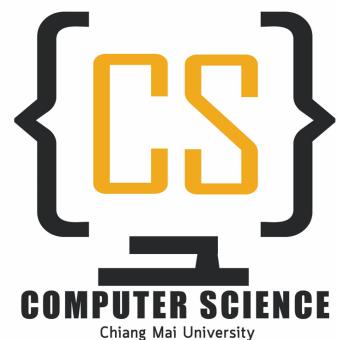


# CS204362 – Object-Oriented Design

## L8: Developing Use Case Diagrams and Models

Kamonphop Srisopha

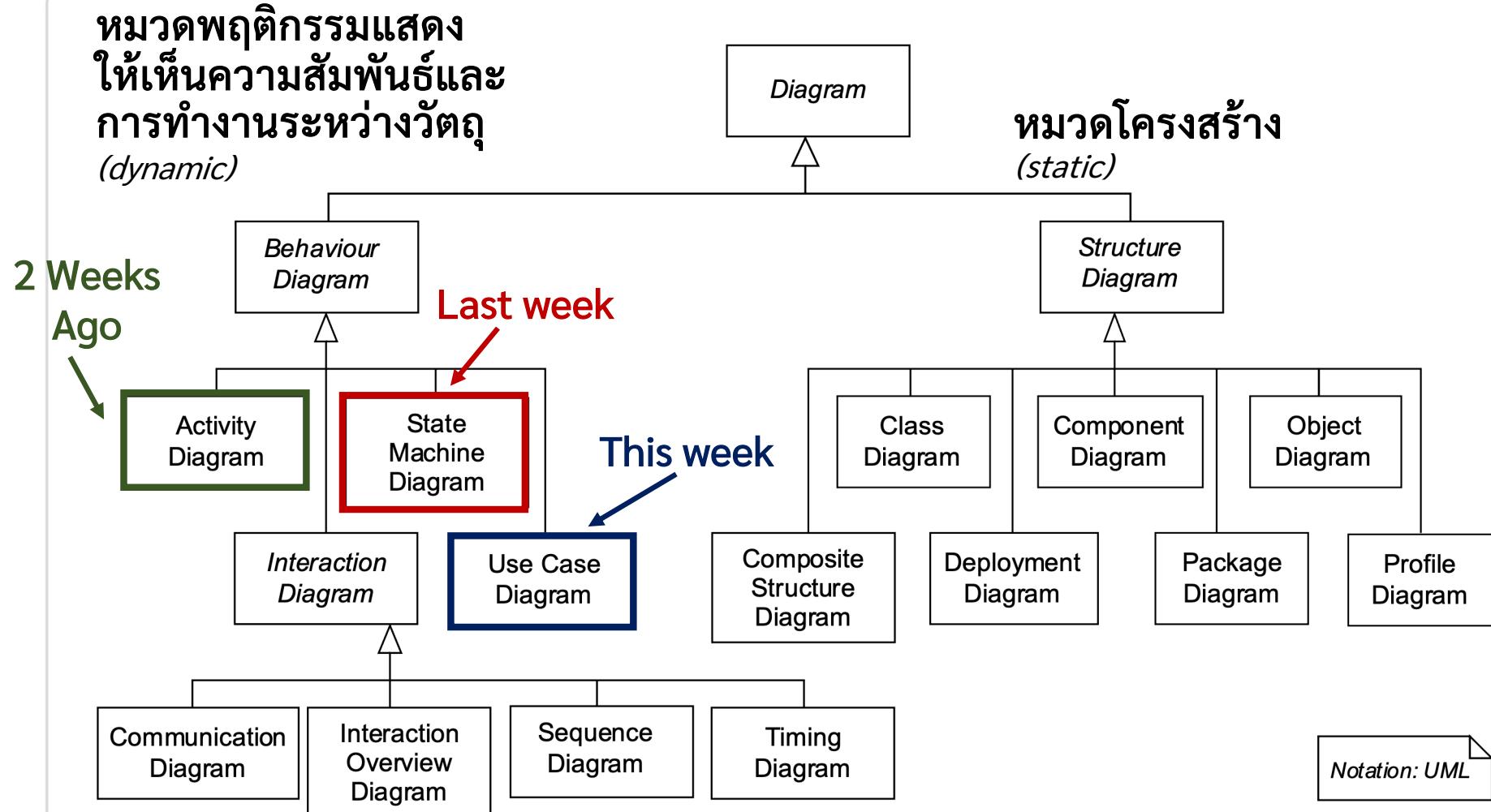


Faculty of Science, Chiang Mai University  
คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

# Agenda

- Type of Requirements
- Use Cases
- **Use Case Specification**
- Prototyping

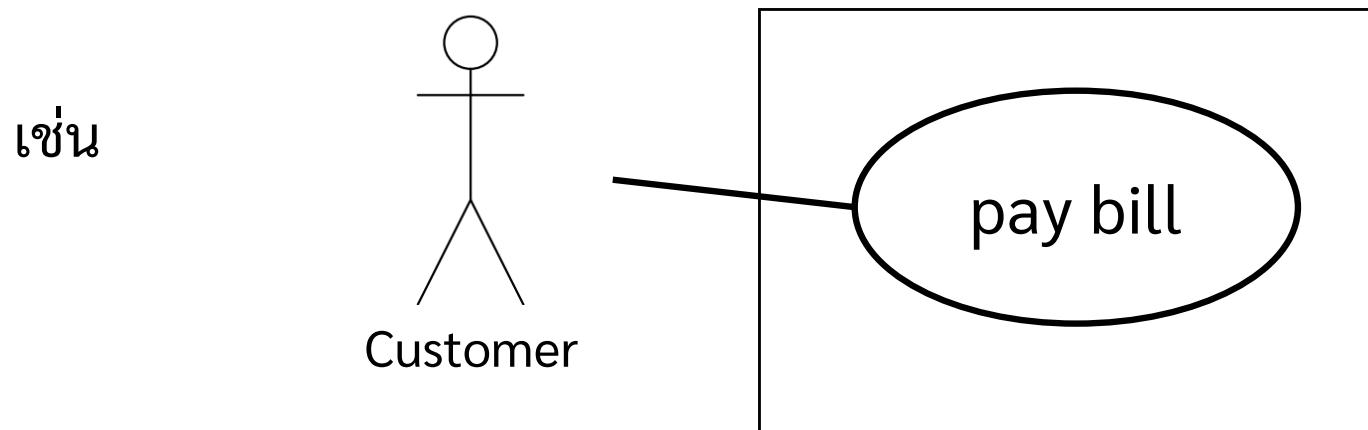
# UML Diagrams



# Use Case Specification

# A Use Case Diagram Is Not Enough

เนื่องจาก use case กล่าวถึง sequence of actions เพื่อให้บรรลุเป้าหมาย หรือวัตถุประสงค์ diagram อย่างเดียวจึงไม่พอที่จะอธิบายมันได้ เพราะฉะนั้น แต่ละ use case ต้องมี description/specification เพื่อที่จะอธิบาย จุดประสงค์และ function การทำงานของระบบต่างๆที่เกี่ยวข้องกับ use case นั้นๆเมื่อมันถูกใช้งาน



