# Diagrams

Chonnam National University
School of Electronics and Computer Engineering

Kyungbaek Kim

# Contents

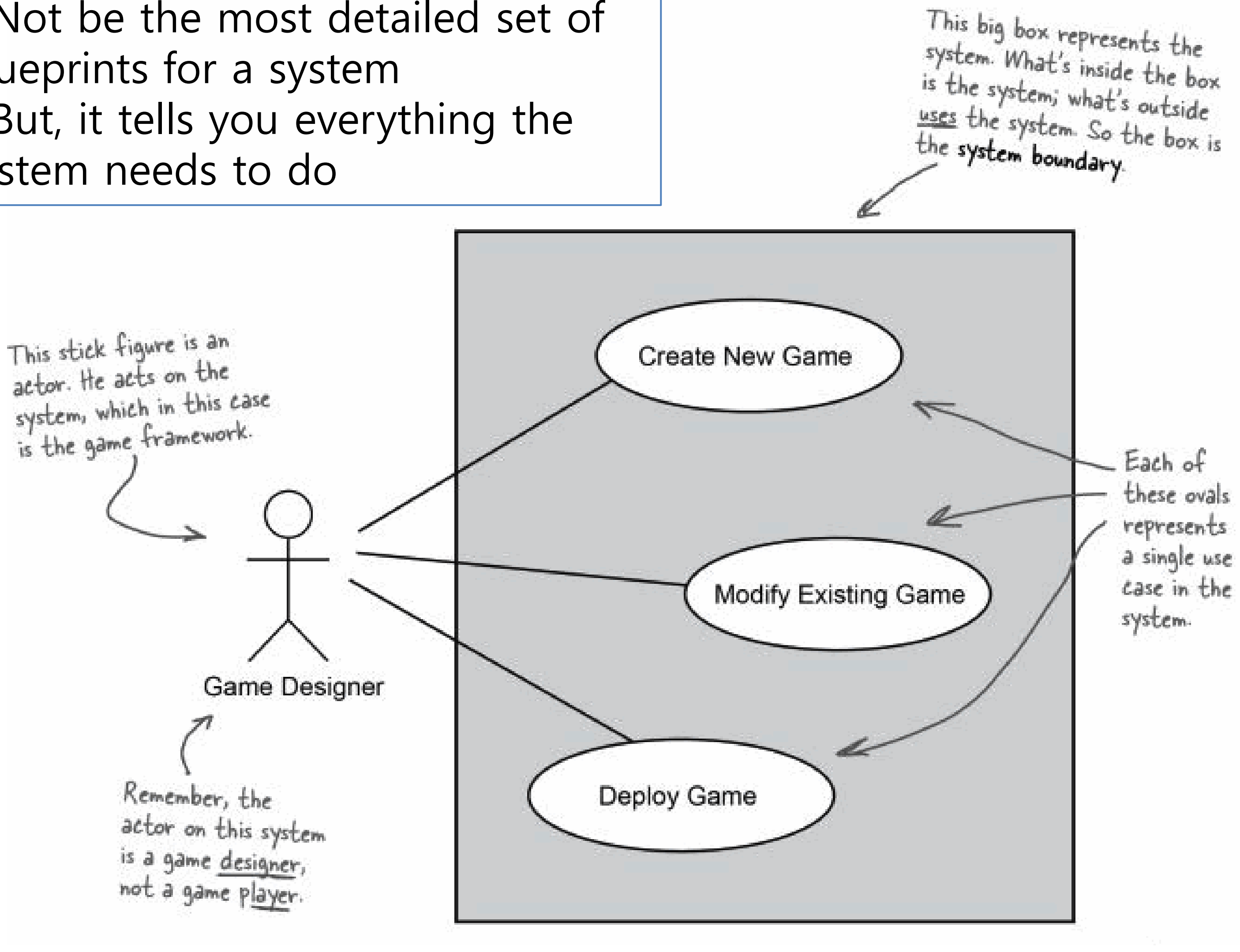
- Usecase Diagram
- Sequence Diagram
- GUI sketch

