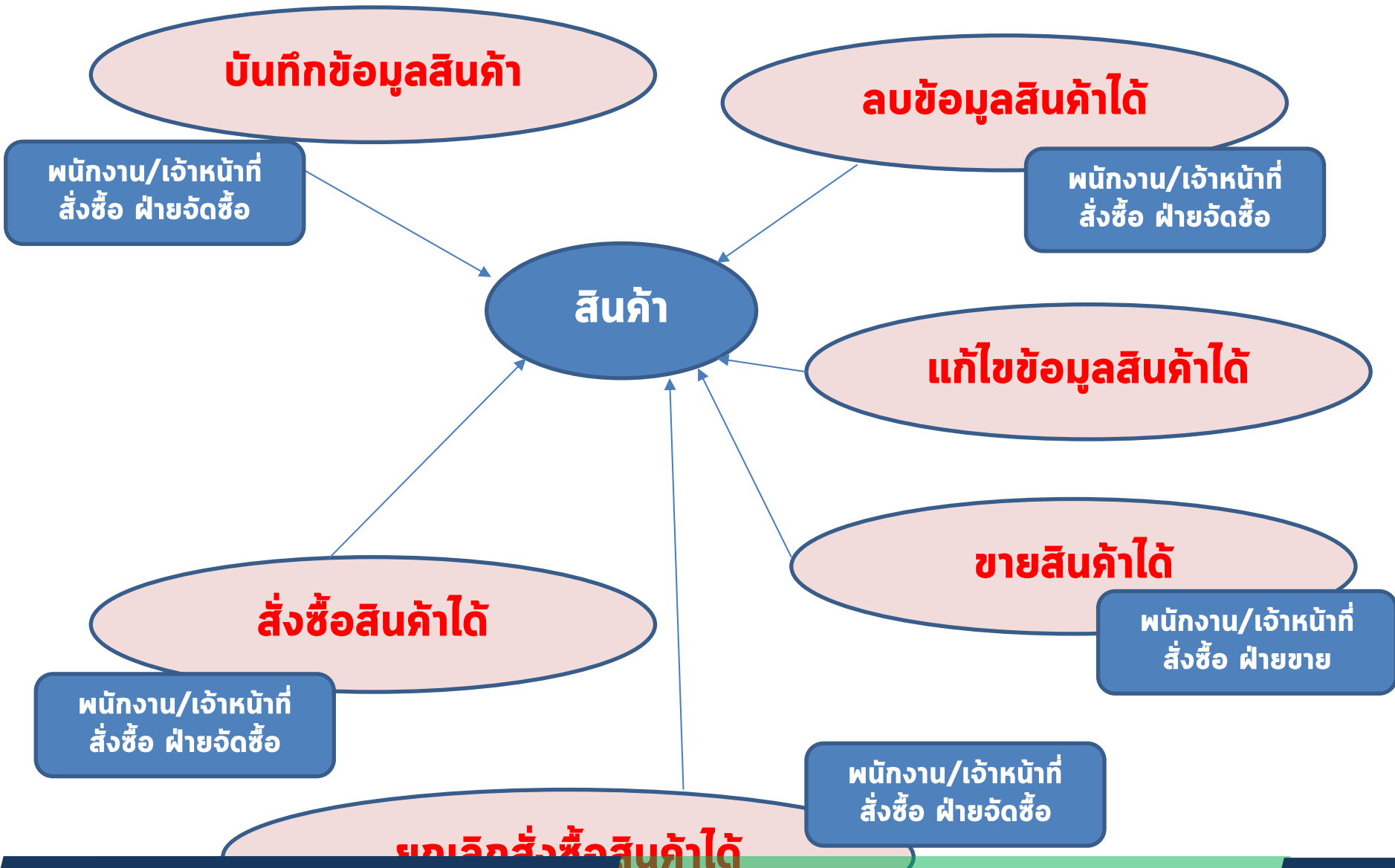
- **เจ้าของระบบ**



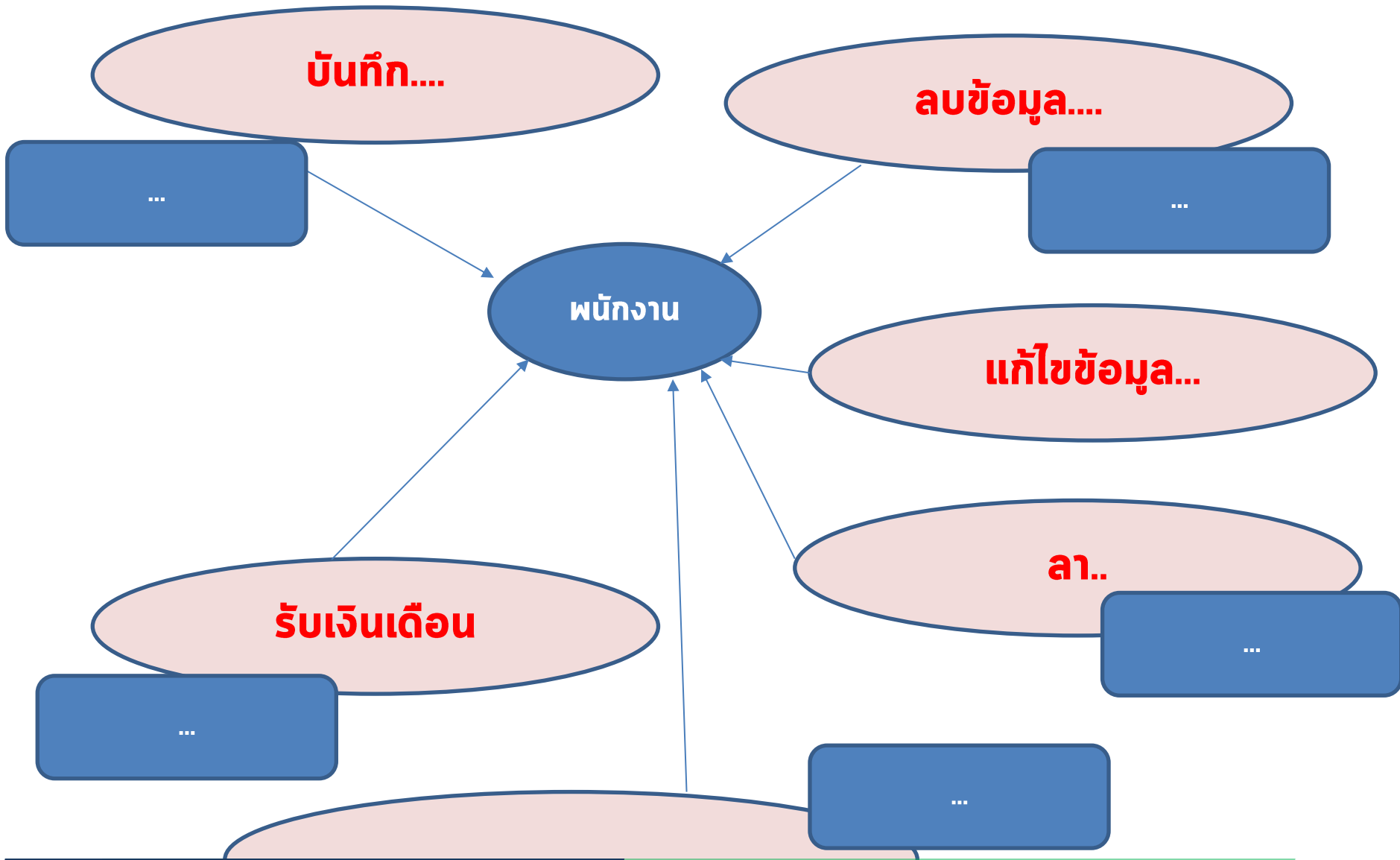
ลูกค้า



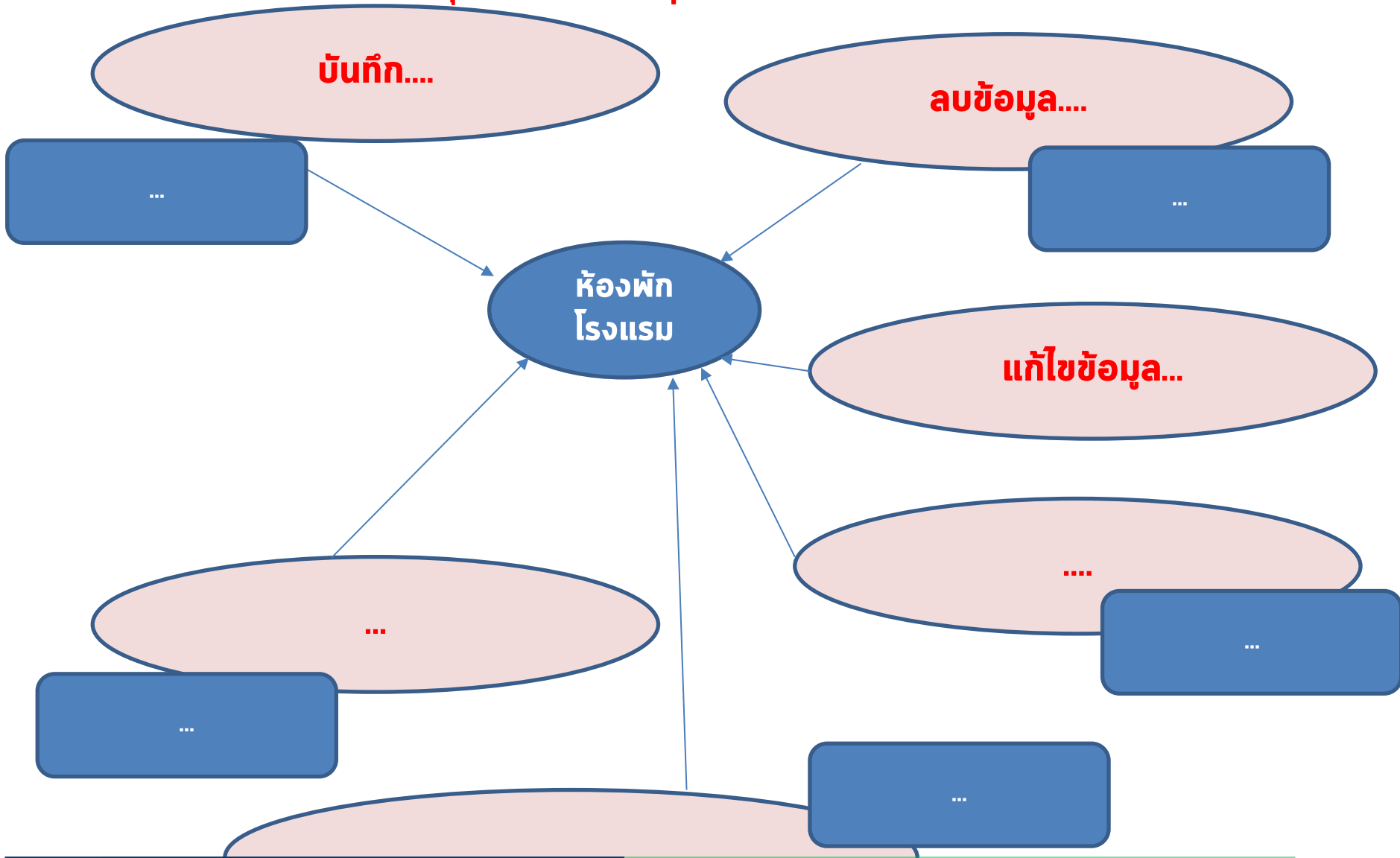
การระบุความต้องการ Requirement Identification



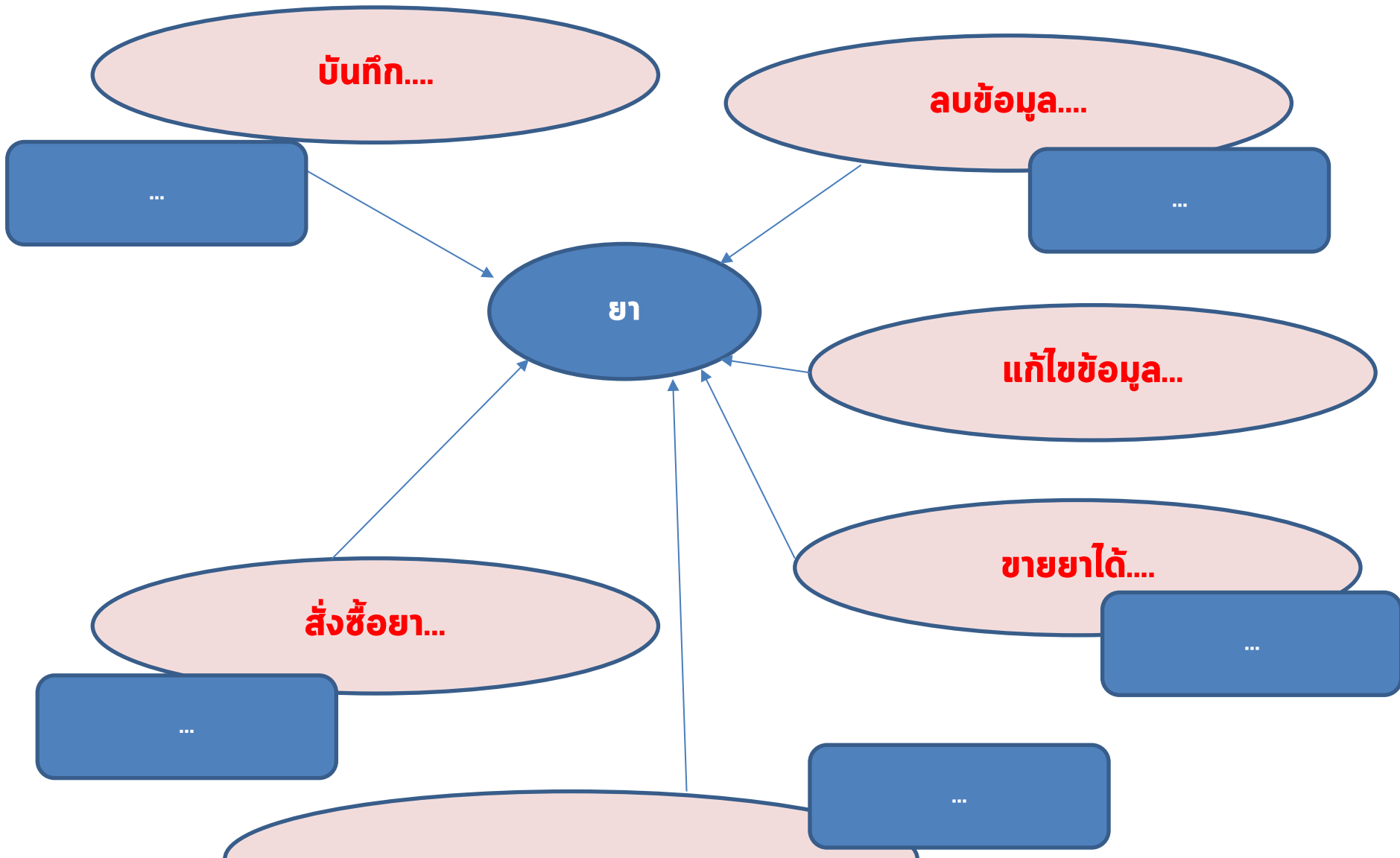
ระบุความต้องการ Requirement Identification



ระบุความต้องการ Requirement Identification



ระบุความต้องการ Requirement Identification



5.4.4 จัดลำดับความสำคัญ (Prioritized)

- กำหนด ลำดับความสำคัญ ความต้องการไหนเร่งด่วน ก็กำหนดให้มีความสำคัญ สูงสุด ความต้องการไหนที่ไม่เร่งด่วน ก็ ลดความสำคัญลงไปตามลำดับ

ตัวอย่าง

0 หมายถึง ไม่สำคัญ

1 หมายถึง ปานกลาง ยังไม่เร่งด่วน

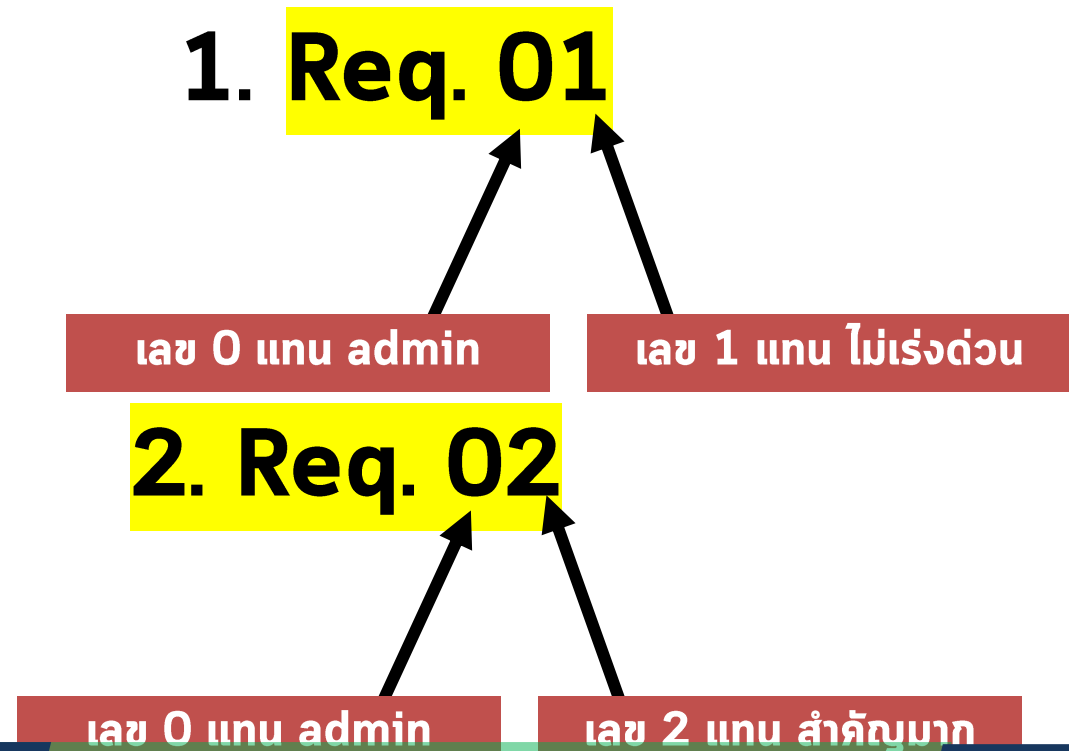
2 หมายถึง สำคัญมาก เร่งด่วนต้องทำก่อน

5.4.4 จัดลำดับความสำคัญ (Prioritized)

- ระบบสามารถบันทึกข้อมูลลูกค้าได้ → 2
- ระบบสามารถสั่งซื้อสินค้าได้ → 2
- ระบบสามารถแก้ไขข้อมูลการสั่งซื้อได้ → 2
- ลูกค้าสามารถ เรียกดูรายการสั่งซื้อได้ → 2
- พนักงานสามารถ ยกเลิกรายการสั่งซื้อได้ → 2
- ระบบแสดงรายการสินค้าไปรษณีย์ -> 1

5.4.4 จัดลำดับความสำคัญ (Prioritized)

1. ลูกค้า หรือคนใช้ ติดต่อเจ้าหน้าที่ → **Req. 01**
2. จนท. สอบถามประวัติ คนใช้ → Req. 02
3. จนท. ออกบัตร คนใช้ → Req. 03
4. จนท. กรอกประวัติ การรักษา
5. จนท. กำหนดคิว คนใช้
6. จนท. แจ้งให้เข้าพบแพทย์
7. คนใช้เข้าพบแพทย์
8. แพทย์วินิจฉัย
9. แพทย์ สั่งยา
10. คนใช้รับยา
11. คนใช้ชำระเงิน
12. คนใช้รับใบเสร็จ

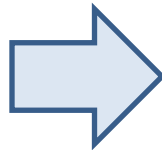


5.4.4 แต่ละความต้องการควรเป็นอิสระในตัวเอง ไม่ผสมปะปนกับความต้องการอื่นๆ

- แยกแต่ละความความต้องการออกจากกัน
- เวลาได้ข้อมูลทั้งหมด จะอยู่รวม ๆ กัน เพราะฉะนั้นต้องแยกแต่ละความความต้องการออกจากกัน

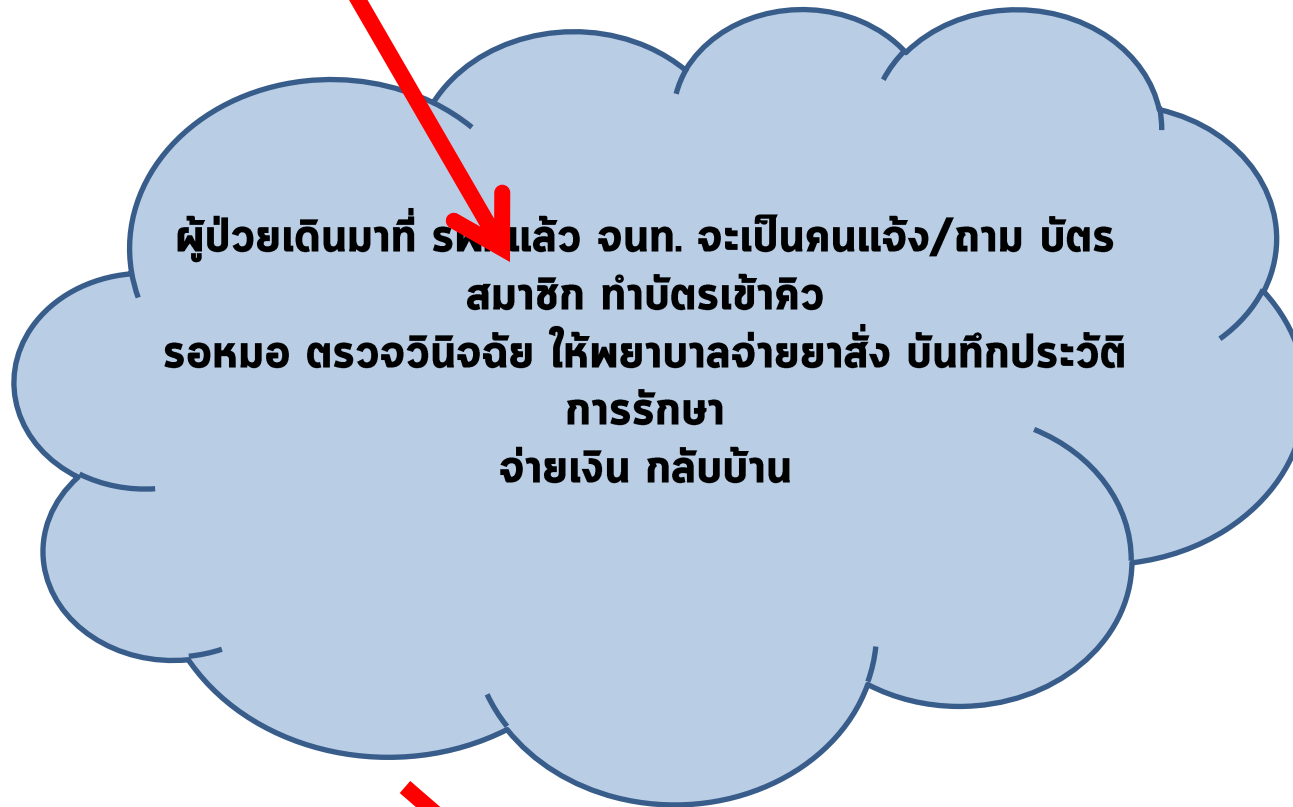
ต.ย.

ผู้ป่วยเดินมาที่ รพ. แล้ว จนท.
จะเป็นคนแจ้ง/ถาม บัตร
สมาชิก ทำบัตรเข้าคิว
รอหมอ ตรวจวินิจฉัย ให้
พยาบาลจ่ายยาสั่ง บันทึก
ประวัติการรักษา
จ่ายเงิน กลับบ้าน



1. ผู้ป่วย ต้องทำบัตรประจำตัวก่อน
2. บันทึกประวัติผู้ป่วย
3. ชำระเงิน
4. วินิจฉัย
5. รับ/จ่ายยา

ใคร.....



- ลำดับก่อน-หลัง
- ความสำคัญ

เขียนเป็นความต้องการ.....

3. คัดเลือกส่วนที่เป็นสาระสำคัญและอยู่ในขอบเขตการพัฒนา (Requirement selection/ Problem Domain)

- สาระ

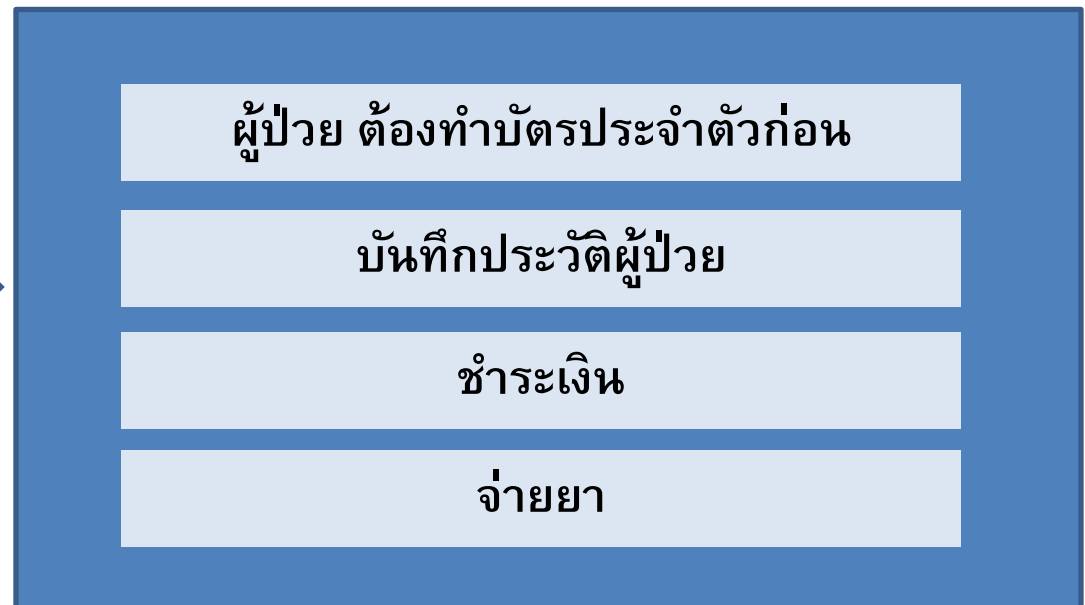
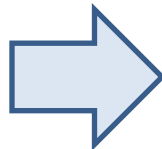
- ไร้สาระ / ไม่จำเป็น / ไม่สำคัญ

- กลับบ้าน

การเข้าคิว จำเป็นต้องมีหรือไม่?

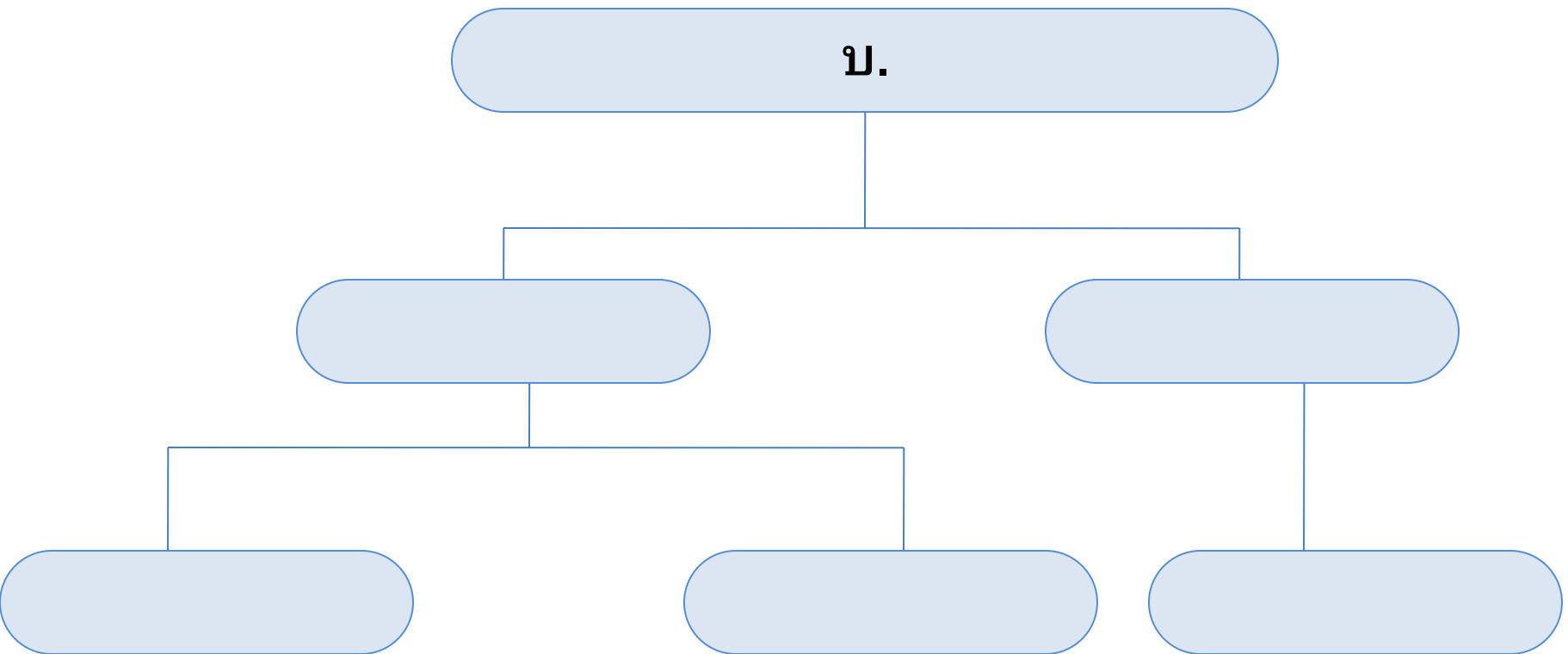
ขึ้นอยู่กับว่าโปรแกรมที่เราจะพัฒนา รวม การทำบัตร คิวหรือไม่ ?

ผู้ป่วยเดินมาที่ห้อง แล้วทำ
บัตร เข้าคิว
รอหมอ ตรวจวินิจฉัยจ่าย
ยา บันทึกประวัติการรักษา
จ่ายเงิน กลับบ้าน



4. จัดจำแนกและจัดโครงสร้างของความต้องการ (Requirement classification and structuring)

ผังงานโครงสร้าง Structure Chart / Organization Chart



5.4.5 ประเภทของความต้องการ (Requirement)

- **ความต้องการเชิงฟังก์ชัน (Functional requirements)** คือการระบุถึงบริการหลักที่ระบบสามารถทำได้ (Statements of Services) เพื่อให้การประยุกต์ใช้สามารถเป็นไปตามวัตถุประสงค์ เช่น **FR**
 - ระบบสามารถบันทึกการสั่งซื้อสินค้าของลูกค้าได้
 - ระบบสามารถออกรายงานสรุปยอดขายประจำเดือนได้
 - ระบบสามารถบันทึกข้อมูลสินค้าได้
- **ความต้องการไม่เป็นเชิงฟังก์ชัน (Non-functional requirements)** คือเงื่อนไข (constraint) ที่ได้กำหนดไว้ต่อการพัฒนาและการนำไปใช้ของระบบ **NFR**
 - เช่น ระบบจะต้องใช้งานบนระบบปฏิบัติการ Linux
 - ความเร็วในการตอบสนอง (response time) ไม่ควรเกิน 3 วินาที
 - สามารถทำงานร่วมกับระบบเก่าได้ เป็นต้น
 - ระบบสามารถรองรับการทำงานหลายหน้าจอได้ (Web Responsive)
 - ทำงานได้กับฐานข้อมูลเก่า
 - ระบบสามารถทำงานได้กับทุกเบราว์เซอร์ (IE , Firefox , Fox)
 - ระบบต้องพัฒนาด้วย MVC ได้

แบบฝึกหัด 5.1

ข้อ 1. จงบอกความต้องการต่อไปนี้ เป็นฟังก์ชัน และไม่เป็นฟังก์ชัน

1. ระบบจะต้องเก็บข้อมูลลูกค้าได้
2. ระบบสามารถเรียกใช้งานได้ทันที
3. ระบบสามารถรองรับการเชื่อมต่อกับ Linux
4. ระบบสามารถใช้งานผ่าน wifi ได้
5. ระบบจะต้องรายงานยอดผู้ป่วย แต่ละวันได้
6. ระบบจะต้องบันทึกการจ่ายยาได้
7. ระบบจะต้องตรวจสอบข้อมูลผู้ป่วยแต่ละโรคได้
8. ระบบจะต้องรันผ่าน iOS ได้
9. ระบบจะต้องใช้งานง่าย
10. ระบบจะต้องรันได้ทุก browser
11. ระบบนี้จะต้องรันบนมือถือได้ทุกแพลตฟอร์ม (android + ios)

FR

Functional requirement

NFR

None-Functional requirement

5.4.5 ประเภทของความต้องการ (Requirement)

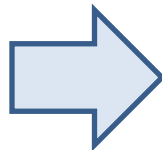
ความต้องการที่เป็นเชิงฟังก์ชัน Functional Requirements

- เป็นความต้องการที่เกี่ยวกับระบบนั้นๆ โดยตรง

FR

Functional requirement

ผู้ป่วยเดินมาที่ห้อง แล้วทำ
บัตร เข้าคิว
รอหมอ ตรวจวินิจฉัยจ่ายยา
บันทึกประวัติการรักษา
จ่ายเงิน กลับบ้าน



Req. 1) ระบบนี้จะต้องสามารถบันทึกข้อมูลผู้ป่วยได้
ได้อย่างไร >

เป็นฟังก์ชัน Function

<input checked="" type="checkbox"/>	ผู้ป่วย ต้องทำบัตรประจำตัวก่อน
<input checked="" type="checkbox"/>	บันทึกประวัติผู้ป่วย
<input checked="" type="checkbox"/>	ชำระเงิน
<input checked="" type="checkbox"/>	จ่ายยา

5.4.5 ประเภทของความต้องการ (Requirement)

ระบบบริหารงานคลินิกผู้ป่วย จะต้อง

- ระบบสามารถบันทึกผลตรวจโรคได้
- ระบบสามารถบันทึกผลวินิจฉัยได้
- ระบบสามารถบันทึกผลรักษาได้
- ทำบัตรสมาชิกได้
- เก็บประวัติการรักษาได้
- ทำจ่ายยาได้
- จ่ายเงินได้
- ทำการออกใบเสร็จได้
-อื่นๆ

Functional requirement

FR

อยู่ในระบบนี้ และเป็นหน้าที่
ที่ซอฟต์แวร์หรือโปรแกรมจะต้องทำได้ เรียกว่าเป็น Functional
requirements

5.4.5 ประเภทของความต้องการ (Requirement)

ต.ย.

functional requirements

Req. 1) ระบบนี้จะต้องสามารถบันทึกข้อมูลผู้ป่วยได้

ID: **Req01**

ผู้ที่เกี่ยวข้อง : พนักงาน(เจ้าหน้าที่/พยาบาล)

ทำอะไร >

1.1 ต้องเก็บ ชื่อ นามสกุล ที่อยู่ เบอร์โทร เลขบัตรประชาชน อีเมล

1.2 เบอร์โทร ต้องเก็บ ตัวอักษร กรณี กด ต่อได้ **025210141#4807**

1.3...เลขบัตรประชาชน ต้องเก็บ 13 หลัก ห้ามเว้นว่าง..

.....

.....

.....

ระดับความสำคัญ : สูง (2)

ข้อมูลด้านบนนี้ ได้มาจากใคร ?

แอดมิน เป็นคนบันทึกประวัติ ผู้ป่วยใช้หรือไม่ ?