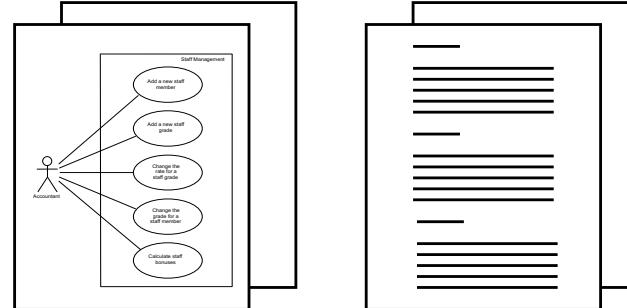
มีทางไหนบ้างที่จะบรรลุเป้าหมายของการจ่ายบิลได้?  
มีทางไหนบ้างที่ไม่สามารถเข้าถึงเป้าหมายได้?

# Development of the Use Case Model Through Successive Iterations

## Iteration 1

Obvious use cases.

**Simple use case descriptions.**

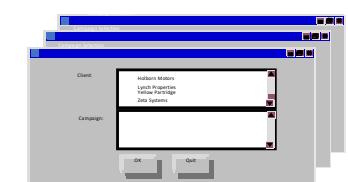
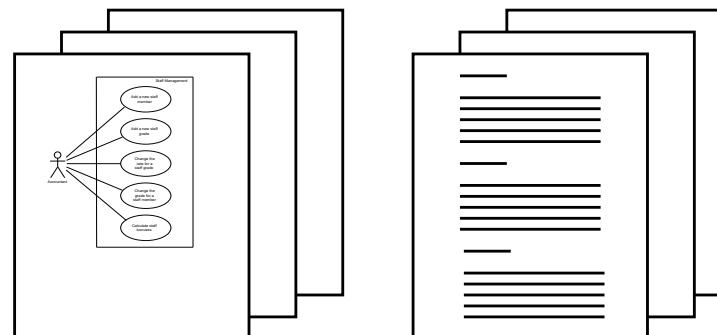


## Iteration 2

Additional use cases.

**Simple use case descriptions.**

Prototypes.

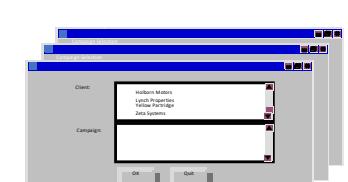
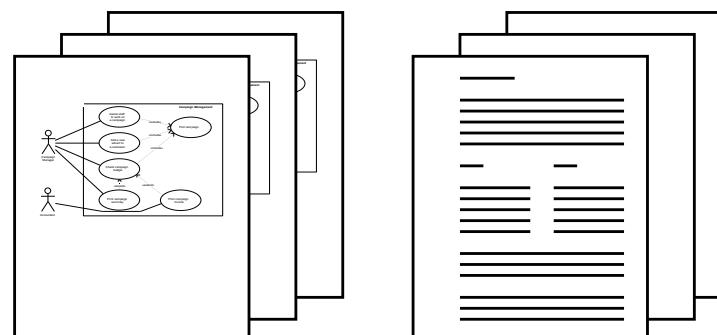


## Iteration 3

Structured use cases.

**Structured use case descriptions.**

Prototypes.



## Iteration n

# The “Guts” of Use Cases

We can now supply the details of a use case

- Define Use Cases (*กล่าวถึงก่อนหน้านี้*)
- Give a simple sentence to describe each use case
- Give a detailed narration of what the use case does, in paragraph form
- Describe the main course scenario/process (main success scenario, main flow)
- Set pre and post conditions
- Identify, then describe alternate scenarios with pose and pre conditions
- List any additional requirements that must be met

# Example

- Using a simple paragraph

- ***Assign staff to work on a campaign***

The campaign manager wishes to record which staff are working on a particular campaign. This information is used to validate timesheets and to calculate staff year-end bonuses.

- Using a step-by-step breakdown of interaction between actor and system

Actor Action	System Response
1. The actor enters the client name.	2. Lists all campaigns for that client.
3. Selects the relevant campaign.	4. Displays a list of all staff members not already allocated to this campaign.
5. Highlights the staff members to be assigned to this campaign.	6. Presents a message confirming that staff have been allocated.
<b>Alternative Courses</b>	
Steps 1–3. The actor knows the campaign name and enters it directly.	

# Anatomy of Use Case Specification

- Actor
- Description
- Pre-condition
- Post-condition
- Flow of events
- Alternate flow of events

# Use Case Specification Example

<b>Use case ID</b>	UC01
<b>Use case Name</b>	Create Assignment
<b>Actor/User</b>	Instructor
<b>Description</b>	Create assignment and their questions for each subject in each semester.
<b>Pre-condition</b>	-
<b>Post-condition</b>	Assignment and question information are stored in the Assignment database.
<b>Flow of events</b>	<ol style="list-style-type: none"><li>1 User inputs assignment data: assignment number, assignment title, number of questions, practice database name, assignment start date, and assignment due date.</li><li>2 The assignment data are validated as follows:<ul style="list-style-type: none"><li>• Length of assignment title must not exceed 80 characters.</li><li>• Length of practice database name must not exceed 30 characters.</li><li>• Assignment number and number of questions must be positive integers.</li></ul></li><li>3 User adds question data until all question data of the assignment are added to the Assignment database as follows:<ol style="list-style-type: none"><li>3.1 Question number is automatic generated by the system starting from 1.</li><li>3.2 User inputs other question data: question description, complication level, SQL statement solution, and question score.</li><li>3.3 The question data are validated as follows:<ul style="list-style-type: none"><li>• Length of question description must not exceed 256 characters.</li><li>• Question score must be positive integer.</li></ul></li><li>3.4 Increasing question number by 1.</li></ol></li><li>...</li></ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"><li>2.1 If there are any invalid data, the system displays error message and forces user to re-enter assignment data.<ol style="list-style-type: none"><li>3.3.1 If there are any invalid data, the system displays error message and forces user to re-enter question data.</li></ol></li></ol>

# Example: ATM System

Use case: Withdraw Cash (ถอนเงิน)

Pre Condition(s): ?

Post Condition(s): ?