# Usecase and Usecase Diagram

- Usecase
  - A case or way of using a module or method
- Usecase Diagram
  - A set of modules or methods of a system
  - The baseline of functional requirements of a system
  - Usually generated at the start up period of a project
  - Usually becomes a basis of other diagrams

# Usecase Diagram

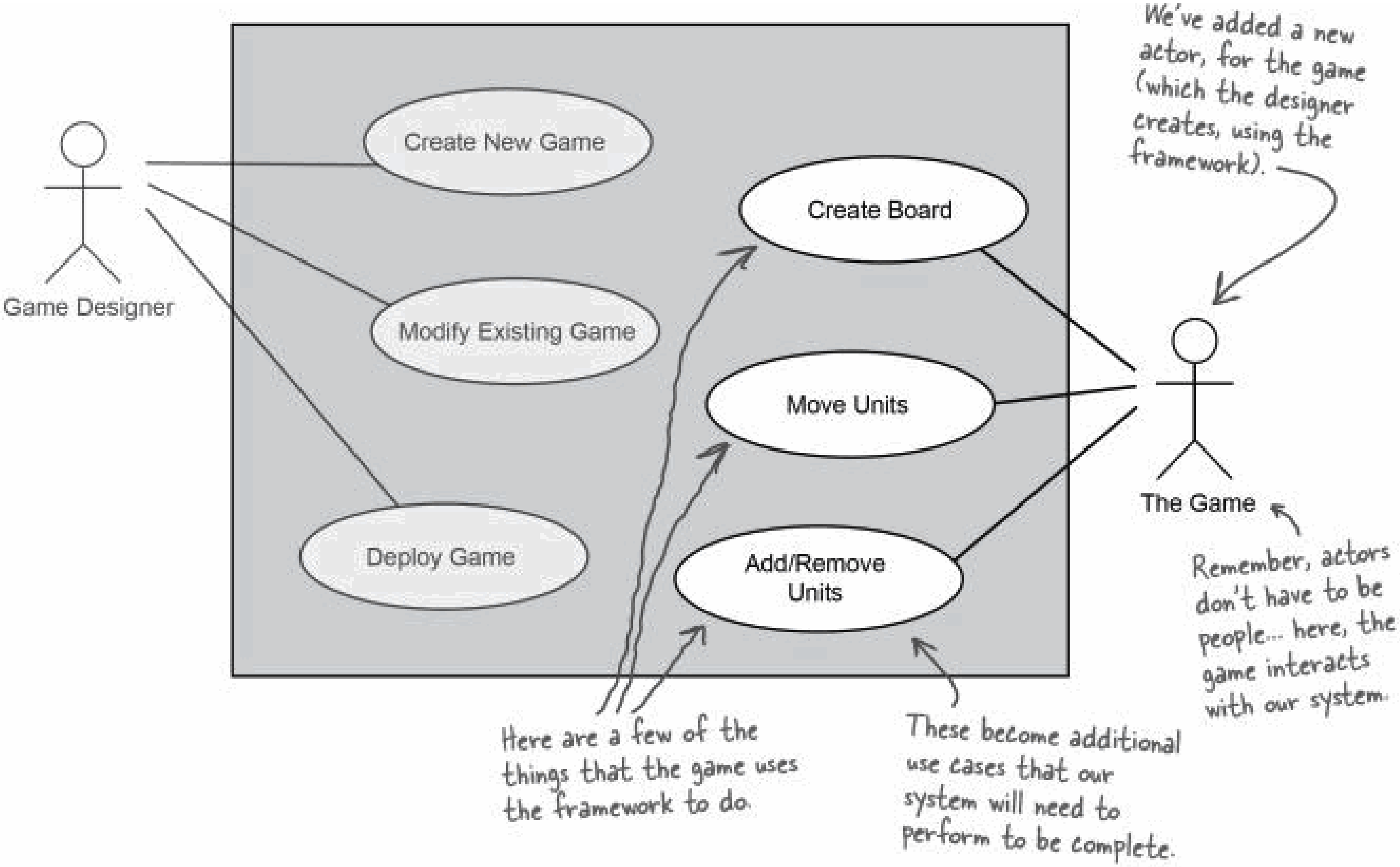
* **Blueprint of your system**
- Not be the most detailed set of blueprints for a system
- But, it tells you everything the system needs to do


This big box represents the system. What's inside the box is the system; what's outside uses the system. So the box is the system boundary.
This stick figure is an actor. He acts on the system, which in this case is the game framework.
Create New Game
Modify Existing Game
Deploy Game
Each of these ovals represents a single use case in the system.
Game Designer
Remember, the actor on this system is a game designer, not a game player.