การชำระเงิน

functional requirements

Req. 2) ระบบจะต้องสามารถชำระ ค่ารักษาพยาบาล ได้ โดย มีข้อกำหนดดังนี้ / มีรายละเอียดดังนี้

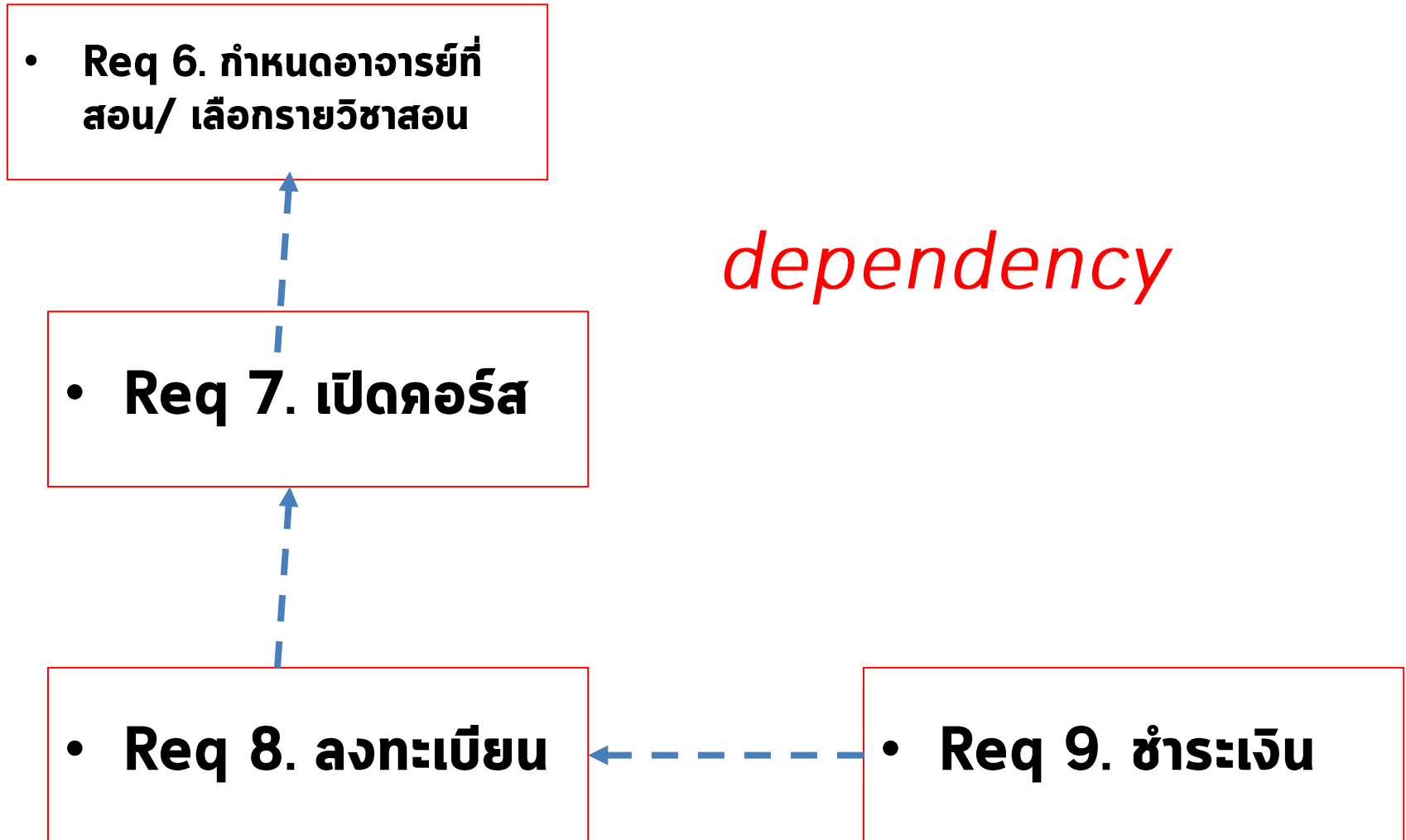
2.1 ระบบสามารถจ่าย/ชำระได้ทั้ง เงินสด และเครดิต

2.2 ไม่อนุญาตให้ผ่อนชำระ

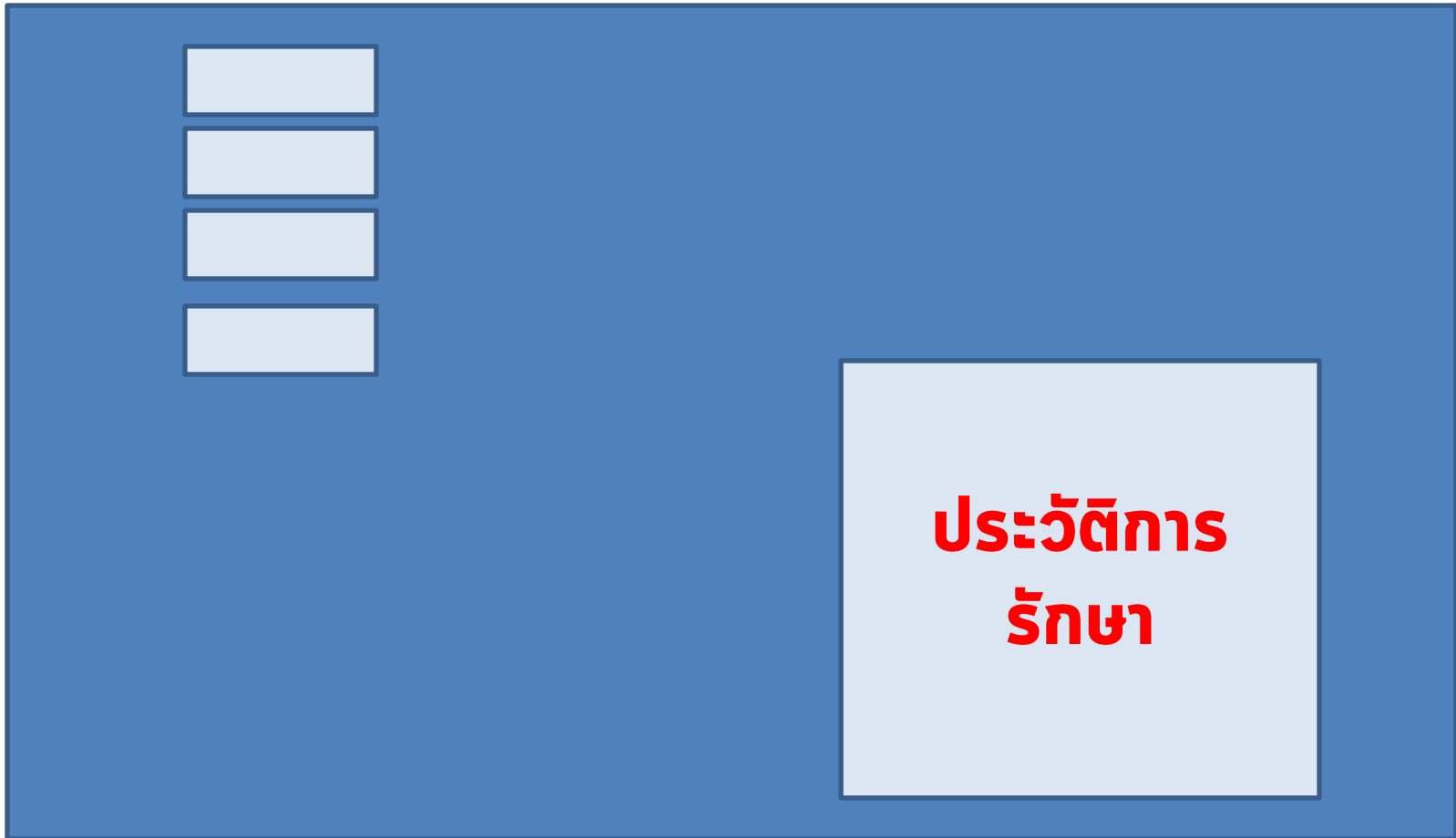
ตอนที่ จะตรวจรับ ซอฟต์แวร์ นี้หรือไม่
ถ้า ไม่สามารถจ่ายเป็นด้วย บัตรเครดิต ได้

functional requirements

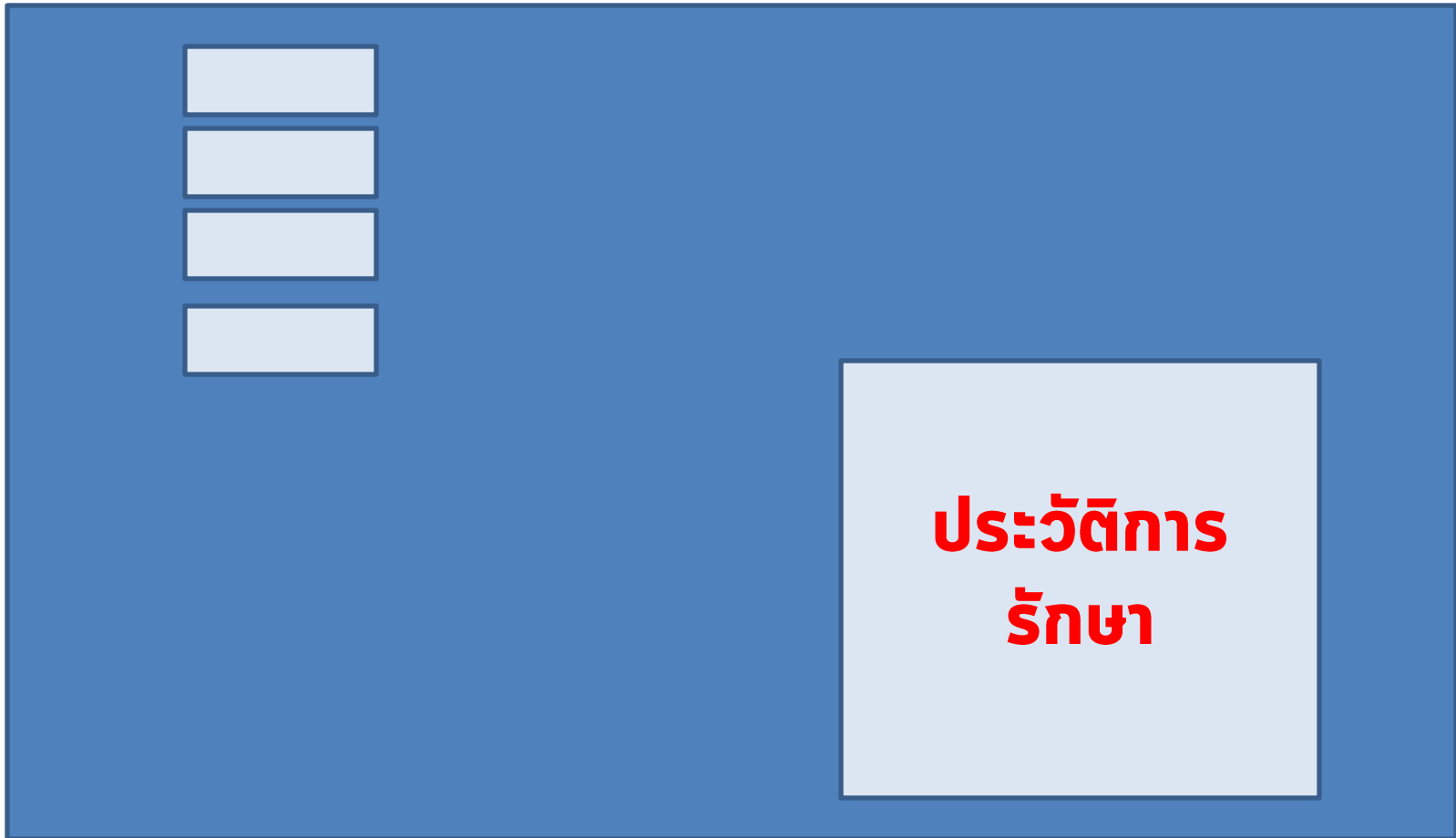
dependency



หน้าจอชำระเงิน



ข้อมูลผู้ป่วย



5.4.5 ประเภทของความต้องการ (Requirement)

การจำแนกประเภทของ

NFR: Non-functional requirements

- **Product requirements**
 - ใช้งานได้อย่างสะดวก (Usability requirements)
 - มีประสิทธิภาพดี (Efficiency requirements): Performance, Speed
 - มีความมั่นคงสูง (Reliability requirements)
 - สามารถใช้งานในสภาพแวดล้อมที่ต่างกันได้ (Portability requirements)
 - สีที่ใช้ของเว็บต้องสวยงามเหมาะสมสำหรับทุกคน ทุกเพศทุกวัย
- **Organizational requirements**
 - สามารถส่งมอบได้ในเวลาที่กำหนด (Delivery requirements)
 - ต้องสร้างด้วยวิธีการและเทคโนโลยีที่กำหนด (Implementation requirements) เช่น responsive web design , Model View Control:MVC
 - ต้องพัฒนาโดยยึดตามมาตรฐานของการพัฒนาที่กำหนด (Standard requirements) เช่น ให้กระบวนการพัฒนามีมาตรฐานตาม ISO เป็นต้น
 - ต้องรองรับมาตรฐาน ISO-14000 เป็นกระบวนการพัฒนาซอฟต์แวร์จะต้องเป็นมิตรต่อสิ่งแวดล้อม
 - รองรับ PDPA
- **External requirements**
 - จะต้องรองรับการเชื่อมต่อจากภายนอกได้ (Interoperability requirements)
 - จะต้องไม่ผิดศีลธรรม (Ethical requirements)
 - จะต้องไม่ผิดกฎหมาย (Legislative (Law) requirements)

Non-functional requirements (NFRs) คือความต้องการที่ไม่เกี่ยวข้องกับฟังก์ชันหรือการทำงานของระบบ แต่มีผลต่อประสิทธิภาพ ประสิทธิภาพ ความปลอดภัย และประสบการณ์ผู้ใช้ของระบบ ต่อไปนี้เป็นตัวอย่างการจำแนกประเภทของ **Non-functional requirements**:

1. ประสิทธิภาพ (Performance)

- ประสิทธิภาพในการประมวลผล: ระบบต้องสามารถประมวลผลข้อมูลด้วยความเร็วที่เพียงพอในสถานการณ์ปกติและสูงสุด ตัวอย่างเช่น ระบบต้องสามารถรองรับจำนวนผู้ใช้พร้อมกันที่มากขึ้นได้โดยไม่ล่มเหลว.
- ความล่าช้า (Latency): ระบบต้องมีความล่าช้าในการตอบสนองต่ำกว่าเวลาที่กำหนด ตัวอย่างเช่น ระบบต้องสามารถคืนข้อมูลหรือแสดงผลให้กับผู้ใช้ในเวลาไม่เกิน 1 วินาที.

2. ความปลอดภัย (Security):

- ความปลอดภัยของข้อมูล: ระบบต้องมีมาตรการในการป้องกันการเข้าถึงข้อมูลที่ไม่เหมาะสมหรือไม่ได้รับอนุญาต และต้องมีการเข้ารหัสข้อมูลที่สำคัญ.
- การรักษาความลับ: ระบบต้องรักษาความลับของข้อมูลและข้อมูลสำคัญ เช่น ข้อมูลลูกค้าหรือข้อมูลบัตรเครดิต.

3. ความมั่นคง (Reliability):

- ความเสถียรภาพ: ระบบต้องสามารถทำงานได้ต่อเนื่องและมีความเสถียรในระยะเวลายาวนาน โดยไม่มีการล่มเหลว.
- การกู้คืนข้อมูล: ระบบต้องสามารถกู้คืนข้อมูลหรือสถานะในกรณีที่เกิดข้อขัดข้องหรือฉุกเฉิน.

4. การใช้งาน (Usability):

- ใช้งานง่าย: ระบบต้องมีอินเตอร์เฟซที่ใช้งานง่ายและไม่ซับซ้อน และคำแนะนำในการใช้งาน.
- การเข้าถึง (Accessibility): ระบบต้องสามารถเข้าถึงได้ง่ายสำหรับผู้ใช้งานรวมถึงผู้พิการ.

5. การบำรุงรักษา (Maintainability):

- ความสามารถในการปรับปรุง: ระบบต้องสามารถปรับปรุงหรือปรับแก้ไขโครงสร้างหรือฟังก์ชันได้โดยง่าย.
- ความเข้าใจง่าย: ระบบต้องมีความเป็นระเบียบและความเข้าใจง่ายต่อผู้ดูแลระบบ.

6. ประสบการณ์ผู้ใช้ (User Experience):

- ประสบการณ์การใช้งานที่น่าพึงพอใจ: ระบบต้องมีการออกแบบให้ผู้ใช้มีประสบการณ์การใช้งานที่ดีและน่าพอใจ.
- ประสบการณ์การใช้งานบนอุปกรณ์ต่าง ๆ: ระบบต้องสามารถให้ประสบการณ์การใช้งานที่ดีบนอุปกรณ์ที่แตกต่างกัน เช่น โทรศัพท์มือถือและคอมพิวเตอร์แท็บเล็ต.

ข้างบนเป็นตัวอย่างของประเภทของ Non-functional requirements ที่มีต่อระบบ ความต้องการเหล่านี้ช่วยให้ระบบมีประสิทธิภาพ ปลอดภัย และมีประสบการณ์ผู้ใช้ที่ดีในการใช้งานอย่างถูกต้องและเหมาะสม

ระดับของความต้องการ

- **ความต้องการของผู้ใช้ (User requirements)**
 - ถ้อยแถลงที่เป็นภาษาธรรมชาติ (Natural language) ตลอดจนแผนภาพต่างๆ ที่แสดงให้เห็นถึงความสามารถในการทำงานของระบบและเงื่อนไขในการทำงานของระบบ
 - ความต้องการของผู้ใช้จะต้องเขียนให้ผู้ใช้เข้าใจได้ง่ายเป็นหลัก
- **ความต้องการของระบบ (System requirements)**
 - คือการระบุถึงสิ่งที่ระบบควรจะมี เพื่อให้สามารถทำงานได้ตรงตามความต้องการของผู้ใช้
 - ทำหลังจากได้ข้อกำหนดความต้องการของผู้ใช้มาแล้ว
 - มักจะเขียนด้วยแบบจำลอง (Models) เพื่อแสดงองค์ประกอบของระบบในแง่มุมต่างๆ
 - ใช้เป็นข้อกำหนดในการออกแบบต่อไป

ตัวอย่างข้อกำหนดความต้องการ

FR: functional requirements

ReqID : 01

Name : ระบบสามารถเก็บข้อมูลของลูกค้าได้ (เป็นความต้องการระดับผู้ใช้)

ผู้ที่เกี่ยวข้อง :พนักงานขาย.....

รายละเอียดของข้อกำหนดความต้องการ (Requirement specification)

1. ข้อมูลลูกค้าจะต้องถูกบันทึกลงในฐานข้อมูล
2. ระบบจะไม่ยอมให้เก็บข้อมูลลูกค้า ถ้าขาดข้อมูลต่อไปนี้ เลขที่บัตรประจำตัว ชื่อ นามสกุล อีเมล วันเดือนปีเกิด เป็นต้น
3. ระบบจะต้องตรวจสอบรูปแบบของ เลขที่บัตรประจำตัว ก่อนจัดเก็บ
4. ระบบจะต้องตรวจสอบรูปแบบของอีเมลก่อนจัดเก็บ
5. ข้อมูลของลูกค้าจะต้องถูกจัดเก็บให้เป็นความลับและปลอดภัย เช่น หมายเลขบัญชี ๘. จะต้องเข้ารหัส
6. จะต้องสามารถเรียกดูได้ในภายหลังว่า พนักงานคนใดเป็นผู้บันทึกหรือแก้ไขข้อมูลลูกค้าครั้งล่าสุด
7. Etc.

ระดับความสำคัญ:สูง.....

วัน เดือน ปีเกิด เป็น ค.ศ. หรือ พ.ศ. วัน ก่อน หรือ เดือนก่อน
หรือ เป็น ภาษาไทย หรือ ภาษาอังกฤษ เดือน เป็นตัวย่อหรือเต็ม

ตัวอย่างข้อกำหนดความต้องการ

- ถ้าออกแบบฐานข้อมูลแล้วป้อนข้อมูล หรือ บันทึกข้อมูล

รหัส	ชื่อ-นามสกุล	อีเมล	เพศ	เบอร์โทร
29101110125				

Not Null

แนวทางการเขียนความต้องการด้วยภาษาธรรมชาติ

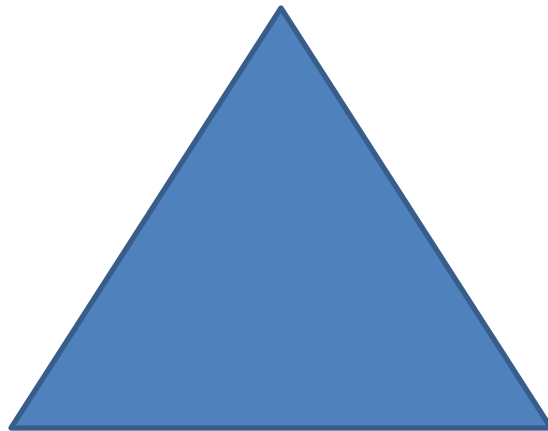
- สร้างรูปแบบมาตรฐานในการเขียน (standard format)
- ใช้ภาษาที่ตรงไปตรงมาอย่างมีเหตุมีผล และไม่มีความขัดแย้งกัน
- เน้นคำที่เป็นส่วนสำคัญของความต้องการนั้นๆ
- เลี่ยงการใช้ศัพท์ทางเทคนิคมากเกินไป (ศัพท์ทางคอมพิวเตอร์)

ข้อเสียของการเขียนความต้องการด้วยภาษาธรรมชาติ

- สื่อความหมายกำกวมไม่ชัด (Lack of clarity)
- มีความสับสน (Confusion)
- ผสมปนเป (amalgamation)

วิธีการอธิบายความต้องการแบบอื่น (นอกเหนือจากภาษาธรรมชาติ)

- ภาษาที่เป็นโครงสร้าง (Structured Language)
- แบบจำลองสัญลักษณ์ภาพ (Graphical model)
- ข้อกำหนดทางคณิตศาสตร์ (Formal/Mathematical specification)



$$1/2 * \text{จู่} * \text{สี่}$$

การจูดคุ่มทุน
POS

การตรวจสอบความถูกต้องของความต้องการ

- การตรวจสอบความถูกต้องของความต้องการ (Requirement validation) คือ การตรวจว่าความต้องการที่ได้มานั้น ถูกต้องและตรงกับความต้องการของผู้ใช้อย่างแท้จริงหรือไม่
- หลักในการพิจารณา
 - **Validity** ความต้องการนั้นตรงกับที่ผู้ใช้ต้องการจริงหรือไม่ สามารถแก้ปัญหาให้ผู้ใช้ได้จริงหรือไม่
 - **Consistency** มีความขัดแย้งกันระหว่างความต้องการหรือไม่
 - **Completeness** เป็นความต้องการที่ครบถ้วนของผู้ใช้ทุกคนหรือไม่
 - **Realism** สามารถสร้างได้จริงตามความต้องการหรือไม่
 - **Verifiability** สามารถตรวจสอบได้หลังจากพัฒนาเสร็จแล้วหรือไม่

ตัวอย่างข้อกำหนดความต้องการ

functional requirements

- ReqID: 01
 - ระบบสามารถเก็บข้อมูลของลูกค้าได้ (เป็นความต้องการระดับผู้ใช้)
- ข้อกำหนดความต้องการ (Requirement specification)
 - ข้อมูลลูกค้าจะต้องถูกบันทึกลงในฐานข้อมูล
 - ระบบจะไม่ยอมให้เก็บข้อมูลลูกค้า ถ้าขาดข้อมูลต่อไปนี้ เลขที่บัตรประจำตัว ชื่อ นามสกุล อีเมล
 - ระบบจะต้องตรวจสอบรูปแบบของ เลขที่บัตรประจำตัว ก่อนจัดเก็บ
 - ระบบจะต้องตรวจสอบรูปแบบของอีเมลก่อนจัดเก็บ
 - ข้อมูลของลูกค้าจะต้องถูกจัดเก็บให้เป็นความลับและปลอดภัย เช่น หมายเลข บัญชี ๘. จะต้องเข้ารหัส
 - จะต้องสามารถเรียกดูได้ในภายหลังว่า พนักงานคนใดเป็นผู้บันทึกหรือแก้ไขข้อมูลลูกค้าครั้งล่าสุด
 - Etc.

วัน เดือน ปีเกิด เป็น ค.ศ. หรือ พ.ศ. วัน ก่อน หรือ เดือนก่อน
หรือ เป็น ภาษาไทย หรือ ภาษาอังกฤษ เดือน เป็นตัวย่อหรือเต็ม

ตัวอย่างข้อกำหนดความต้องการ

ReqID: 010

Power User

ระดับความสำคัญ
0 ไม่สำคัญ
1 ปานกลาง
2 สำคัญมาก/
เร่งด่วน

ระดับความต้องการ
0 : User
1 : System

ปล. เป็นแค่แนวทางเท่านั้น

**วัน เดือน ปีเกิด เป็น ค.ศ. หรือ พ.ศ. วัน ก่อน หรือ เดือนก่อน
หรือ เป็น ภาษาไทย หรือ ภาษาอังกฤษ เดือน เป็นตัวย่อหรือเต็ม**

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบการจองห้องพักโรงแรม

Req. 01: การลงทะเบียนและเข้าสู่ระบบ

- ระบบจะต้องมีหน้าเข้าสู่ระบบสำหรับผู้ใช้ทั้งเจ้าของโรงแรมและลูกค้า.
- ผู้ใช้ต้องมีบัญชีผู้ใช้และรหัสผ่านเพื่อลงทะเบียนและเข้าสู่ระบบ.

Req. 02: การค้นหาห้องพัก

- ระบบจะต้องมีการค้นหาห้องพักโรงแรมโดยให้ผู้ใช้ระบุวันที่เข้าพักและวันที่ออก.

Req. 03: รายละเอียดห้องพัก

- ผู้ใช้จะต้องสามารถดูรายละเอียดของห้องพักรวมถึงรูปภาพและสิ่งอำนวยความสะดวก.

Req. 04: การจองห้องพัก

- ผู้ใช้จะต้องสามารถเลือกห้องพักและทำการจองโดยระบุจำนวนผู้เข้าพักและรายละเอียดการชำระเงิน.

Req. 05: การชำระเงิน

- ระบบจะต้องรองรับการชำระเงินออนไลน์ผ่านวิธีการต่าง ๆ เช่น บัตรเครดิต, PayPal, หรือโอนเงินผ่านธนาคาร.

Req. 06: การยกเลิกการจอง

- ผู้ใช้จะต้องสามารถยกเลิกการจองกับค่าธรรมเนียมหรือเงื่อนไขที่กำหนด.

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบการจองห้องพักโรงแรม (ต่อ)

Req. 07: การยืนยันการจอง

- ระบบจะต้องส่งอีเมลยืนยันการจองพร้อมรายละเอียดการเข้าพักถึงผู้ใช้และโรงแรม.

Req. 08: การบริหารจัดการการจอง

- โรงแรมจะต้องมีหน้าบริหารจัดการเพื่อยืนยันและจัดการการจอง, เช่น การกำหนดสถานะการจองและการตรวจสอบความพร้อมในการเข้าพัก.

Req. 09: การรายงานและสถิติ

- ระบบจะต้องสร้างรายงานเกี่ยวกับการจอง, รายได้, และสถิติการใช้บริการของโรงแรม.

Req. 10: ความปลอดภัยและความเป็นส่วนตัว

- ระบบจะต้องมีมาตรการรักษาความปลอดภัยของข้อมูลส่วนบุคคลและข้อมูลการชำระเงิน.

ข้อกำหนดเหล่านี้เป็นตัวอย่างเพียงแต่บางส่วนของความต้องการที่อาจปรากฏในระบบการจองห้องพักโรงแรม ความต้องการที่แท้จริงจะต้องเน้นไปยังการทำงานและประสิทธิภาพของระบบที่ต้องมีให้บรรลุได้ในระดับสูงสุดและมีความปลอดภัยในการใช้งาน.

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบร้านค้าแคร์:

Req. 01: ระบบการจัดการลูกค้า

- ระบบจะต้องสามารถลงทะเบียนลูกค้าใหม่และบริหารจัดการข้อมูลลูกค้าเช่น ชื่อ, ที่อยู่, และข้อมูลติดต่อ.

Req. 02: การนัดหมายและตารางการบริการ

- ระบบจะต้องสนับสนุนการจัดการการนัดหมายลูกค้าและสร้างตารางการบริการสำหรับร้านค้าแคร์.

Req. 03: การบันทึกข้อมูลรถยนต์

- ระบบจะต้องสามารถบันทึกข้อมูลรถยนต์ของลูกค้า เช่น ยี่ห้อ, รุ่น, และหมายเลขทะเบียน.

Req. 04: การตรวจสอบสถานะบริการ

- ลูกค้าจะต้องสามารถตรวจสอบสถานะของบริการรถยนต์ที่อยู่ในกระบวนการ.

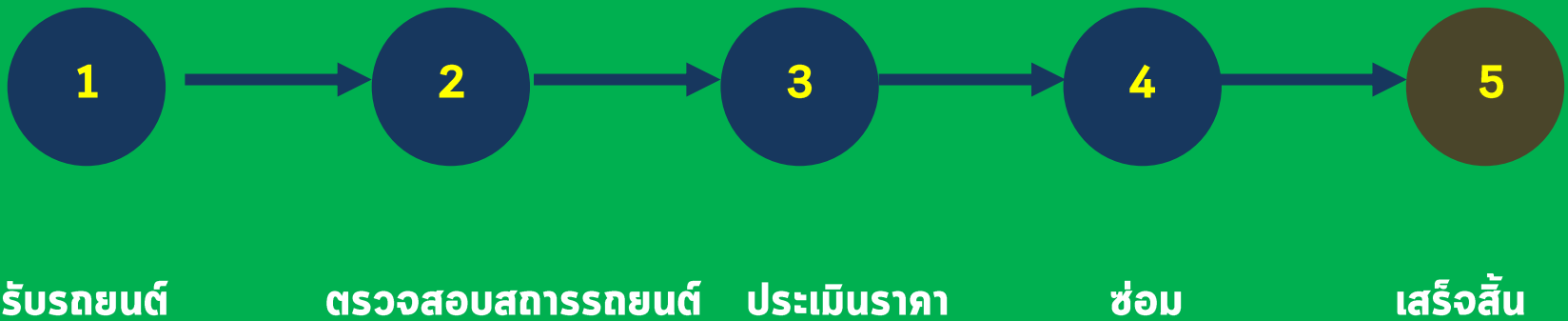
Req. 05: การคำนวณค่าบริการ

- ระบบจะต้องสามารถคำนวณค่าบริการของร้านค้าแคร์ตามบริการที่เลือกและรายละเอียดของรถยนต์.

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบร้านค้ารีแคร์:

Req. 04: การตรวจสอบสถานะบริการ

- ลูกค้าจะต้องสามารถตรวจสอบสถานะของบริการรถยนต์ที่อยู่ในกระบวนการ.



ตัวอย่างข้อกำหนดความต้องการสำหรับระบบการจอง ห้องพักโรงแรม (ต่อ)

Req. 06: การจัดการการชำระเงิน

- ระบบจะต้องรองรับการชำระเงินออนไลน์หรือหน้าร้านค้าบัตรเครดิต.

Req. 07: ระบบความปลอดภัย

- ระบบจะต้องมีมาตรการรักษาความปลอดภัยของข้อมูลลูกค้าและการทำธุรกรรม.

Req. 08: ระบบการสื่อสาร

- ระบบจะต้องส่งอีเมลหรือข้อความแจ้งเตือนถึงลูกค้าเกี่ยวกับสถานะการบริการและการนัดหมาย.

Req. 09: การสร้างรายงาน

- ระบบจะต้องสร้างรายงานเกี่ยวกับประสิทธิภาพการบริการ, รายได้, และความเคลื่อนไหวของร้านค้าบัตรเครดิต.

Req. 10: การสนับสนุนผู้ดูแลระบบ

- ระบบจะต้องมีบทบาทผู้ดูแลระบบที่สามารถเพิ่มหรือแก้ไขข้อมูลลูกค้าและบริการ.

ข้อกำหนดเหล่านี้เป็นตัวอย่างเพียงแค่ว่าบางส่วนของความต้องการที่อาจปรากฏในระบบร้านค้าบัตรเครดิต ความต้องการที่แท้จริงจะต้องเน้นไปยังการทำงานและประสิทธิภาพของระบบที่ต้องมีให้บรรลุได้ในระดับสูงสุดและมีความปลอดภัยในการใช้งาน.

รายละเอียดข้อกำหนดความต้องการสำหรับ "Req. 01: การลงทะเบียนและเข้าสู่ระบบระบบร้านค้าแคร์":

คำอธิบาย:

- ระบบต้องสามารถอนุญาตให้ผู้ใช้ลงทะเบียนเพื่อสร้างบัญชีของตนเองในระบบร้านค้าแคร์ ซึ่งจะใช้ในการเข้าสู่ระบบเพื่อใช้บริการต่าง ๆ ของร้านค้าแคร์.

รายละเอียดข้อกำหนดความต้องการ:

1. Req. 01.01: หน้าทีลงทะเบียน

- ระบบจะต้องมีหน้าที่ให้ผู้ใช้ลงทะเบียนโดยให้กรอกข้อมูลส่วนตัว เช่น ชื่อ, ที่อยู่, อีเมล, และรหัสผ่าน.

2. Req. 01.02: ตรวจสอบความถูกต้องของข้อมูล

- ระบบจะต้องทำการตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้ลงทะเบียนโดยตรวจสอบว่าข้อมูลต้องไม่ว่างเปล่าและมีรูปแบบที่ถูกต้อง.

3. Req. 01.03: การสร้างบัญชีผู้ใช้

- เมื่อข้อมูลการลงทะเบียนถูกตรวจสอบและถูกต้อง ระบบจะต้องสร้างบัญชีผู้ใช้ใหม่และเก็บข้อมูลลงในฐานข้อมูล.

4. Req. 01.04: การส่งอีเมลยืนยัน

- ระบบจะต้องส่งอีเมลยืนยันการลงทะเบียนไปยังอีเมลที่ผู้ใช้ลงทะเบียนเพื่อให้ผู้ใช้ทำการยืนยันบัญชีของตน.

รายละเอียดข้อกำหนดความต้องการสำหรับ "Req. 01: การลงทะเบียนและเข้าสู่ระบบระบบร้านค้าคาร์แคร์":

5. Req. 01.05: การเข้าสู่ระบบ

- ระบบจะต้องมีหน้าที่ให้ผู้ใช้เข้าสู่ระบบโดยกรอกชื่อผู้ใช้และรหัสผ่านที่ลงทะเบียนไว้.

6. Req. 01.06: การลืมหรหัสผ่าน

- ระบบจะต้องรองรับกระบวนการกู้คืนรหัสผ่านโดยการส่งลิงก์กู้คืนรหัสผ่านไปยังอีเมลที่ลงทะเบียน.

7. Req. 01.07: การป้องกันการบุกรุก

- ระบบจะต้องมีมาตรการความปลอดภัยเพื่อป้องกันการบุกรุกบัญชีผู้ใช้.

8. Req. 01.08: การบันทึกการเข้าสู่ระบบ

- ระบบจะต้องบันทึกข้อมูลเกี่ยวกับการเข้าสู่ระบบเพื่อติดตามประวัติการใช้งาน.

9. Req. 01.09: การลงทะเบียนด้วยบัญชีสังคมออนไลน์

- ระบบจะต้องรองรับการลงทะเบียนโดยใช้บัญชีสังคมออนไลน์ เช่น บัญชี Google หรือ Facebook

Non-functional requirements

รายละเอียดข้อกำหนดความต้องการสำหรับ "Req. 01: การลงทะเบียนและเข้าสู่ระบบระบบร้านค้าเร่":

10. Req. 01.10: การตรวจสอบความปลอดภัยของรหัสผ่าน

- ระบบจะต้องมีการตรวจสอบความปลอดภัยของรหัสผ่านโดยกำหนดเกณฑ์ความปลอดภัย เช่น ความยาวขั้นต่ำ, การใช้ตัวอักษรและตัวเลข **Non-functional requirements**

ข้อกำหนดความต้องการนี้เป็นการจัดเตรียมสำหรับการลงทะเบียนและเข้าสู่ระบบในระบบร้านค้าเร่ มันระบุว่าระบบต้องมีความสามารถในการจัดการข้อมูลลงทะเบียนของผู้ใช้และต้องม

แบบฝึกหัด 5.1

ข้อ 2. การเขียนข้อกำหนดความต้องการ

- ReqID: 2
 - **ระบบสามารถเก็บข้อมูลการซื้อสินค้าของลูกค้าได้**
- ข้อกำหนดความต้องการ (Requirement specification)
 1.
 2.
 3.
 4.
 5.
 6.
 7.
 8.
 9.
 10.

แบบฝึกหัด 5.1

ข้อ 3. การเขียนข้อกำหนดความต้องการ

- ReqID: 2
 - **ระบบออกใบกำกับภาษีให้ลูกค้า**
- ข้อกำหนดความต้องการ (Requirement specification)
 1. จะต้องมีข้อมูล เลขที่ วันที่ ชื่อลูกค้า หน่วยงาน ฯลฯ
 2. จะต้องมี ตราประทับ บ.
 3. จะต้องมี VAT
 4. กรณีที่เป็นสมาชิก จะมีส่วนลด ...%
 5.
 6.
 7.
 8.
 9.
 10.

แบบฝึกหัด 5.1

ข้อ 4. การเขียนข้อกำหนดความต้องการ

- ReqID: 2
 - ระบบรองรับการชำระเงินทั้งเงินสด โอนเข้าบัญชี และบัตรเครดิตได้
- ข้อกำหนดความต้องการ (Requirement specification)
 1.
 2.
 3.
 4.
 5.
 6.
 7.
 8.
 9.
 10.

