



How the customer explained it



How the Project Leader understood it



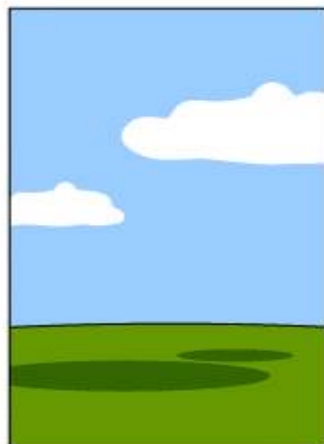
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



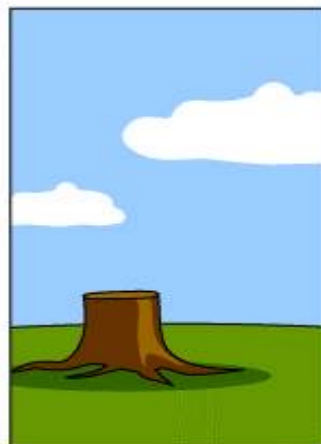
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

# CPEN 321

## *More on Requirements*

# Recap

- Why we need to identify, capture, and manage *functional and non-functional* requirements
  - To avoid ambiguity, estimate workload, drive design and testing
- What a good requirement is
  - Specific and precise, testable and measurable
- How to collect requirements
  - Talk to users, analyze their needs, “simulate” users with personas
- How to document requirements
  - User stories, formal use cases, structured text, mockups, prototypes

# ***Defining Requirements in This Course***

## **Functional**

- Actors, functional requirement names, use case diagram
- Lightweight formal use case specification of each requirement:
  - name, short description, primary actors, success scenarios, failing scenarios
- Mockups, if helpful/needed

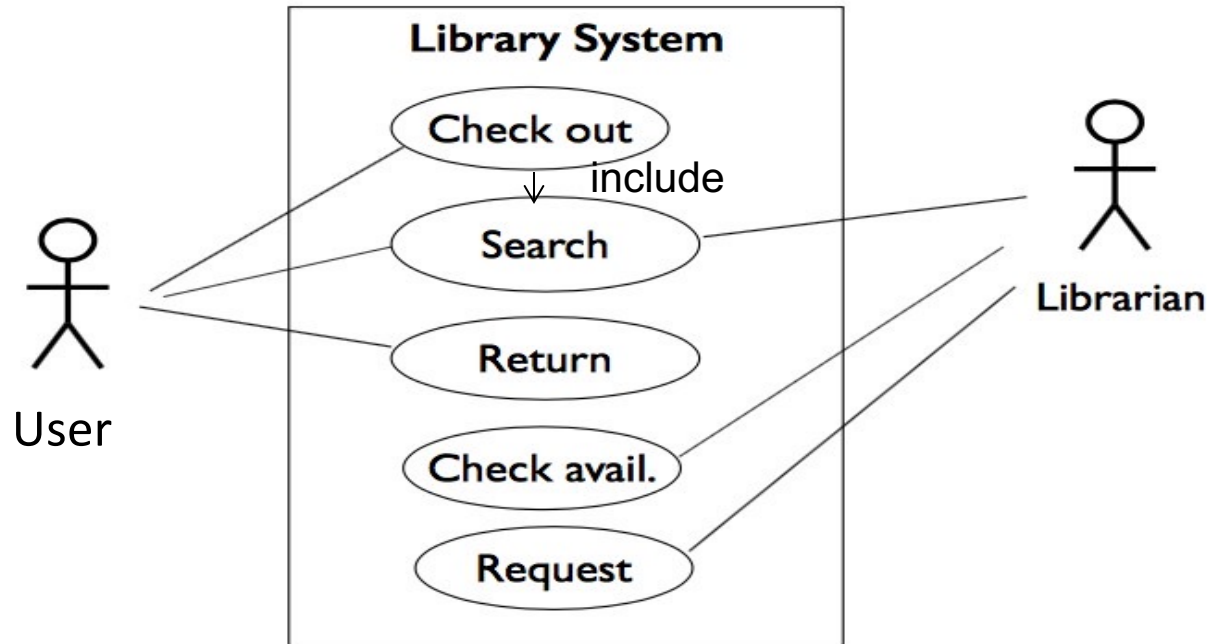
## **Non-Functional (at least 2)**

- Specification for each requirement:
  - Textual description
  - Justification (why needed)
  - Validation approach (how to confirm)

# *Use Case Diagram*

- Think about external actors  $\neq$  personas
  - Remember: multiple personas can correspond to one actor!
- Focus on behaviors you build into the software to meet the user's needs (verbs)
- If needed, add info about relationships between actors and between use cases (but do not abuse the mechanism, if not necessary)

# Example: Use Case Diagrams (UML)



# ***Example: Search Formal Use Case – 1/3***

**Title:** Search

**Description:** The actor searches for a particular article by providing (possibly partial) information about any of the following:

- article title
- author names
- time range (in years)

**Primary Actor:** User, Librarian

**Preconditions:** The actor is logged into the system

**Postconditions:** A list of identified articles and their availability is presented

# ***Example: Search Formal Use Case – 2/3***

## **Main Success Scenario:**

1. The actor opens search from the menu.
2. System displays search criteria: title, authors, years
3. The actor enters a search string for one of more of the criteria and presses enter
4. The system displays all found articles with the availability info for each



# ***Example: Search Formal Use Case – 3/3***

## **Extensions:**

**3a.** The actor did not enter any search string.

- 3a1. System displays error message saying at least one search string is mandatory.

