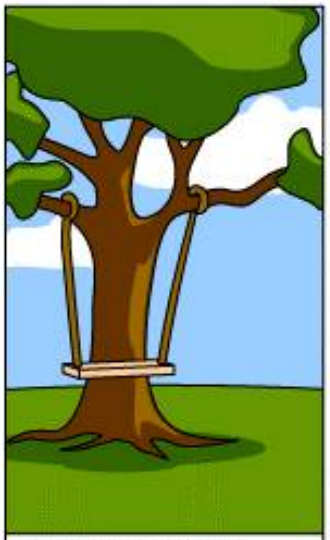
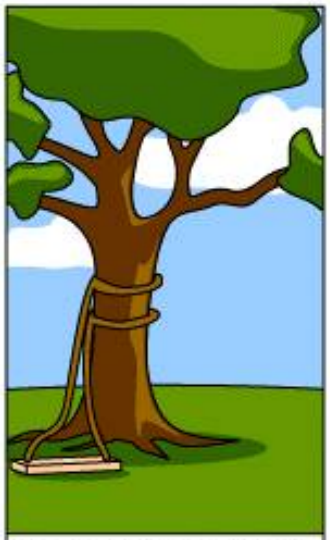
How the customer explained it



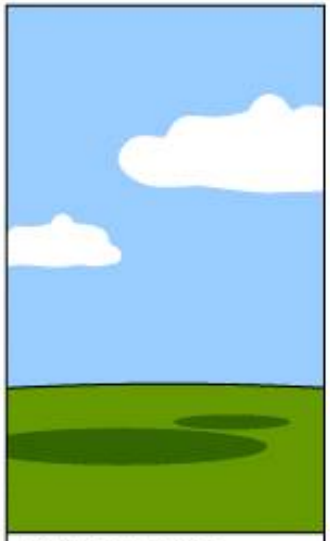
How the Project Leader understood it

How the Analyst designed it

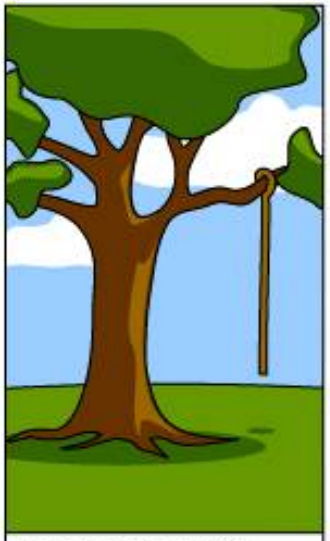


How the Programmer wrote it

How the Business Consultant described it

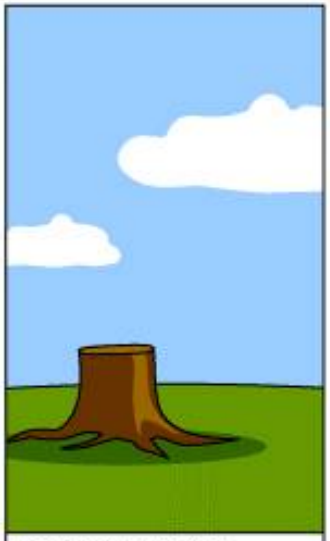


How the project was documented



What operations installed

How the customer was billed



How it was supported

What the customer really needed

CPEN 321

More on Requirements, Examples

Recap: Basic Terminology

- **The user:** Persona, actor, etc.
  - Multiple personas can correspond to one actor

- **What the user wants to accomplish:** user story, use case, etc.