เทคนิคในการหาความต้องการ

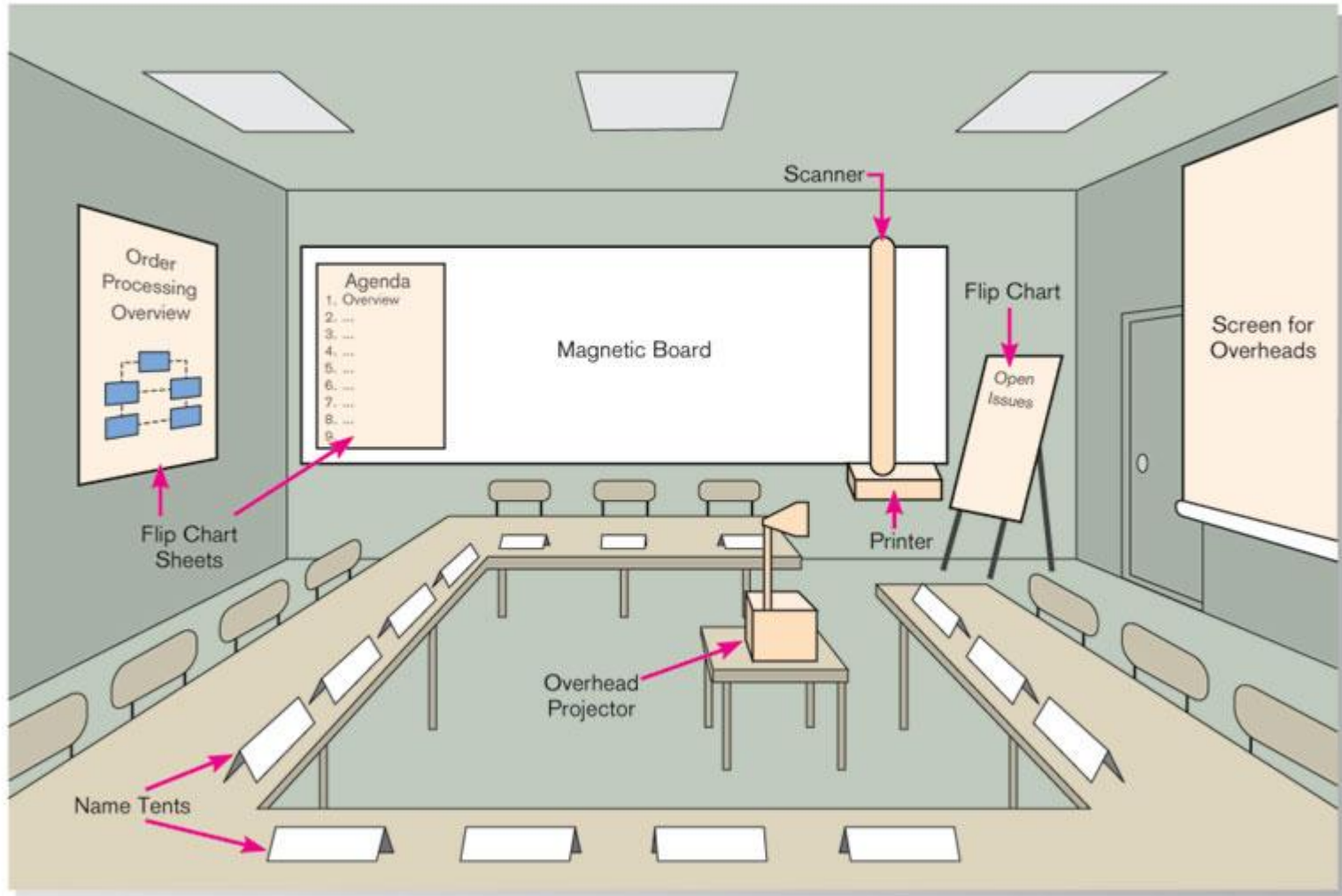
- Joint Application Development (JAD)
- Rapid Application Development (RAD)
 - Prototyping
- **Agile**
- **Scrum**

Joint Application Design (JAD)

- คือเทคนิคการกำหนดความต้องการของระบบโดยเน้นการรวบรวมบุคคลที่เกี่ยวข้องกับระบบ (stakeholders) มากำหนดความต้องการร่วมกัน
 - User , Manager, Sponsor, System Analysis
 - เป้าหมาย คือ หาความต้องการไปพร้อม ๆ กัน จากแต่ละ viewpoints
- การดำเนินการเป็นไปในลักษณะการประชุม (Meeting/Brain Storming) {Zoom , Google Meet}
- จนสุดท้ายได้ข้อสรุปร่วมกันเป็น ข้อกำหนดความต้องการ (SRS) ซึ่งแสดงคุณลักษณะของระบบที่ต้องการ

JAD Technique

Figure 6-6 Illustration of the typical room layout for a JAD



Source: Adapted from Wood and Silver, 1995.

JAD Technique

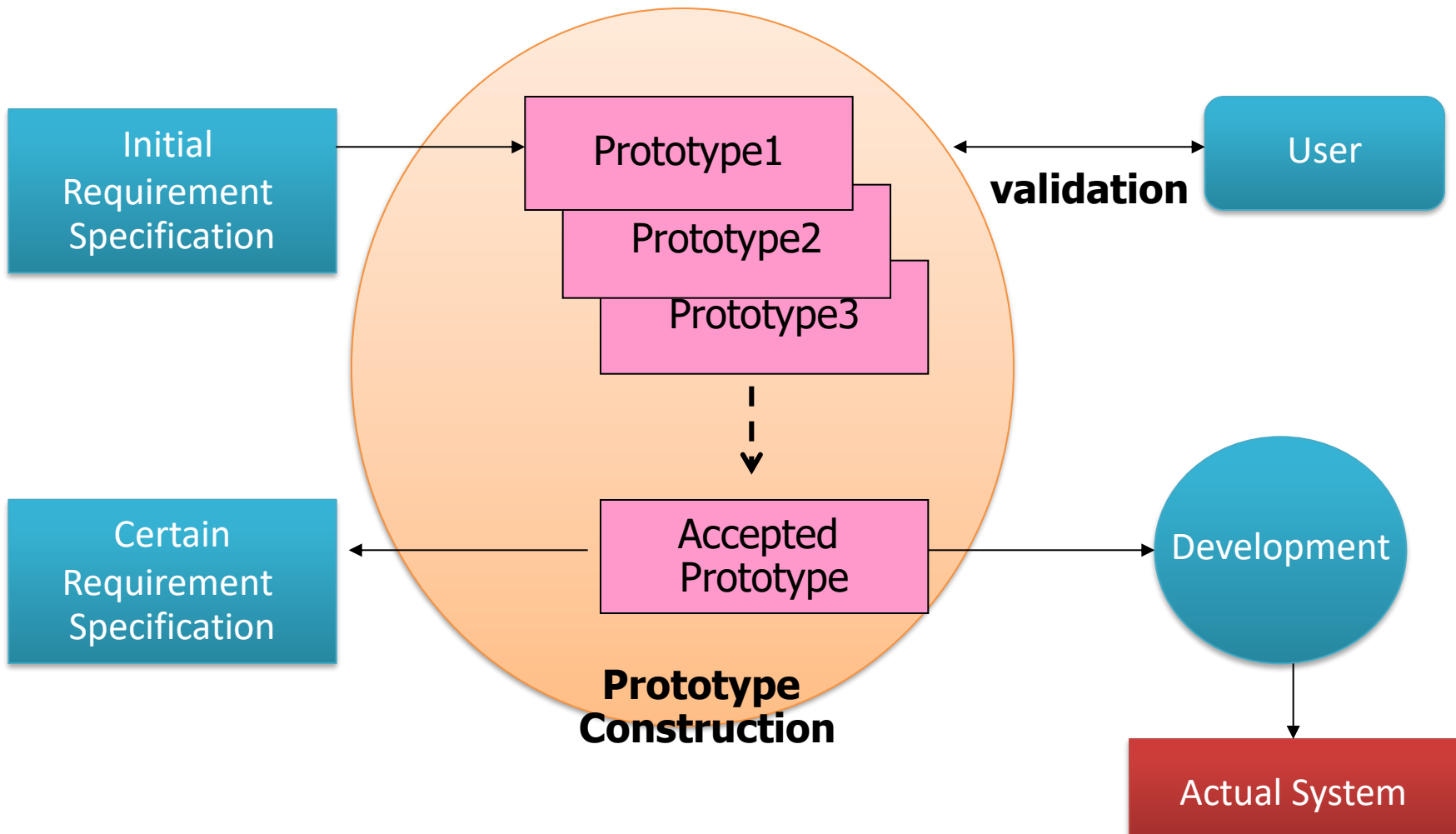
ถ้าองค์กรเล็ก ไม่เกิน 10 จะโอเค
ถ้าขนาดกลาง 10 -100 พอได้
ถ้าใหญ่ เกิน ไม่เหมาะสม แต่อาจจะคัดเลือกตัวแทนได้

ถ้าระบบใหญ่มีคนเกี่ยวข้องจำนวนมาก อาจจะแบ่งการประชุมออกเป็นห้องย่อย หรือจัดประชุมหลายครั้งแทน

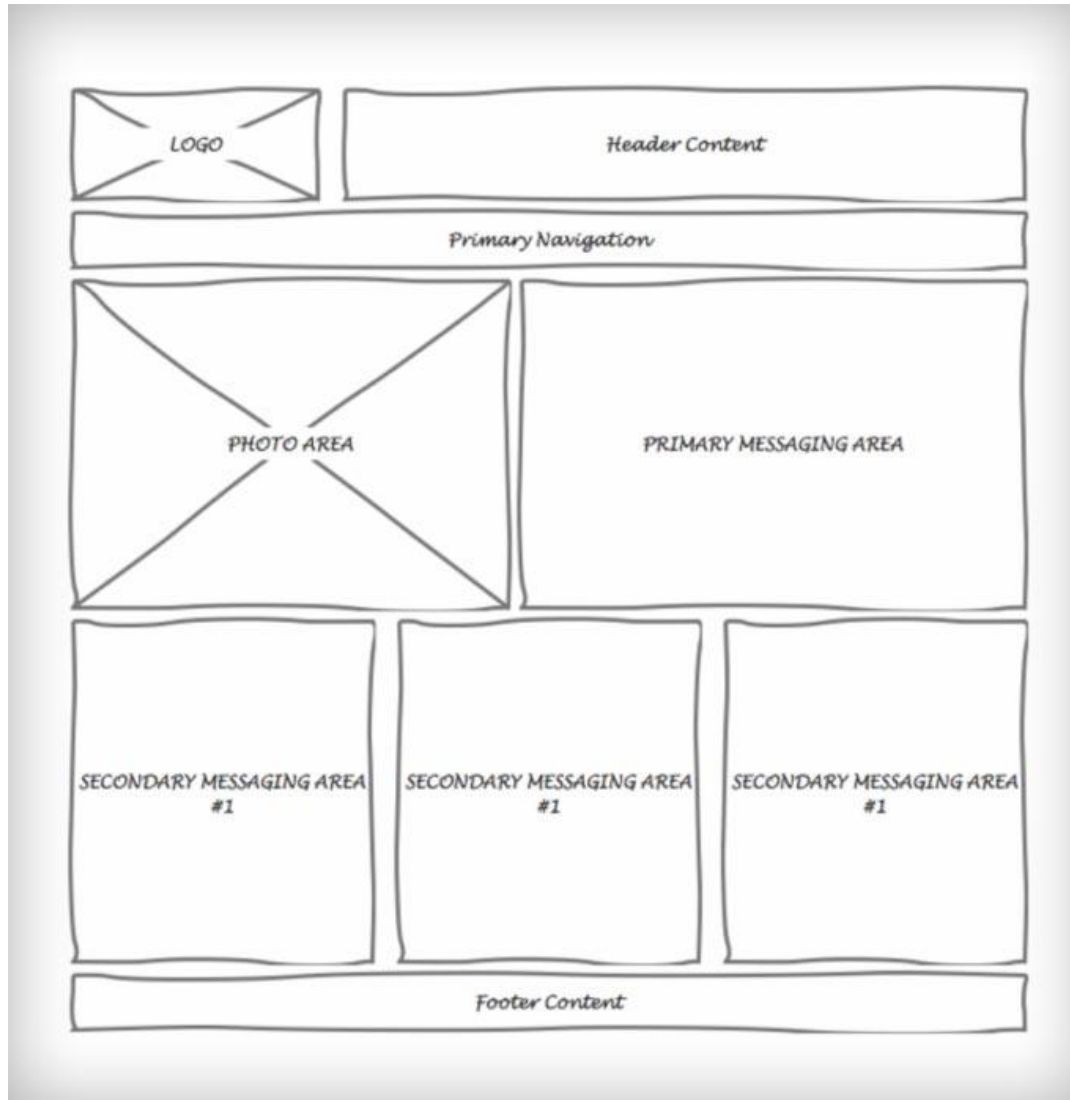
Prototyping

- ใช้ในกรณีที่ผู้ใช้ยังไม่ทราบความต้องการของตนเองอย่างชัดเจน
- สร้างต้นแบบของระบบอย่างรวดเร็วจากความต้องการที่มีอยู่เบื้องต้น (Mockup หรือ Wireframe)
- ให้ผู้ใช้ประเมินและเสนอแนะเพื่อทำการปรับปรุงจนกว่าผู้ใช้อยอมรับ
- นำต้นแบบที่เป็นที่ยอมรับของผู้ใช้มาพัฒนาต่อให้สมบูรณ์จนสามารถใช้งานได้จริง

Prototyping



Mockup



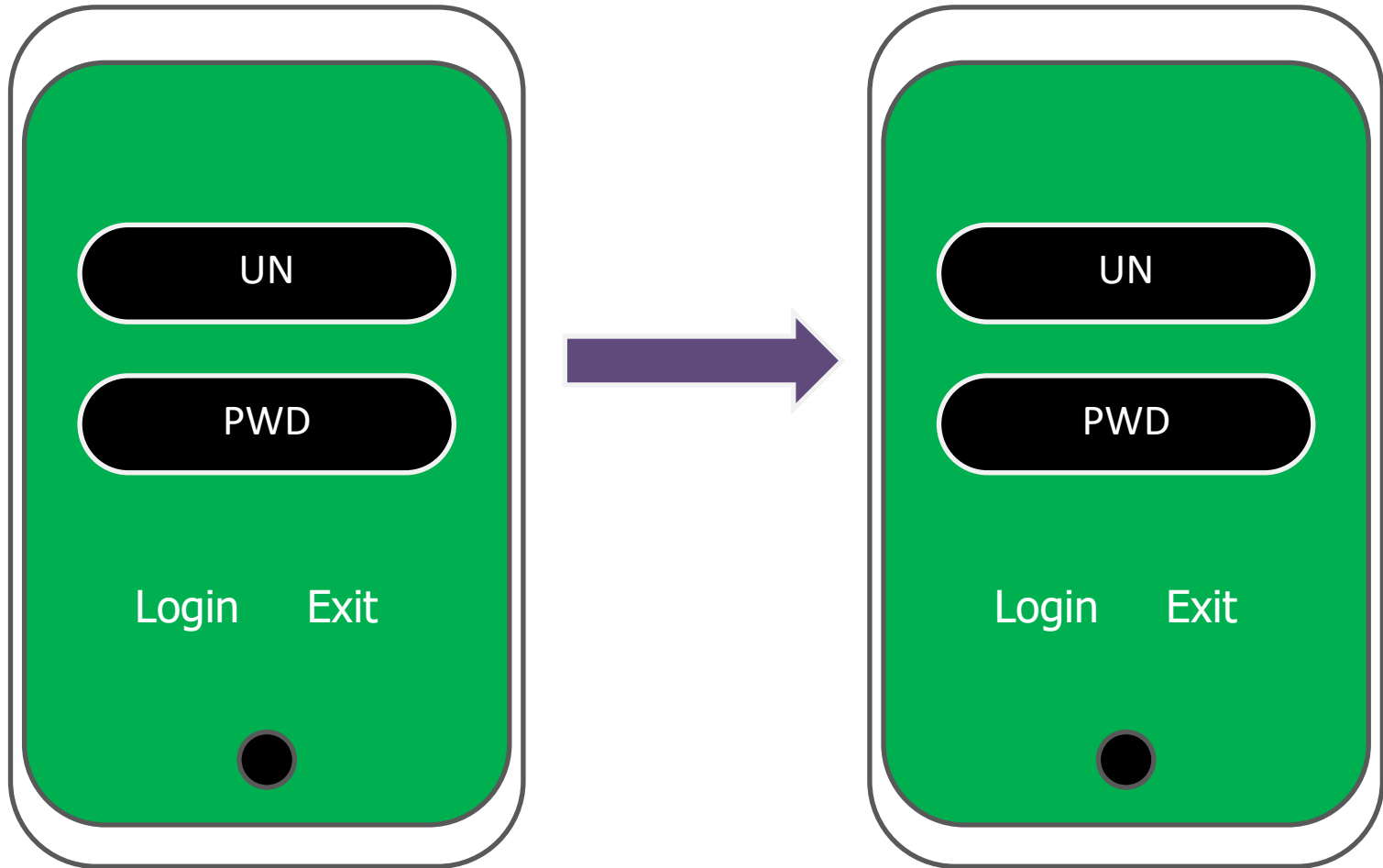
MOTIONGRAPHICPLUS.COM

Mockup คือแบบจำลองของงานที่จะช่วยให้เรา
และลูกค้าสามารถเห็นภาพและเข้าใจงานจริงได้ชัดเจนมากขึ้น
ว่างานจะออกมาแบบไหนและทำให้เข้าใจมากกว่าแค่การมองในงาน
Artwork หรือแค่การอธิบาย

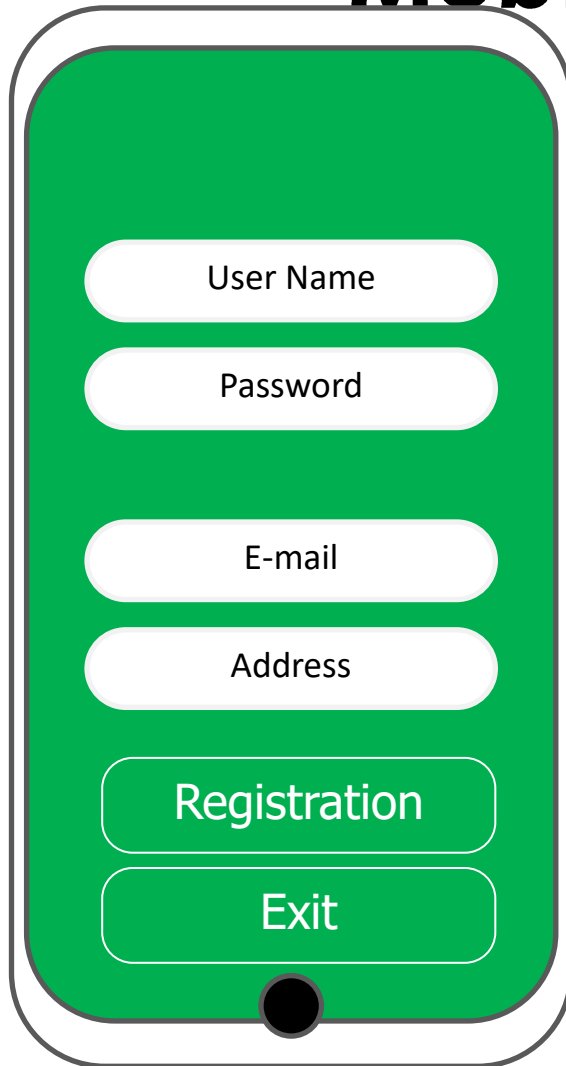


Line ID : @MotionGraphic
Facebook.com/MotionGraphicTH

Mobile Login



Mobile : Register



A mobile registration form with a green background and rounded corners. It contains the following fields and buttons:

- User Name
- Password
- E-mail
- Address
- Registration
- Exit



Mobile : Shopping Cart



วิเคราะห์ความต้องการเชิงวัตถุด้วย แบบจำลองยูสเคส Use Case Model

Requirement Analysis
การวิเคราะห์ความต้องการ

2. System Analysis

- **กระบวนการวิเคราะห์ระบบ (system analysis phase)**
 - มุ่งเน้น “what” ที่ระบบจะต้องมี และต้องทำให้กับผู้ใช้ โดยยังไม่เน้น “how” ว่าจะทำอย่างไร (**ในขั้นตอนนี้เป็นการ User Requirement**)
- **กระบวนการวิเคราะห์ความต้องการของผู้ใช้ระบบ (Requirement analysis phase)**
 - ใช้ในการสร้างแบบจำลองหน้าที่การทำงานของระบบซอฟต์แวร์ จากมุมมองของผู้ใช้ภายนอก หรือ ระบบภายนอก
 - จะได้แบบจำลองของความต้องการของผู้ใช้ระบบ (**Requirement Model**) เป็น Output

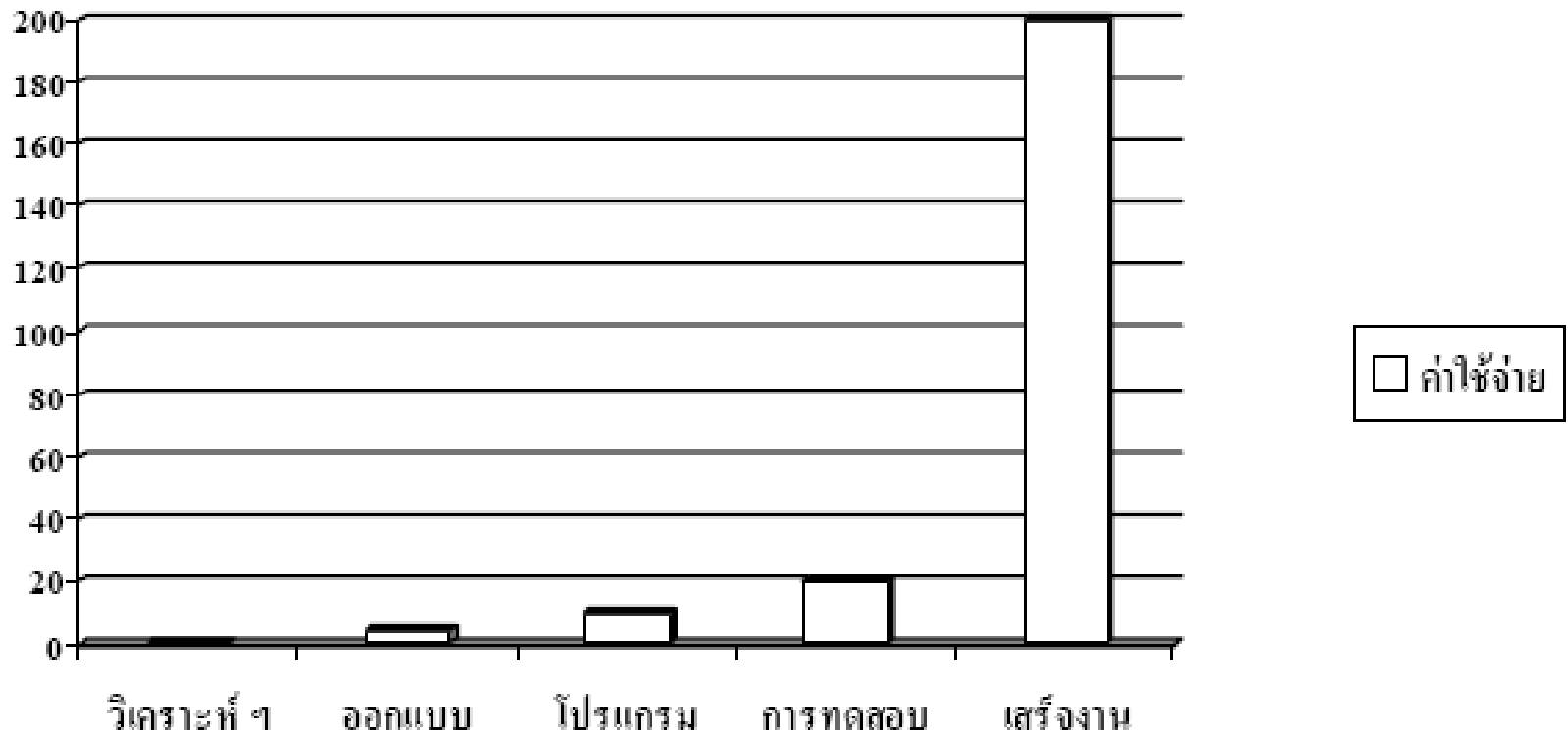
จาก UP ในเฟส ที่ 2 สิ่งจะต้องได้ หรือเสร็จ ก็คือ “เมื่อสิ้นสุด Elaboration ควรเข้าใจ Requirement ส่วนสำคัญและ Use Case หลักได้ค่อนข้างสมบูรณ์ ประมาณ 80% เพื่อใช้กำหนดสถาปัตยกรรมระบบ”

ความสำคัญของการวิเคราะห์ความต้องการ

- สาเหตุของการยกเลิกพัฒนาซอฟต์แวร์
 - ความต้องการไม่สมบูรณ์
 - ไม่มีผู้ใช้ระบบร่วมพัฒนา
 - ขาดทรัพยากร
 - ความคาดหวังในระบบเป็นไปไม่ได้
 - ไม่มีการสนับสนุนจากผู้บริหาร
 - มีการปรับความต้องการขณะพัฒนา
 - ขาดการวางแผน
 - **ไม่มีความต้องการระบบที่กำลังพัฒนา**

ค่าใช้จ่ายในการแก้ไขข้อผิดพลาด

แสดงให้เห็นว่าในการที่จะพัฒนาระบบ จำเป็นจะต้องวิเคราะห์และออกแบบระบบให้ดี มิฉะนั้น ค่าใช้จ่ายจะสูงมากเมื่อดำเนินโครงการไปแล้วเสร็จ



วัตถุประสงค์หลักของการวิเคราะห์ความต้องการ

- กำหนดข้อตกลงกับผู้ว่าจ้าง ผู้ใช้ระบบ ว่าระบบที่จะพัฒนาควรมีความสามารถทำอะไรได้บ้าง (มี function และ none-function อะไรบ้าง)
- ให้ผู้พัฒนาระบบ (System Developers) เข้าใจขีดความสามารถของระบบชัดเจนยิ่งขึ้น
- ปรับขอบเขตของระบบ
- จัดการ กำหนดแผนการด้านเทคนิคต่าง ๆ สำหรับการทำงานตามรอบการพัฒนา
- กำหนดส่วนเชื่อมต่อของผู้ใช้งานระบบงานกับระบบ

วิธีการหาความต้องการที่แท้จริง

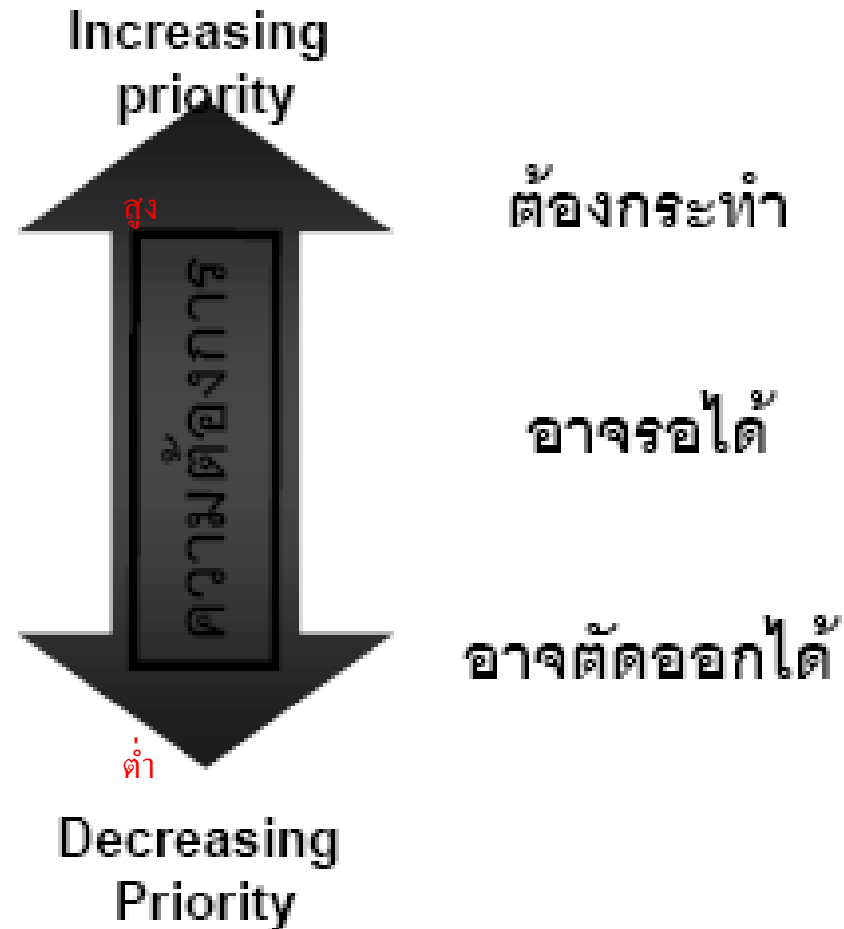
- การสัมภาษณ์ (Interviewing)
- การระดมสมอง (Brainstorming)
- การใช้แบบสอบถาม (Questionnaire)
- การสังเกต (Observation)

ระดับของความต้อการ

ระดับความสำคัญของความต้องการ
เมื่อเก็บรวบรวมความต้องการมาแล้ว
ต้องมาจัดลำดับความสำคัญของ
ความต้องการ

โดยจัดเป็น

- ความต้องการแบบเร่งด่วน
- ความต้องการที่รอ
- ความต้องการที่สามารถตัดออกได้



เอกสารการวิเคราะห์ความต้องการ

สิ่งที่ต้องส่ง ในเล่มรายงานทั้งหมด ของแต่ละกลุ่ม
Requirement Specification

1. บทนำ

1.1 วัตถุประสงค์

1.2 ขอบเขต

1.3 คำจำกัดความ คำเหมือน คำย่อ (Req ID = Requirement Identification)

1.4 เอกสารอ้างอิง

2. การบริหารความต้องการ

2.1 การบริหารโครงการ หน้าทีภายในทีมงาน

2.2 โครงสร้างพื้นฐานของระบบ และเครื่องมือที่ใช้

เอกสารการวิเคราะห์ความต้องการ

3. รายการบริหารความต้องการ

3.1 ระบุความต้องการ ขั้นตอนการปฏิบัติงาน

3.2 ระบุกฎข้อบังคับของความต้องการ(ถ้ามี)

3.3 กำหนดคุณสมบัติ (Attributes) ในข้อ 3.1

3.4 รูปแบบเนื้อหาของรายงานและวิธีการวัด

3.5 การบริหารความเปลี่ยนแปลง

 ขั้นตอนการขอเปลี่ยนแปลงระบบ

 โครงสร้างของคณะกรรมการพิจารณา

 จุดตรวจสอบในแต่ละเฟส

3.6 ขั้นตอนการทำงานและกิจกรรม

4. แผนการดำเนินงาน

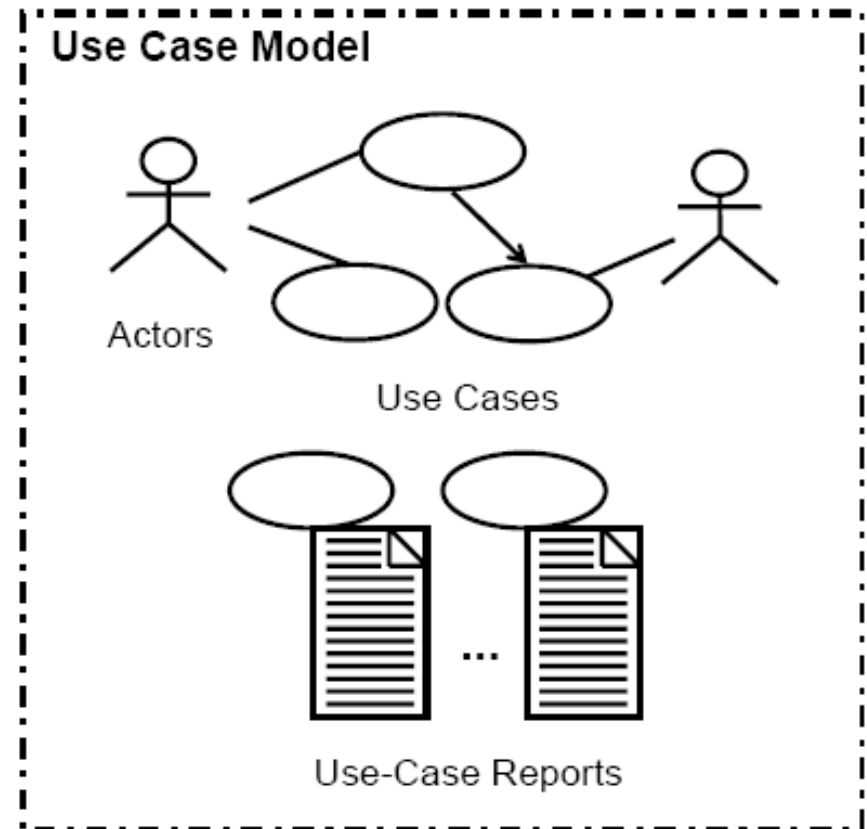
5. เครื่องมือในการพัฒนาระบบที่ใช้และการฝึกอบรม

การพัฒนาระบบเล็ก ๆ อาจเขียนเอกสารดังนี้

- ขอบเขตของระบบที่จะพัฒนา(Problem Domain)
- Glossary พจนานุกรม
- เอกสารรายละเอียดของระบบ(Requirements Specification)
- Use-Case Diagram

Use Case Model

- Introduction
- Survey Description
- Use Case Packages
- Use Case
- Actors
- Relationships
- Diagrams
- Use Case View



Use case Model

Use case diagram แผนภาพยูสเคส

Use case description คำอธิบายยูสเคส

Use case diagram แผนภาพยูสเคส

ในขั้นตอนการวิเคราะห์ระบบเชิงวัตถุ OOA

- เมื่อเก็บรวบรวมความต้องการ Information
 - จัดโครงสร้างของความต้องการ ตามมุมมอง view point เช่น
 - มุมมองผู้ใช้
 - มุมมองผู้ดูแลระบบ
 - มุมมองบุคคลทั่วไป
 - จัดลำดับความสำคัญของแต่ละความต้องการ
 - จัดทำเอกสาร SRS
 - นำมาเขียน เป็น ยูสเคส

Use case diagram แผนภาพยูสเคส

- เป็นแผนภาพของยูเอ็มแอล สำหรับการวิเคราะห์ระบบเชิงวัตถุ
- เป็นแบบจำลองที่แสดงให้เห็นถึงฟังก์ชันของระบบ ว่าระบบมีฟังก์ชันที่ตอบสนองกับแอ็กเตอร์อย่างไร
- ส่วนประกอบของยูสเคสไดอะแกรม หรือแผนภาพยูสเคส
 - Actor
 - Use case
 - System boundary
 - Flow / Communication Flow
 - Relationship


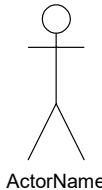

Use case diagram แผนภาพยูสเคส

- นำเสนอฟังก์ชันหรือ Use Case และการปฏิสัมพันธ์โต้ตอบกันระหว่างระบบ และ ผู้ใช้ภายนอก (someone / something อาจเป็นคน หรือระบบก็ได้)
- ส่วนประกอบของ use case diagram ด้วย
 - **Use Case** - ฟังก์ชัน/ความสามารถ/หน้าที่ของระบบ
 - **Actor** - ผู้ที่มีบทบาท/ ผู้กระทำ/ผู้ใช้งาน Use Case นั้นๆ
 - **Relationship** - เส้นแสดงความสัมพันธ์ระหว่าง Use Case กับ Actor
 - **System / System Boundary** - ระบบที่กำลังพัฒนา

Functional Requirement

Use case diagram แผนภาพยูสเคส

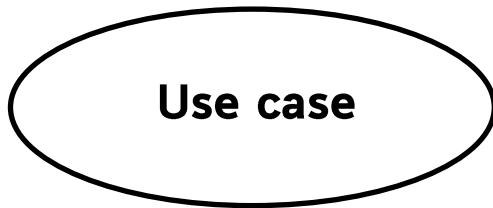
Use Case Modeling : Core Elements

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 diagram แผนภาพยูสเคส

Use case แทน Functional requirement

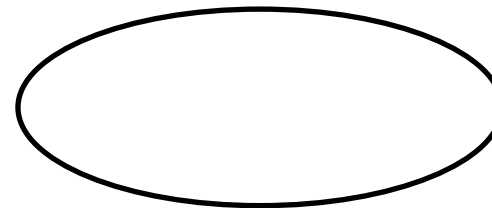
สัญลักษณ์ของยูสเคส



ตัวอย่าง

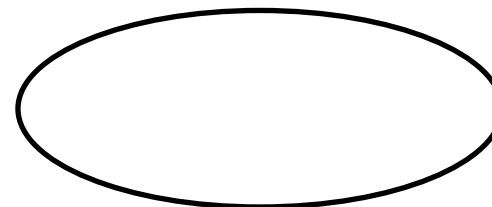


สัญลักษณ์ของยูสเคส



Use case

ตัวอย่าง



เข้าสู่ระบบ

*** เนื่องจากยูสเคสเป็นฟังก์ชัน ควรเขียนเป็นคำกริยา ไม่ใช่คำนาม

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบร้านค้าแคร์:

Req. 01: ระบบการจัดการลูกค้า

- ระบบจะต้องสามารถลงทะเบียนลูกค้าใหม่และบริหารจัดการข้อมูลลูกค้าเช่น ชื่อ, ที่อยู่, และข้อมูลติดต่อ.

Req. 02: การนัดหมายและตารางการบริการ

- ระบบจะต้องสนับสนุนการจัดการการนัดหมายลูกค้าและสร้างตารางการบริการสำหรับร้านค้าแคร์.

Req. 03: การบันทึกข้อมูลรถยนต์

- ระบบจะต้องสามารถบันทึกข้อมูลรถยนต์ของลูกค้า เช่น ยี่ห้อ, รุ่น, และหมายเลขทะเบียน.