# Actor

- Actor locates outside of a system and interact with the system.
- Types of Actor
  - Users of a system
  - Other systems interacting with a system
- Naming of Actor
  - Focus on the **Role**

# Actors are people, not always


Game Designer
Create New Game
Modify Existing Game
Deploy Game
Create Board
Move Units
Add/Remove Units
The Game

We've added a new actor, for the game (which the designer creates, using the framework).

Remember, actors don't have to be people... here, the game interacts with our system.

Here are a few of the things that the game uses the framework to do.

These become additional use cases that our system will need to perform to be complete.

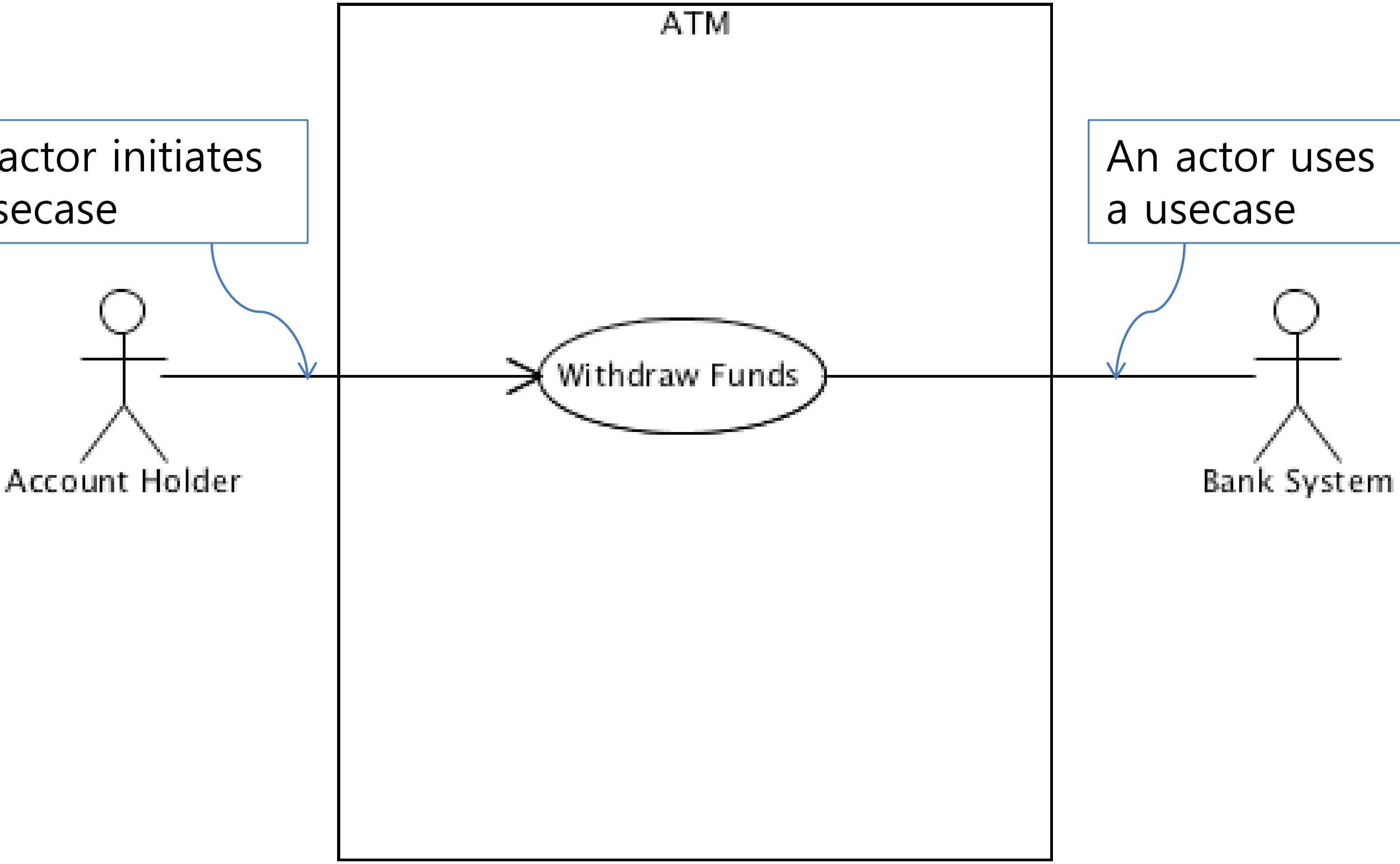
# How to identify actors

- Use the following questions
  - Who use the functions of a system?
  - Which needs the resources of a system?
  - Who manages a system?
  - Which hardware is required?
  - Which other systems are required?
  - Which is interesting of the output of a system?

# How to identify usecases

- Use the following questions
  - What is the main functionality of a system?
  - Which information is modified (store, remove, search…)?
  - Which events are requested from an actor to a system, or vise versa?
  - Which input/output is used by a system?