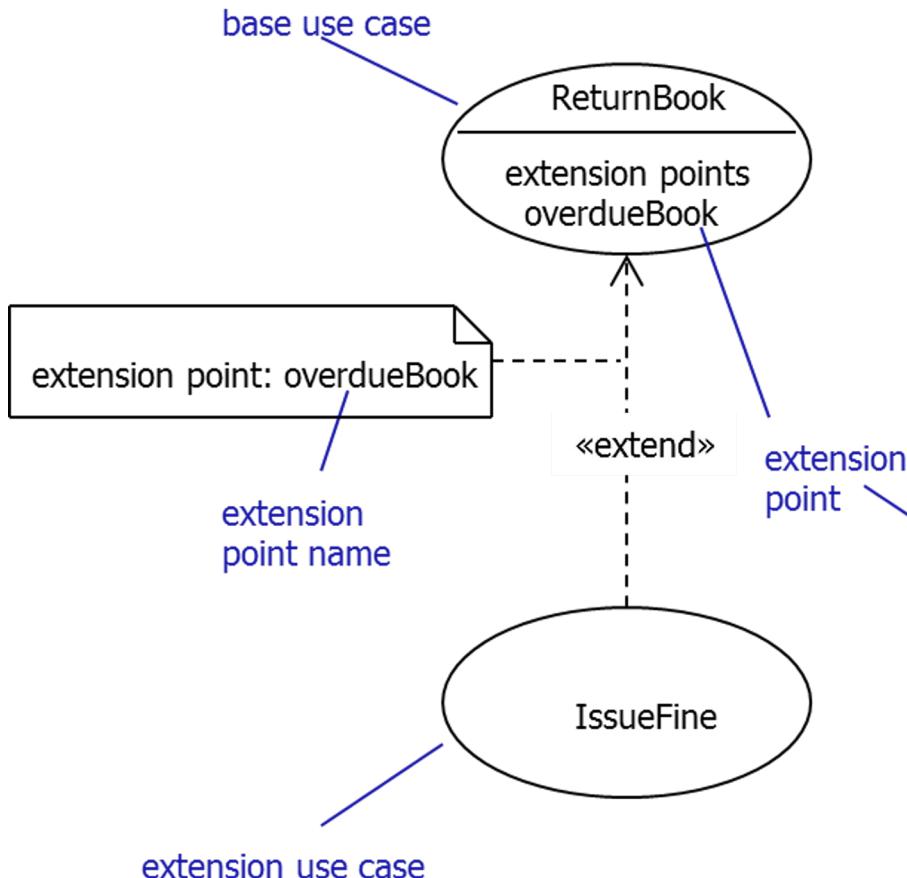
Normal Path: ?

Alternative Path(s): ?

Non-Functional Requirements: ?

# Example: ATM System

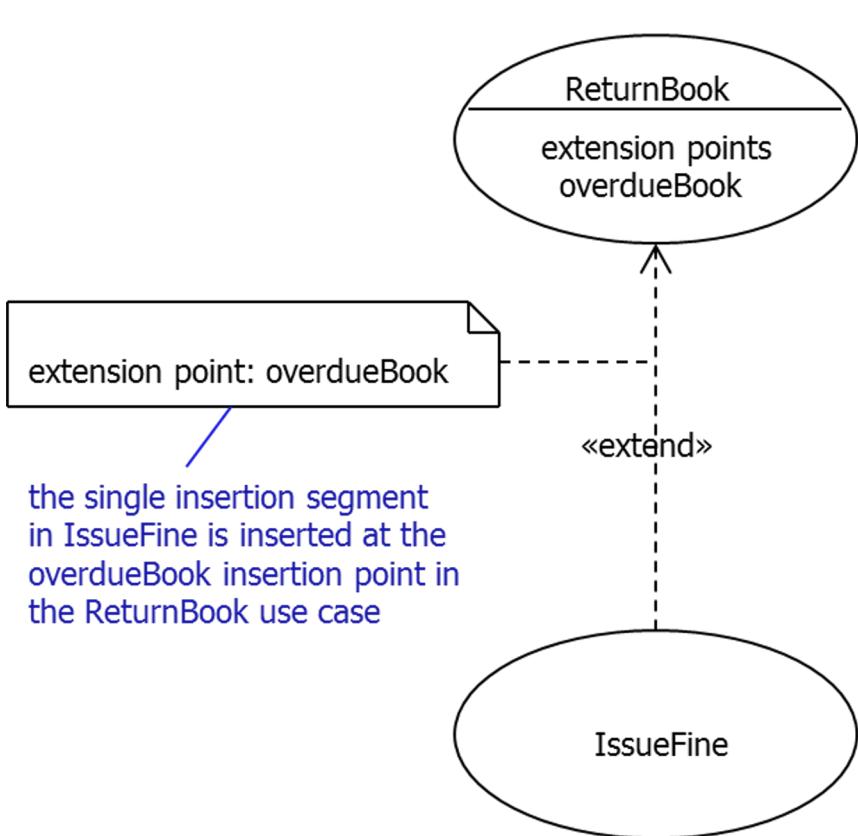
# Base use case



Use case: ReturnBook
ID: 9
Brief description: The Librarian returns a borrowed book.
Primary actors: Librarian
Secondary actors: None.
Preconditions: 1. The Librarian is logged on to the system.
Main flow: 1. The Librarian enters the borrower's ID number. 2. The system displays the borrower's details including the list of borrowed books. 3. The Librarian finds the book to be returned in the list of books. extension point: overdueBook 4. The Librarian returns the book. ...
Postconditions: 1. The book has been returned.
Alternative flows: None.

- There is an extension point **overdueBook** just before step 4 of the flow of events
- Extension points are *not* numbered, as they are *not* part of the flow