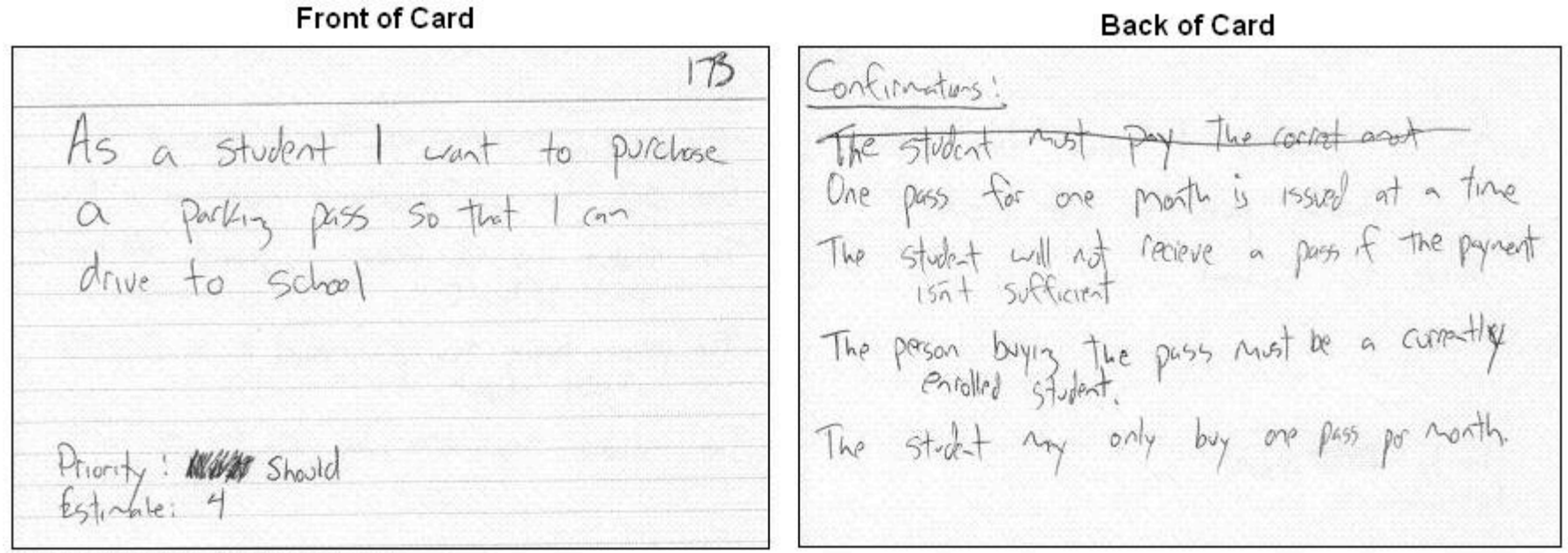
- **A sequence of steps to accomplish the user's goal:** scenario**,** story description, sequence of steps, etc.

Recap: Where do user-level requirements come from?

- Interviewing real users
- Identifying “fictional” personas
- Completing the requirements extracted from the users by describing actions of each persona

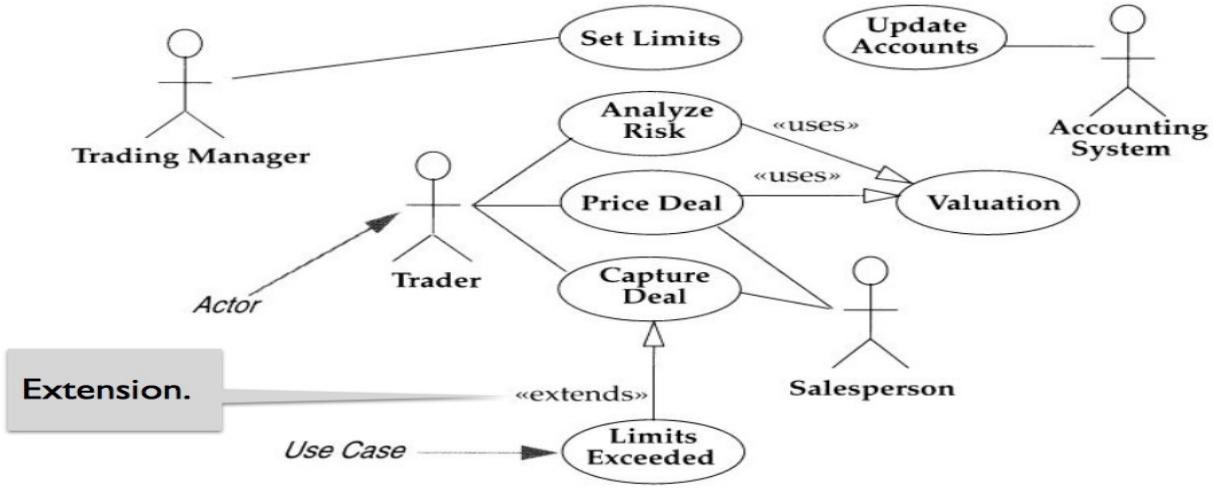
Recap: How to capture such requirements?