# Communicates


ATM
Withdraw Funds
Account Holder
Bank System

An actor initiates a usecase

An actor uses a usecase

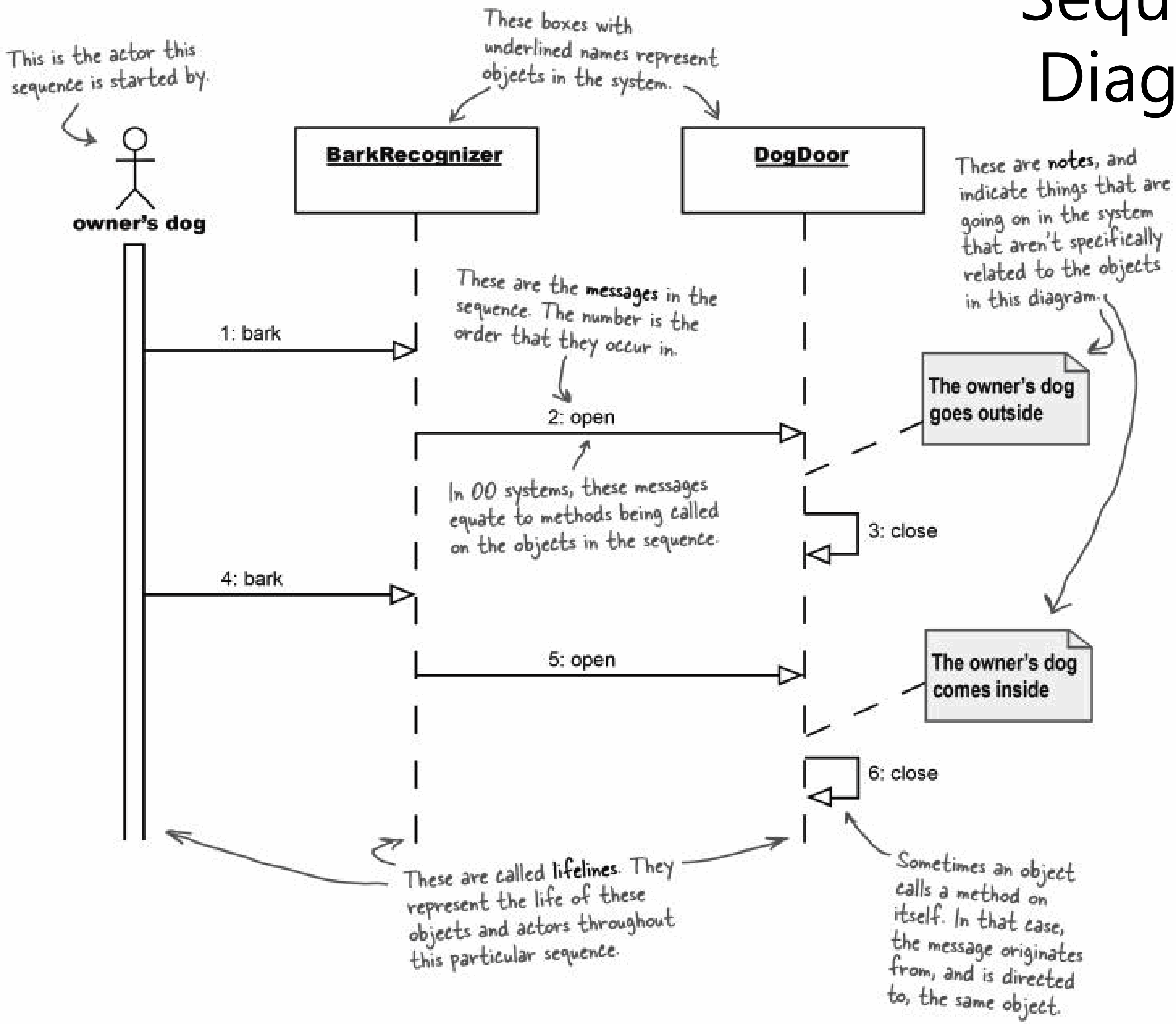
# Sequence of drawing a usecase diagram

- Identify Actors
- Identify Usecases
  - Every usecase should interact at least one actor
  - Granularity of usecases should be similar
- Define Relationships
  - Between actors → generalization
  - Between actors and usecases → communicates
  - Between usecases → include, extend
- Factoring Usecases

# Sequence Diagram

- A visual way to show the things that happen in particular interaction between an actor and your system
  - Focus on the timing sequence and the messages
  - Dynamic modeling
- Realization of a usecase diagram
  - Define operations and properties of objects of a system

# Sequence Diagram



owner's dog
BarkRecognizer
DogDoor
1: bark
2: open
3: close
4: bark
5: open
6: close
The owner's dog goes outside
The owner's dog comes inside

This is the actor this sequence is started by.

These boxes with underlined names represent objects in the system.

These are **notes**, and indicate things that are going on in the system that aren't specifically related to the objects in this diagram.

These are the **messages** in the sequence. The number is the order that they occur in.

In OO systems, these messages equate to methods being called on the objects in the sequence.

These are called **lifelines**. They represent the life of these objects and actors throughout this particular sequence.

Sometimes an object calls a method on itself. In that case, the message originates from, and is directed to, the same object.

