Requirements Overview

Requirements Engineering

- AKA Requirements Capture, Requirements Elicitation, etc.

- Refers to the activities we use to determine what the customer needs and/or what we are going to build to address those needs

ITS GROUNDHOG DAY....
AGAIN
memegenerator.ne

But Wait! Didn’t We Already Do This?

- User Stories are indeed a way to express a customer’s needs
  - are merely one possible artifact for expressing needs
  - They aren’t the only, nor necessarily the best way!

- But User Stories were easy to learn weren’t they! And both your customers and business managers can understand them!

Confusing Terminology:

User Requirements
vs.
System Requirements

Confusing Terminology: User Requirements

- What does the customer/user need?
- Authors have used the term *requirement* in several ways
  - Sommerville refers to these as *User Requirements*
  - Some authors simply call these *Requirements*
  - e.g., *User Stories* expressed with the Role/Goal/Benefit template

- Most useful for discussing and expressing the customer or end-user's business needs in their language

Confusing Terminology: System Requirements

- What do we plan to build to address the customer/user’s needs?
- Authors have used the term *requirement* in several ways
  - Sommerville refers to these as *System Requirements*
  - Others call these a *Specification* or a *Functional Specification*
- Most useful for discussing what you plan to build with other developers in engineering language

Example *User Requirement*

- Role-Goal-Benefit template
- As a new user, I need to create an account on the server that will be associated with my data on that server

- Stories are an excellent way to express *user requirements*

Example *User Requirement* (Sommerville)

- “The Mentcare system shall generate monthly management reports showing the cost of drugs prescribed by each clinic during that month”

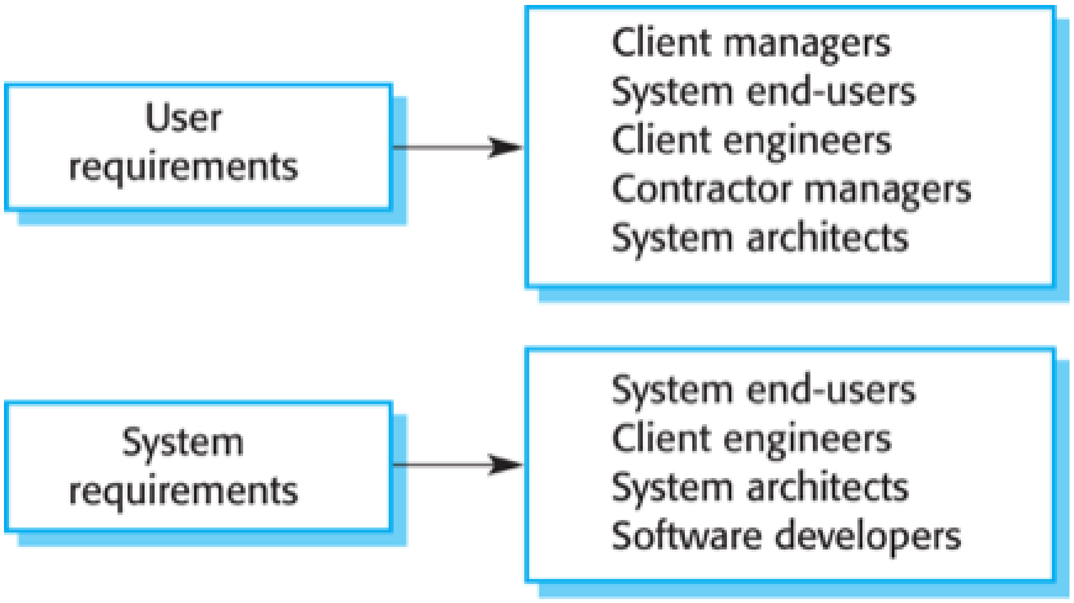
- Sommerville provides alternative examples in Ch4

Example *System Requirements*

- The CreateNewAccount screen will display text fields for the new user’s Account Name, Password, and an OK and Cancel button
- The new user’s Account Name entry must be a valid eMail address
- The length of the new user’s Password entry must be at least 6 characters
- The new user’s Password entry must contain both upper and lower case letters
- The new user’s Password entry must contain at least 1 digit or symbol

Who is the Audience?