**3b.** The actor entered illegal characters in at least one search string.

- 3b1. System displays error message identifying the erroneous criterion and characters.

**4a.** No articles are found.

- 4a1. System displays message saying that no articles were found, along with a suggestion to refine the search criteria.

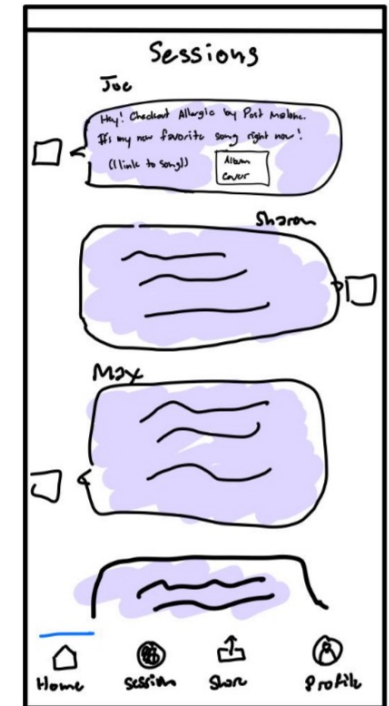
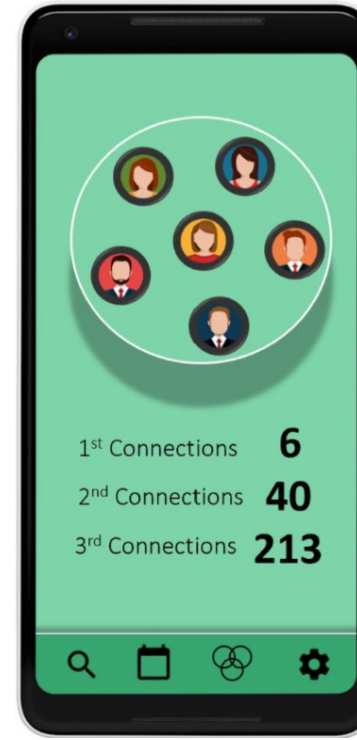
**Very important to consider failing cases!**

# Example: Search Mockup

**Find articles**  
with **all** of the words

Return articles **authored**  
by  e.g., "PJ Hayes" or McCarthy

Return articles **dated**  
between  —  e.g., 1996



# Exercise

- You want to build an online dating system.
- The client will attract customers by providing free browsing and matching functionality, but charging for allowing users to contact other users.
  - For example, a user should be able to register, create a profile, and search for “soul mates”, all without paying.
  - Then, if they want to send a message to another user, or receive a message from another user, they need to upgrade their membership by making a payment.
- The client should be able to find and ban “offensive” users.

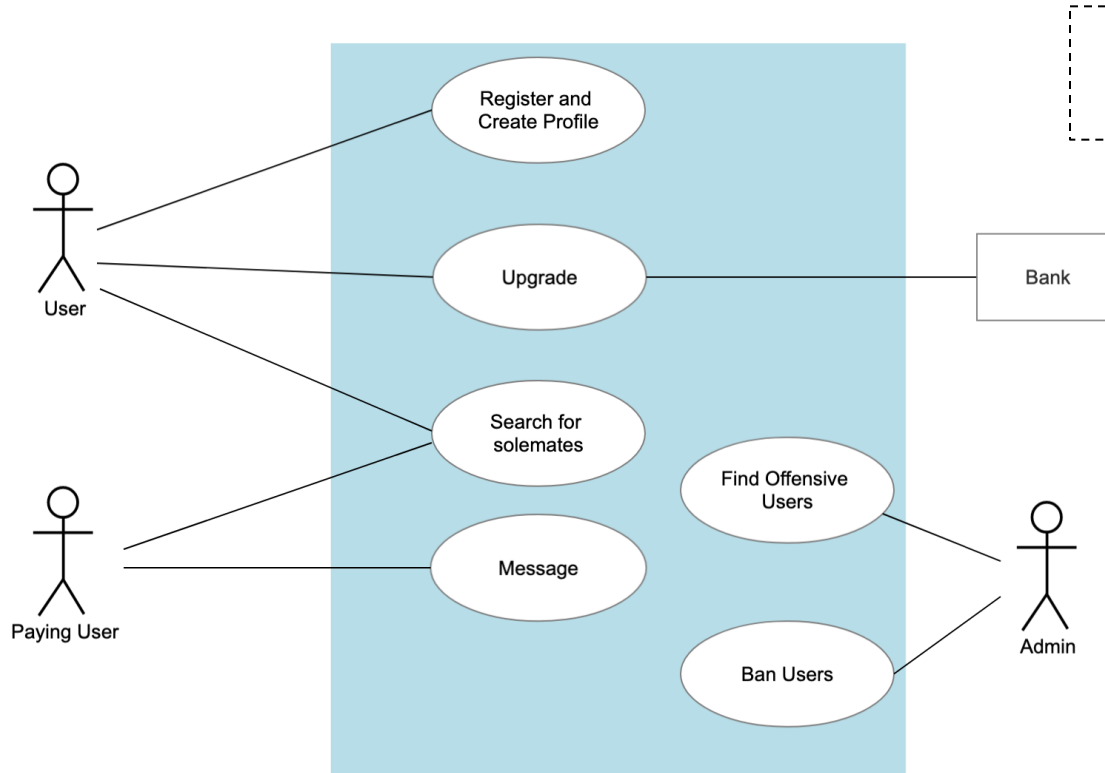
1. **Name at least six use cases of the system**
2. **Identify actors**
3. **Draw a use case diagram**
4. **Write a formal use case description for each use case**

- You want to build an online dating system.
- The client will attract customers by providing free browsing and matching functionality, but charging for allowing users to contact other users.
  - For example, a user should be able to register, create a profile, and search for “soul mates”, all without paying.
  - Then, if they want to send a message to another user, or receive a message from another user, they need to upgrade their membership by making a payment.
- The client should be able to find and ban “offensive” users.