- Agile: user story cards
  - story, steps, confirmations



Recap: How to capture such requirements?

- UML: use case diagrams
  - Users, use cases, relationships between users, relationships between use cases



Recap: How to capture such requirements?

- UML: use case diagrams
  - Users, use cases, relationships between users, relationships between use cases
  - Steps?
    - Formal scenarios

Main success scenario	1. Patron enters account and password 2. System verifies and logs patron in 3. System presents catalog with search screen 4. Patron enters book title 5. System finds match and presents location choices 6. Patron selects location and reserves book 7. System confirms reservation and re-presents catalog

Recap: How to capture such requirements?

- UML: use case diagrams
  - Users, use cases, relationships between users, relationships between use cases
  - Steps?
    - Formal scenarios
    - Sequence diagrams

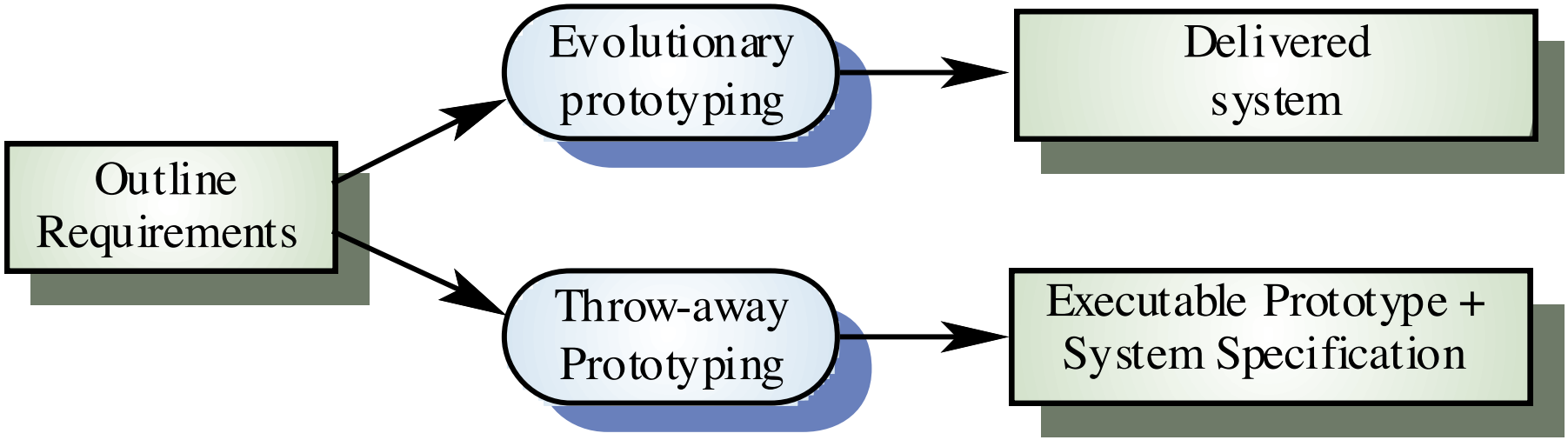
User Stories / Use Cases

	Agile-like	More formal/ UML-like
The user	User, Persona	User, Actor
What the user wants to accomplish	User story (front of the card) e.g., "As Alice I want to borrow a book from the library"	Use case (ellipse) e.g., "borrow a book"
A sequence of steps to accomplish the user's goal	• Steps and confirmations (back of the card)	• Formal use case description • Sequence diagram