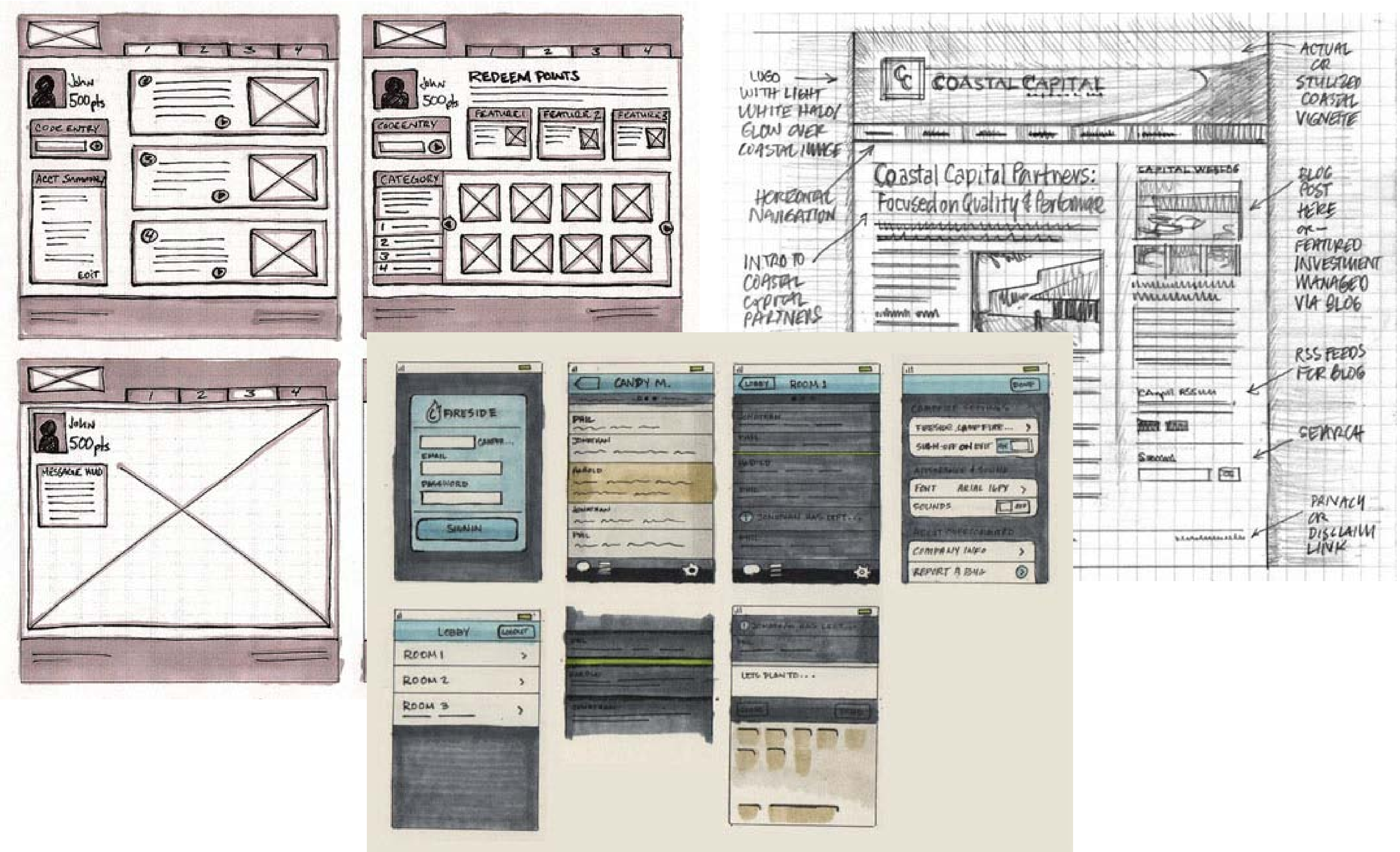
# GUI Sketch

- Useful to implement a window applications
  - Swing based applications
  - Android/iPhone applications
- Identify the standard User Interface of a system