1. **Name at least six use cases of the system**
2. **Identify actors**
3. **Draw a use case diagram**
4. **Write a formal use case description for each use case**

- You want to build an online dating system.
- The client will attract customers by providing free browsing and matching functionality, but charging for allowing users to contact other users.
  - For example, a user should be able to register and create a profile, and search for “soul mates”, all without paying.
  - Then, if they want to send a message to another user, or receive a message from another user, they need to upgrade their membership by making a payment.
- The client should be able to find and ban “offensive” users.

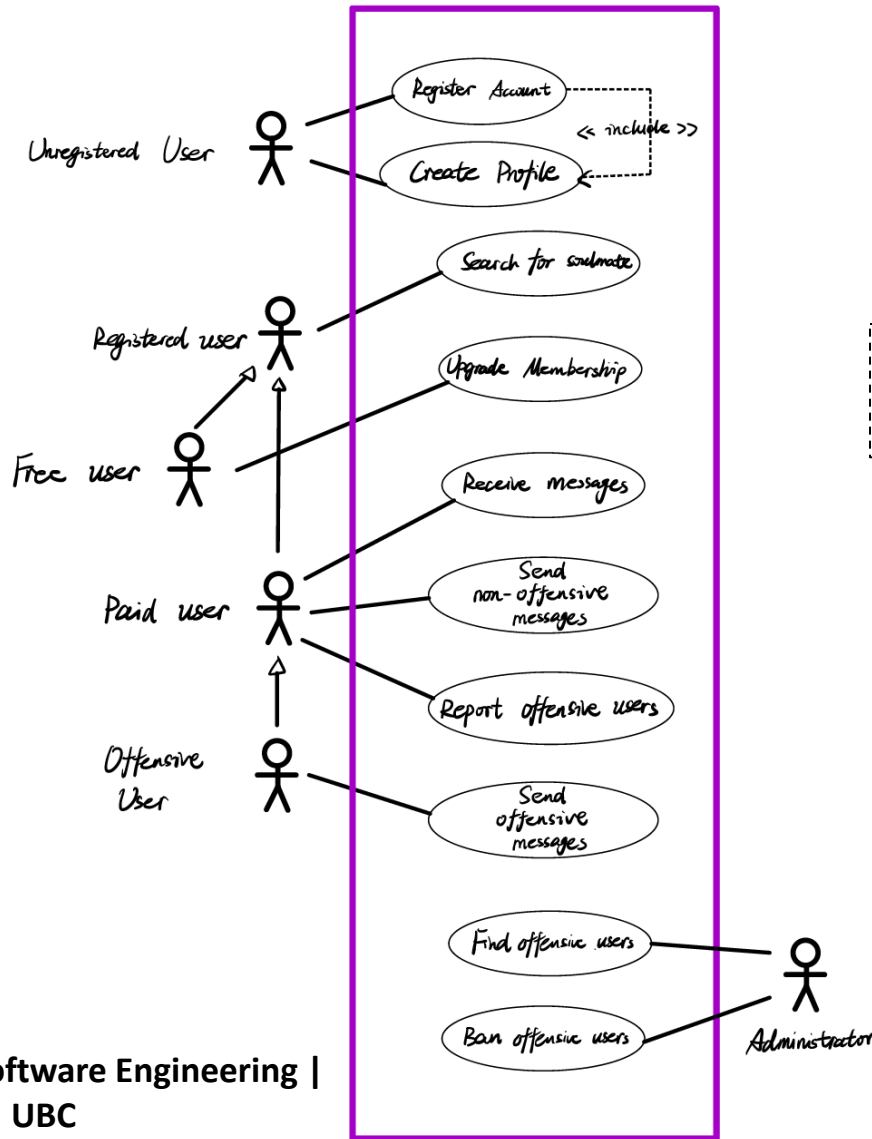
# ***Now Your Turn: Use Case Diagram***



*Pretty Good and Simple!*

# ***Use Case Diagram: Common Mistakes***

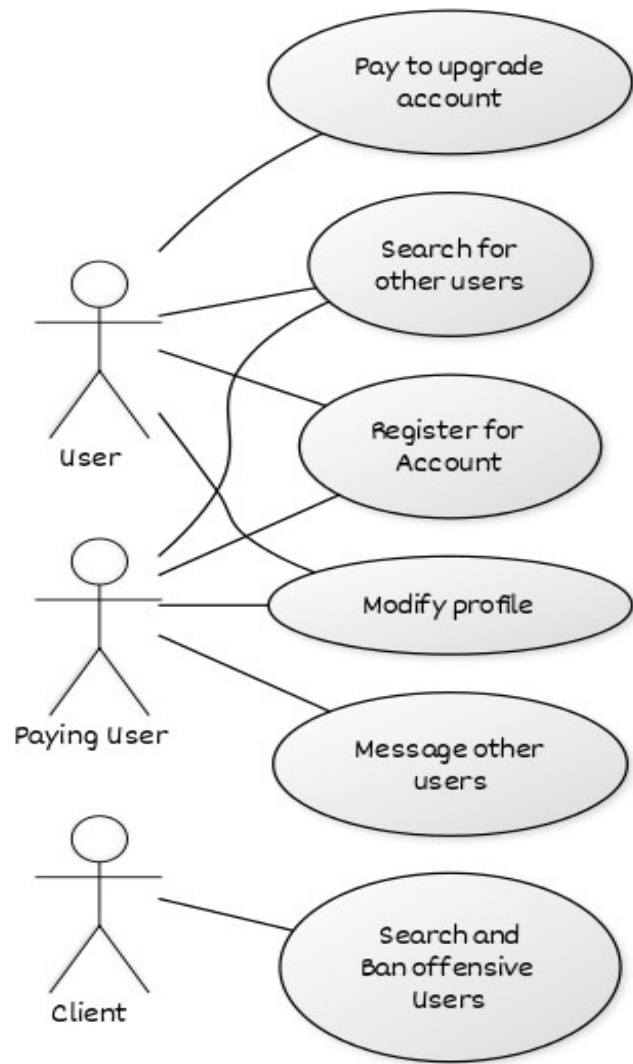




*Pretty Good!*

Send offensive and non-offensive messages are not separate use cases. Make it “Send Message”.

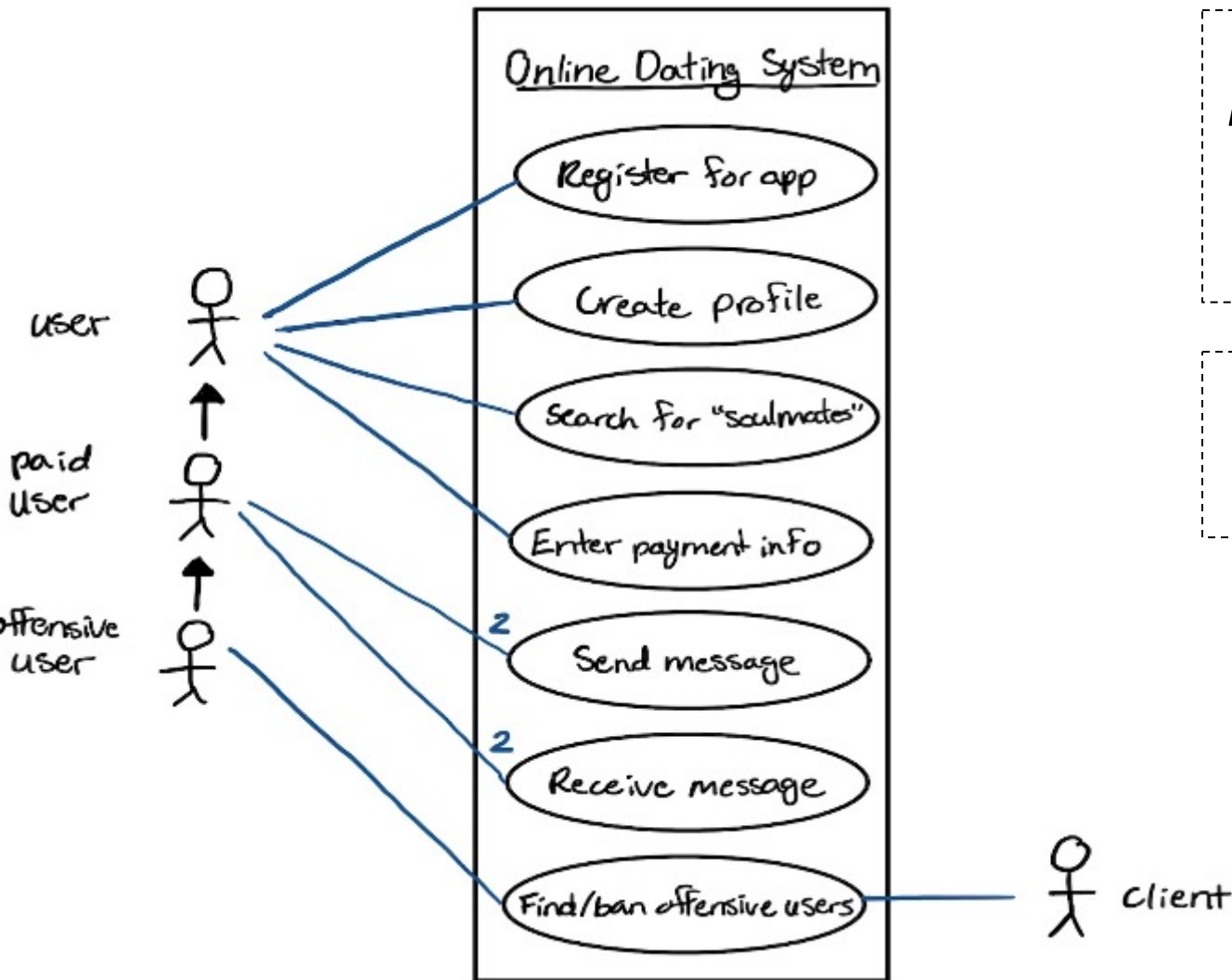
There is no “Offensive User” actor, it is just User.