Specifying Requirements

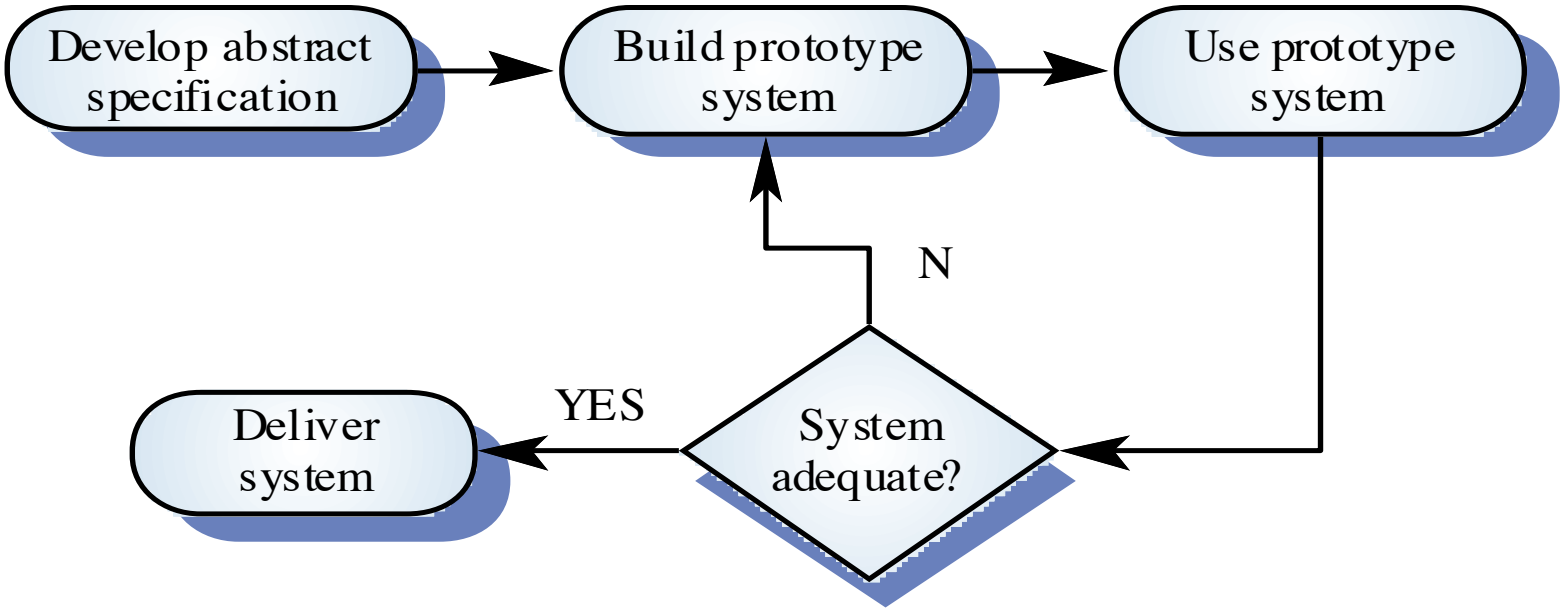
- Formal Specification (document)
- User Stories / Use Cases
- **Prototype**