Req. 04: การตรวจสอบสถานะบริการ

- ลูกค้าจะต้องสามารถตรวจสอบสถานะของบริการรถยนต์ที่อยู่ในกระบวนการ.

Req. 05: การคำนวณค่าบริการ

- ระบบจะต้องสามารถคำนวณค่าบริการของร้านค้าแคร์ตามบริการที่เลือกและรายละเอียดของรถยนต์.

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบร้านค้าแคร์:

Req. 01: ระบบการจัดการลูกค้า

- ระบบจะต้องสามารถลงทะเบียนลูกค้าใหม่และบริหารจัดการข้อมูลลูกค้าเช่น ชื่อ, ที่อยู่, และข้อมูลติดต่อ.

Req. 02: การนัดหมายและตารางการบริการ

- ระบบจะต้องสนับสนุนการจัดการการนัดหมายลูกค้าและสร้างตารางการบริการสำหรับร้านค้าแคร์.

Req. 03: การบันทึกข้อมูลรถยนต์

- ระบบจะต้องสามารถบันทึกข้อมูลรถยนต์ของลูกค้า เช่น ยี่ห้อ, รุ่น, และหมายเลขทะเบียน.

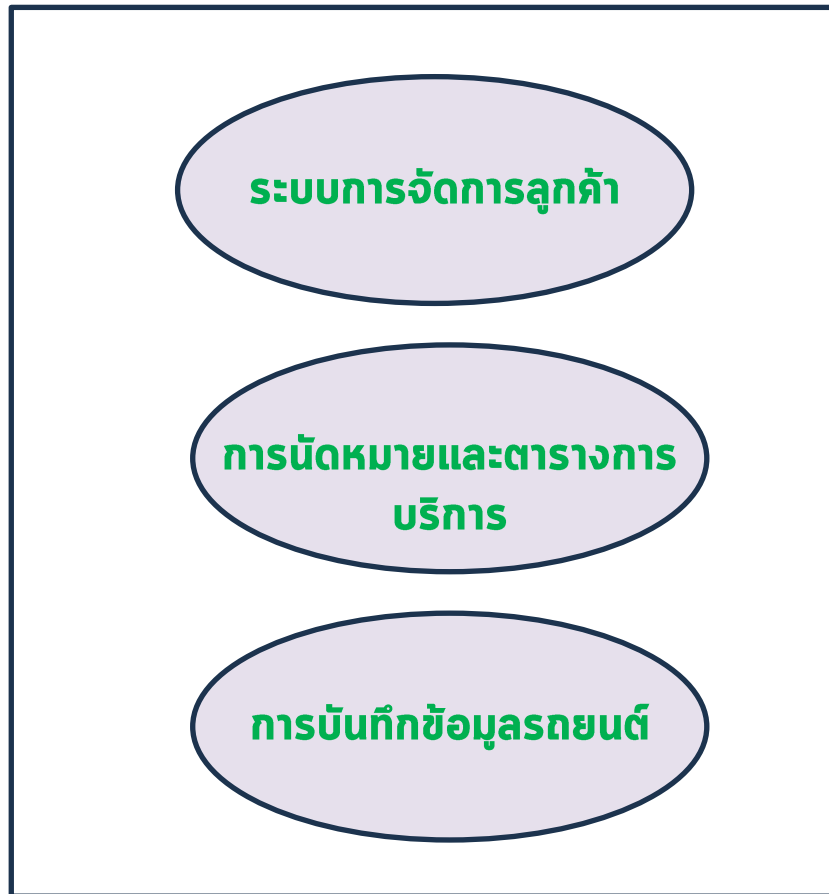
Req. 04: การตรวจสอบสถานะบริการ

- ลูกค้าจะต้องสามารถตรวจสอบสถานะของบริการรถยนต์ที่อยู่ในกระบวนการ.

Req. 05: การคำนวณค่าบริการ

- ระบบจะต้องสามารถคำนวณค่าบริการของร้านค้าแคร์ตามบริการที่เลือกและรายละเอียดของรถยนต์.

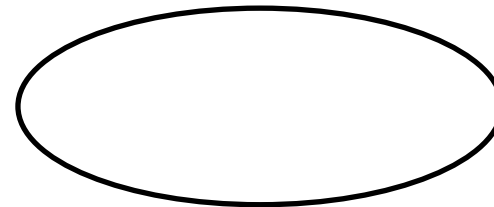
Use case diagram แผนภาพยูสเดส



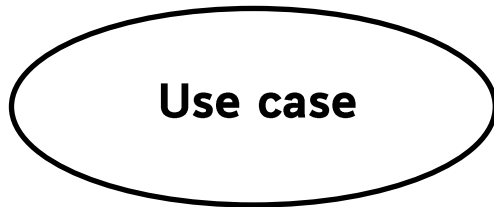
Use case diagram แผนภาพยูสเคส

Use case

สัญลักษณ์ของยูสเคส



สัญลักษณ์ของยูสเคส

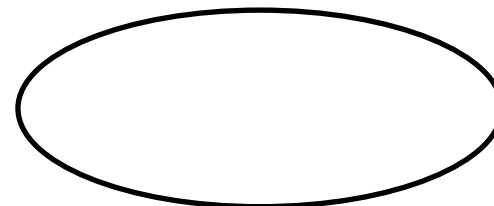


Use case

ตัวอย่าง



ตัวอย่าง



ระบบการจัดการลูกค้า

*** เนื่องจากยูสเคสเป็นฟังก์ชัน ควรเขียนเป็นคำกริยา ไม่ใช่คำนาม

ตัวอย่างข้อกำหนดความต้องการสำหรับระบบการจองห้องพักโรงแรม (ต่อ)

Req. 06: การจัดการการชำระเงิน

- ระบบจะต้องรองรับการชำระเงินออนไลน์หรือหน้าร้านค้าเร่

Req. 07: ระบบความปลอดภัย

functional requirements

- ระบบจะต้องมีมาตรการรักษาความปลอดภัยของข้อมูลลูกค้าและการทำธุรกรรม

Req. 08: ระบบการสื่อสาร

functional requirements

- ระบบจะต้องส่งอีเมลหรือข้อความแจ้งเตือนถึงลูกค้าเกี่ยวกับสถานะการบริการและการนัดหมาย

Req. 09: การสร้างรายงาน

functional requirements

- ระบบจะต้องสร้างรายงานเกี่ยวกับประสิทธิภาพการบริการ, รายได้, และความเคลื่อนไหวของร้านค้าเร่

Req. 10: การสนับสนุนผู้ดูแลระบบ

functional requirements

- ระบบจะต้องมีบทบาทผู้ดูแลระบบที่สามารถเพิ่มหรือแก้ไขข้อมูลลูกค้าและบริการ.

ข้อกำหนดเหล่านี้เป็นตัวอย่างเพียงแต่บางส่วนของความต้องการที่อาจปรากฏในระบบร้านค้าเร่ ความต้องการที่แท้จริงจะต้องเน้นไปยังการทำงานและประสิทธิภาพของระบบที่ต้องมีให้บรรลุได้ในระดับสูงสุดและมีความปลอดภัยในการใช้งาน.

Use case diagram แผนภาพยูสเคส

CASE Tool

- Dia
- Star UML
- Rational rose
- **Draw.IO** เป็นเว็บไซต์ ออนไลน์
- **Visio**

Untitled Diagram.drawio - drawio X

https://www.draw.io

Untitled Diagram.drawio

File Edit View Arrange Extras Help

100%

Search Shapes

Scratchpad

Drag elements here

General

Diagram

View

- Grid 10 pt
- Page View
- Background Image
- Shadow

Options

- Connection Arrows
- Connection Points
- Guides

Paper Size

A4 (210 mm x 297 mm)

Portrait Landscape

Edit Data

Clear Default Style

Page-1

Download draw.io Desktop...

Windows Taskbar: OOAD, Calculator, VLC, File Explorer, npx, Untitled D..., Chrome, Lec06_Re..., EN, 10:19 21/9/2562

Library of shapes and symbols for diagramming, including rectangles, circles, and various connectors.

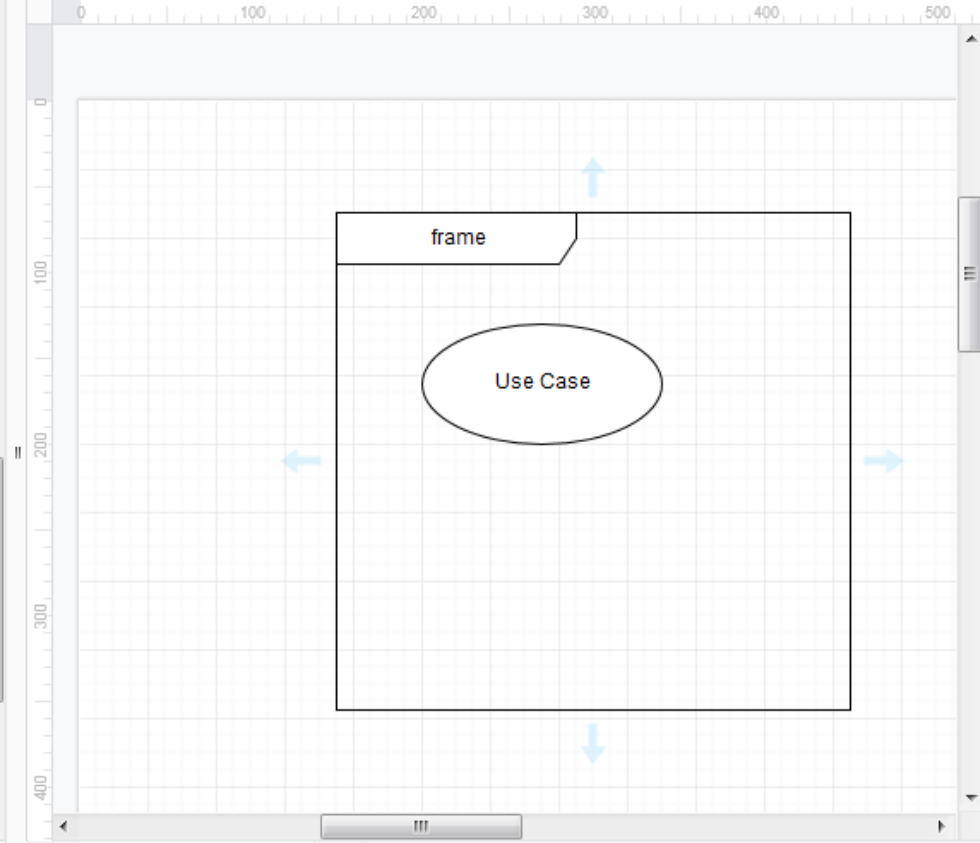


Diagram settings panel with options for View (Grid, Page View, Background, Shadow), Options (Connection Arrows, Connection Points, Guides), and Paper Size (A4, Portrait/Landscape). Includes buttons for Edit Data and Clear Default Style.

Use case diagram แผนภาพยูสเคส

ตัวอย่าง use case

เพิ่มลูกค้า

สมัครสมาชิก

ถอนเงิน

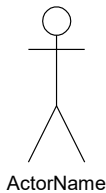
ฝากเงิน

***** เนื่องจากยูสเคสเป็นฟังก์ชัน ควรเขียนเป็นคำกริยา ไม่ใช่คำนาม
เช่น การฝากเงิน ไม่ควรใส่ การ
การถอนเงิน
การลงทะเบียน**

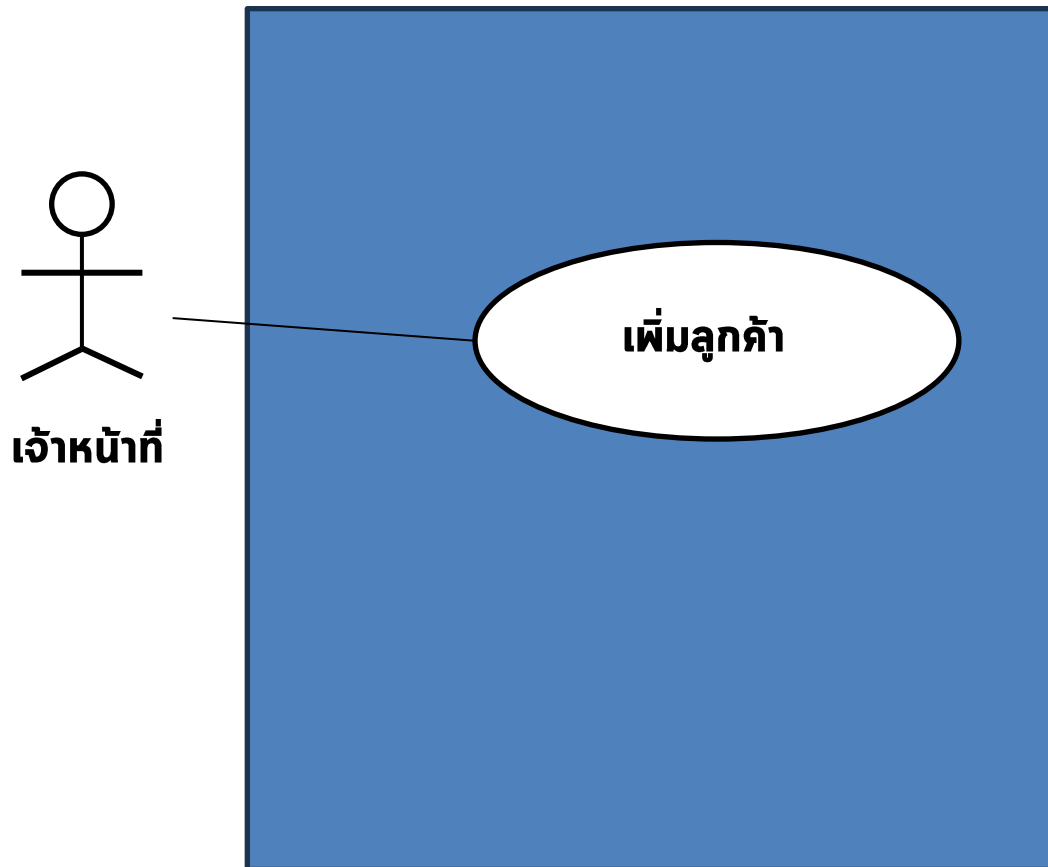
Use case diagram แผนภาพยูสเคส

Actor

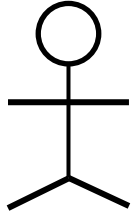
- ผู้ที่มีบทบาท/ ผู้กระทำ / ผู้ใช้งาน Use Case นั้นๆ
- เป็น someone or something เป็นคน ระบบ ซอฟต์แวร์ หรือ ฮาร์ดแวร์ใด ๆ ก็ได้ ที่มีความต้องการ หรือมีปฏิสัมพันธ์ หรือ จะให้ระบบทำงาน หรือใช้งานระบบ หรือรับผลลัพธ์จากการทำงาน บางอย่างระบบ หรือ ให้ข้อมูลกับระบบ



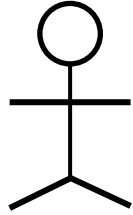
Use case diagram แผนภาพยูสเคส



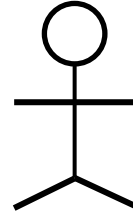
สัญลักษณ์ Actor



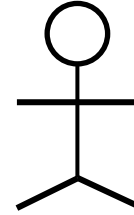
Actor



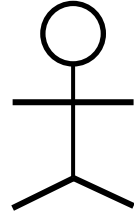
เจ้าหน้าที่



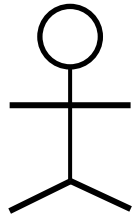
ลูกค้า



ผู้ดูแลระบบ

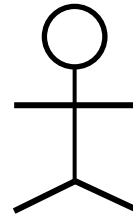


พนักงาน

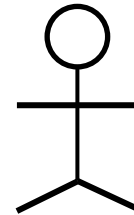


**Someone หรือ
Somethings**

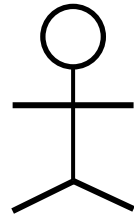
สรรพากร



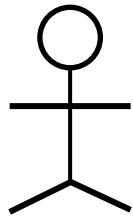
ATM



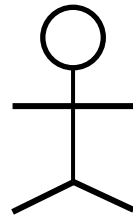
Bank Consortium



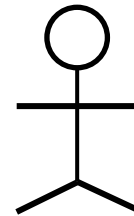
Stock Exchange



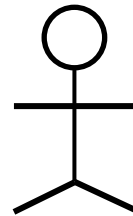
คน



sensor






Print server



Temp Sensor

Use case diagram แผนภาพยูสเคส

ความสัมพันธ์ Relationship

- ระหว่าง Actor กับ Actor 
- ระหว่าง Actor กับ use case 
- ระหว่าง Use case กับ Use case 

`<<include>>`

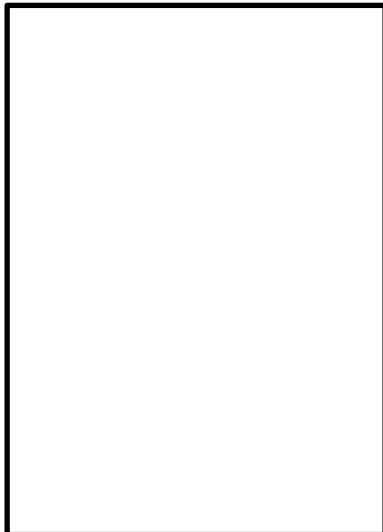

`<<extend>>`


Use case diagram แผนภาพยูสเคส

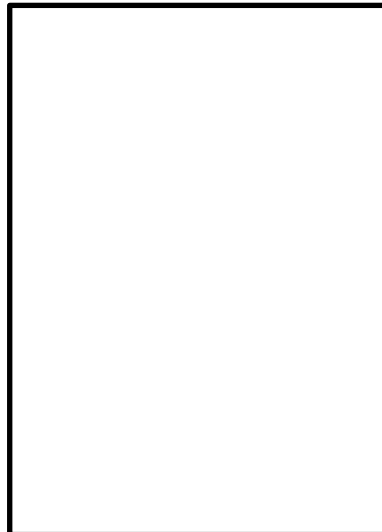
System boundary

- แสดงถึงระบบที่กำลังพัฒนา ซึ่งอาจเป็นส่วนหนึ่งส่วนใดของระบบ หรือระบบย่อย (Sub system) หรือทั้งระบบ (Whole system)
- แสดงขอบเขตของระบบ

ชื่อระบบ



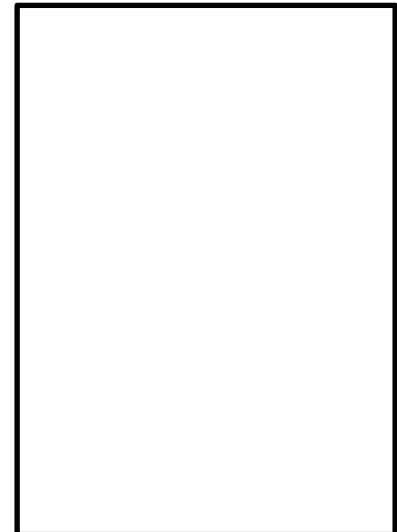
ระบบสั่งซื้อสินค้า



ระบบจองห้องพักโรงแรม



ระบบการลงทะเบียน

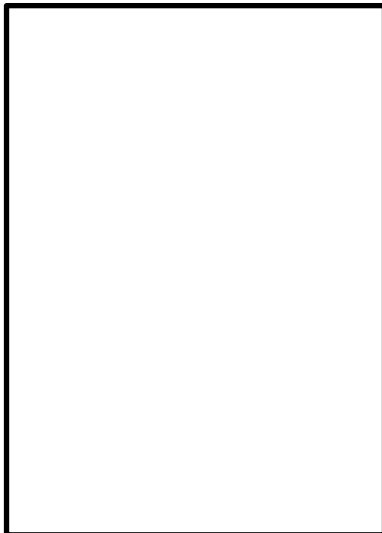


Use case diagram แผนภาพยูสเคส

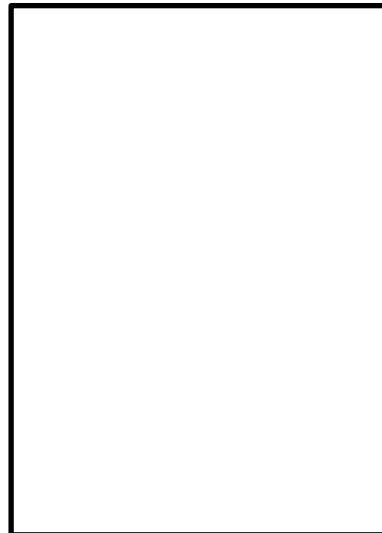
System boundary

- แสดงถึงระบบที่กำลังพัฒนา ซึ่งอาจเป็นส่วนหนึ่งส่วนใดของระบบ หรือระบบย่อย (Sub system)หรือทั้งระบบ (Whole system)
- แสดงขอบเขตของระบบ

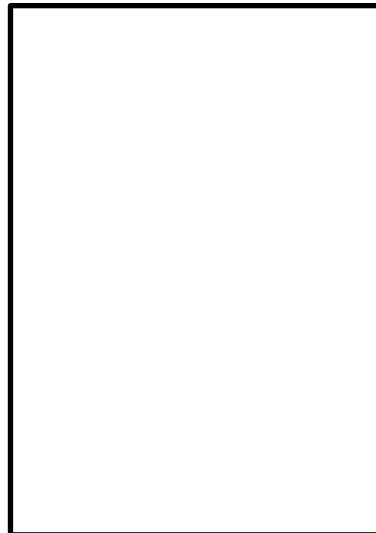
ระบบยืม-คืนหนังสือ



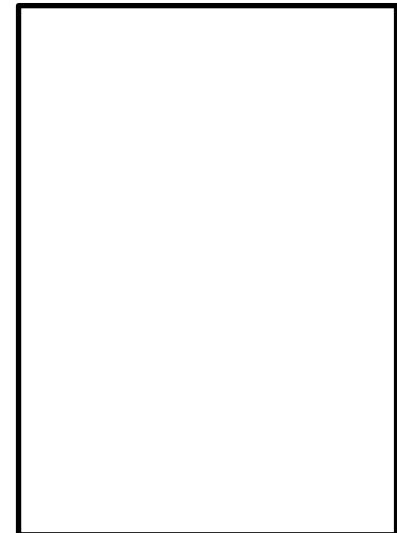
ระบบร้านค้าบัตรเครดิต



ระบบจองสนามกีฬา

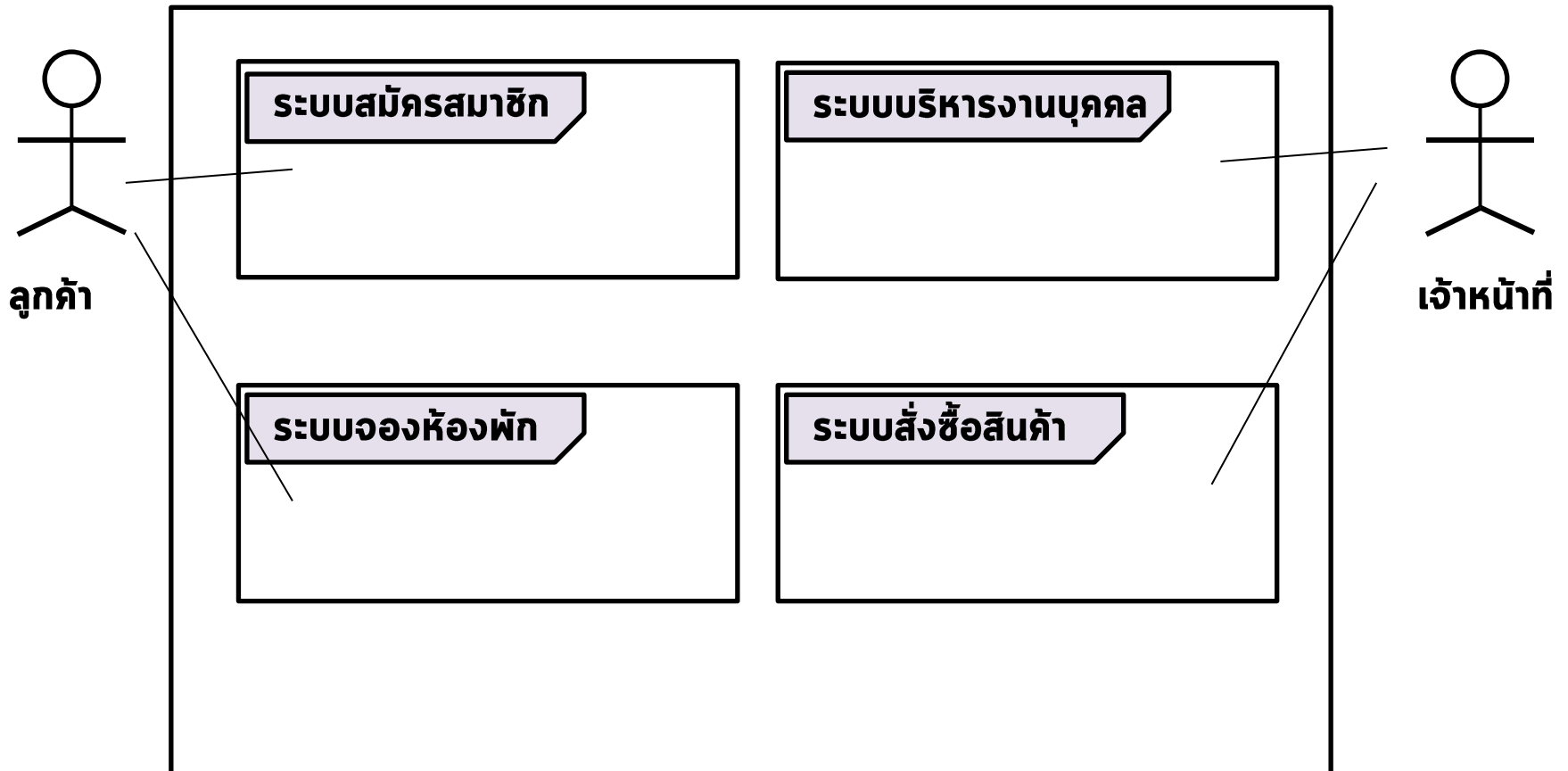


ระบบการจองตั๋วภาพยนตร์



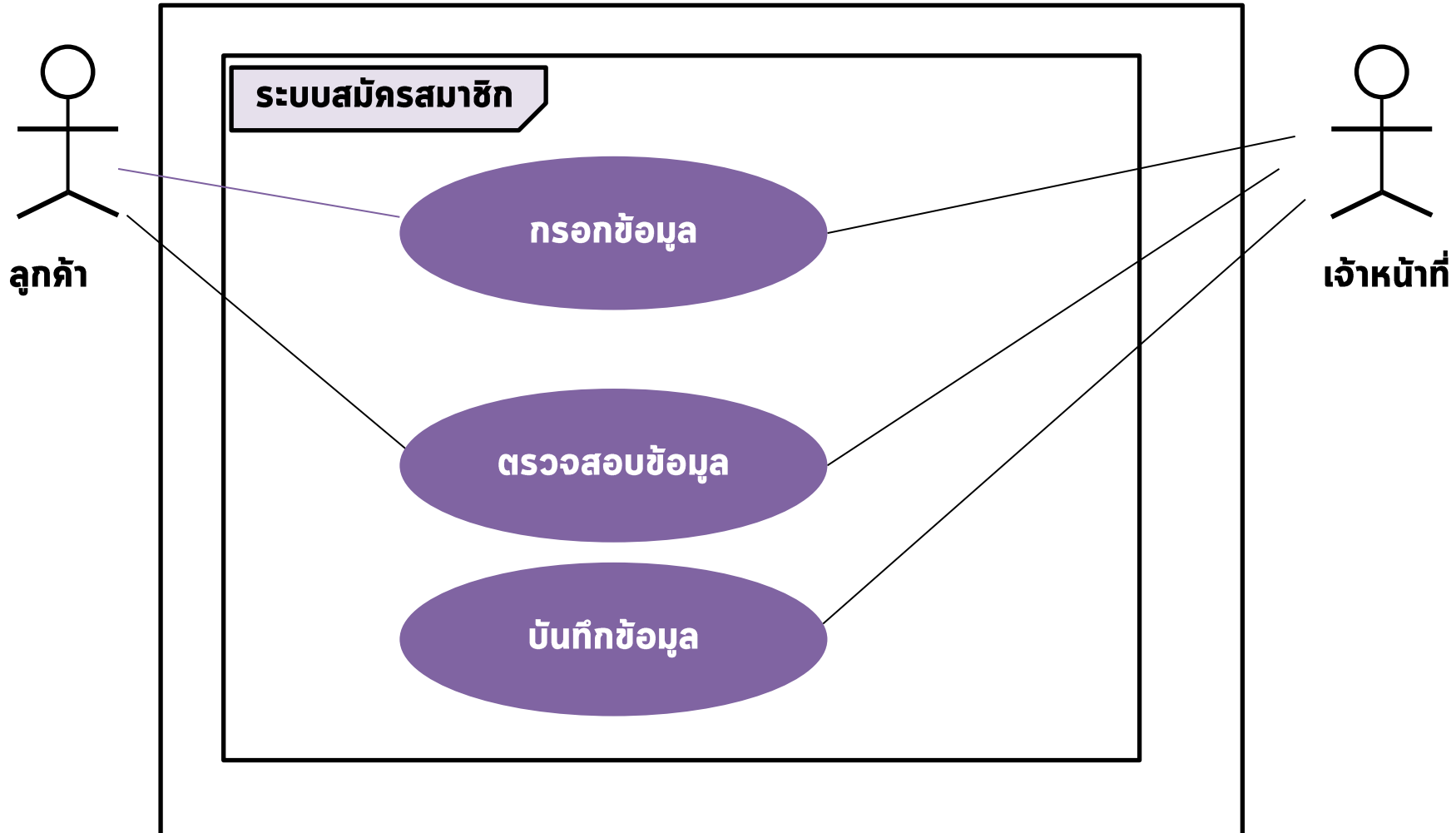
Whole system /Sub system

ระบบบริหารจัดการโรงแรม



Whole system /Sub system

ระบบบริหารจัดการโรงแรม



ความสัมพันธ์ Relationship

- ระหว่าง Actor กับ Actor

association relationship



Generalization relationship

- ระหว่าง Actor กับ use case



- ระหว่าง Use case กับ Use case



Generalization relationship

`<<include>>`



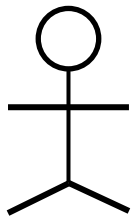
`<<extend>>`



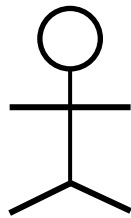
ความสัมพันธ์ระหว่าง Actor กับ Actor

ความสัมพันธ์แบบทั่วไป Generalization

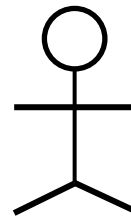
ความสัมพันธ์แบบมีความเกี่ยวข้องกัน(แอสโซซิเอชัน) Association



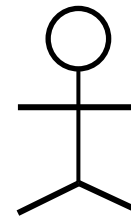
Actor



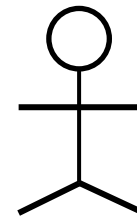
เจ้าหน้าที่



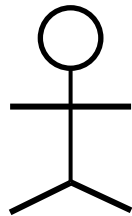
ลูกค้า



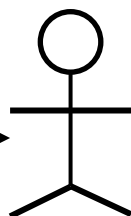
ผู้ดูแลระบบ



พนักงาน



ลูกค้า

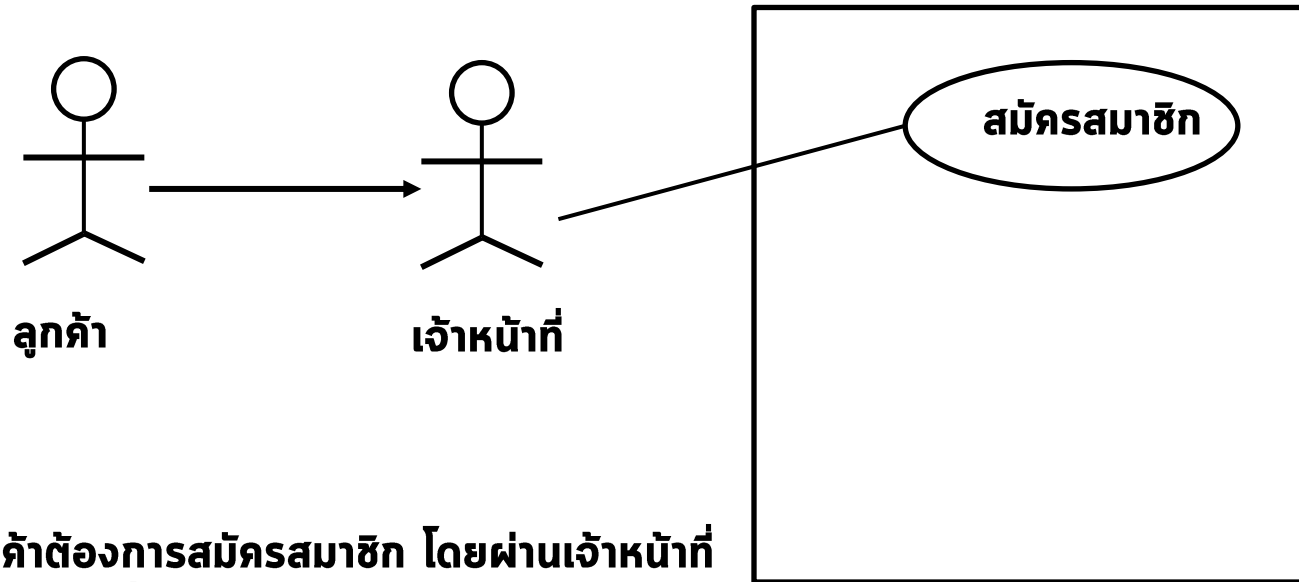


เจ้าหน้าที่

association relationship

ลูกค้าต้องการสมัครสมาชิก โดยผ่านเจ้าหน้าที่

ความสัมพันธ์แบบ Association

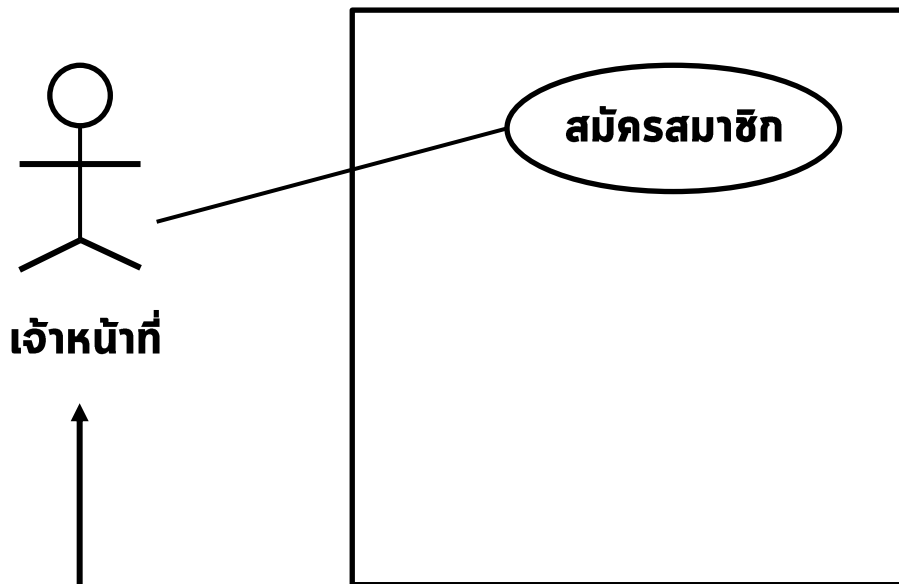


ลูกค้าต้องการสมัครสมาชิก โดยผ่านเจ้าหน้าที่
ระบบแบบนี้จะเป็นระบบแบบ offline

ตัวอย่าง
สมัครสมาชิก MK

สมัครสมาชิก CJ Mall
สมัครสมาชิก Macro

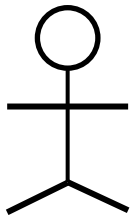
ความสัมพันธ์แบบ Association



เจ้าหน้าที่



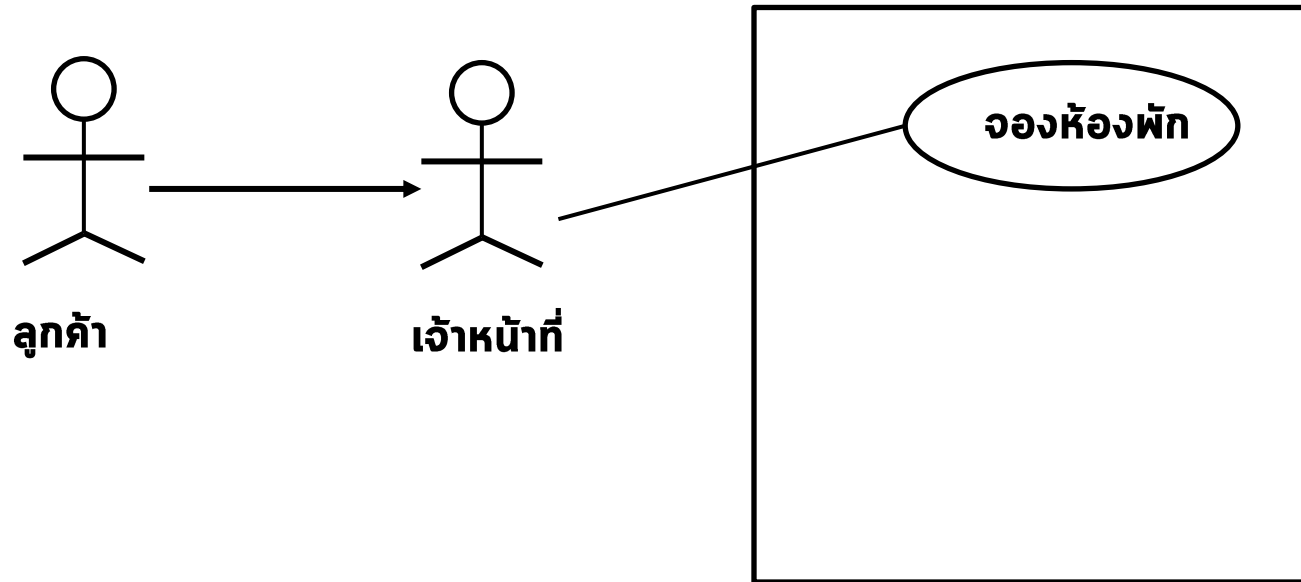
ลูกค้าไม่สามารถกระทำเองได้โดยตรง จะต้องให้เจ้าหน้าที่ทำการสมัครให้