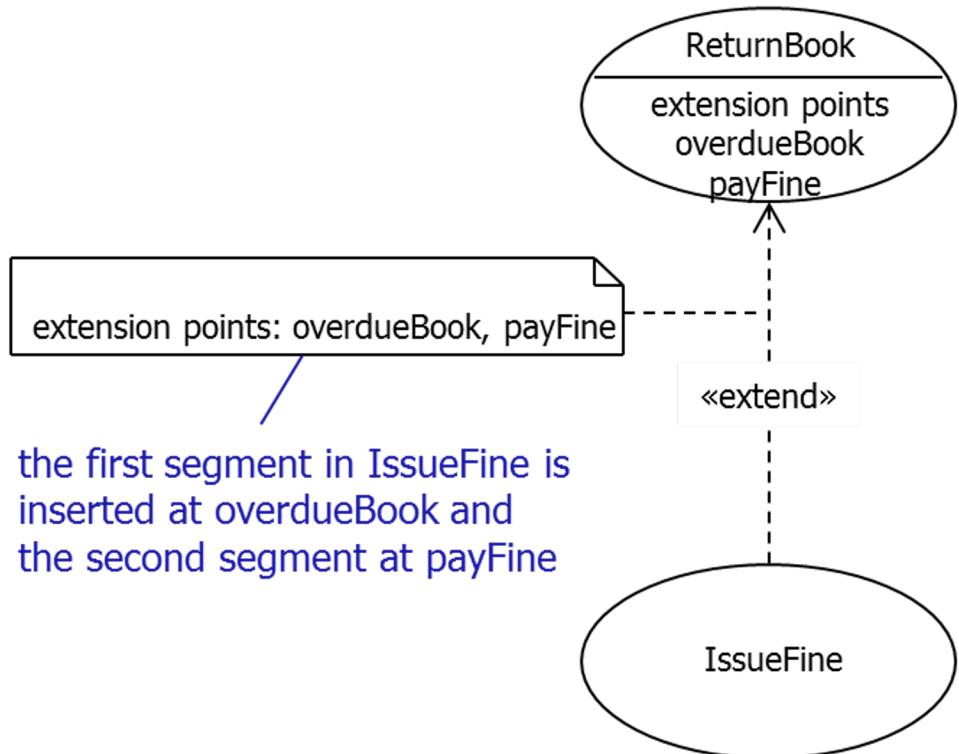
# Extension use case



Extension Use case: IssueFine	
ID: 10	
Brief description:	Segment 1: The Librarian records and prints out a fine.
Primary actors:	Librarian
Secondary actors:	None.
Segment 1 preconditions:	
1. The returned book is overdue.	
Segment 1 flow:	
1. The Librarian enters details of the fine into the system. 2. The system prints out the fine.	
Segment 1 postconditions:	
1. The fine has been recorded in the system. 2. The system has printed out the fine.	

- Extension use cases have one or more *insertion segments* which are behaviour fragments that will be inserted at the specified extension points in the base use case

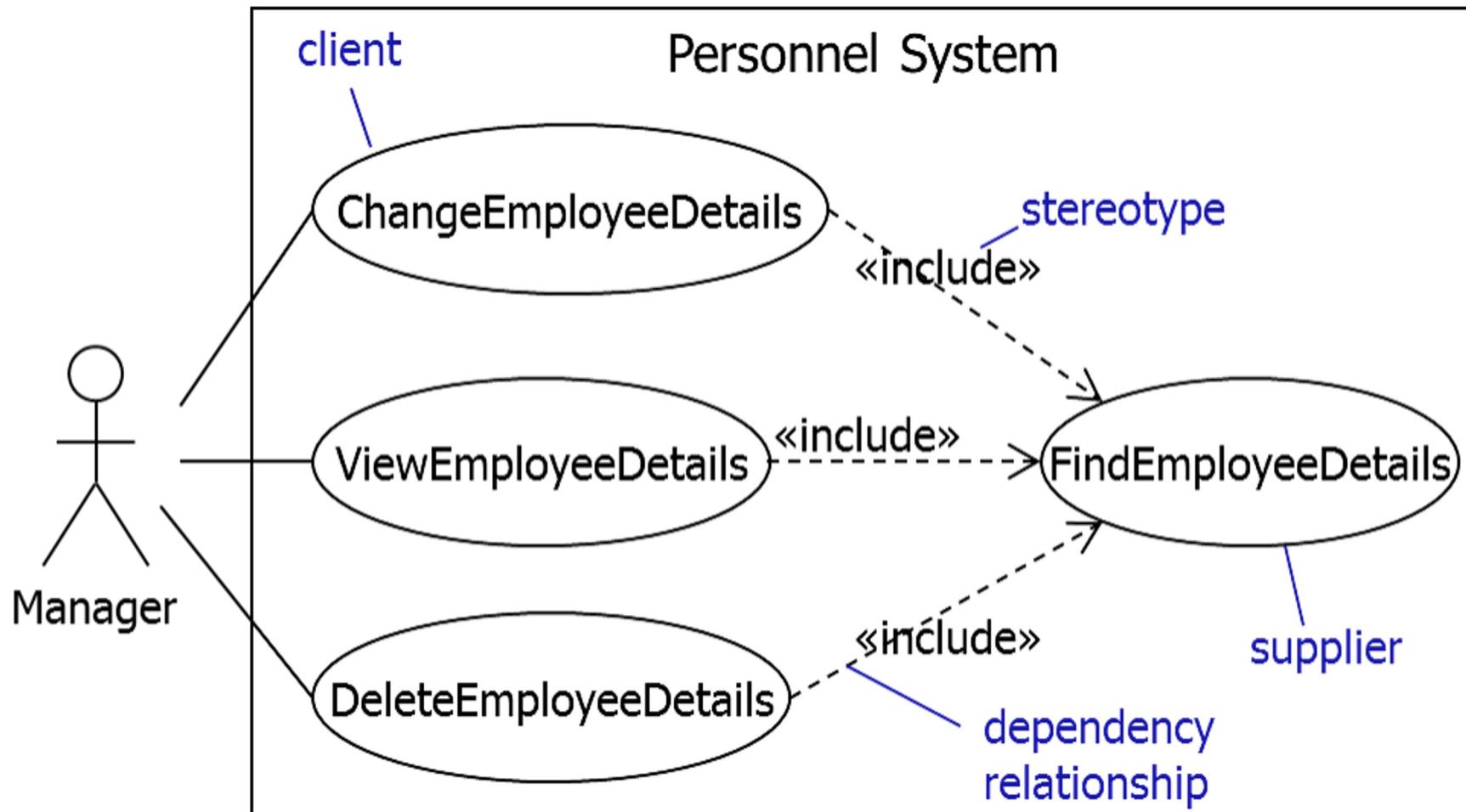
# Multiple insertion points



- If more than one extension point is specified in the «extend» relationship then the extension use case must have the *same number* of insertion segments

Extension Use case: IssueFine	
ID: 10	
Brief description:	Segment 1: The Librarian records and prints out a fine. Segment 2: The Librarian accepts payment for a fine.
Primary actors:	Librarian
Secondary actors:	None.
Segment 1 preconditions:	1. The returned book is overdue.
Segment 1 flow:	1. The Librarian enters details of the fine into the system. 2. The system prints out the fine.
Segment 1 postconditions:	1. The fine has been recorded in the system. 2. The system has printed out the fine.
Segment 2 preconditions:	1. A fine is due from the borrower.
Segment 2 flow:	1. The Librarian accepts payment for the fine from the borrower. 2. The Librarian enters the paid fine in the system. 3. The system prints out a receipt for the paid fine.
Segment 2 postconditions:	1. The fine is recorded as paid. 2. The system has printed a receipt for the fine.