Approaches to Prototyping



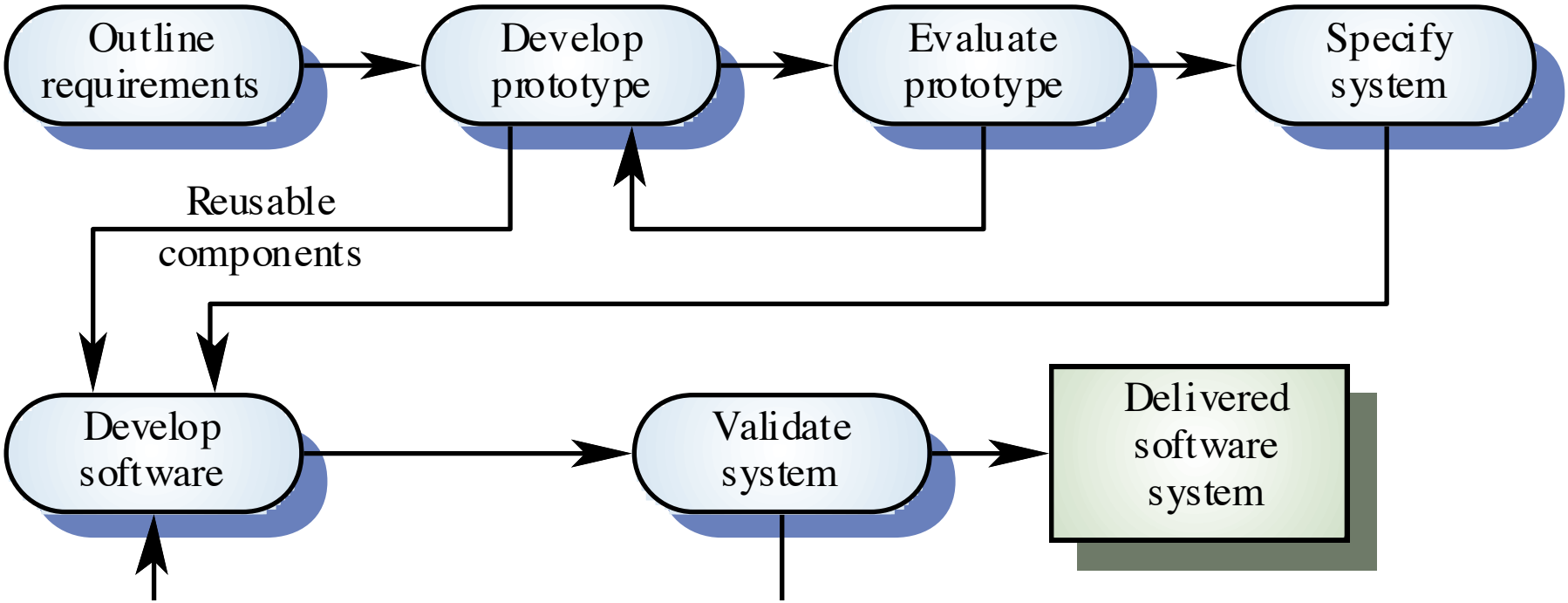
Appropriate Use

- Evolutionary prototyping is used for systems where the specification cannot be developed in advance
  - Make careful design decisions



Appropriate Use

- Throw-away prototyping is used for defining the specification
  - The system might be poorly structured and difficult to maintain



Advantages of Prototypes as Specifications

- Appealing to the users; gives the users a concrete impression of the system's capabilities
- Clear and easy to understand
- Useful for parts of the system which cannot be effectively described in words, such as user interfaces

Problems with Prototypes as Specifications

- Non-functional requirements cannot be adequately tested in a system prototype
- Some functional requirements (e.g., safety-critical functions) may be difficult to prototype
- An implementation has no legal standing as a contract
- Time-consuming

My Recommendation for Requirements Milestone

- Use case diagram for actors and high-level use cases
- Sequence diagram for interactions
- List of non-functional requirements (measurable!!)
  - The system should perform well 
- (bonus) Screen prototypes

Example

- You build an online dating system.
- The client plans to 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.

Identify personas

- Alice is a college student interested in browsing profiles in order to snoop on her friends. Not willing to pay.
  - Needed to represent a population of **non-paying users**
- Bob is a software engineer looking to find a younger male date.
  - Needed to highlight **different search criteria**
- Cynthia is a retired nurse who is looking for a soul-mate. She faces challenges using mobile technology but is too shy to go to a blind date.
  - Needed to represent a population **of technically-impaired people**, who **will not easily look for help**
- David is the owner of the system who wants to make sure the system is ethical and legally-compliant.
  - Needed to represent the requirements of the owner, e.g., to delete users