ลูกค้า

ลูกค้าต้องการสมัครสมาชิก โดยผ่านเจ้าหน้าที่
ระบบแบบนี้จะเป็นระบบแบบ offline

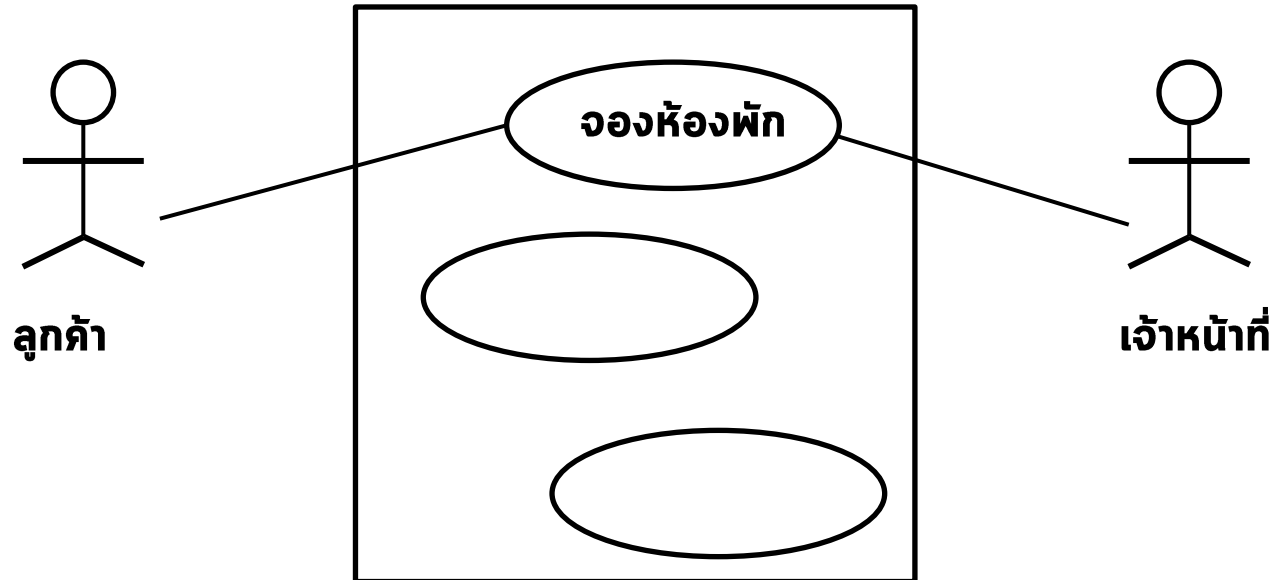
จองห้องพักโรงแรม



ลูกค้าต้องการจองห้องพักโรงแรม โดยผ่านเจ้าหน้าที่
ระบบแบบนี้จะเป็นระบบแบบ **offline**

จองห้องพักโรงแรม

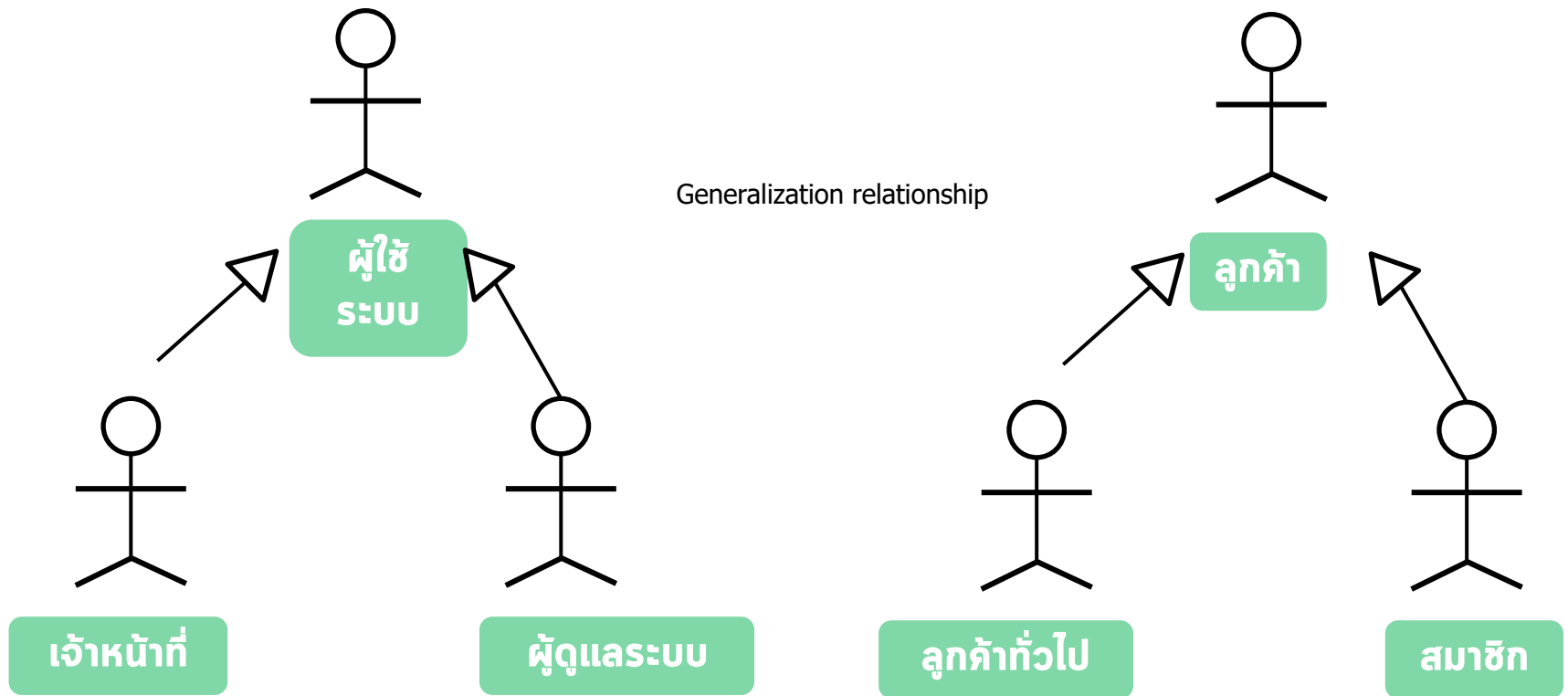
ลูกค้าต้องการจองห้องพักโรงแรม โดยไม่ผ่านเจ้าหน้าที่หรือผ่านเว็บไซต์
ระบบแบบนี้จะเป็นระบบ แบบ online



จัดความต้องการตาม View point of user

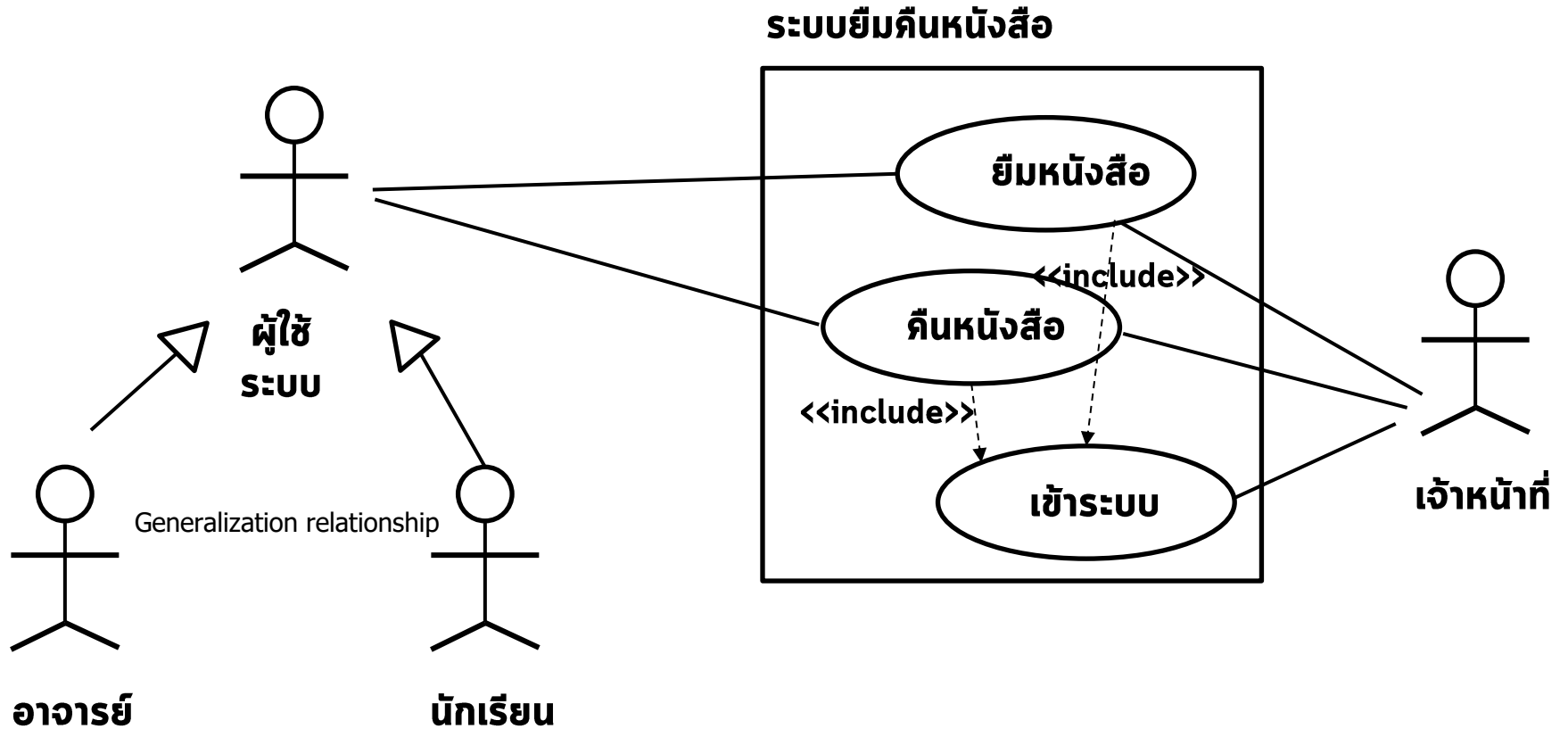
ความสัมพันธ์ระหว่าง Actor กับ Actor

- ความสัมพันธ์แบบ ทั่วไป Generalization

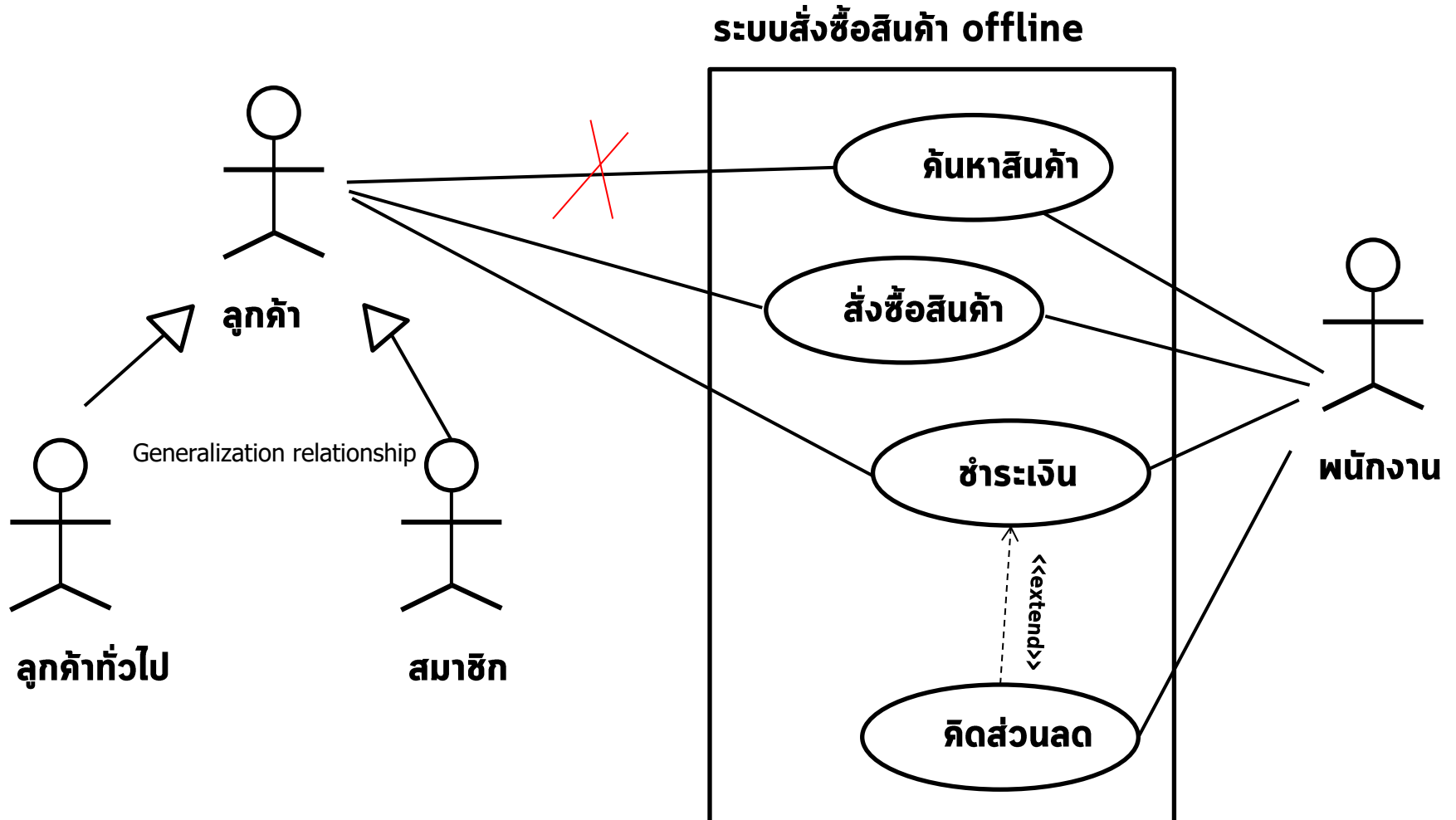


ความสัมพันธ์แบบ ทั่วไป Generalization มักใช้ในการแบ่งประเภทของแอกเตอร์

ตัวอย่างระบบยืม-คืนหนังสือ

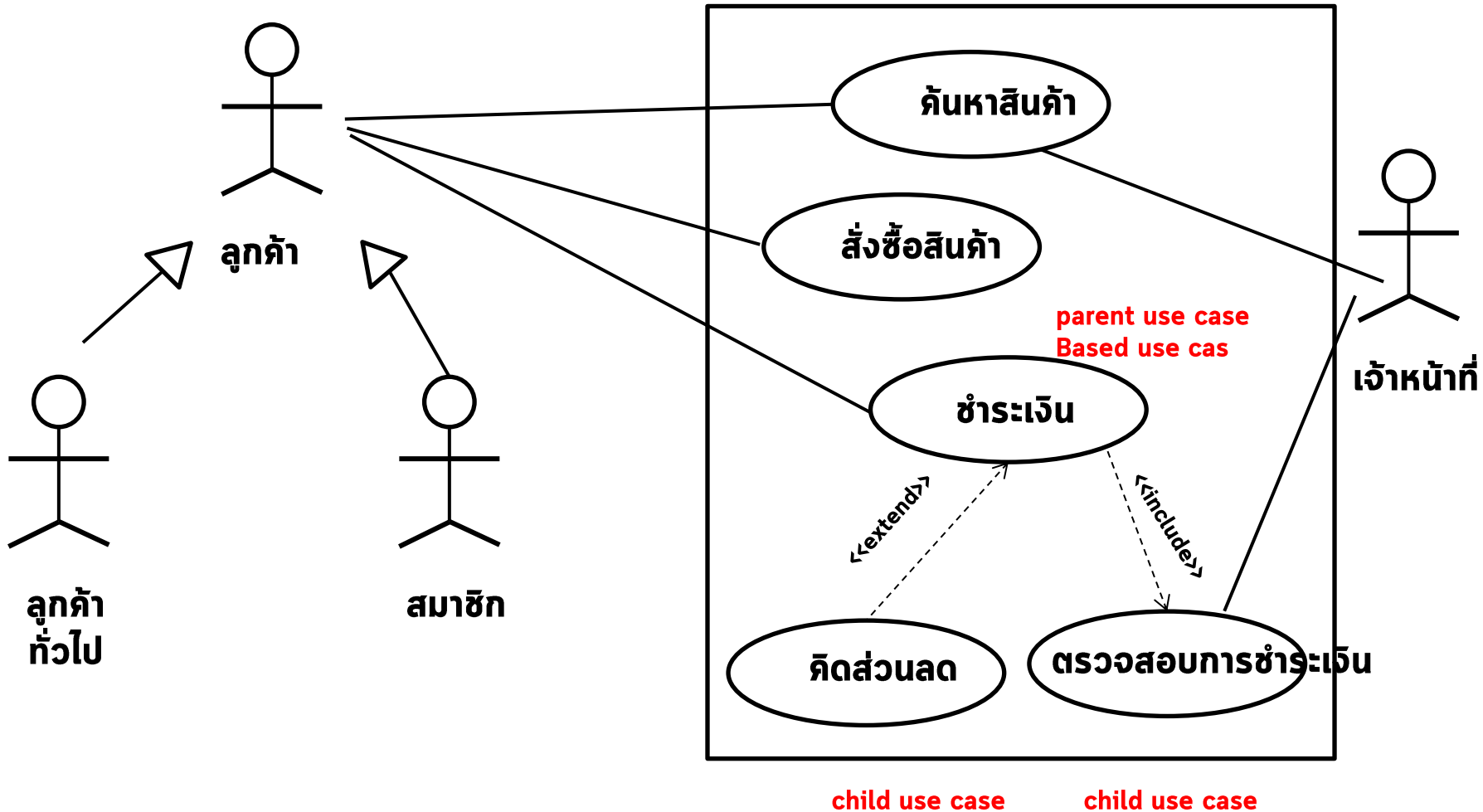


ตัวอย่าง การซื้อ-ขายสินค้า



ตัวอย่าง การซื้อ-ขายสินค้า

ระบบสั่งซื้อสินค้า Online



Use case diagram แผนภาพยูสเคส

ความสัมพันธ์ระหว่าง use case กับ use case

- แบบ include และ extend
- แบบทั่วไป generalize

Use case diagram แผนภาพยูสเคส

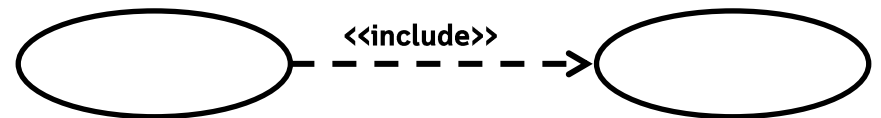
Parent use case
Based use case

child use case

ความสัมพันธ์ระหว่าง use case กับ use case

- **แบบ include**

- **<<include>>** **<<uses>>**

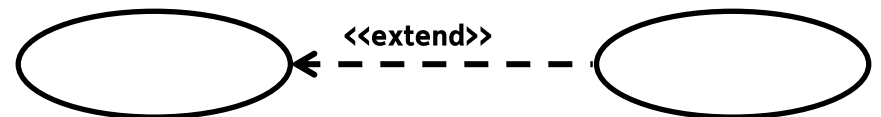


Use case ก้าน

Use case ย่อย

- **แบบ extend**

- **<<extend>>**



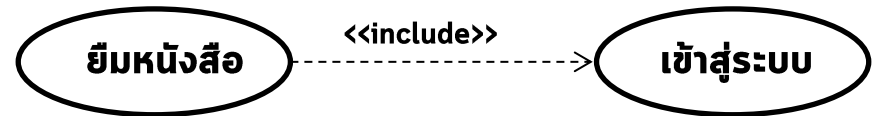
Use case ก้าน

Use case ย่อย

ตัวอย่างความสัมพันธ์ระหว่าง use case กับ use case

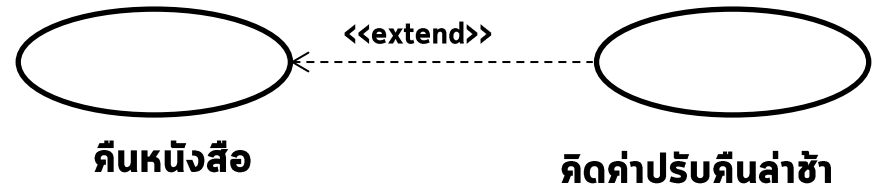
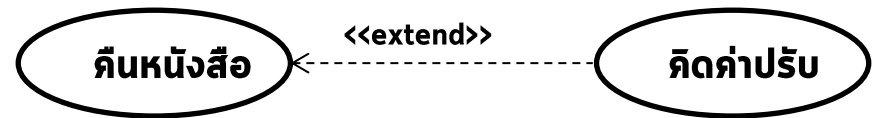
- **แบบ include**

- **<<include>>** **<<uses>>**

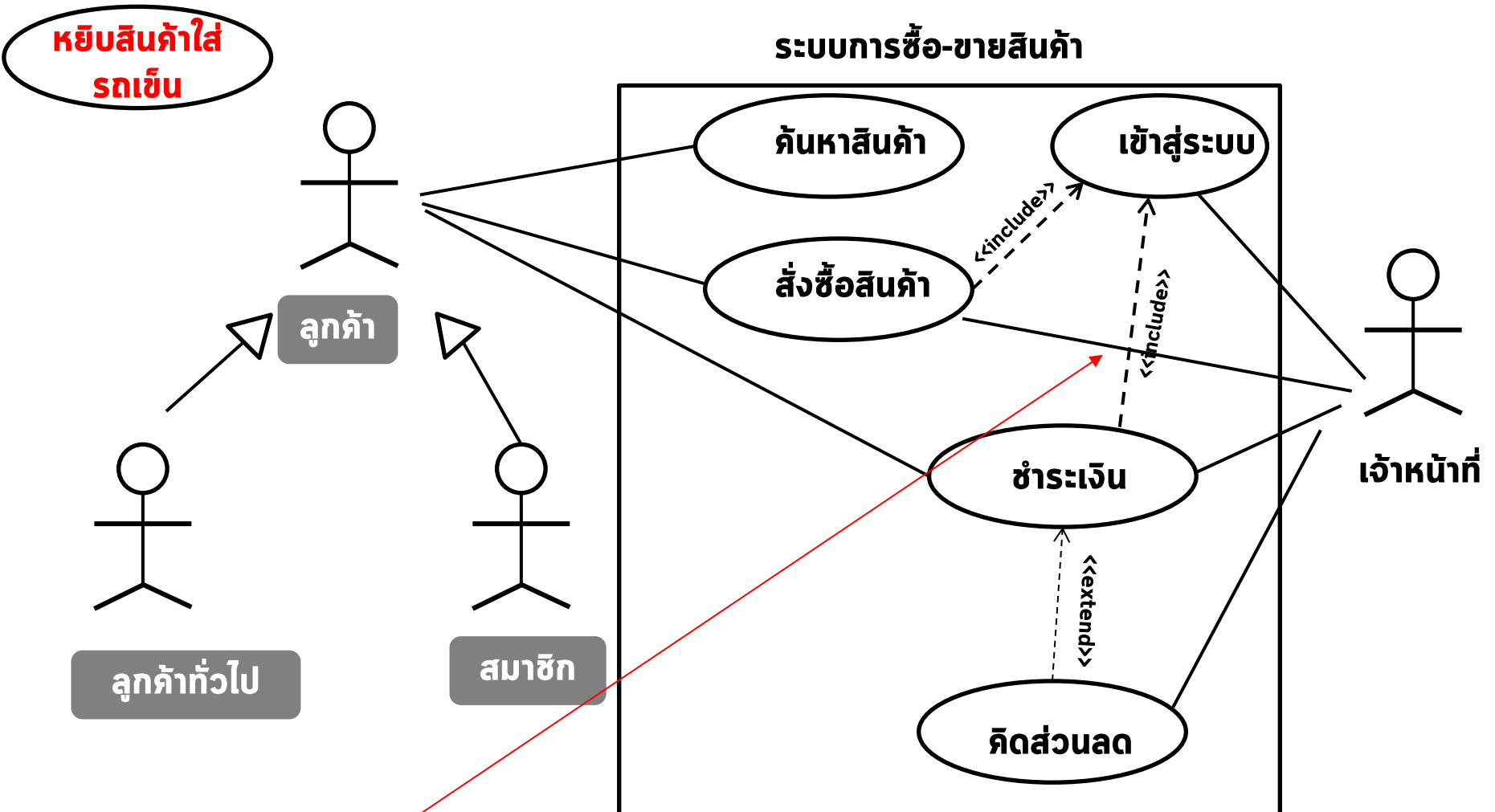


- **แบบ extend**

- **<<extend>>**



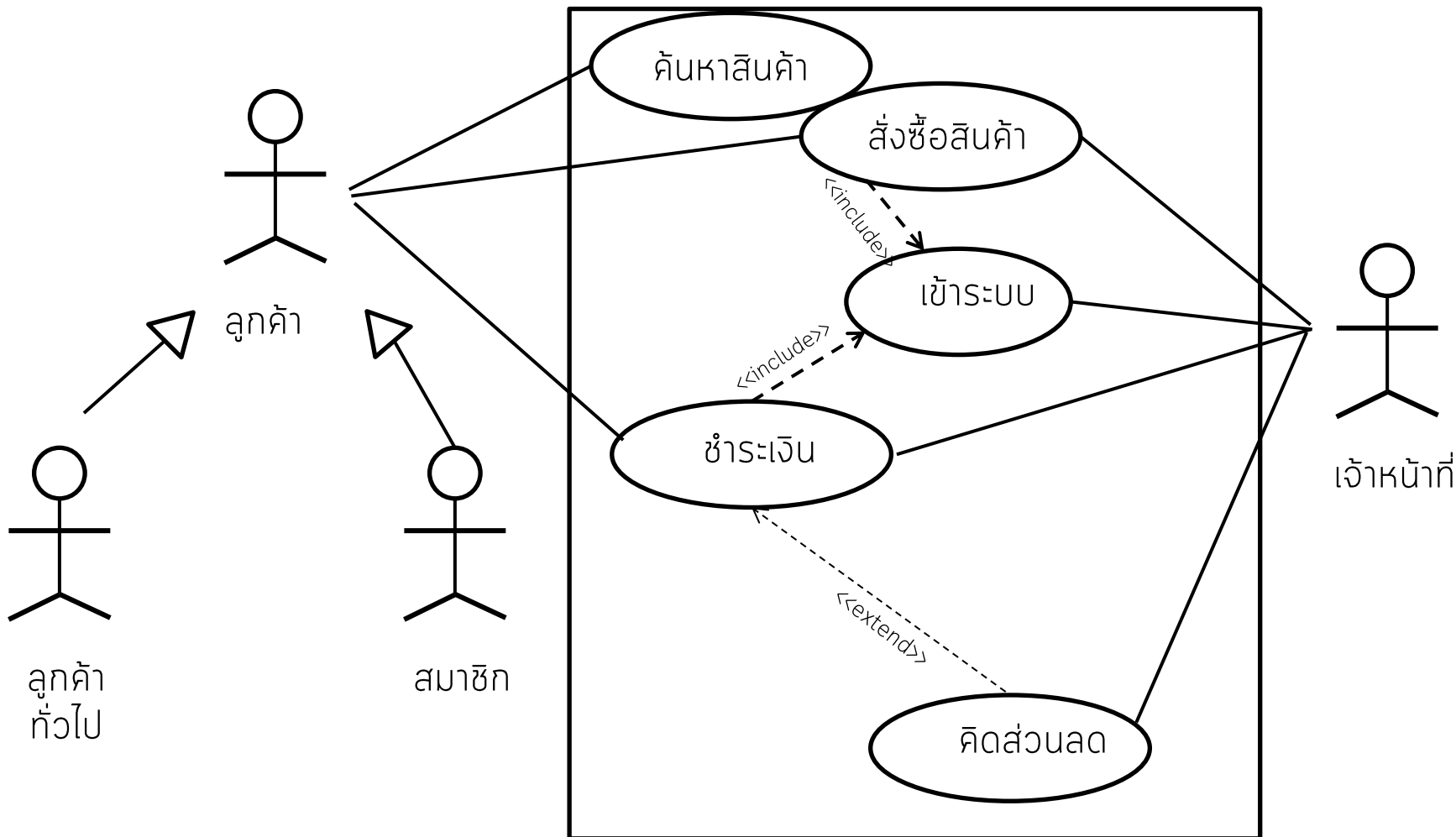
ตัวอย่าง การซื้อ-ขายสินค้า



*** ในการเขียนแผนภาพ ยูเอ็มแอลทุกแผนภาพ นอกจากจะเขียนให้ถูกต้องตามหลักการแล้ว จะเขียนให้อ่านได้ง่ายด้วย ไม่ควรมีเส้นซ้อนทับกัน หรือถ้าจะมีก็ให้มีให้น้อยที่สุด

ตัวอย่าง การซื้อ-ขายสินค้า

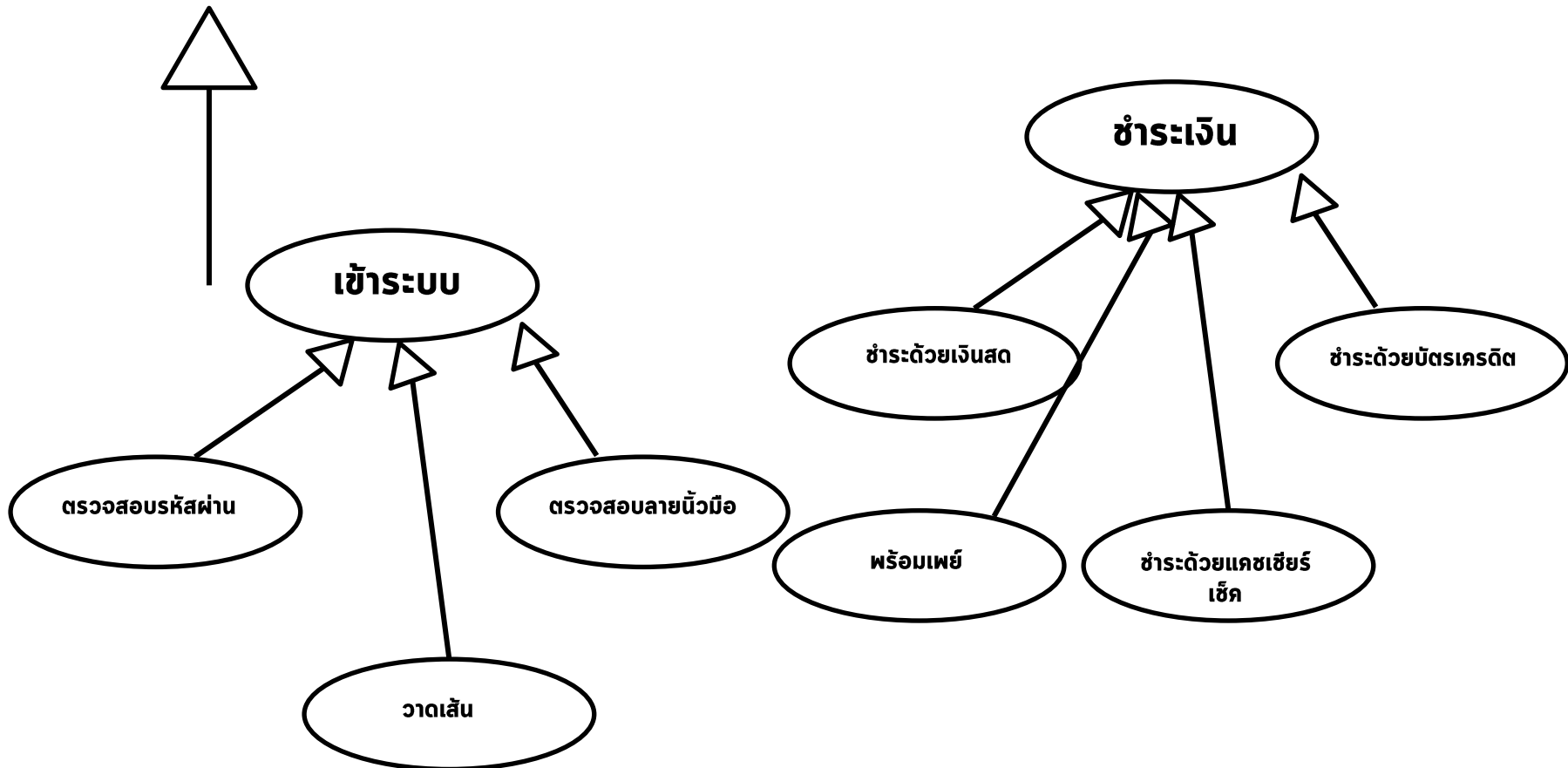
ระบบการซื้อ-ขายสินค้า



*** ในการเขียนแผนภาพ ยูเอ็มแอลทุกแผนภาพ นอกจากจะเขียนให้ถูกต้องตามหลักการแล้ว จะเขียนให้อ่านได้ง่ายด้วย ไม่ควรมีเส้นซ้อนทับกัน หรือถ้าจะมีก็ให้มีให้น้อยที่สุด

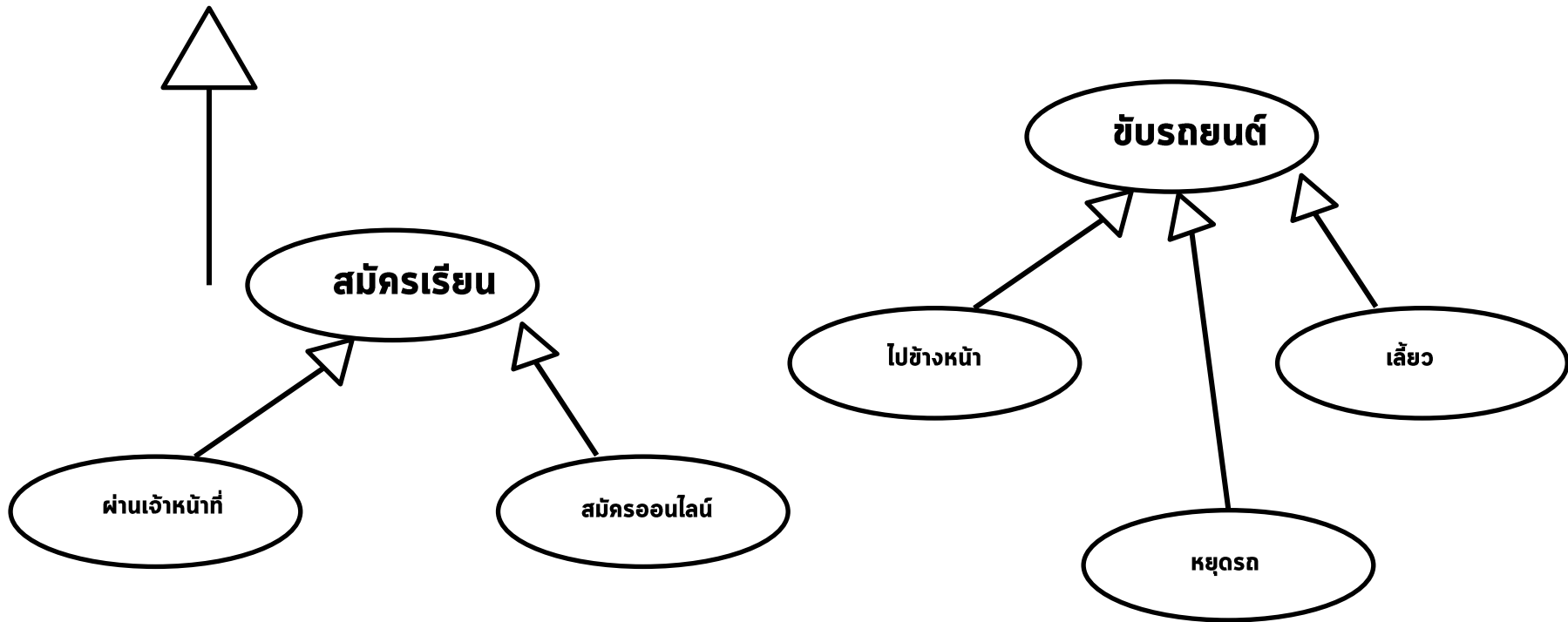
ความสัมพันธ์ระหว่าง use case กับ use case