# Use Case Specification for Includes



# «include» Example

<p>Use case: ChangeEmployeeDetails</p> <p>ID: 1</p> <p>Brief description: The Manager changes the employee details.</p> <p>Primary actors: Manager</p> <p>Secondary actors: None</p> <p>Preconditions: 1. The Manager is logged on to the system.</p> <p>Main flow: 1. <b>include( FindEmployeeDetails ).</b> 2. The system displays the employee details. 3. The Manager changes the employee details. ...</p> <p>Postconditions: 1. The employee details have been changed.</p> <p>Alternative flows: None.</p>	<p>Use case: FindEmployeeDetails</p> <p>ID: 4</p> <p>Brief description: The Manager finds the employee details.</p> <p>Primary actors: Manager</p> <p>Secondary actors: None</p> <p>Preconditions: 1. The Manager is logged on to the system.</p> <p>Main flow: 1. The Manager enters the employee's ID. 2. The system finds the employee details.</p> <p>Postconditions: 1. The system has found the employee details.</p> <p>Alternative flows: None.</p>
---	---

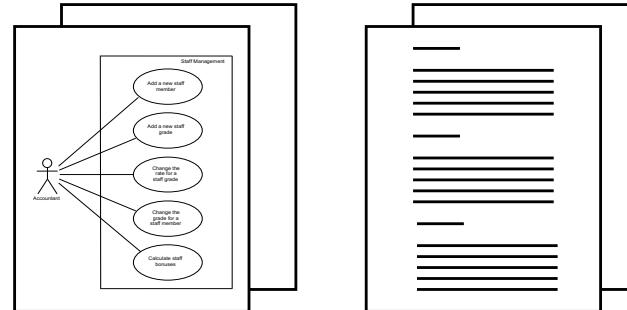
# Prototype

# Development of the Use Case Model Through Successive Iterations

## Iteration 1

Obvious use cases.

Simple use case descriptions.

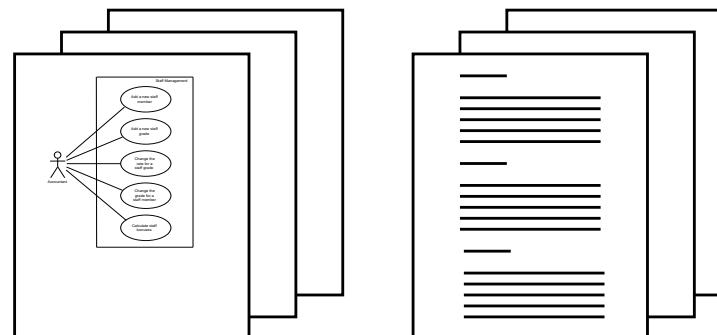


## Iteration 2

Additional use cases.

Simple use case descriptions.

**Prototypes.**

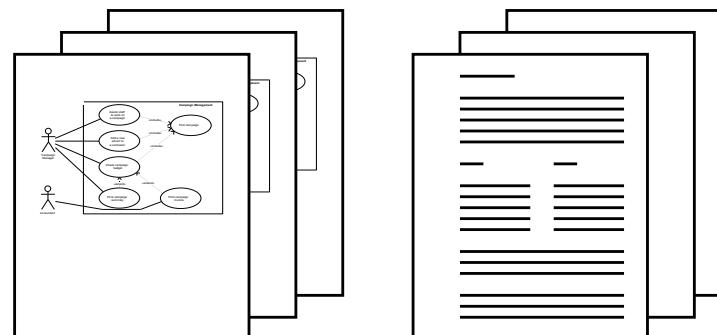


## Iteration 3

Structured use cases.

Structured use case descriptions.

**Prototypes.**

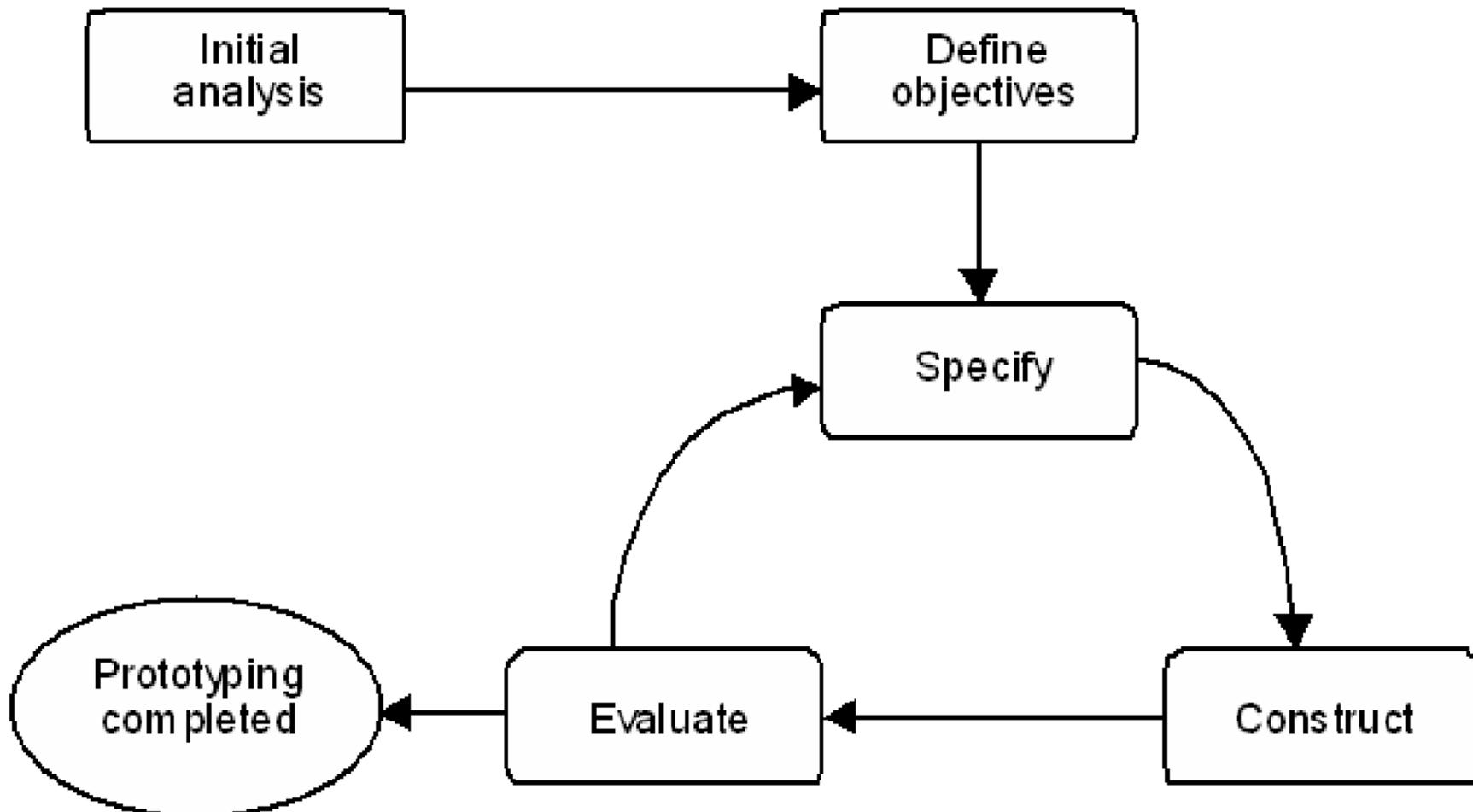


## Iteration n

# Prototype

- A prototype is a system or a partially complete system that is **built quickly to explore some aspects** of the system requirements and that is not intended as the final working system.
- Prototyping can be used to **support use case modeling**
- Help **elicit functional requirements**
- Test out system architectures based on the use cases in order to **meet the non-functional requirements**