Figure 4.2 Readers of different types of requirements specification



User requirements

Client managers
System end-users
Client engineers
Contractor managers
System architects

System requirements

System end-users
Client engineers
System architects
Software developers

- The text suggests both user and system requirements should be accessible to the system's end-users
- User requirements, in particular, appear in the customer's business language
- System requirements are in a "transition between customer and engineer"

User and System Requirements Expressed as User Stories and their Acceptance Criteria

1. As a new user, I need to create an account on the server that will be associated with my data on that server
    1. Given the account name does not already exist, when the user clicks OK, then an entry will be created in the accounts table for the supplied account name and password
    2. Given the user is creating an account, when the user clicks OK, then verify the account name is a valid eMail address
    3. Given the user is creating an account, when the user clicks OK, verify the length of the new user’s password > 6 chars
    4. Given the user is creating an account, when the user clicks OK, verify the password contains upper & lower case letters
    5. Given the user is creating an account, when the user clicks OK, verify the password contains at least 1 digit or symbol

Requirements vs. Product Backlog

- Sommerville's introduction to Ch4 merely presents use of a traditional terminology and templates for requirements
  - Compare *User Requirements* with *User Stories*
  - Compare *System Requirements* with *Acceptance Criteria*
  - Compare *Stakeholders* with *User Story Roles*

Requirements vs. Product Backlog

- Scrum incorporates additional information in the Product Backlog that was often omitted in traditional approaches
  - User Story Priorities
  - User Story Estimates

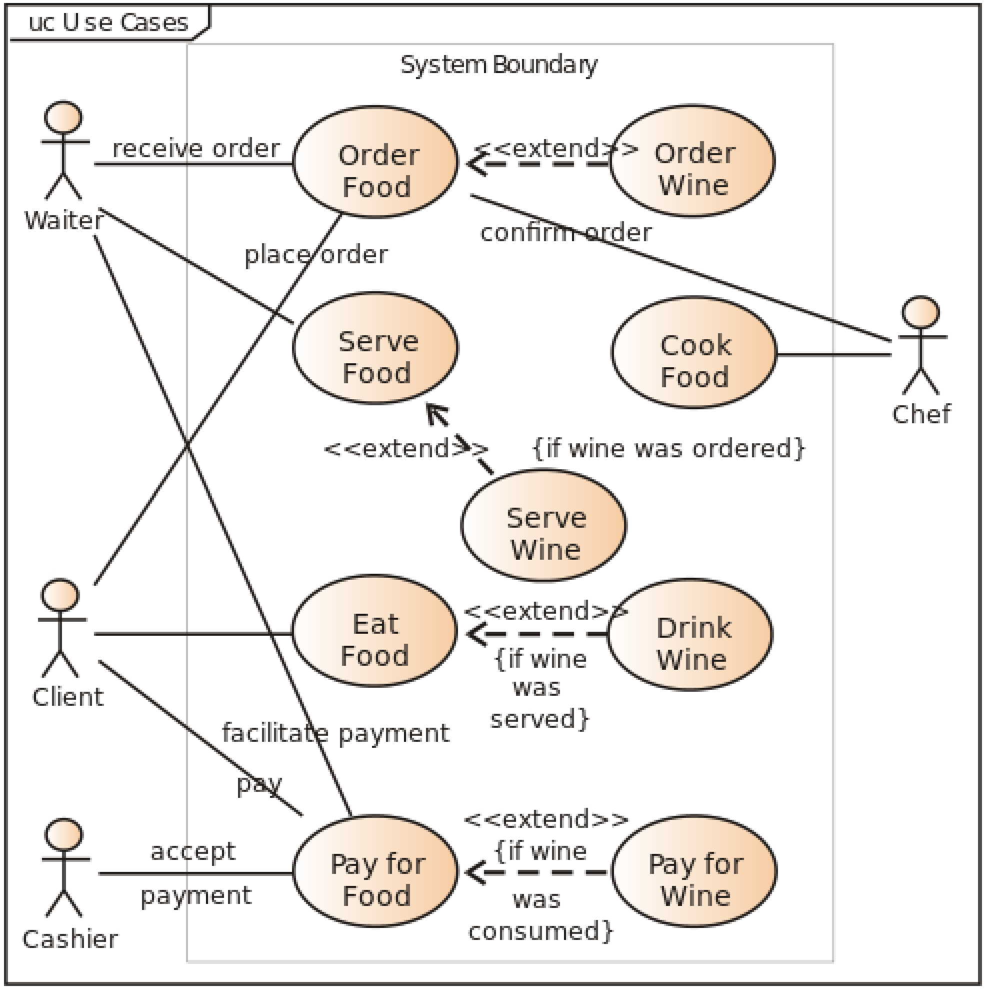
- And in CS471, we promote the use of the Role/Goal/Benefit and Given/When/Then templates