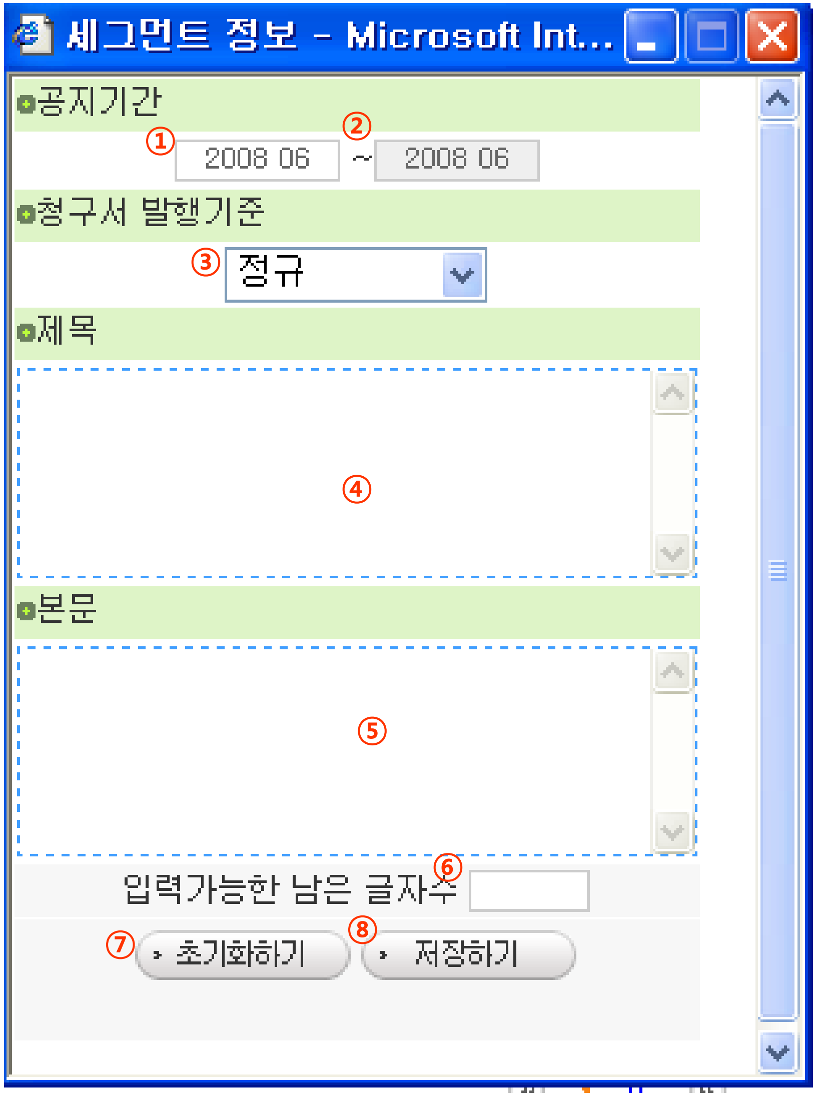
# Notes of GUI sketch

- The standard User Interface
  - The types of components
  - The types of messages
- Required input parameters of a component
- Properties of each input parameter
  - Name : better to be standardized
  - Maximum/minimum length, types of input
  - Handling of errors
- Business process of given inputs

# Example of GUI sketch


John
500pts
CODE ENTRY
ACCT SUMMARY
EDIT
REDEEM POINTS
FEATURE 1
FEATURE 2
FEATURE 3
CATEGORY
MESSAGE HUD
COASTAL CAPITAL
Coastal Capital Partners: Focused on Quality & Performance
CAPITAL WEBLOG
LOGO WITH LIGHT WHITE HALO/ GLOW OVER COASTAL IMAGE
HORIZONTAL NAVIGATION
INTRO TO COASTAL CAPITAL PARTNERS
ACTUAL OR STYLIZED COASTAL VIGNETTE