- ความสัมพันธ์แบบทั่วไป generalize เป็นการแบ่งประเภทของยูสเคส ดังตัวอย่าง



ความสัมพันธ์ระหว่าง use case กับ use case

- ความสัมพันธ์แบบทั่วไป generalize เป็นการแบ่งประเภทของยูสเคส ดังตัวอย่าง

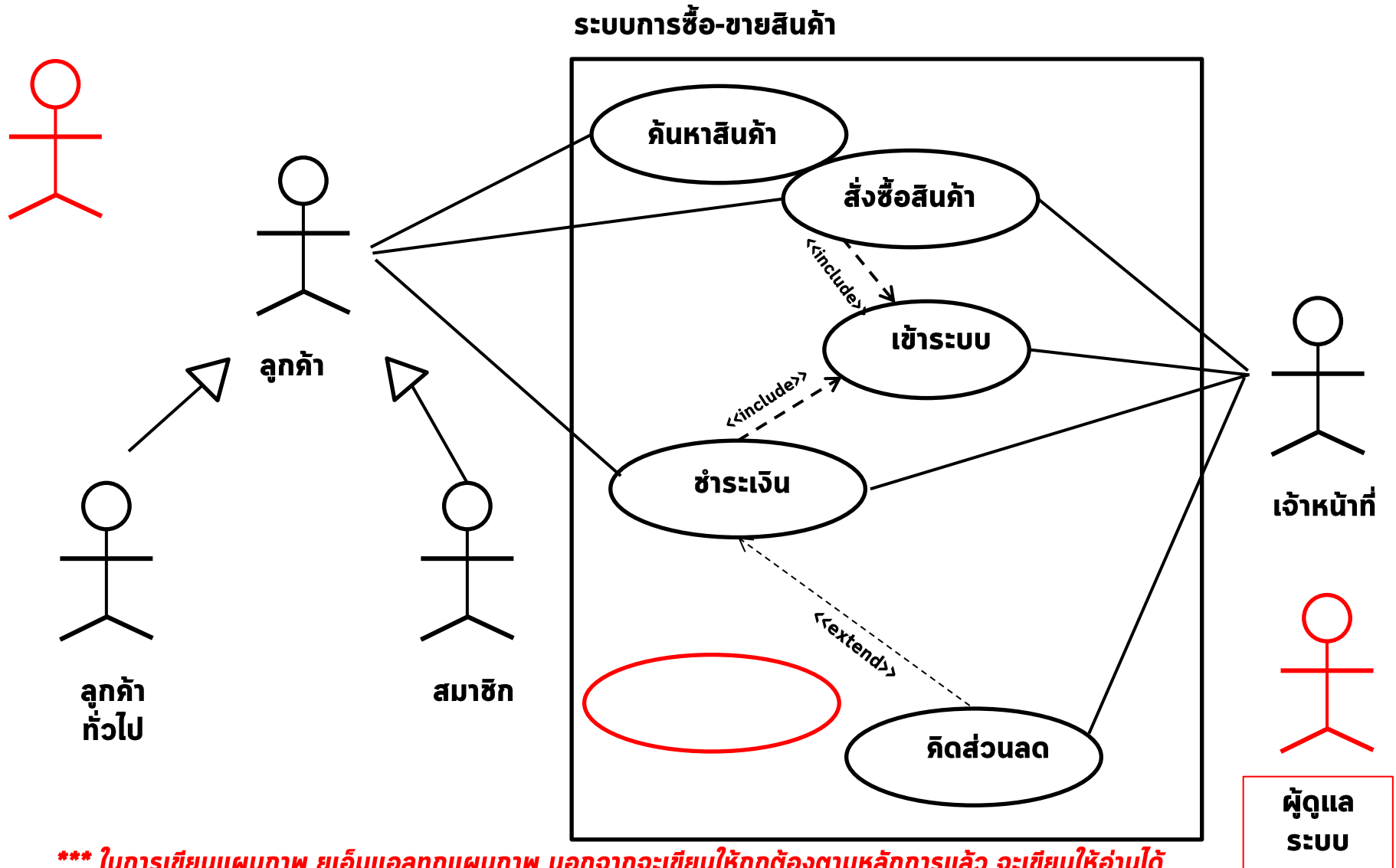


ขั้นตอนการสร้างแผนภาพยูสเคส (Use Case Diagram)

เมื่อเริ่มต้นการสร้างแผนภาพยูสเคส(Use Case Diagram) ด้วยการวิเคราะห์หาขอบเขตของระบบ (Problem Domain) ซึ่งประกอบไปด้วยการค้นหาแอกเตอร์(Actor) ที่ควรมีในระบบ และยูสเคสที่มีปฏิสัมพันธ์โดยตรงกับแอกเตอร์เหล่านั้นขึ้นมาก่อนจากนั้นจึงเพิ่มเติม Use Case อื่นๆ เข้าไปจนครบหน้าที่การทำงานของระบบ

1. ค้นหาแอกเตอร์ Actor
2. ค้นหา Use Case ที่มีปฏิสัมพันธ์กับ Actor นั้นโดยตรง
3. ค้นหาและสร้างความสัมพันธ์ระหว่าง Use Case หรือ Actor (ถ้ามี) แล้วเพิ่มเติม Use Case ใหม่ซึ่งอาจเป็น Included Use Case, Extending Use Case ที่เพิ่มเติมจาก Base Use Case ที่มีอยู่แล้ว หรือจะเพิ่ม Base Use Case ใหม่ก็ได้ (ถ้ามี)
4. ต้องไม่มี Actor ใดเลยที่ไม่มีปฏิสัมพันธ์กับ Use Case
5. ต้องไม่มี Use Case ใดเลยที่ไม่มีปฏิสัมพันธ์กับ Actor
6. Use Case ทุกตัวต้องมีปฏิสัมพันธ์อย่างใดอย่างหนึ่งกับ Actor หรือ Use Case ตัวอื่นๆเสมอ
7. เขียนคำอธิบาย (use case description) แต่ละ Use Case จนครบถ้วน

ตัวอย่าง การซื้อ-ขายสินค้า



*** ในการเขียนแผนภาพ ยูเอ็มแอลทุกแผนภาพ นอกจากจะเขียนให้ถูกต้องตามหลักการแล้ว จะเขียนให้อ่านได้ง่ายด้วย ไม่ควรมีเส้นซ้อนทับกัน หรือถ้าจะมีก็ให้มีให้น้อยที่สุด

Use case diagram แผนภาพยูสเคส

ตัวอย่าง Use Case Diagram ของระบบลงทะเบียน

ตัวอย่าง Use Case Diagram ของระบบลงทะเบียน

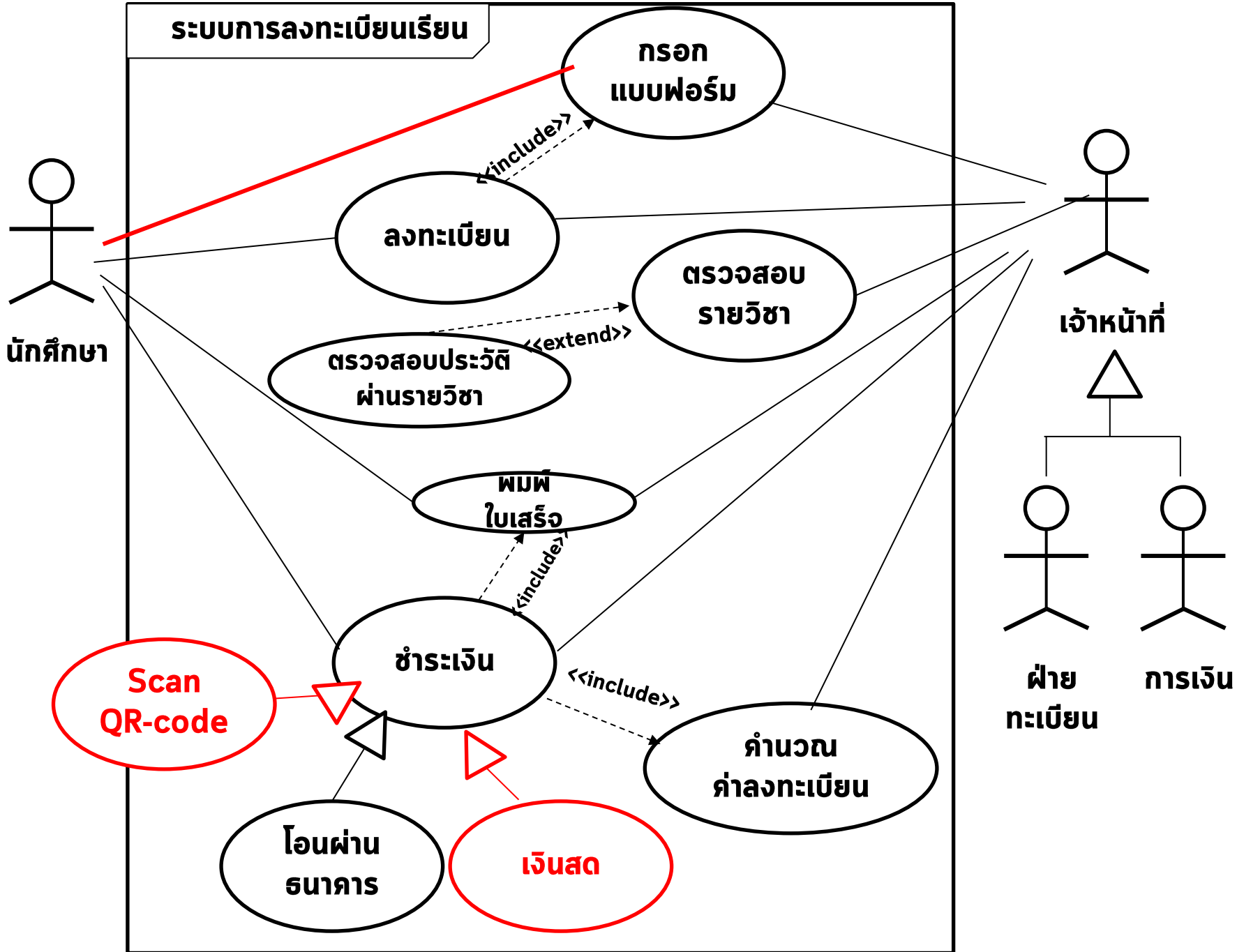
ระบบลงทะเบียนมีกลุ่มบุคคลที่เกี่ยวข้อง 2 กลุ่ม ได้แก่ นักศึกษา และพนักงานของมหาวิทยาลัย(เจ้าหน้าที่ฝ่ายทะเบียนและเจ้าหน้าที่ฝ่ายการเงิน) ในแต่ละเทอมจะต้องมีนักศึกษามาลงทะเบียนเรียนของภาคเรียนปกติ โดยนักศึกษาจะต้องกรอกแบบฟอร์มลงทะเบียนให้เรียบร้อยแล้วนำไปยื่นกับเจ้าหน้าที่ฝ่ายทะเบียนในวันและเวลาที่ประกาศไว้ เมื่อเจ้าหน้าที่รับแบบฟอร์มลงทะเบียนมาแล้ว จะทำการตรวจสอบวิชาที่นักศึกษาได้ลงไว้ในแบบฟอร์มกับประวัติการเรียนว่าถูกต้องหรือไม่ เนื่องจากบางวิชาของแต่ละเทอมมีเงื่อนไขว่าจะลงทะเบียนได้ก็ต่อเมื่อสอบผ่านอีกวิชาหนึ่งมาก่อน(pre-requiresit) เมื่อตรวจสอบพบว่าถูกต้องแล้ว เจ้าหน้าที่ฝ่ายทะเบียนจะคำนวณเงินค่าลงทะเบียนเรียน แล้วบันทึกลงในฐานข้อมูล ส่งพิมพ์ใบรับลงทะเบียนโดยแบ่งออกเป็น 2 ส่วน ส่วนที่ 1 นักศึกษาเก็บไว้เอง ส่วนที่ 2 นำไปชำระเงินโดยโอนผ่านทางธนาคารแล้วนำไปรับชำระเงินกลับมาให้เจ้าหน้าที่ฝ่ายการเงินบันทึกสถานะการชำระเงิน เป็นขั้นตอนสุดท้าย

Use case diagram แผนภาพยูสเคส

การวิเคราะห์

- Actor ได้แก่นต. เจ้าหน้าที่
- Use case
 - ลงทะเบียน
 - ตรวจสอบวิชา
 - จำนวนเงินค่าลงทะเบียน
 - บันทึกข้อมูลการลงทะเบียน
 - พิมพ์ใบลงทะเบียน
 - ชำระเงิน
 - บันทึกสถานะการชำระ
- Relationship
 - ...
 - ...
 - ...
 - ...

ระบบการลงทะเบียนเรียน



2. วิเคราะห์ระบบการจองห้องพักออนไลน์

เนื่องจากการจองห้องพักโรงแรมเดิมนั้นลูกค้าที่เข้าพักจะเดินทางมาจองห้อง ที่โรงแรมโดยตรงหน้าเคาท์เตอร์ บางครั้งห้องเต็มแล้ว ลูกค้าต้องเสียเวลาในการสอบถามสถานะ ซึ่งเดิมโรงแรมไม่มีป้ายแสดงสถานะห้องเต็มไว้หน้าทางเข้าจึงทำให้ลูกค้าต้อง เสียเวลา และในเว็บไซต์ของโรงแรมไม่มีบริการจองห้องพัก มีเพียงตัวอย่างห้องและค่าบริการต่อวันเท่านั้นเพื่อพัฒนาให้โรงแรมมี ประสิทธิภาพมากขึ้นคืออำนวยความสะดวกให้แก่ลูกค้า สร้างความได้เปรียบกับคู่แข่งในธุรกิจเดียวกัน ดังนั้นการนำระบบการจองห้องพักโรงแรมออนไลน์เข้ามาช่วยจัดการระบบงานจึงเป็น ทางเลือกที่เหมาะสม

ระบบจองห้องพัก (Booking online, reservation online) ให้ลูกค้าสามารถจองห้องพัก ตรวจสอบราคาของห้องพักในวันที่ต้องการเข้าพักได้พร้อมทั้งระบุความต้องการพิเศษลงในช่อง บริการเสริม/เพิ่มเติมพิเศษได้อีกด้วย เช่น ต้องการเตียงเสริม ห้องที่สามารถสูบบุหรี่ได้ เป็นต้น ในส่วนของข้อมูลลูกค้าเมื่อลูกค้ากรอกข้อมูลลงไปทำให้โรงแรมสามารถค้นหาข้อมูลผู้จองได้รวดเร็วโดยไม่ต้องกรอกข้อมูลซ้ำเมื่อไปถึงโรงแรม ช่วยจัดการด้านเอกสาร และทำให้เจ้าหน้าที่ตรวจสอบห้องว่างได้โดยไม่เกิดความซ้ำซ้อน



ปัญหาที่เกิดขึ้นของระบบงานเดิม ได้แก่

1. เกิดความยุ่งยากในการจัดเก็บและค้นหาเนื่องจากเอกสารการจองห้องพักโรงแรมที่มากขึ้นเรื่อยๆ
2. ลูกค้าไม่ได้รับความสะดวกเท่าที่ควรในการตรวจสอบการจองห้องพัก
3. เสียเปรียบคู่แข่งในธุรกิจเดียวกัน
4. เกิดความวุ่นวายในกรณีที่มีการจองห้องพักซ้ำซ้อนเนื่องจากเจ้าหน้าที่อาจตรวจสอบห้องว่างผิดพลาด
5. ลูกค้าต้องเสียเวลาในการตรวจสอบสถานะที่โรงแรมโดยตรง

Use case diagram แผนภาพยูสเคส

ระบบสารสนเทศที่เสนอแนะและความสามารถของระบบ

ทำระบบจองห้องพัก (Booking online/ Reservation online) ให้ลูกค้าสามารถจองห้องพัก ตรวจสอบอัตราค่าเช่าห้องพักในวันที่ต้องการเข้าพักได้พร้อมทั้งระบุความต้องการพิเศษลงในช่อง บริการเสริม/เพิ่มเติมพิเศษได้อีกด้วย เช่น ต้องการเตียงเสริม ห้องที่สามารถสูบบุหรี่ได้ เป็นต้น ในส่วนของข้อมูลลูกค้าเมื่อลูกค้ากรอกข้อมูลลงไปทำให้โรงแรมสามารถค้นหา ข้อมูลผู้จองได้รวดเร็วโดยไม่ต้องกรอกข้อมูลซ้ำเมื่อไปถึงโรงแรม ช่วยจัดการด้านเอกสาร และทำให้เจ้าหน้าที่ตรวจสอบห้องพักว่างได้โดยไม่เกิดความซ้ำซ้อน

Use case diagram แผนภาพยูสเคส

วัตถุประสงค์

1. เพื่อให้ลูกค้าสามารถตรวจสอบสถานะห้องพักได้
2. เพื่อให้ลูกค้าได้รับความสะดวกในการใช้บริการ
3. เพื่อได้เปรียบคู่แข่งในธุรกิจเดียวกัน
4. เพื่อให้เอกสารไม่สูญหายและเป็นระเบียบในการตรวจสอบ

การวาด Use Case Diagram สำหรับระบบการจอง ห้องพักออนไลน์

1. กำหนดแอกเตอร์ (Actor)

ผู้ดำเนินการในระบบประกอบด้วย ลูกค้า และ พนักงาน

2. กำหนดยูสเคส (Use case)

การกำหนดยูสเคสของระบบการจองห้องพัก ลูกค้า (Customer) สามารถตรวจสอบสถานะของห้องพัก (Check) ในกรณีของการพิจารณาพบว่าลูกค้า (Customer) ต้องการจะจองห้องพัก (Reservation) จะต้องมีการลงทะเบียนเพื่อเข้าสู่ระบบ (Log in) โดยหลังจากที่ลูกค้าทำการจองห้องพักแล้ว จะต้องมีการชำระเงิน (Payment) ลูกค้าสามารถเลือก (Type of payment) รูปแบบการชำระเงินได้ 2 แบบ คือ การโอนเงินผ่านธนาคาร หรือ จ่ายผ่านบัตรเครดิต จากนั้น พนักงาน (Officer) ทำการตรวจสอบสถานะการจองห้องพัก (Reservation status) และ จัดการห้องพัก (Reservation management)

วิเคราะห์

- หาความต้องการเชิงฟังก์ชัน Functional Requirement
- หาความต้องการเชิงฟังก์ชัน None-Functional Requirement
- หาแอกเตอร์
- หาユスเคสหลัก (based use case) และ ยูสเคสย่อย (sub use case)
- หาความสัมพันธ์ระหว่างแอกเตอร์กับแอกเตอร์ (ถ้ามี)
- หาความสัมพันธ์ระหว่างแอกเตอร์กับユスケス (ถ้ามี)
- หาความสัมพันธ์ระหว่างユスケスหลักกับユスケスย่อย
- เขียนแผนภาพ

requirement

- Functional requirement

- ลูกค้าสามารถจองห้องพักได้
- ลูกค้าสามารถทำการยกเลิกการจองได้
- ลูกค้าสามารถตรวจสอบสถานะการจองได้
- ลูกค้าสามารถตรวจสอบสถานะของห้องได้
- ลูกค้าสามารถลงทะเบียนได้
- ลูกค้าสามารถชำระเงินได้
- ลูกค้าสามารถเลือกประเภทการชำระเงินได้
- เจ้าหน้าที่สามารถตรวจสอบห้องพักได้
- เจ้าหน้าที่สามารถออกใบเสร็จได้
- เจ้าหน้าที่สามารถทำการยกเลิกการจองได้
- ฯลฯ

- None Functional requirement

- ระบบจะต้องสามารถใช้งานได้กับทุก browser
- ระบบสามารถรองรับการใช้งานได้หลายอุปกรณ์ (responsive web design) เช่น iPad มือถือ และคอมพิวเตอร์ การแสดงผลต้องไม่มีความแตกต่าง

จัดโครงสร้างความต้องการ view point

กำหนด ID ของ Requirement ตามประเภทหรือระดับของความ
ต้องการ

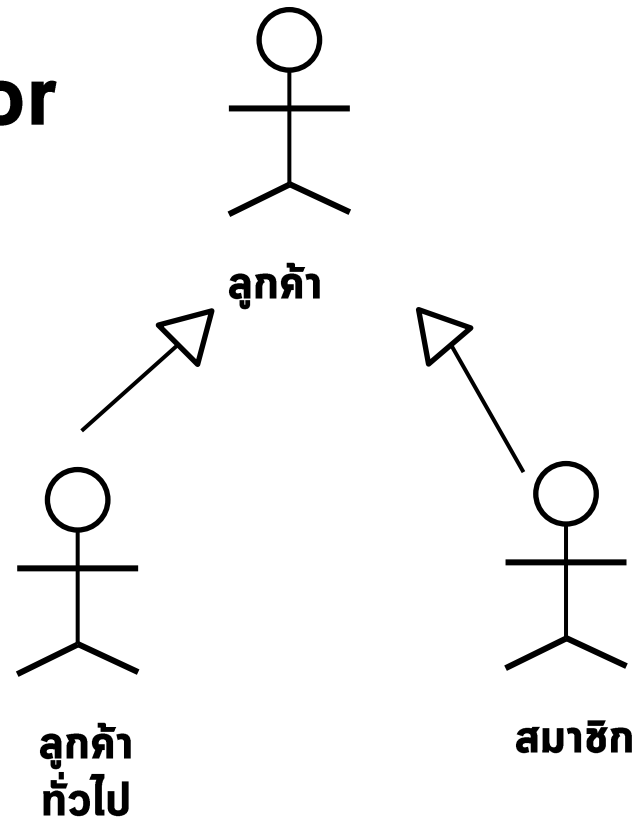
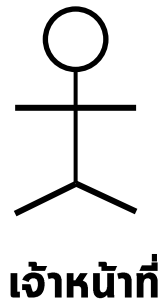
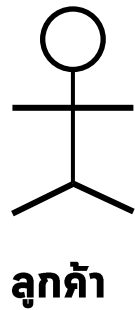
- ผู้ใช้ URS01 - URS99
- เจ้าหน้าที่ SRS01 - SRS99
- ระบบ SYSRS01 - SYSRS99
- ผู้ดูแลระบบ ARS01 - ARS99

ผู้ใช้ (User Requirement) URS01 - URS99

- URS01 : ลูกค้าสามารถจองห้องพักได้
- URS02 : ลูกค้าสามารถตรวจสอบสถานะการจองได้
- URS03 : ...

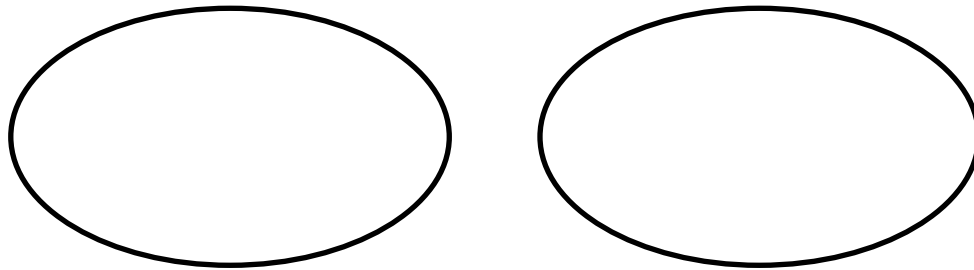
กา Actor

- กา actor
- กาความสัมพันธ์ ระหว่าง actor



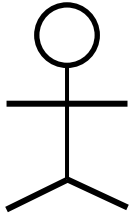
การหา use case

- หา use case
 - Main use case
 - Sub use case
- หาความสัมพันธ์ระหว่างยูสเคส
- หาความสัมพันธ์ระหว่าง แอกเตอร์ กับ ยูสเคส



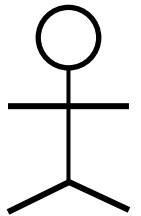
แผนภาพยูสเคส

ระบบการจองห้องพักโรงแรมออนไลน์



ลูกค้า

แอกเตอร์ หลักที่ กระทำหรือใช้งานยูสเคสหลัก จะต้องอยู่
ด้านซ้าย

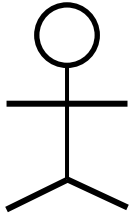


เจ้าหน้าที่

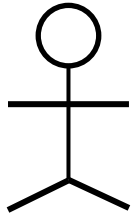
แผนภาพยูสเคส

ระบบการจองห้องพักโรงแรมออนไลน์

สมัครสมาชิก



ลูกค้า



เจ้าหน้าที่

ขอบเขตของระบบที่จะพัฒนา(Problem Domain)

- **เอกสารที่ระบุถึงระบบงานที่กำลังจะพัฒนา โดยบอกถึงปัญหาของงานที่ทำอยู่ ข้อจำกัดของระบบที่กำลังดำเนินงาน หรือความต้องการของระบบที่คาดหวัง ตลอดจนเทคนิคการทำงาน ระเบียบ กฎเกณฑ์ที่จำเป็น**

Glossary พจนานุกรมศัพท์เทคนิค

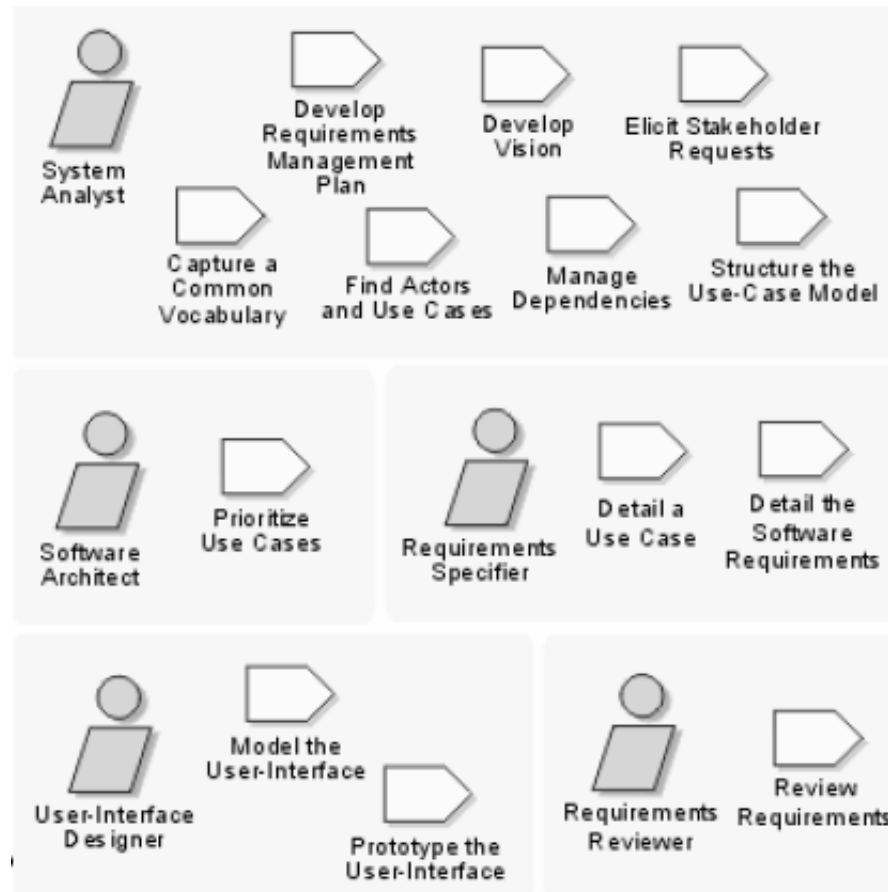
- เป็นเอกสารที่รวบรวมคำนิยามของระบบที่ใช้กัน หลักการเลือกคำก็คือ จะต้องเลือกคำที่เป็นที่เข้าใจกันภายในหน่วยงาน หรือเป็นคำศัพท์ที่มีความหมายเฉพาะผิดแปลกไปจากความเข้าใจโดยทั่วไป หรือเป็นคำศัพท์ที่นิยามใช้เฉพาะระบบ เป็นต้น

เอกสารรายละเอียดของระบบ (Requirements Specification)

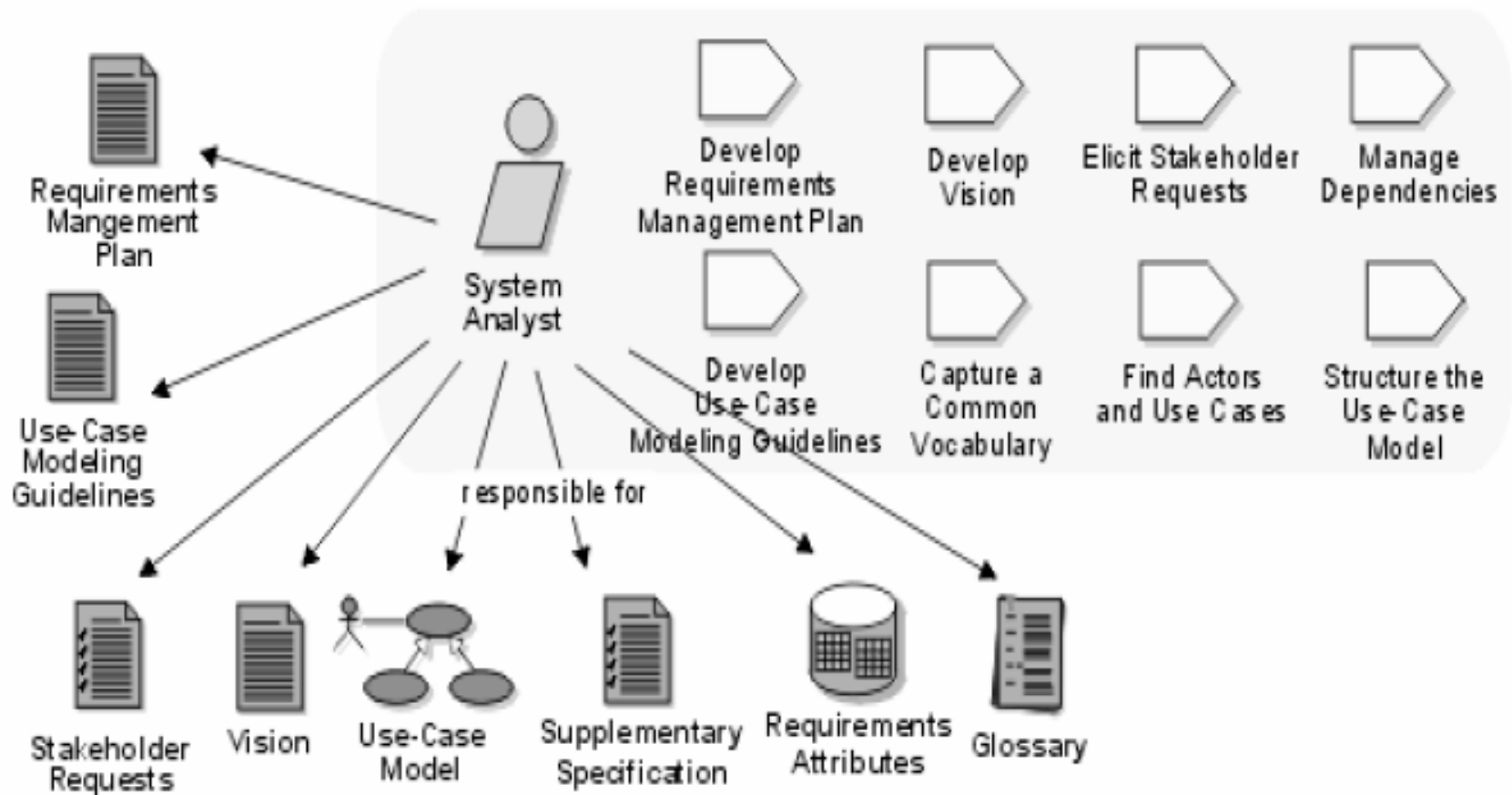
- บทนำ - กล่าวถึงภาพกว้างๆ ของระบบ หลักการและความสำคัญของปัญหา หรือ ที่มาและความสำคัญของปัญหา (บทที่ 1)
- ภาพรวมทั้งหมด - อธิบายขอบเขตของระบบที่จะต้องติดต่อกับส่วนอื่นๆ ของระบบทั้งภายในและภายนอก รวมถึงการทำงานร่วมกับระบบเดิม
- ความต้องการโดยละเอียด - อธิบายความต้องการโดยละเอียด เพื่อให้นักออกแบบสามารถออกแบบได้ และผู้ทดสอบระบบทดสอบได้
- เอกสารสนับสนุนอื่นๆ เช่น ดัชนี ภาคผนวก

หน้าที่เกี่ยวข้อง

- เอกสารประกอบการพัฒนาระบบเป็นหน้าที่ของใคร



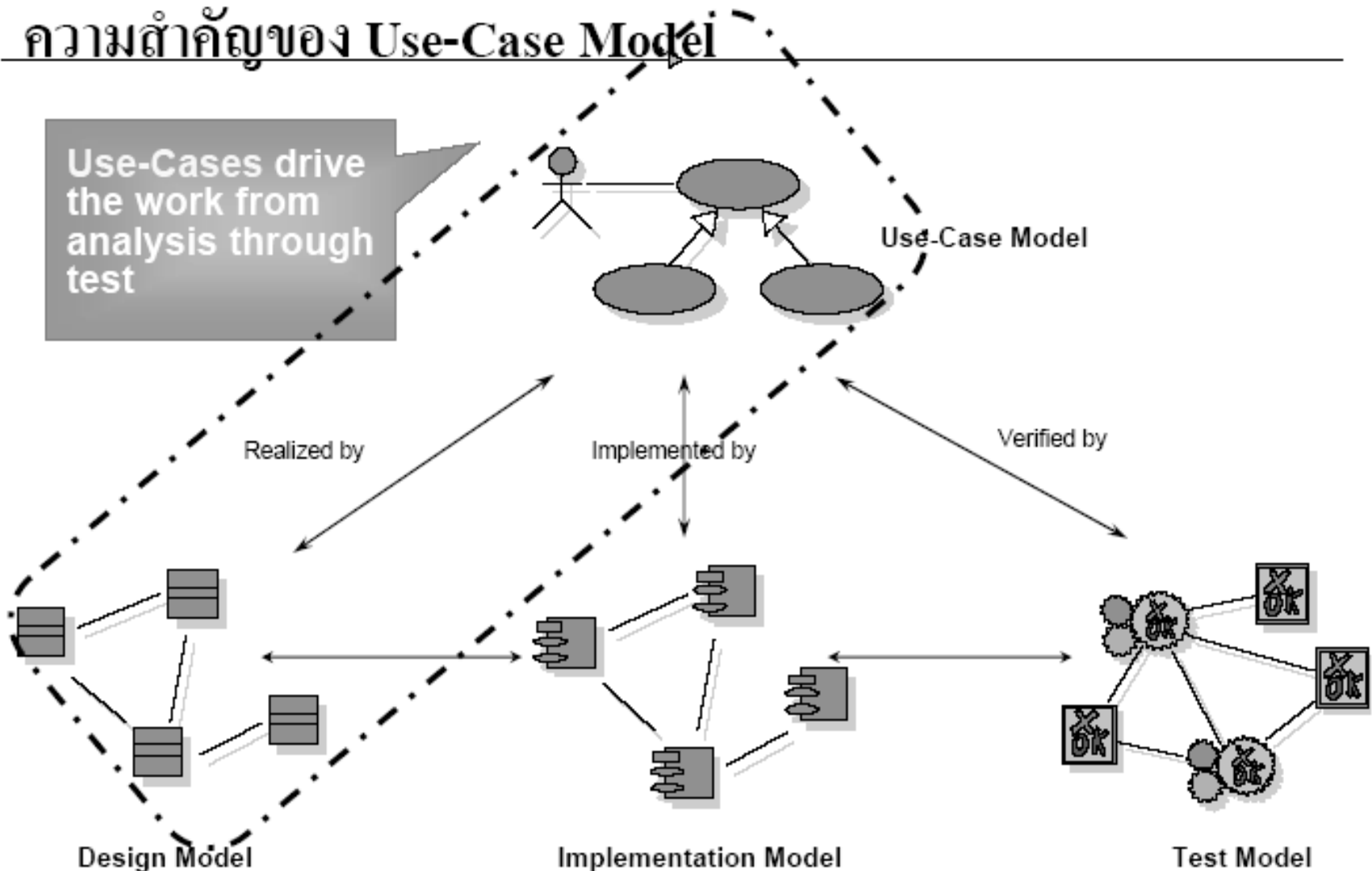
หน้าที่ของนักวิเคราะห์ระบบ



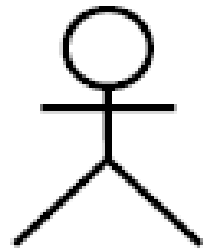
Concepts in Use Case Model

ความสำคัญของ Use-Case Model

Use-Cases drive the work from analysis through test



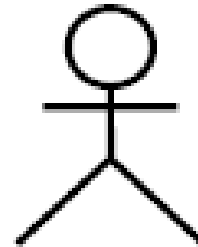
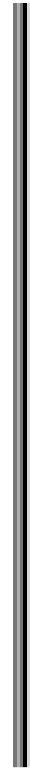
Concepts in Use Case Model