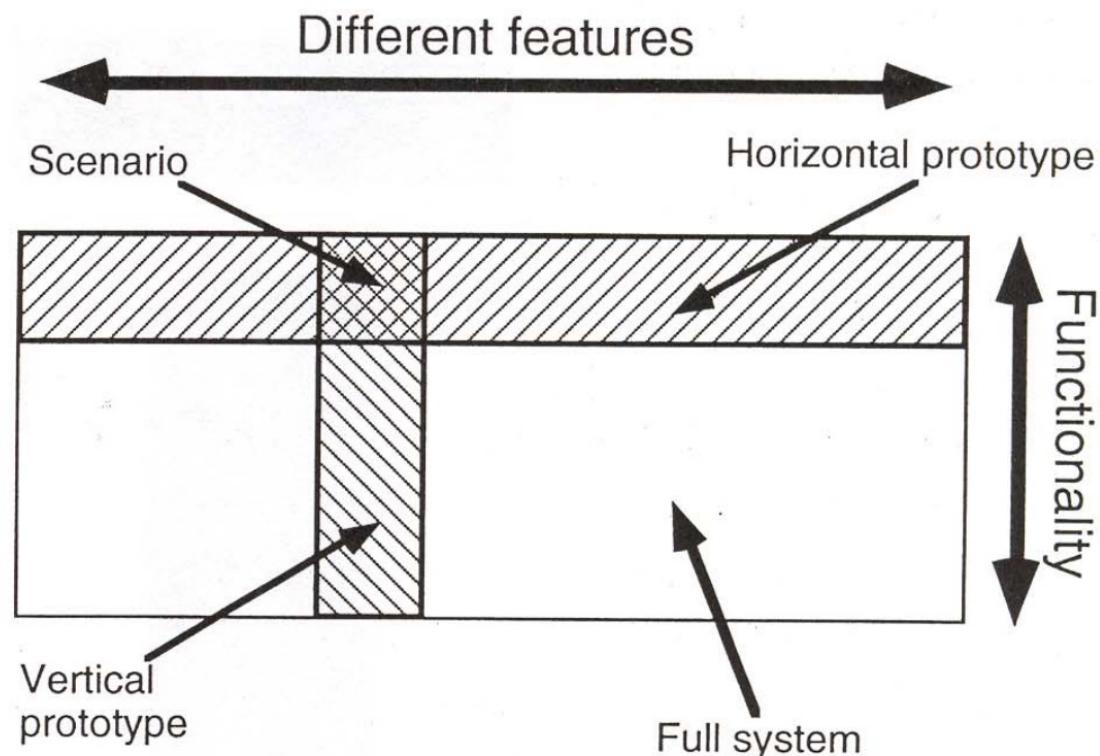
# Prototyping Workflow



# Dimension of Prototypes

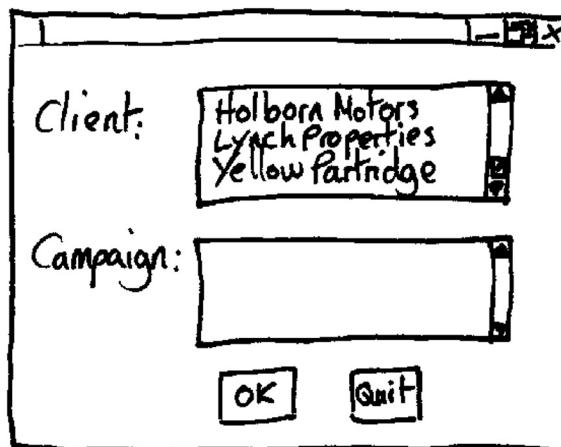
**Horizontal** - broad view of the entire system, focusing on user interaction, but no underlying functionality.

**Vertical** - complete elaboration of a single subsystem or function (in-depth functionality, but only for selected few features)

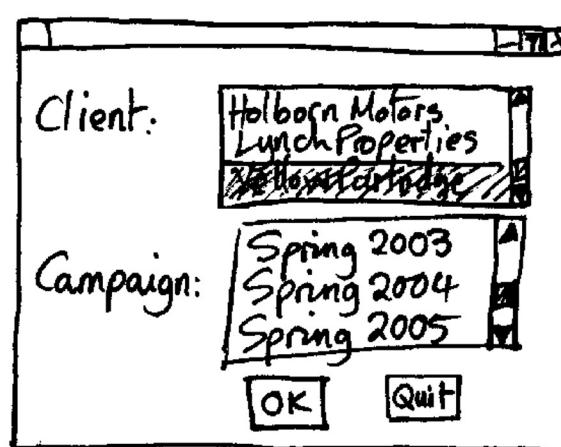


# User Interface Prototyping

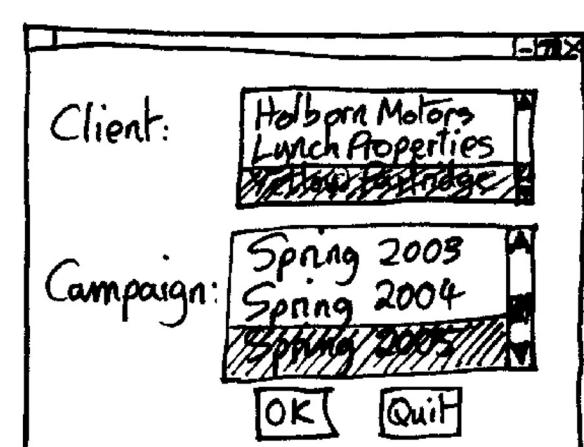
- Horizontal Prototyping
- For visualizing and validating requirements/**flows of use cases**
- นิยมจำลองการใช้งานจริง เช่น การกรอกข้อมูล การเรียกใช้ข้อมูล แต่ใช้งานจริงไม่ได้ หรือการแสดง flow/step ของการทำงานจากหน้าหนึ่งไปหน้าหนึ่ง



Dialogue initialized.



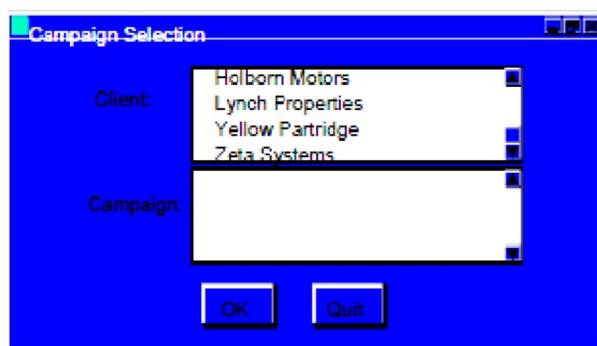
User selects Client. Campaigns listed.