BLOG POST HERE OR — FEATURED INVESTMENT MANAGED VIA BLOG
RSS FEEDS FOR BLOG
SEARCH
PRIVACY OR DISCLAIM LINK
FIRESIDE
EMAIL
PASSWORD
SIGN IN
CANDY M.
PHIL
JONATHAN
HAROLD
ROOM 1
LOBBY
ROOM 2
ROOM 3
FONT
ARIAL 16PT
SOUNDS
COMPANY INFO
REPORT A BUG
LETS PLAN TO...

| 화면ID | MMSW006 | 화면 명 | 공지사항. 입력.수정 |
|---|---|---|---|

| 화면 layout | 설 명 |
|---|---|
|  세그먼트 정보 - Microsoft Int...<br>공지기간<br>①<br>2008 06<br>②<br>~<br>2008 06<br>청구서 발행기준<br>③<br>정규<br>제목<br>④<br>본문<br>⑤<br>입력가능한 남은 글자수<br>⑥<br>⑦<br>초기화하기<br>⑧<br>저장하기 | 1.화면설명<br>공지사항 입력.수정<br><br>2.주요화면 설명<br>1)공지기간-시작월부터<br>2)공지기간 -종료월까지<br>3)청구서 발행기준-정규/반송.재발행<br>4)제목-공지사항 제목<br>5)본문-공지사항내용<br>6)입력가능한 남은 글자수-본문에 입력 가능한 글자수<br>7)초기화-다시 원래대로 빈칸으로.<br>8) 저장하기- 저장하기 |

*From : 화면 정의서* **LG텔레콤 Mobile 청구서**