CREATED WITH YUML

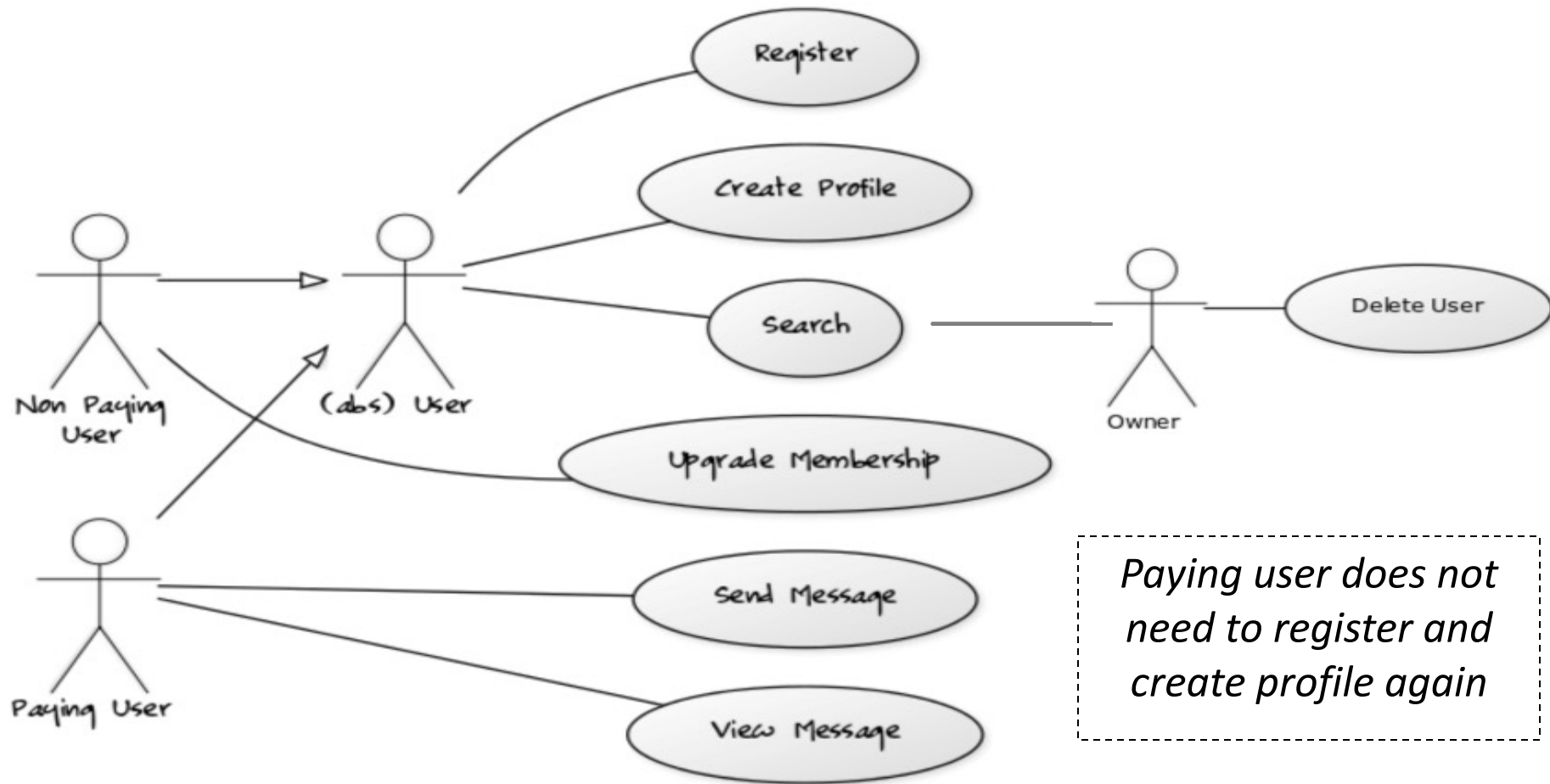
*Avoid duplications (and clutter)  
with proper generalization*