Name at least six use cases and draw a use case diagram

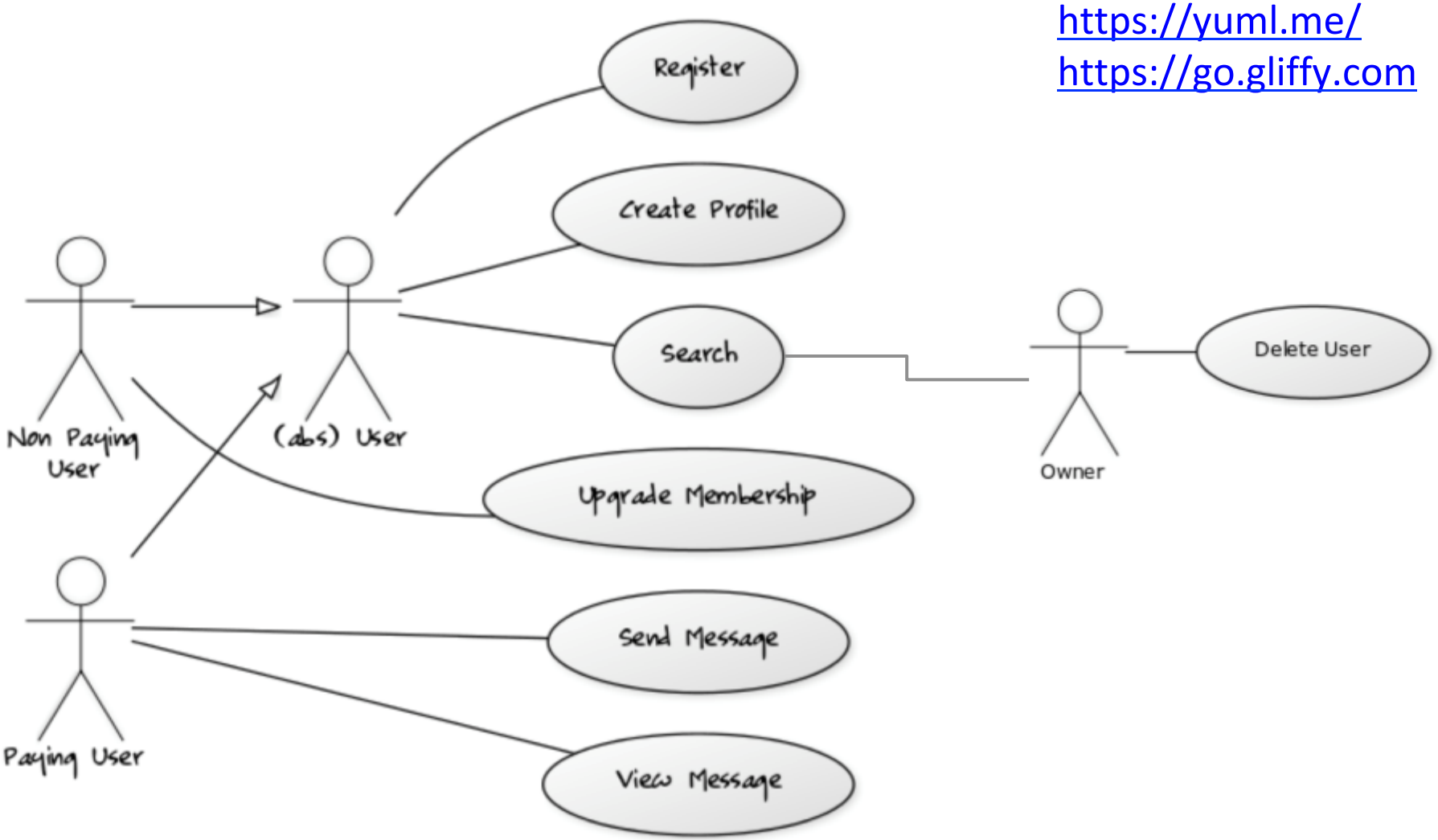
- You build an online dating system.
- The client plans to attract customers by providing free browsing and matching functionality, but charging for allowing users to contact other users.
- For example, a user should be able to register, create a profile, and search for “soul-mates”, all without paying.
- 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.

Name at least six use cases and draw a use case diagram

- You build an online dating system.
- The client plans to attract customers by providing free browsing and matching functionality, but charging for allowing users to contact other users.
- For example, (3) user should be able to register (1), create a profile (2), and search for "soul-mates" (4), all without paying.
- Then, if they want to send a message to another user, or (5) receive a message from another user, they need to upgrade their membership by making a payment (6).

Use Case Diagram

https://yuml.me/
https://go.gliffy.com



Identify the main modules needed to implement your system. Model their relationships using a class diagram

Home assignment...

Weekly Deliverables

- Submitted via Canvas in PDF format: https://canvas.ubc.ca/courses/13059/assignments
- Due **two days before** your Lab time.
- For example, the deliverable for W5 (requirements) is due:
  - End of the day on Saturday W4, Sept 29 for **Monday** groups.
  - End of the day on Sunday W4, Sept 30 for **Tuesday** groups.
  - End of the day on Monday W4, Oct 1 for **Wednesday** groups.

Content of the Deliverable

- 1-page progress report that includes:
  - a high-level description of the progress for the week,
  - the plans for next week,
  - major decisions and changes in the scope of the project,
  - the contributions of individual team members

- The deliverable itself, when required (like in W5 and W6).