Actor



Use-Case

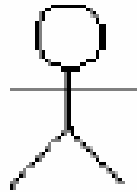


Actor

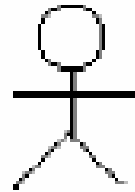


Use-Case

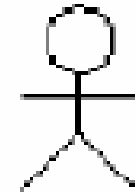
หลักการกำหนดชื่อ Actor



DATABASE



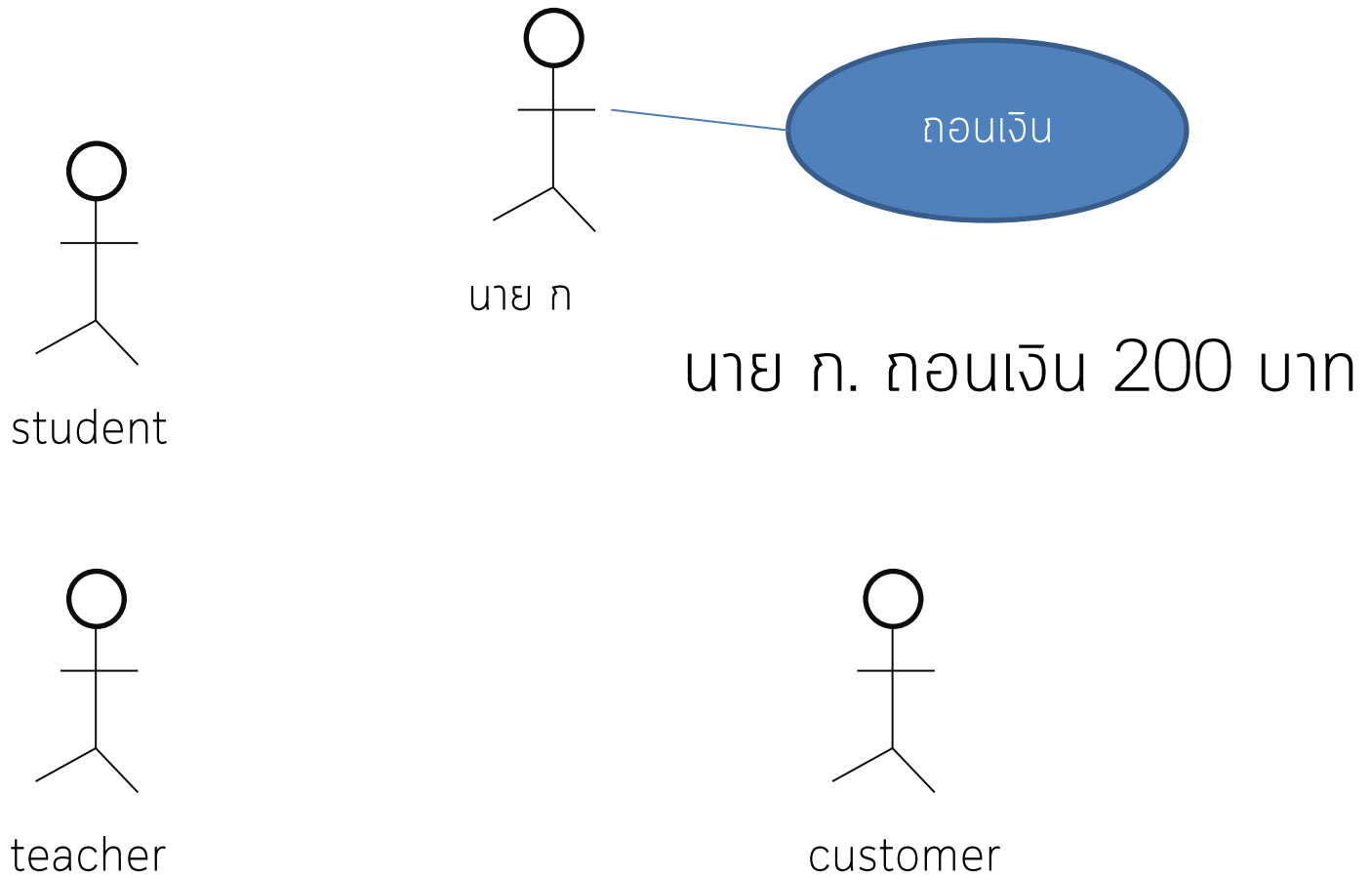
Student



User

ชื่อ Actor ต้องสื่อความหมาย

หลักการกำหนดชื่อ Actor



การลงทะเบียน

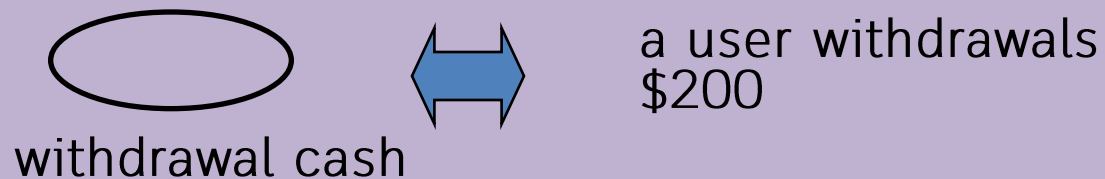


scenario




นายสมชาย ลงทะเบียนเรียน ภาคการศึกษา 2/2560

Scenario

- สถานการณ์ หรือตัวอย่างเรื่องราวการใช้งานระบบ
- Scenario จัดเป็น instance ของ use case



ความสัมพันธ์ใน Use Case Diagram

- ความสัมพันธ์เชิงโครงสร้าง(Association)
 - แทนด้วย 
- ความสัมพันธ์แบบสืบทอด(Generalization)
 - แทนด้วย 
- ความสัมพันธ์แบบพึ่งพา(Dependency)
 - แทนด้วย 

ความสัมพันธ์ระหว่าง Actor



Charlie

Charlie as
professor



Professor

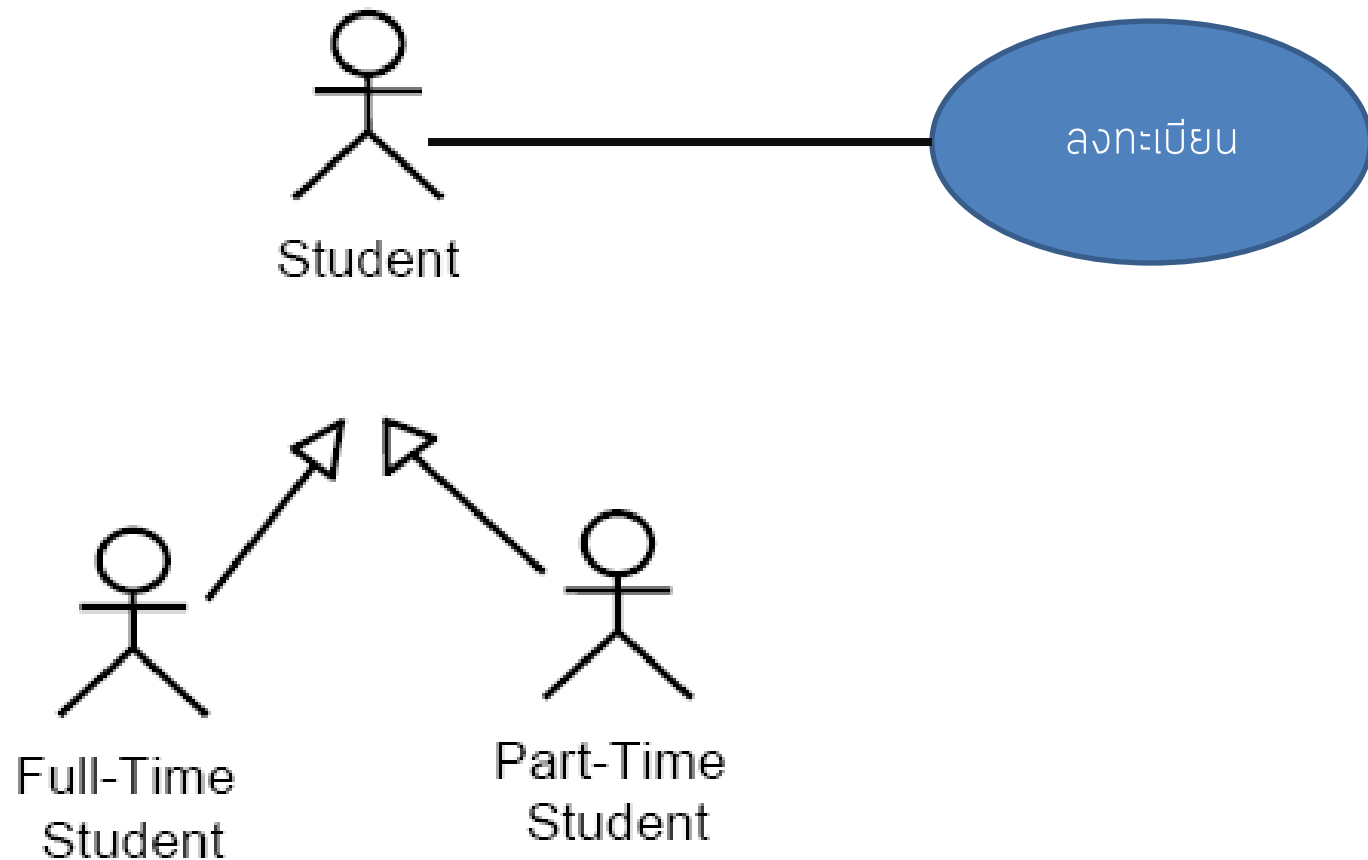
Charlie as
student



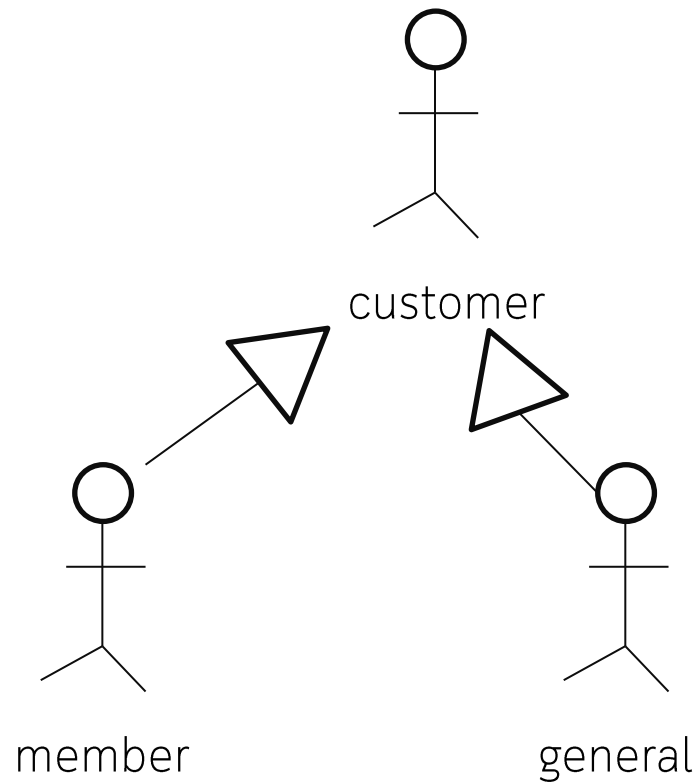
Student

Actor Generalization

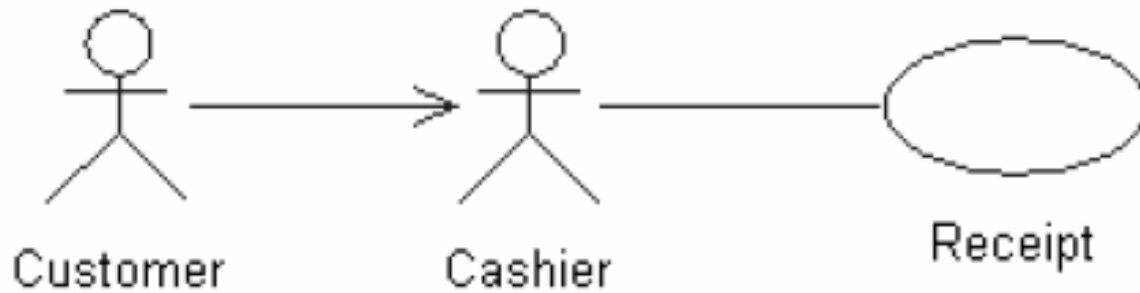
แบ่งนักเรียน เป็นภาคปกติ และ ภาค กศ.พว.



Actor generalization



ความสัมพันธ์ระหว่าง Actor กับ Use Case

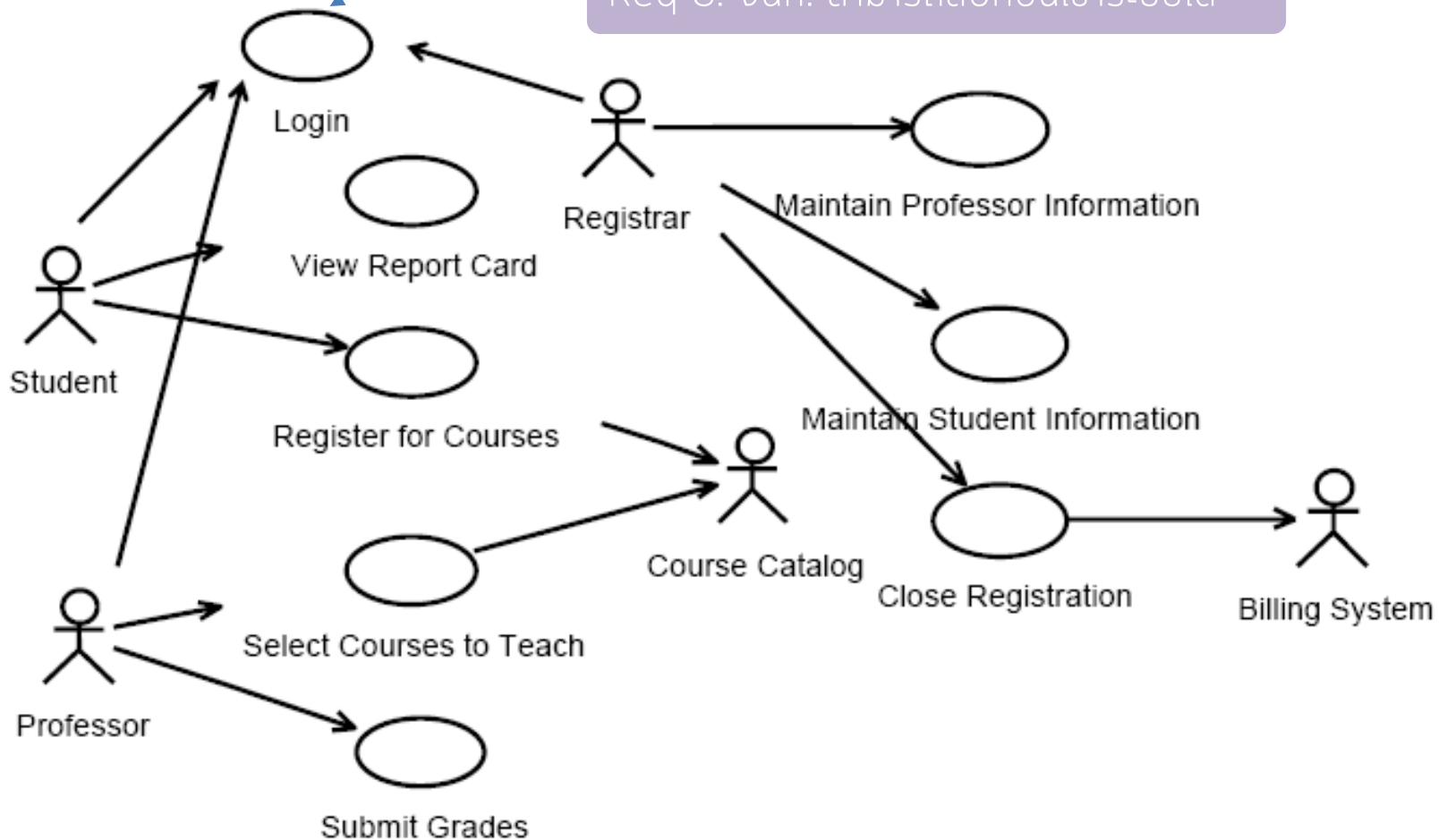


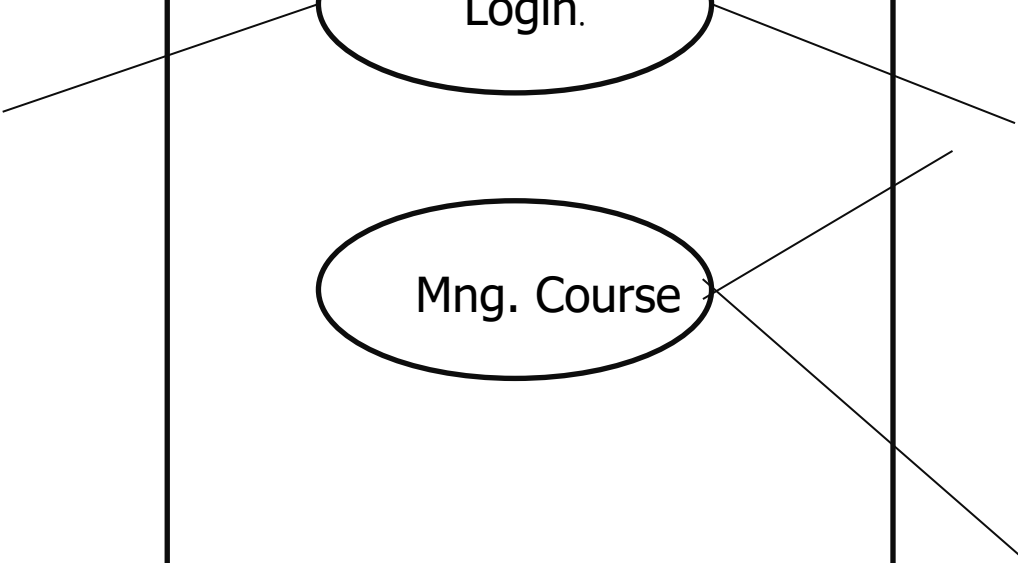
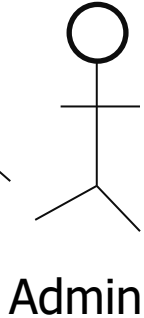
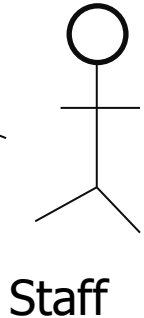
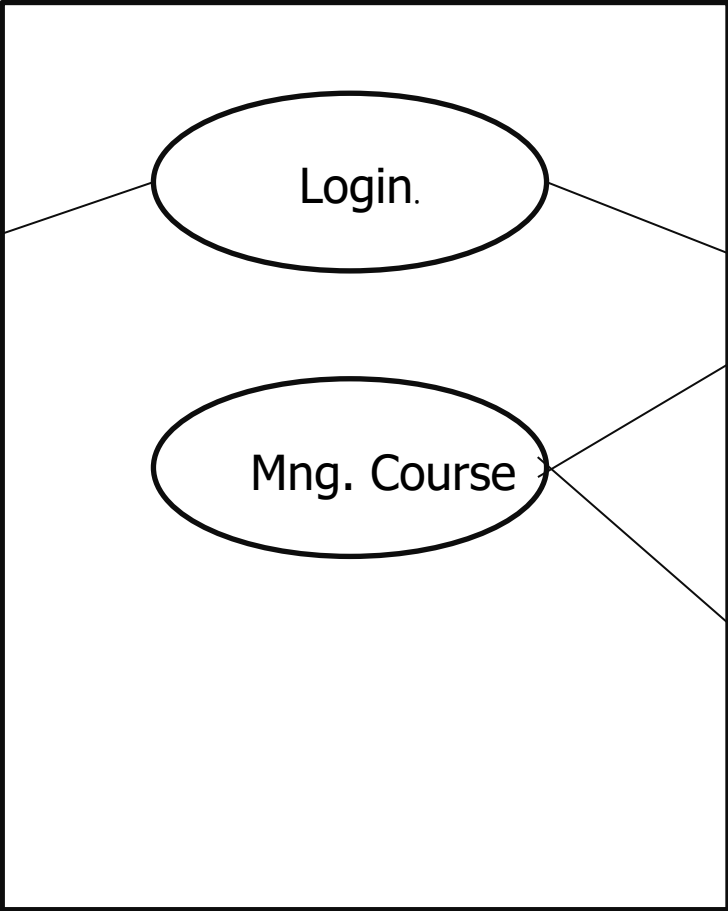
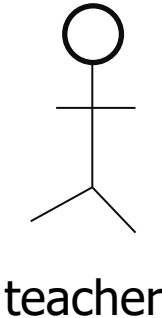
Example: Use Case Diagram

Req 1. นักเรียนสามารถล็อกอินเข้าระบบได้

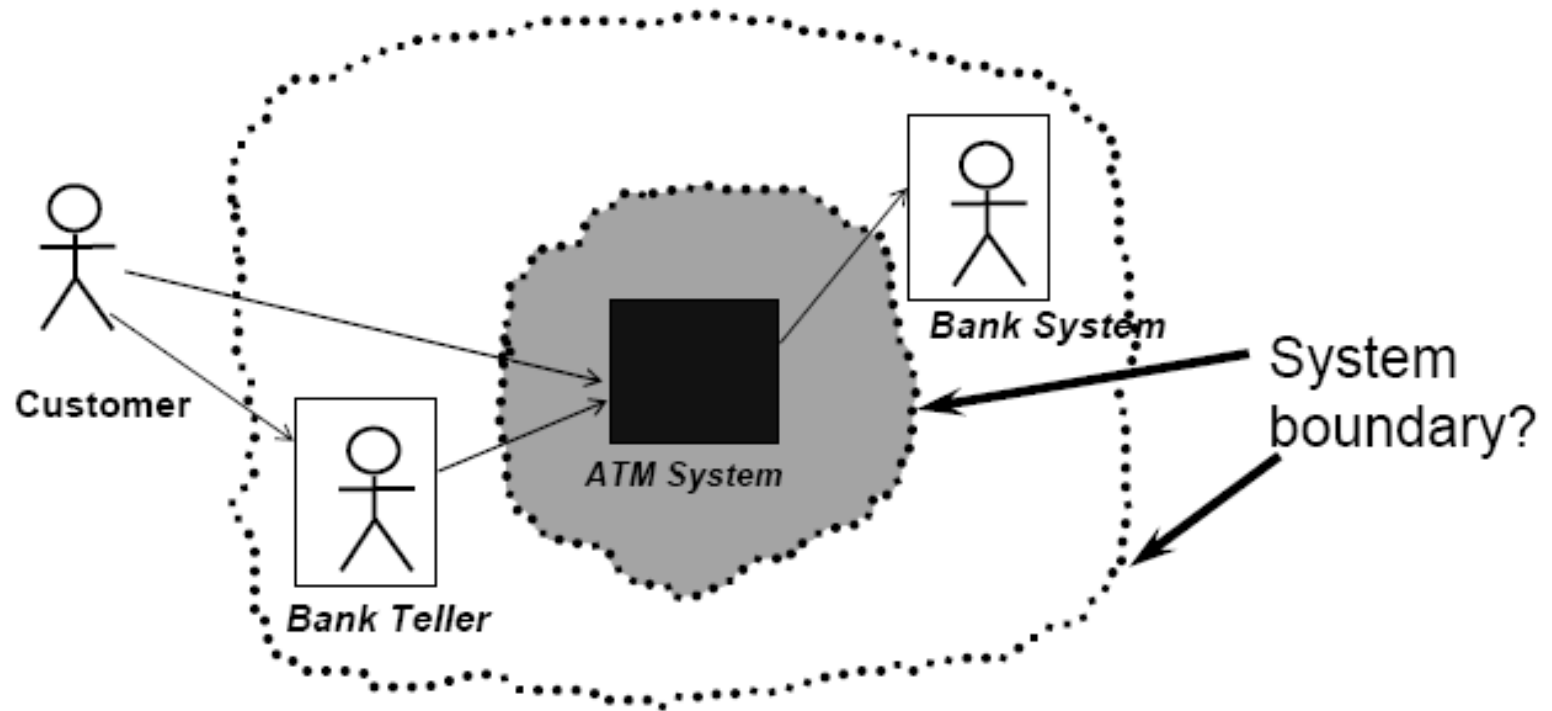
Req 2. อาจารย์สามารถล็อกอินเข้าระบบได้

Req 3. จนท. สามารถล็อกอินเข้าระบบได้

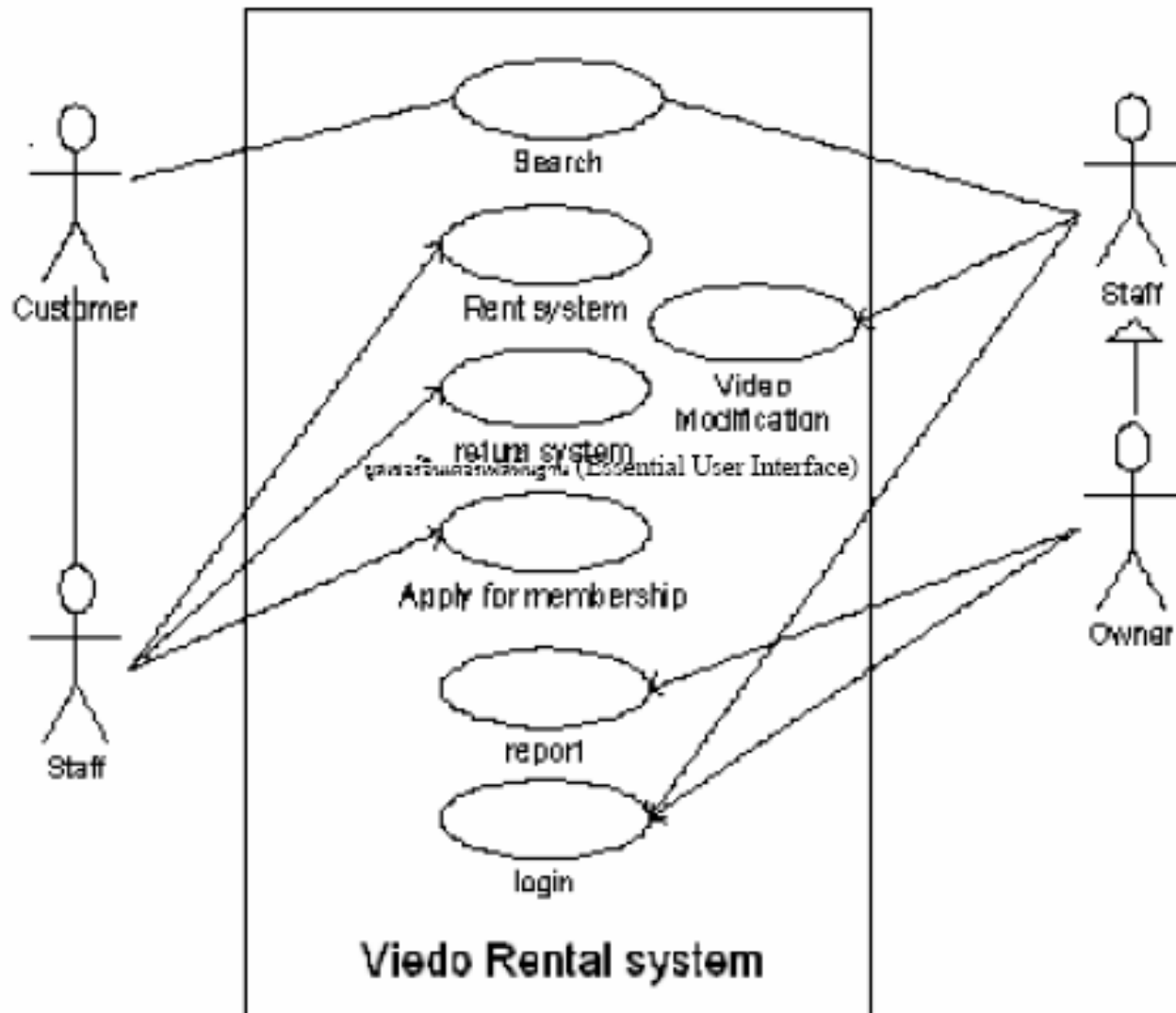




Actor กับขอบเขตของระบบ

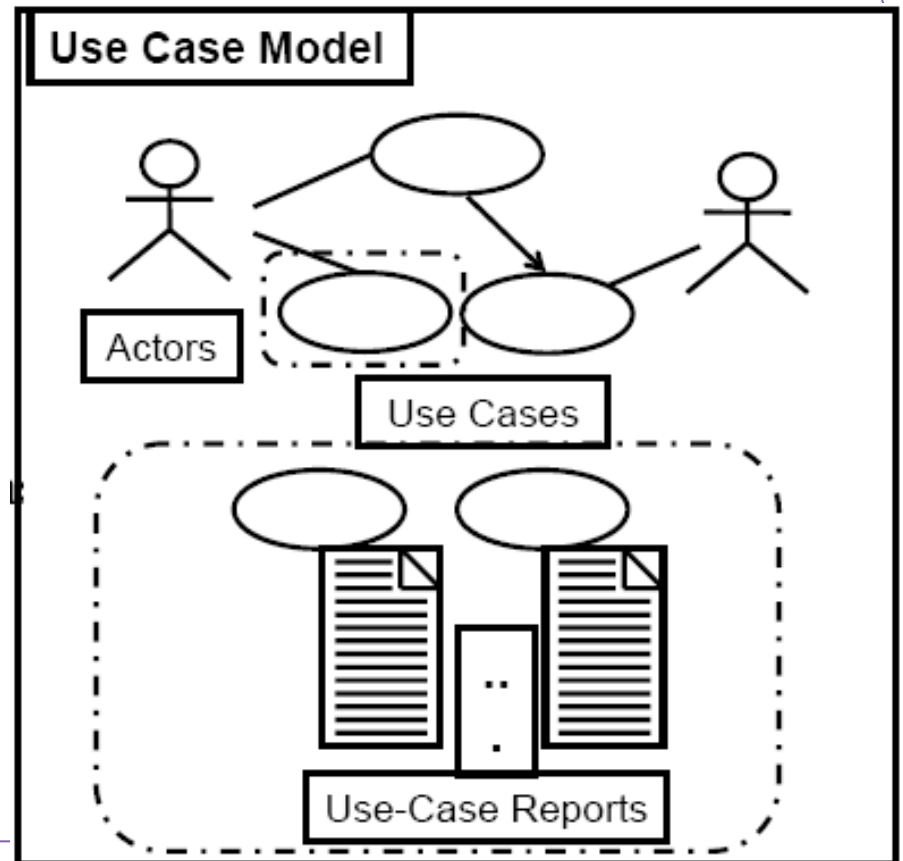


ตัวอย่าง Use Case

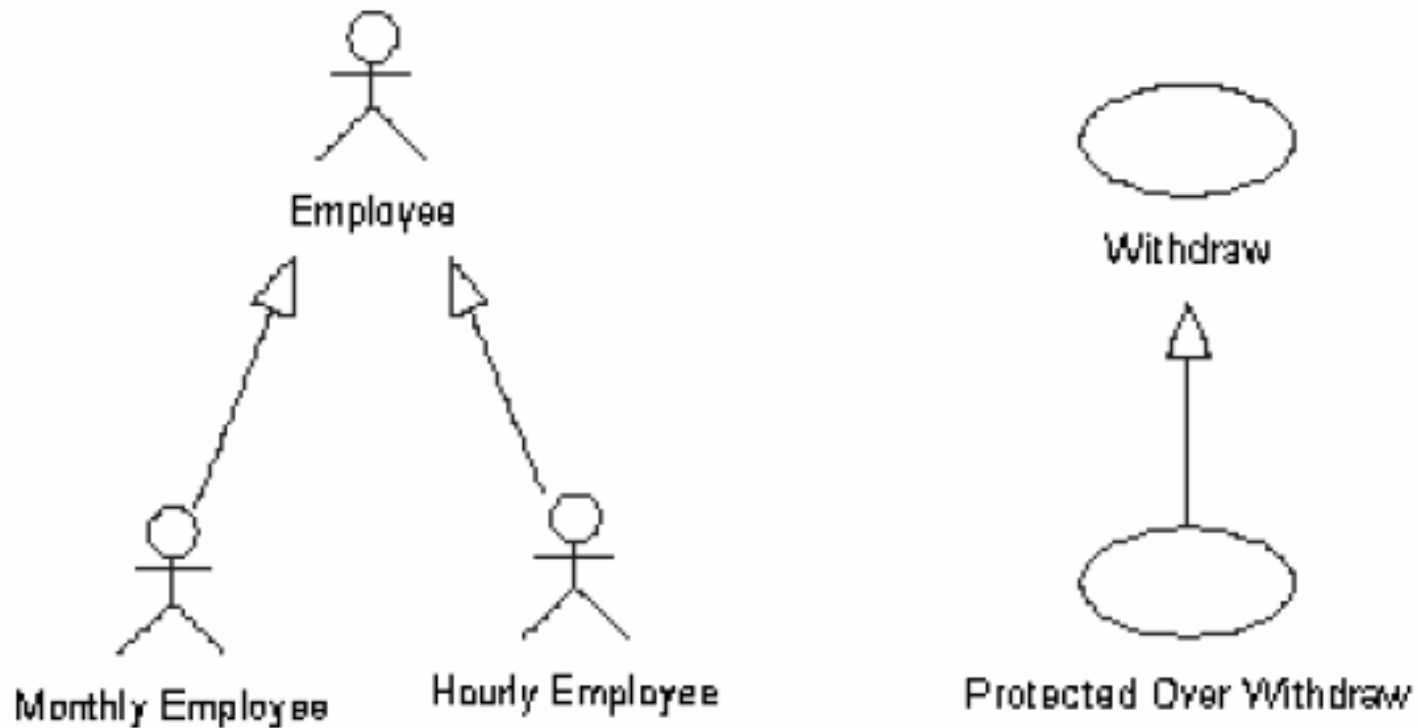


หลักการตั้งชื่อ Use Case

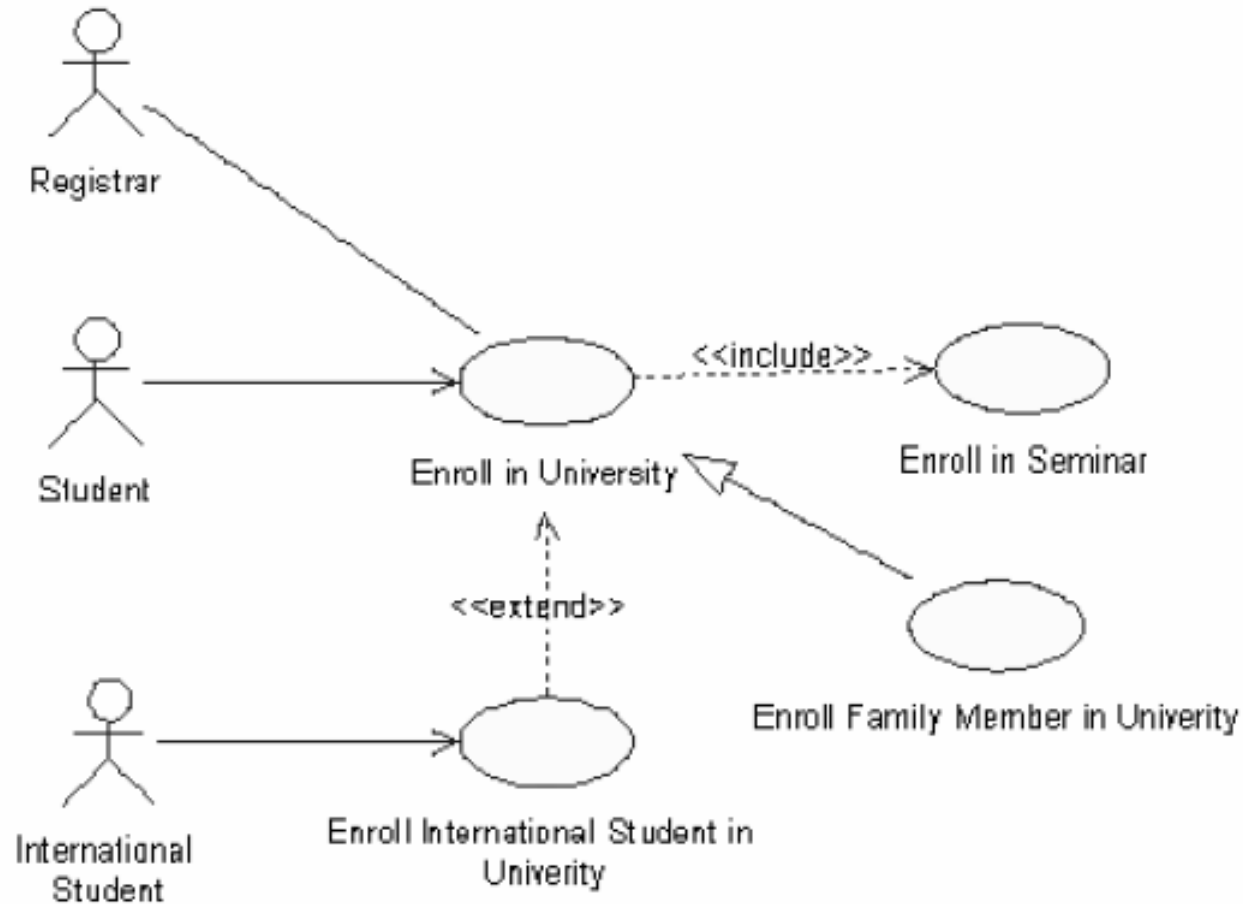
- ชื่อยูสเคส
- การทำงานโดยย่อ
- ลำดับเหตุการณ์ที่เกิดขึ้น
- ความสัมพันธ์
- เงื่อนไขก่อนยูสเคสทำงาน
- เงื่อนไขหลังยูสเคสทำงาน