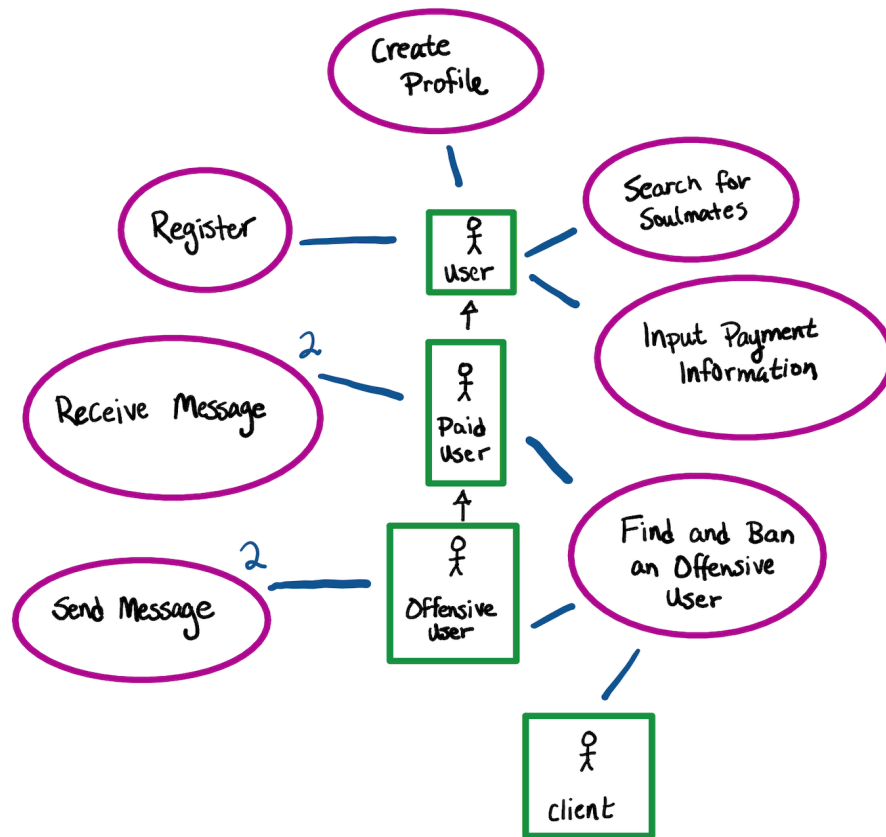
*“Client” and “User” names are  
too similar*



*Paid user does not need to enter payment info again (overuse of generalization)*

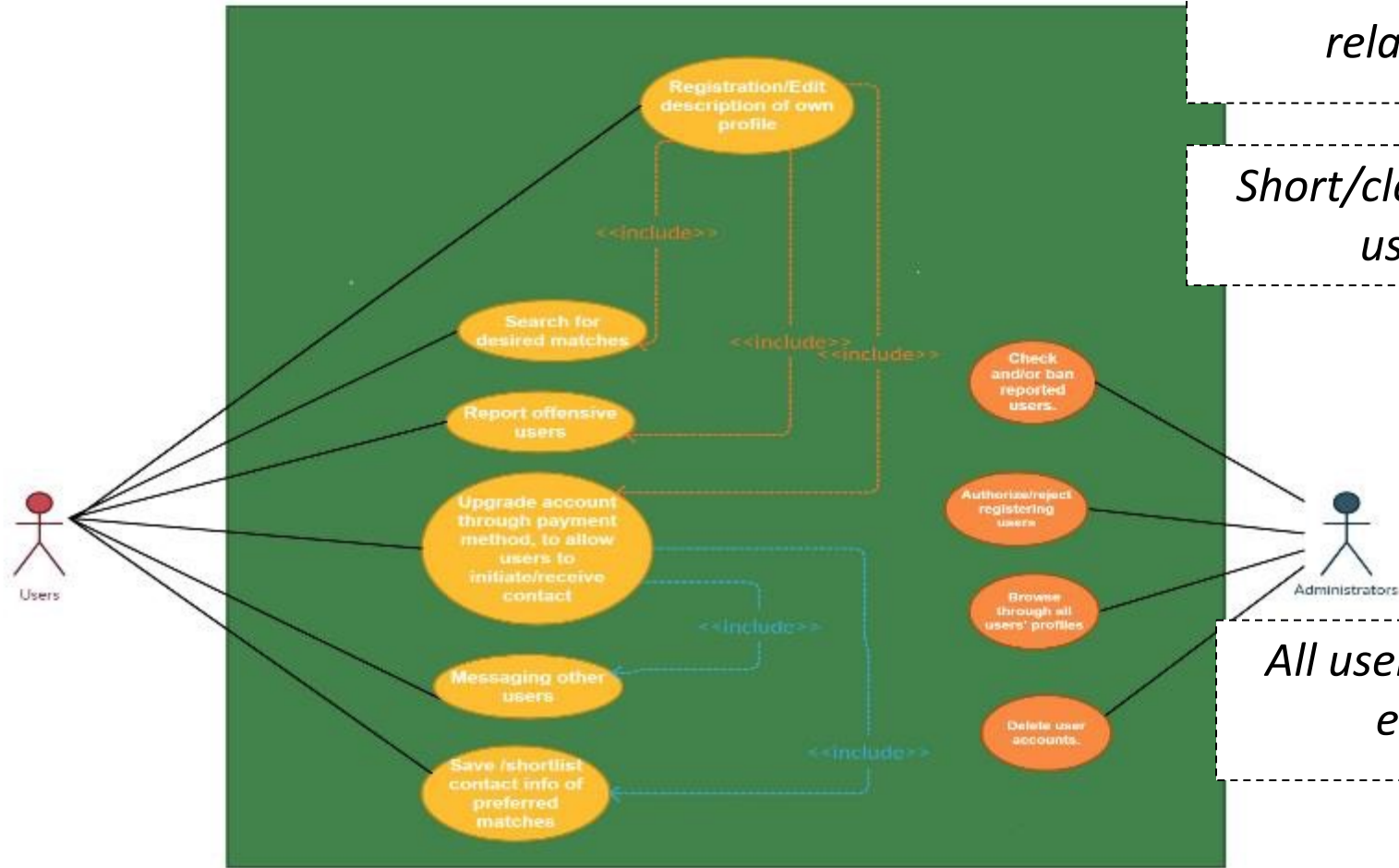
*"Client" and "User" names are too similar*