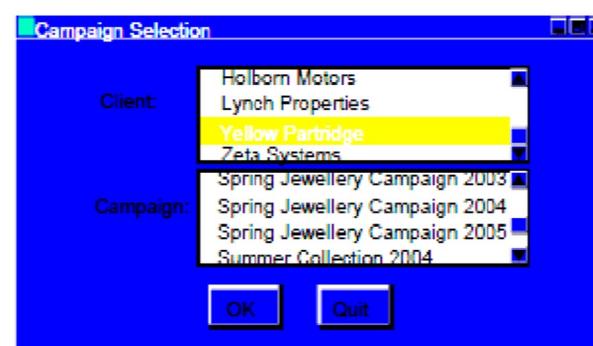
User selects Campaign.

# User Interface Prototyping

UI prototype สามารถพัฒนาขึ้นมาจากการภาษาที่ระบบที่จะพัฒนา  
ไม่ได้ใช้ก็ได้



Dialogue initialized.



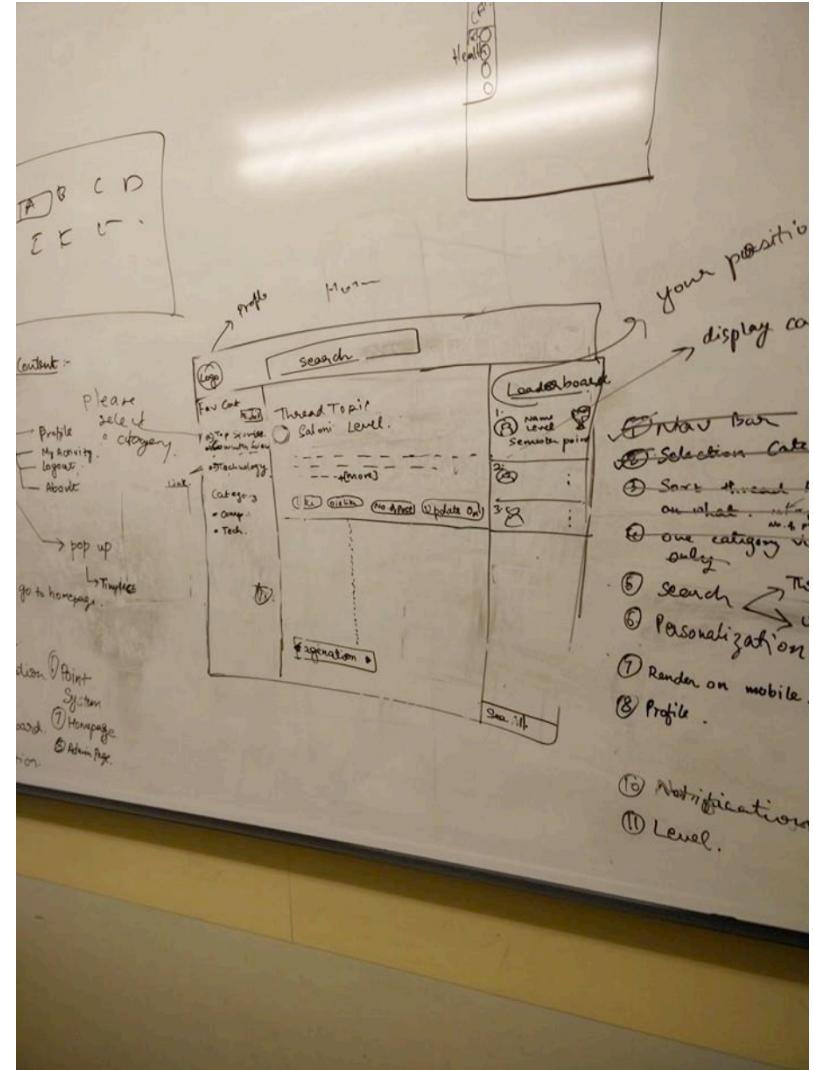
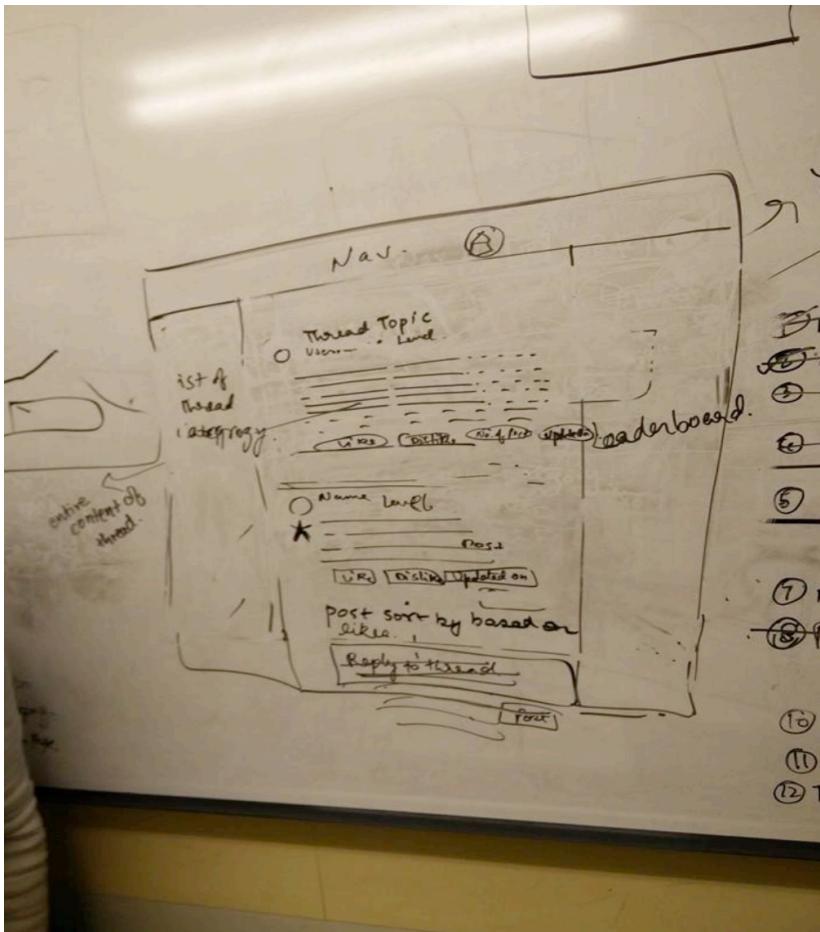
User selects Client.  
Campaigns listed.



User selects Campaign.