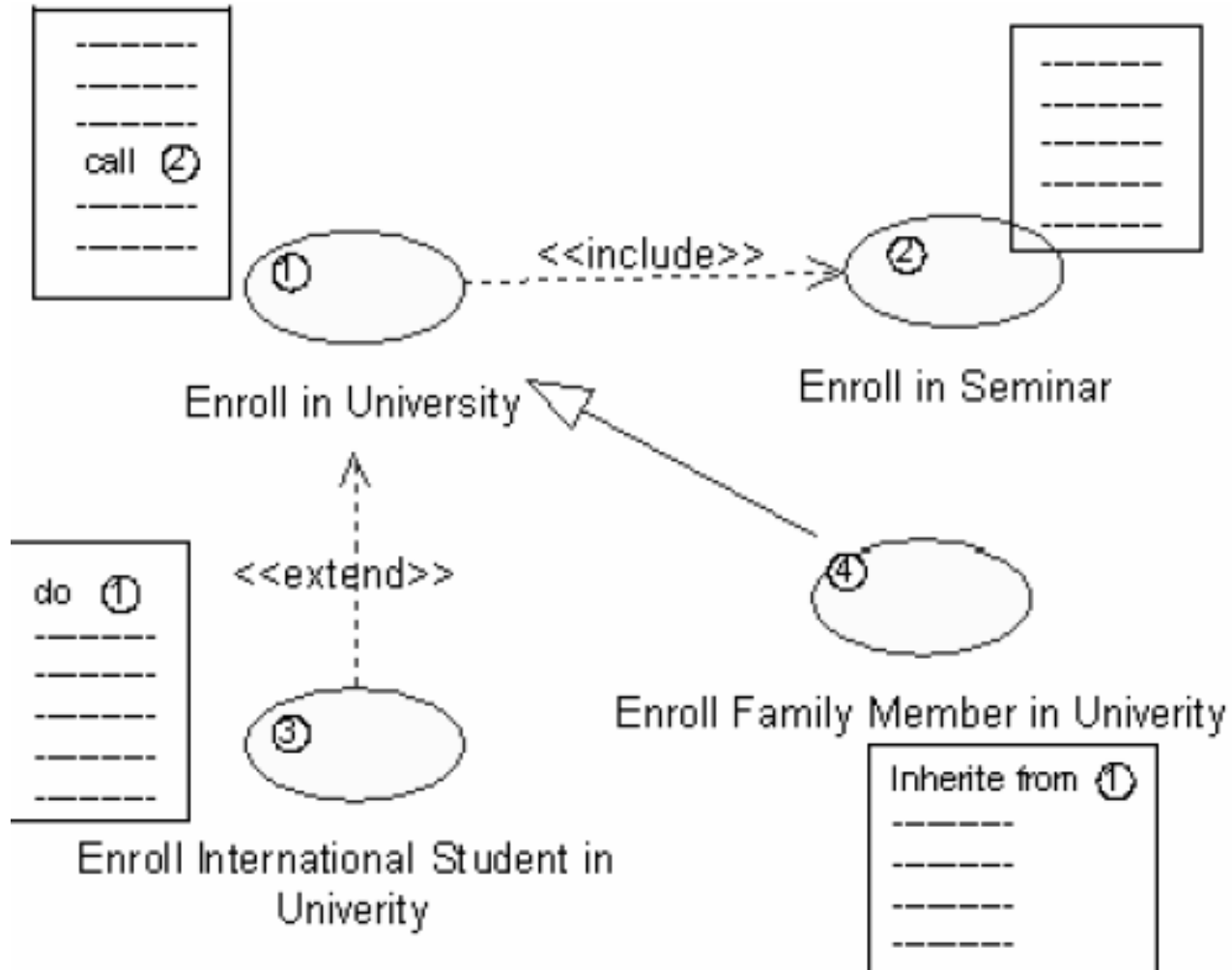
Generalization နှင့် Use Case Diagram



ความสัมพันธ์ <<include>> และ <<extend>>



สรุปความสัมพันธ์ของ Use Case Diagram



ข้อควรระวังในการใช้ Use Case Diagram

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

GUI ð vs GUI Ìðð

before

"Grep" what?



after

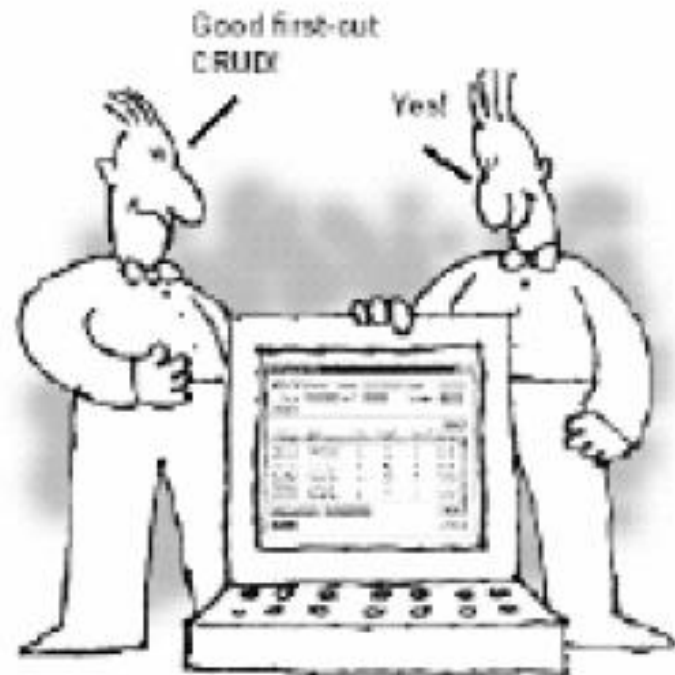
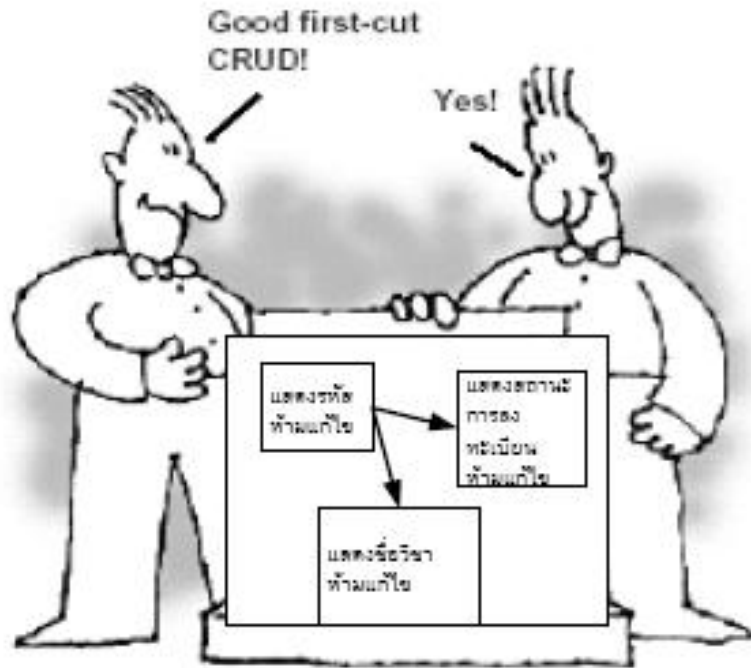
I get it!
Easy!



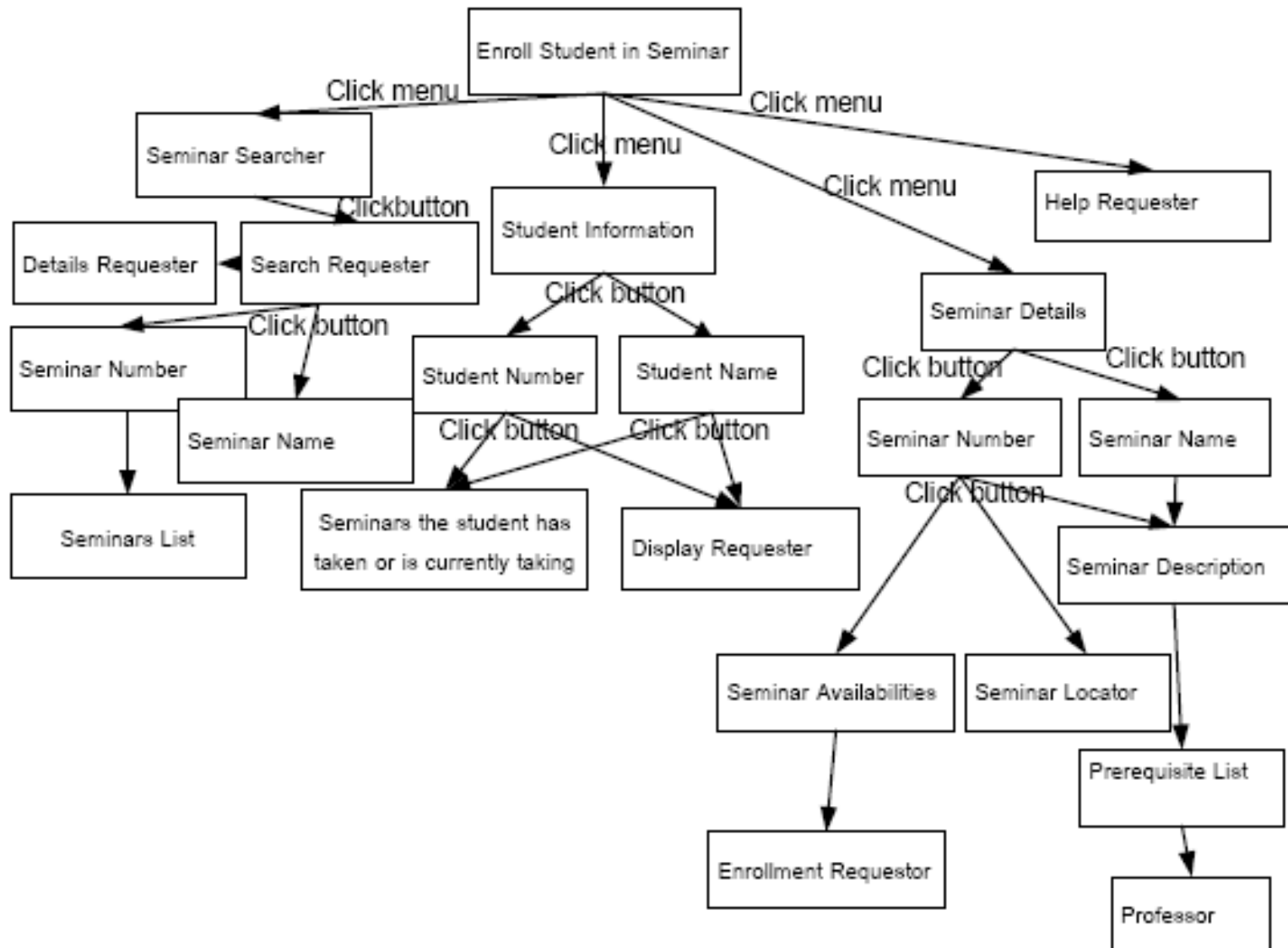
ขั้นตอนการสร้างอินเทอร์เน็ตพื้นฐาน

- ค้นหาการใช้ระบบ
- โมเดลยูสเซอร์อินเทอร์เน็ตหลัก
- โมเดลยูสเซอร์อินเทอร์เน็ตย่อย
- สำรวจความง่ายของการใช้อินเทอร์เน็ต
- สำรวจความสัมพันธ์ของส่วนต่างๆ

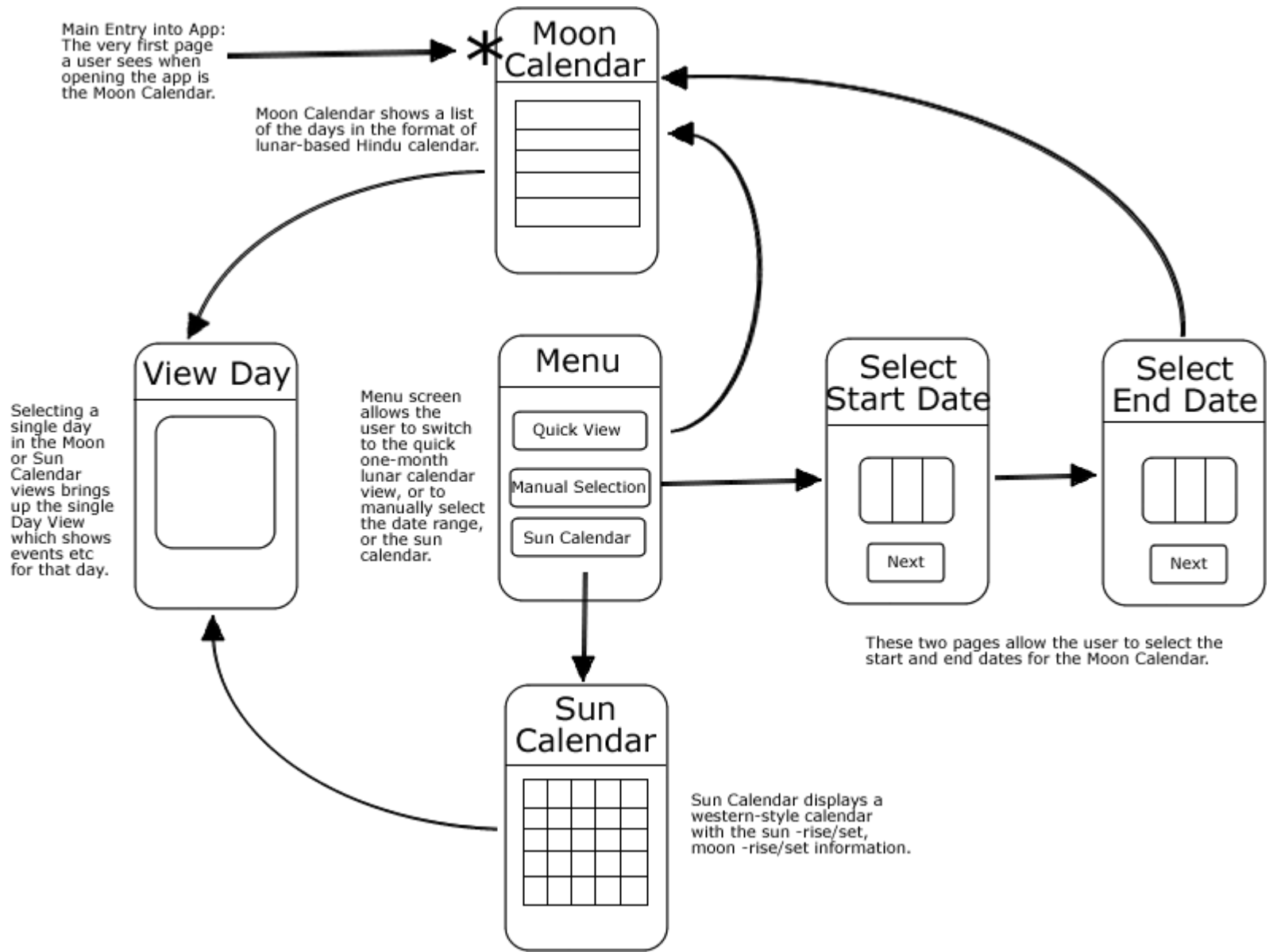
Interface Flow Diagram vs GUI Design



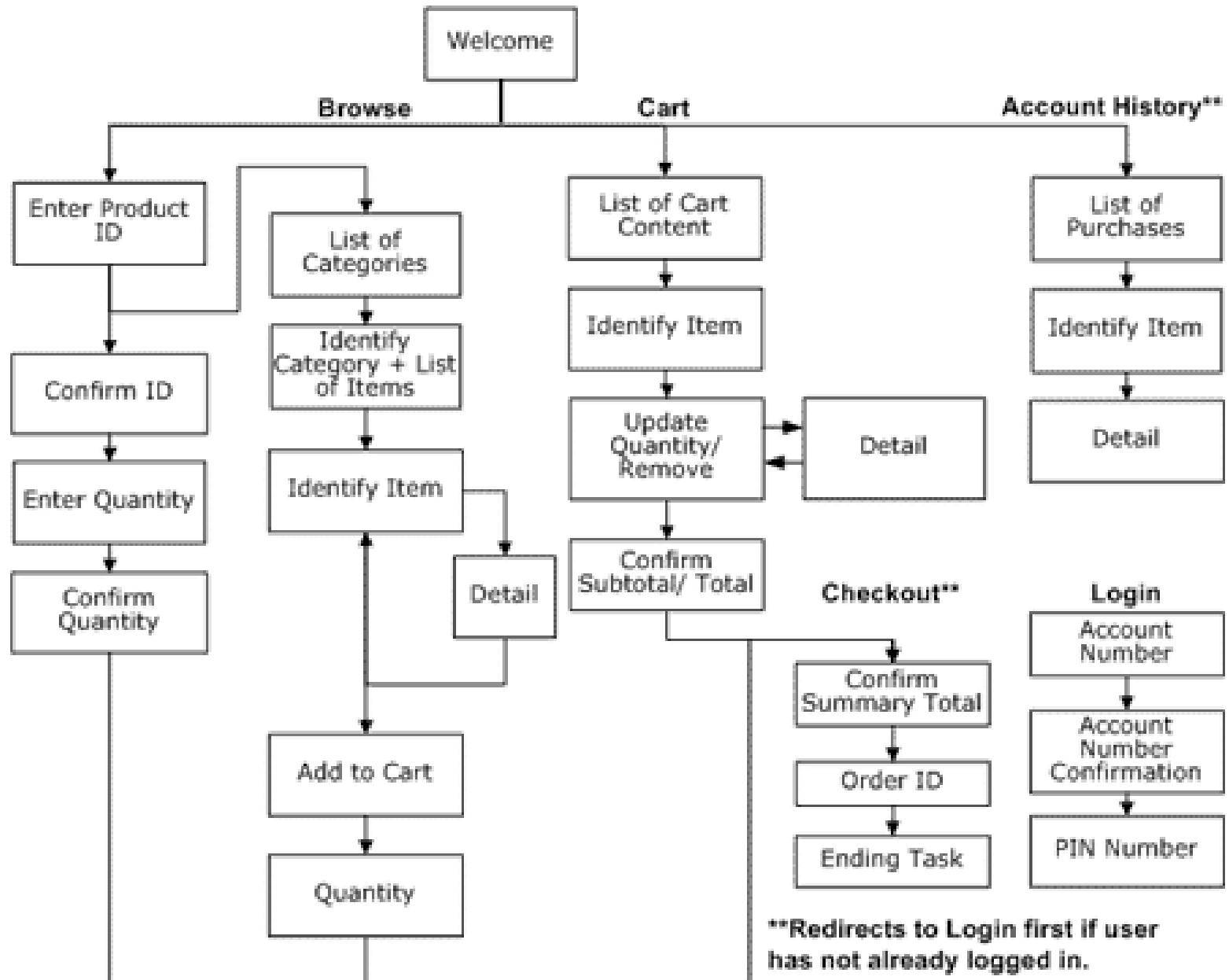
ตัวอย่าง Interface Flow Diagram



ตัวอย่าง Interface Flow Diagram

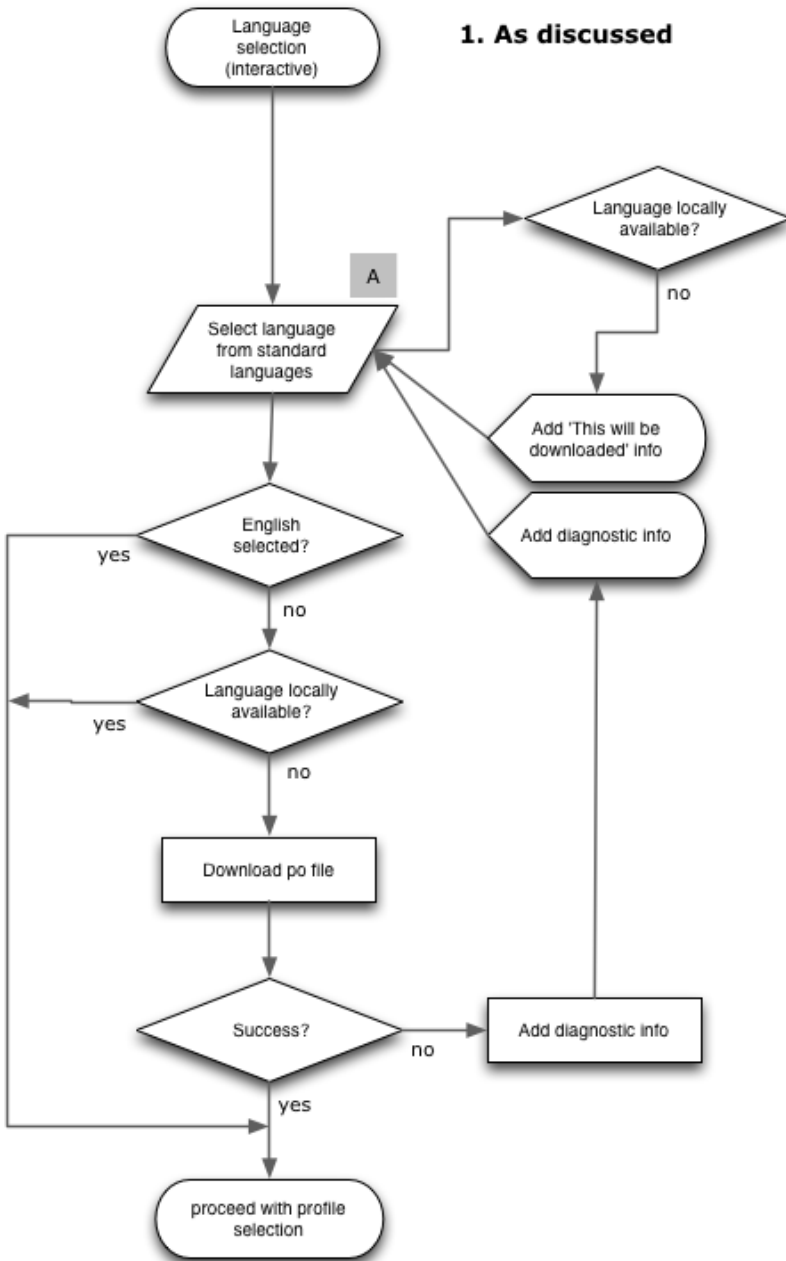


ตัวอย่าง Interface Flow Diagram

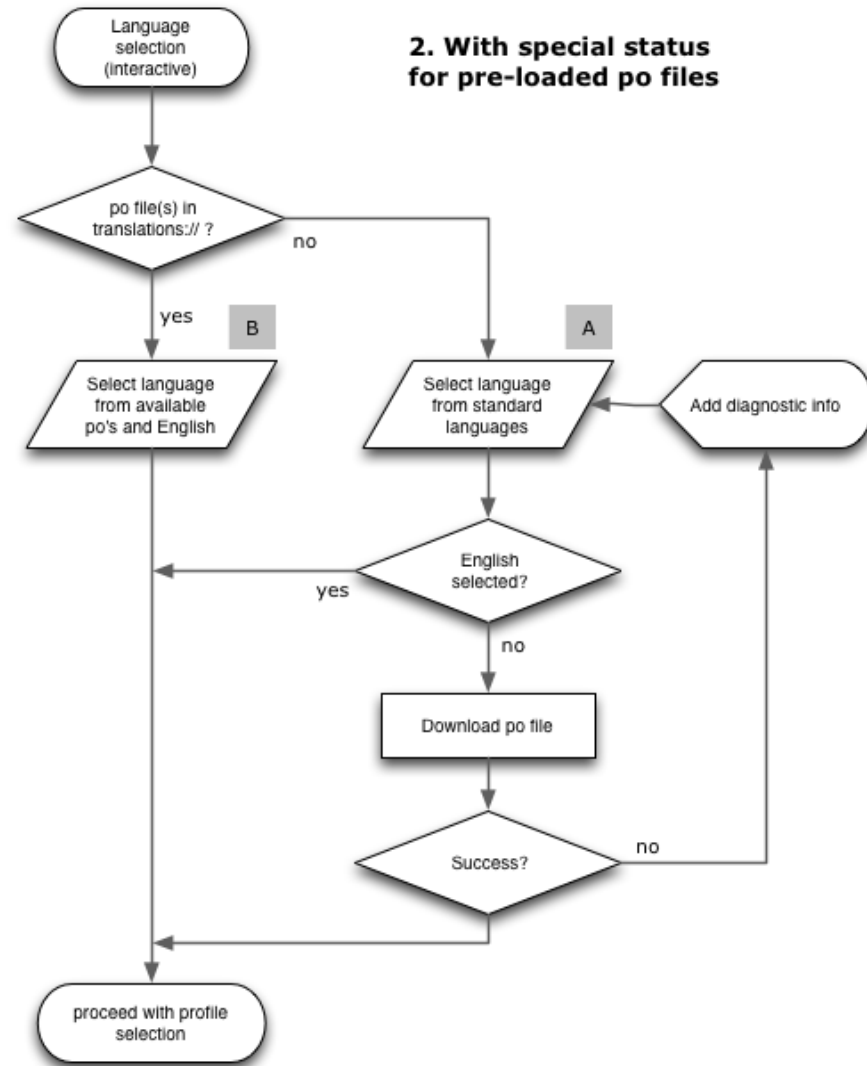


ตัวอย่าง Interface Flow Diagram

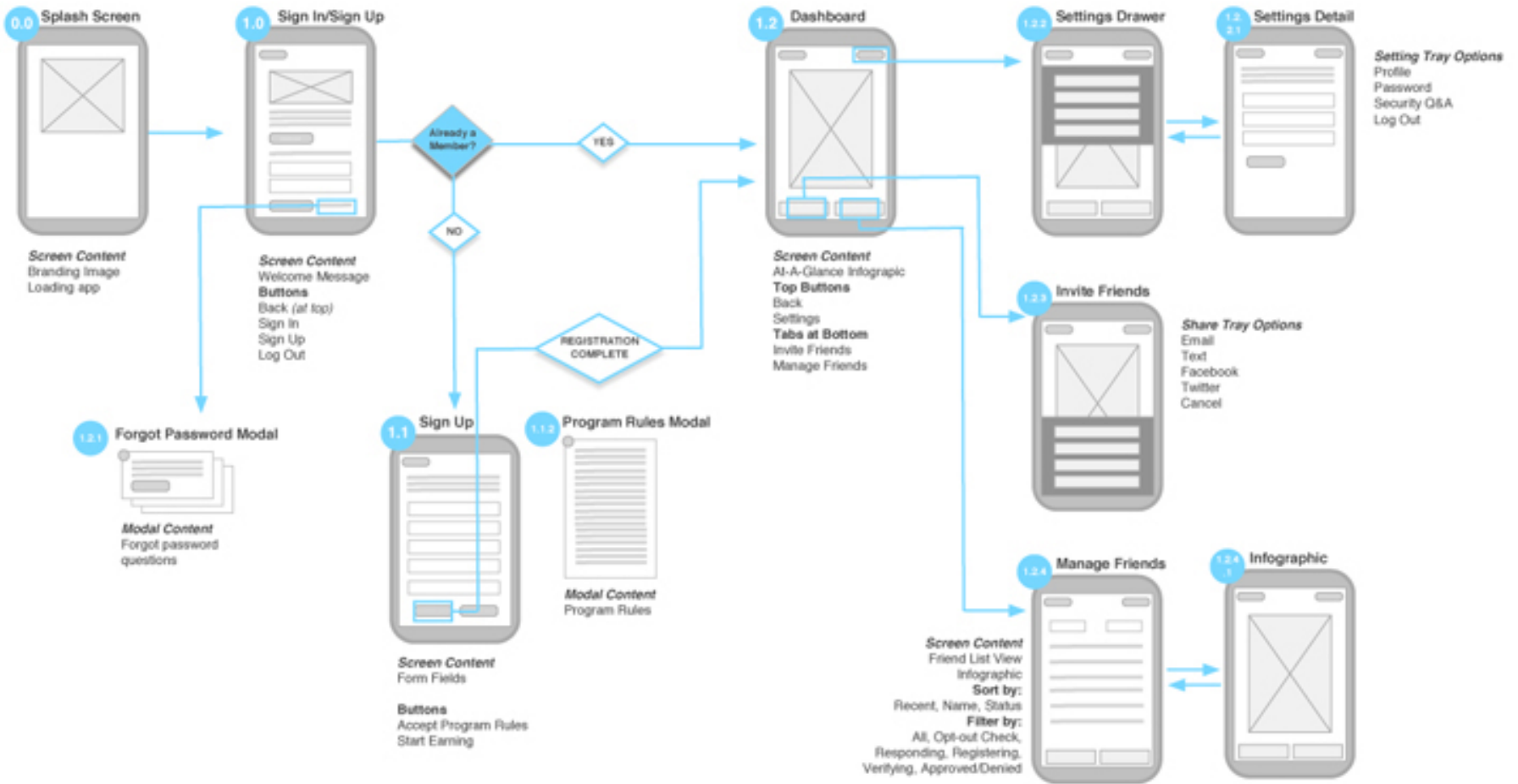
1. As discussed



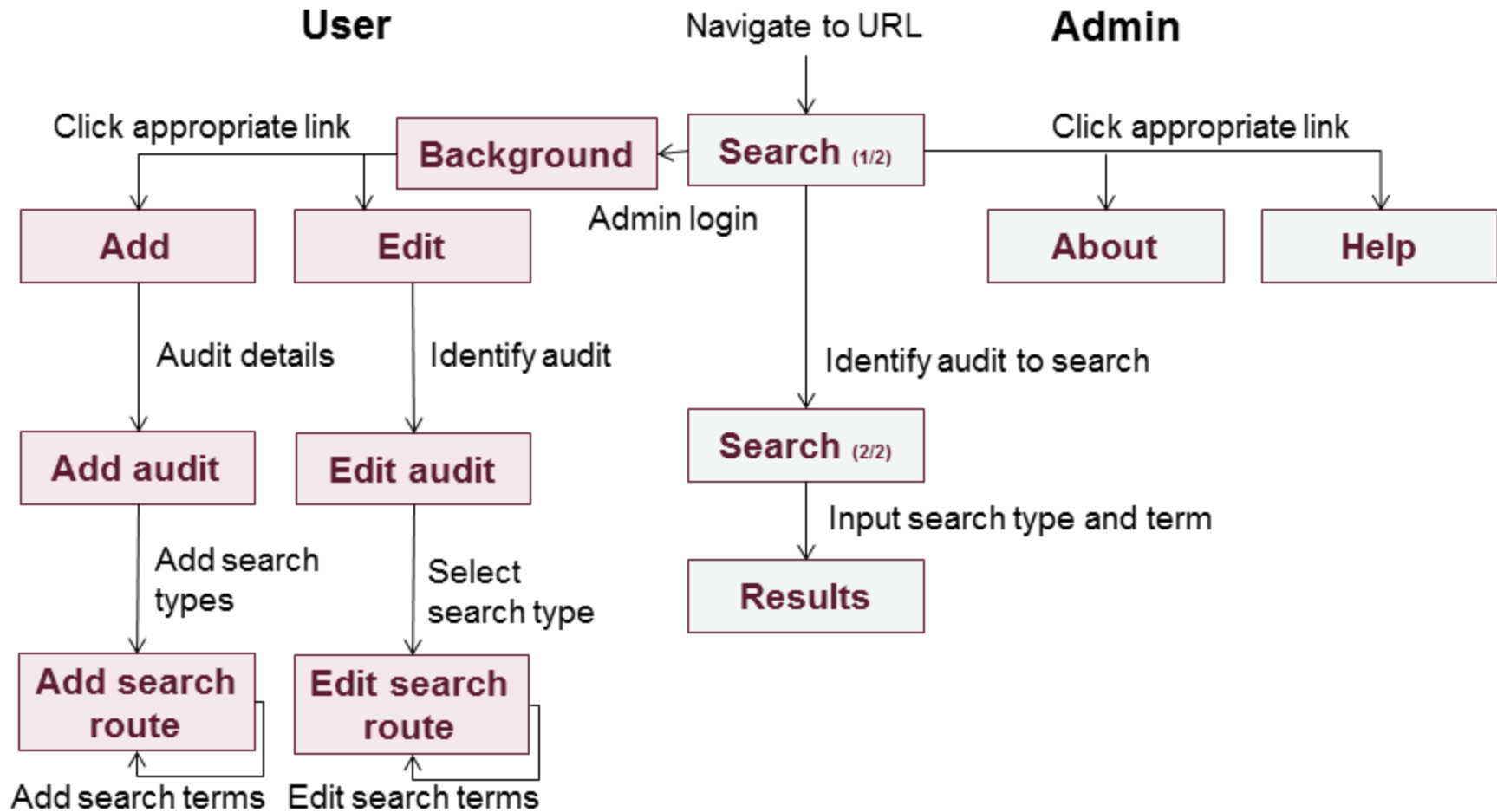
2. With special status for pre-loaded po files



ตัวอย่าง Interface Flow Diagram



Interface flow diagram



Use case diagram แผนภาพยูสเคส

แบบฝึกหัด 5.4

- ให้ศึกษารายละเอียดของระบบงานใด ๆ ที่พบเห็น เช่น การลงทะเบียน การตัดเกรด แล้วให้นำมาวิเคราะห์ความต้องการ พร้อมทั้งเขียน Use Case Model

แบบฝึกหัด 5.4

1. จงสร้างแบบจำลองยูสเคสจากรายละเอียดของระบบการเข้าพักห้องพักเริ่มจากผู้ค้นหาข้อมูลห้องว่าง ตรวจสอบสถานะของห้องว่างหรือไม่ จากเพิ่มข้อมูลห้องพัก ถ้ามีห้องว่างก็จะทำสัญญา ถ้าไม่มีห้องว่างก็ทำการจอง รับชำระเงินค่าประกันทำการบันทึกและออกใบเสร็จ จบการทำงาน เจ้าของห้องพักจะได้รับรายงานข้อมูลห้องพักและข้อมูลการเข้าพัก
2. จงสร้างแบบจำลองยูสเคสจากรายละเอียดของระบบโรงงานแห่งหนึ่งประกอบด้วยระบบการผลิต ระบบควบคุมคุณภาพ ซึ่งทำหน้าที่ในการควบคุมระบบการผลิต โดยในระบบการผลิตจะมีคนงานเป็นผู้ปฏิบัติงาน นอกจากนี้โรงงานยังมีระบบการตลาดทำหน้าที่ขายผลิตภัณฑ์ต่างๆ ของโรงงาน โดยระบบการตลาดและระบบควบคุมคุณภาพจะมีผู้จัดการทำหน้าที่จัดการ
3. จงสร้างแบบจำลองยูสเคสจากรายละเอียดของระบบการซื้อขายเงินตราต่างประเทศของสถาบันการเงิน จำแนกเป็นการซื้อขายเช็คเดินทางและการแลกเปลี่ยนเงินต่างประเทศ การแลกเปลี่ยนเงินต่างประเทศ จำแนกออกเป็น การแลกเปลี่ยน ณ เวลานั้น และการแลกเปลี่ยนล่วงหน้า
4. จงสร้างแบบจำลองยูสเคสของระบบห้องสมุดในมหาวิทยาลัยแห่งหนึ่ง ซึ่งผู้เข้าใช้บริการได้แก่ นักศึกษา อาจารย์ และพนักงานของมหาวิทยาลัย ในกระบวนการทำงาน เจ้าหน้าที่ห้องสมุดจะทำหน้าที่ให้บริการยืม/คืนหนังสือของแก่บุคคลดังกล่าว นอกจากนี้ยังจะต้องจัดการกับทรัพยากรในห้องสมุด เช่น การเพิ่ม, แก้ไข, ลบข้อมูลหนังสือและวารสาร รวมทั้งข้อมูลเกี่ยวกับสมาชิกของห้องสมุดอีกด้วย ทุกเดือน เจ้าหน้าที่ห้องสมุดจะต้องทำรายงานต่างๆ ส่งให้กับผู้อำนวยการศูนย์สารสนเทศ เช่น รายงานจำนวนสมาชิก, รายงานจำนวนหนังสือและวารสารใหม่, รายงานการยืม/คืนหนังสือ, รายงานค่าปรับล่าช้า เป็นต้น

อ้างอิง

วุฒิพงษ์ เรือนทอง

ภาควิชา วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

คณะวิทยาศาสตร์ มหาวิทยาลัยนเรศวร

www.uml.org

www.siam2dev.com