*Use standard notations:  
the central part is the  
system*

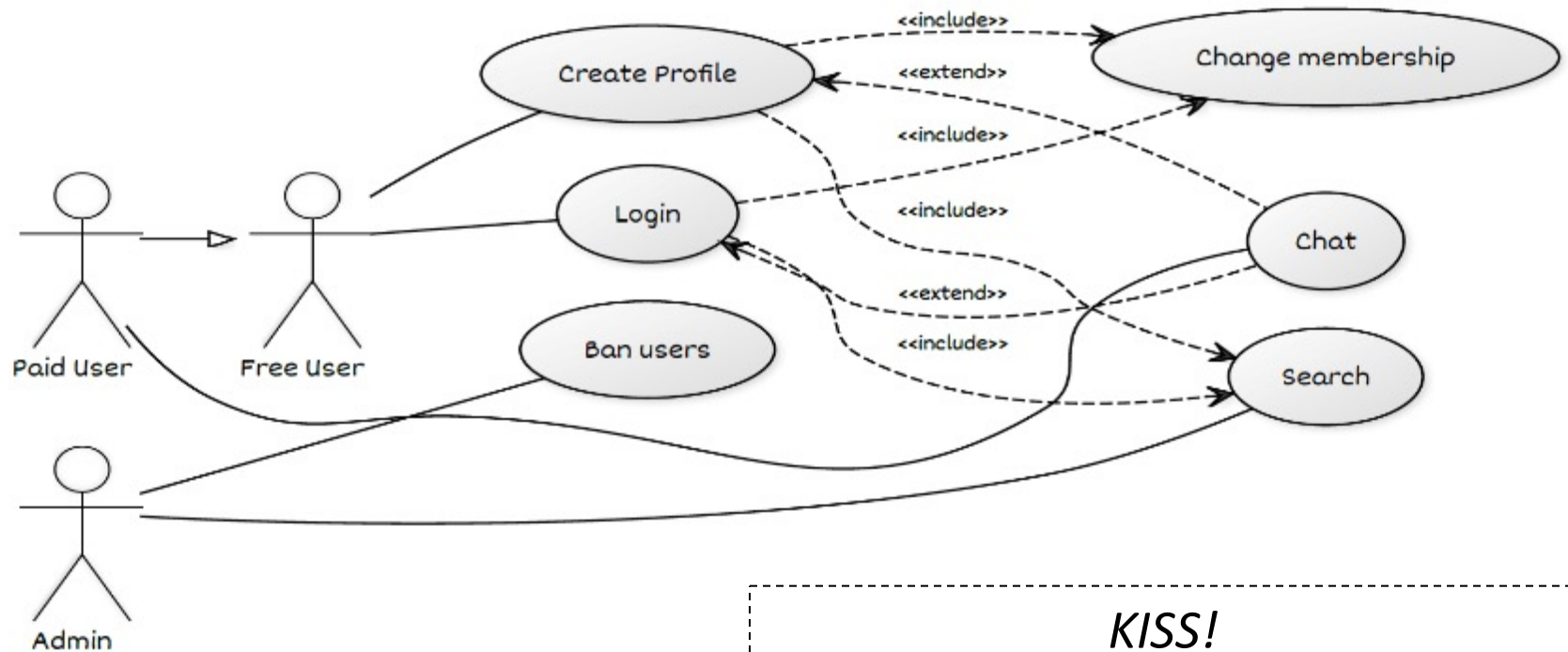
*“Paid user” only receives  
messages but  
cannot send;  
What is 2?*



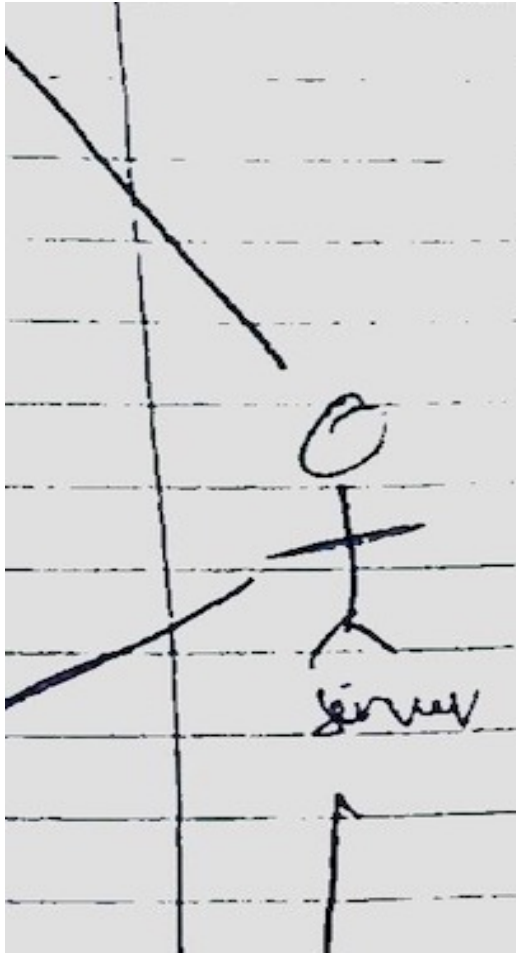
*Incorrect use of include relationships*