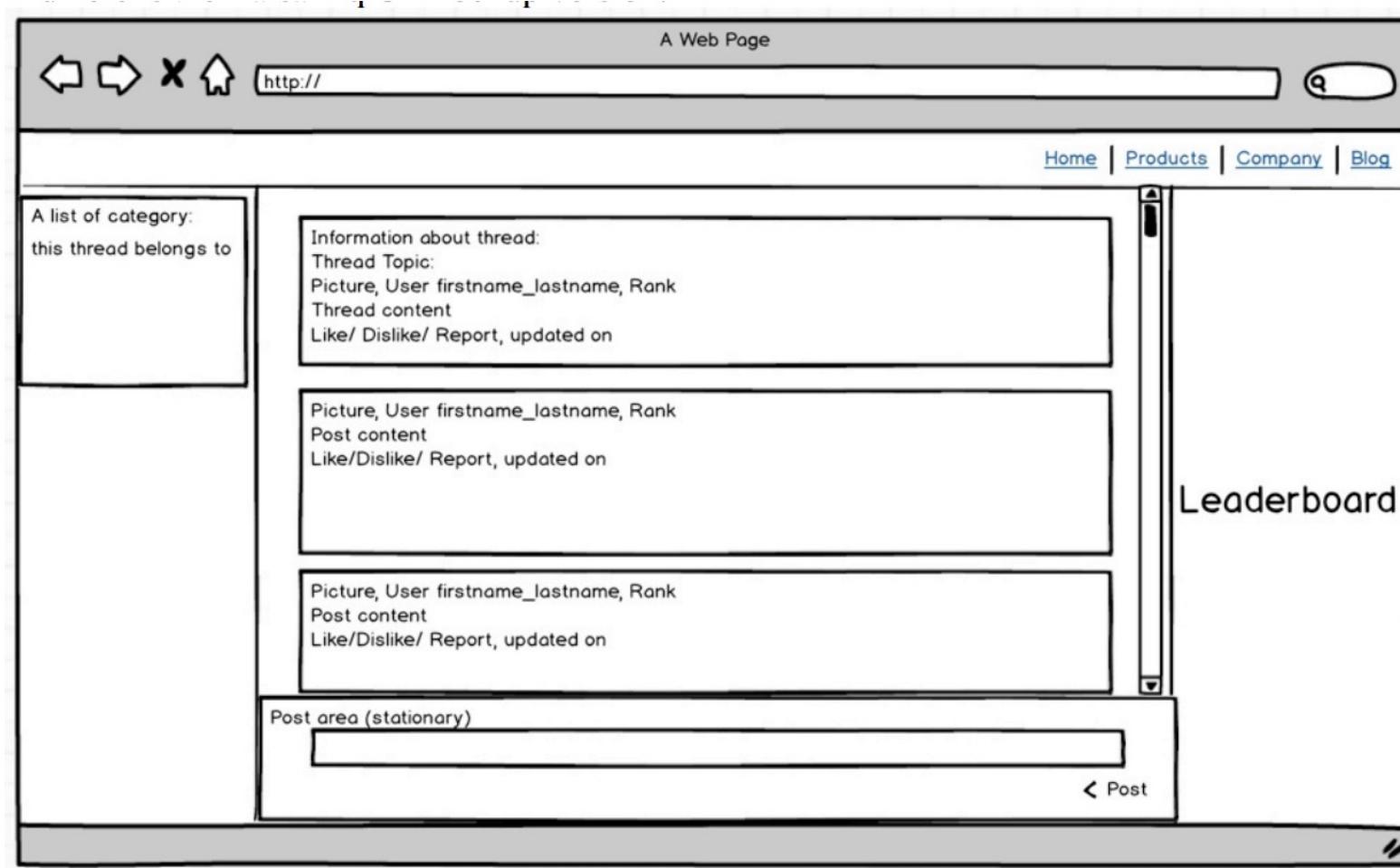
# User Interface Prototyping

## Paper (pen) prototyping



# User Interface Prototyping

## The Mock-up



# User Interface Prototyping

## The Finished Product

The screenshot shows a user interface for a forum or Q&A platform. At the top, there is a navigation bar with the logo "We Are Trojans", a search bar, and links for "Home", "Punyawee", "Event", "Notification" (with a red badge), and "Add a thread". On the left, a sidebar titled "TOPIC FEEDS" lists various categories: All Threads (selected), Hot Threads, Education, Technology, Art, General, Tips and Tricks, Business, Engineering, and Traveling. In the center, there is a section titled "★ Hot Threads" containing several threads:

- Is Snape in love with Lily, or is he just obsessed with her?** by Punyawee Pakdiying (about a month ago). The post content reads: "I know it is described as love in the books, but is it really? Frankly it seems more obsessive than caring to me. This may be because I have had experience with a guy who was defiantly obsessed with me". It has 4 likes and 0 dislikes.
- Hello world!!!!** by Punyawee Pakdiying (15 minutes ago). The post content reads: "testing testing testing testing testing testing". It has 3 likes and 0 dislikes.
- Does Google hire people with a Ph.D in humanities?** by Punyawee Pakdiying (about a month ago). The post content reads: "I would like to know whether Google hire people with a Ph.D in humanities or not. Thank you in advance,". It has 5 likes and 1 dislike.
- Tennis league** (partially visible at the bottom).

On the right side, there is a "LEADERBOARD Top#10" section showing the top 10 users based on Semester Points:

Rank	User	Semester Points
1	Punyawee Pakdiying	1034
2	Linda Suen	25000
3	Suleyman Erten	2273
4	Linda Suen	2004
5	Bowen Wang	1252
6	Punyawee Pakdiying	1034
7	Pittawat Pamornchaisirikij	624
8	Li Min	450
9	Eirik Skogstad	442
9	Kamonphop Srisopha	370

# Advantages and Disadvantages



- ช่วยให้เห็นว่า developer และผู้ว่าจ้างเข้าใจตรงกันหรือไม่
- ช่วยให้เห็นความต้องการซัดเจนขึ้น
- ช่วยระบุความต้องการของผู้จ้างที่หายไปได้
- ช่วยลดอัตราความเสี่ยงต่างๆ ของ การพัฒนาระบบ (ทดสอบความเป็นไปได้)
- ช่วยในการสื่อสารระหว่าง stakeholders



- ผู้ว่าจ้างหรือผู้ใช้มันจะเข้าใจว่า prototype เสร็จแล้ว
- มักจะไม่คำนึงถึง non-functional aspects
- มักจะต้องการส่วนร่วมจากผู้ว่าจ้างหรือผู้ใช้อย่างมาก
- prototype อาจจะทำให้เกิด expectation ที่เกินความจริง (เช่น prototype เร็วของจริงช้า)

# Summary

- We have learned about techniques for:
- Use case Modeling
- Advanced use case modeling:
  - Actor generalization
  - Use case generalization
  - «include»
  - «extend»
- Use advanced features with discretion only where they simplify the model!