*Short/clear names for use cases*

*All users can message each other*

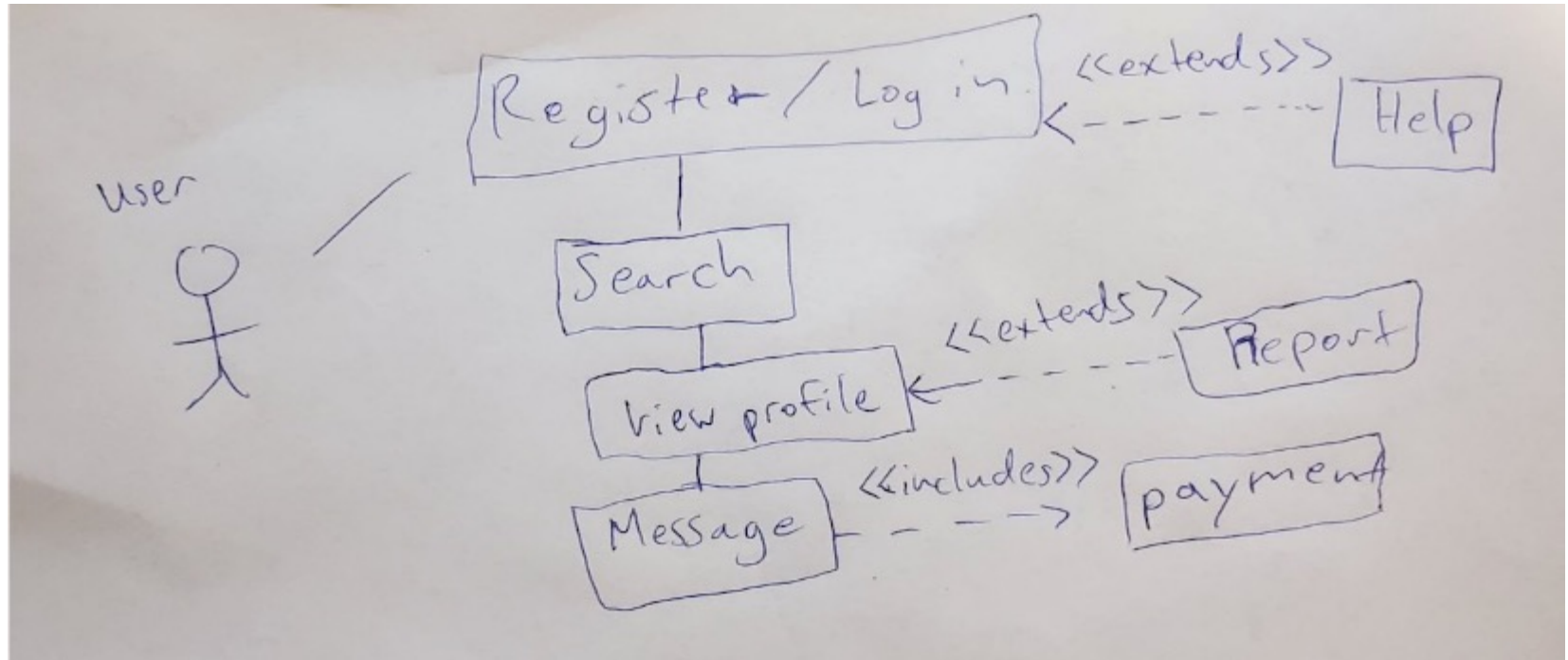


**KISS!**  
(Why would “login” include  
“search”, etc.)



*Illegible*





*Not a use case diagram:*

- *Missing associations between actors and use cases*
- *Undefined relationships between use cases*

# ***Project Presentations Next Week – 1/3***

**Goal:** Describe your project ideas to the rest of the class + receive suggestions for improvement

**Structure:** 3 minutes for each idea + 2 minutes for questions (8 minutes in total),  
4 slides per idea (8 slides in total):

1. The name of the project and group members
2. Why needed? The project's high-level idea and target audience (1 minute)
3. High-level description of the main actors and three most central use cases (1.5 minutes)
4. Your project highlight (0.5 minutes)

# ***Project Presentations Next Week – 2/3***

- Submit your presentation in the “Project Presentation” assignment on Canvas by **Friday, September 22, 9 pm**
  - The exact spec for the assignment, with project requirements is on Canvas
- The order of presentations will be posted on Piazza before the class on Monday
- To give everyone the chance to present, the time allocation is strict, i.e., we will stop you after 8 minutes sharp.
  - The class might run 10 minutes longer, till 5pm → hope there is no class afterwards

# ***Project Presentations Next Week – 3/3***

**Vote on the projects!**

**Awards:** Top 3 ideas will receive 5%, 4%, and 3% bonus for M2

**Procedure:** 50% instructor vote, 50% peer-evaluation

You can vote on multiple projects, excluding your own.

Only one student vote per project counts.

**Success Criteria:**

- The target audience and the need for the project are clearly articulated
- The scope of the project is significant, i.e., not too simplistic
- The “highlight” of the project is interesting
- The presentation is engaging and is visually appealing, includes appropriate graphics (when needed), and is easy to follow

# ***(Optional Home Work): Formal Use Case Description for “Upgrade Membership”***

- name,
- short description,
- primary actor(s),
- success scenario,
- failing scenarios

If you want to get **feedback**, share your solution on Piazza;  
you will get a reply from the course staff

- Public or private message, up to you!
- Subject: *“Formal Use Case Description – Feedback”*

***Looking forward to your  
presentations next week!***