Requirements Analysis: User Stories and Use Cases

Use Case Diagrams

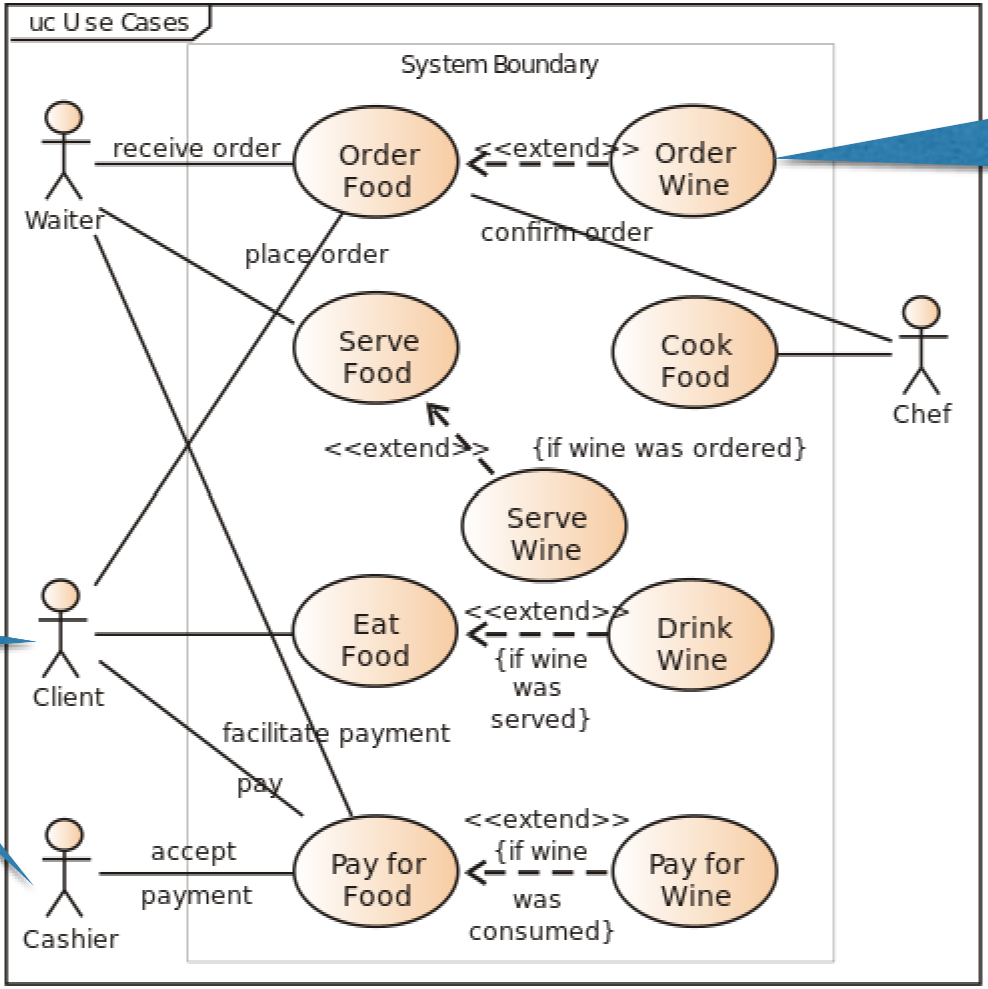
Use Case Diagrams

- Represent the user interactions with the system
- Example:



Use Case Diagrams

- Represent the user interactions with the system
- Example:

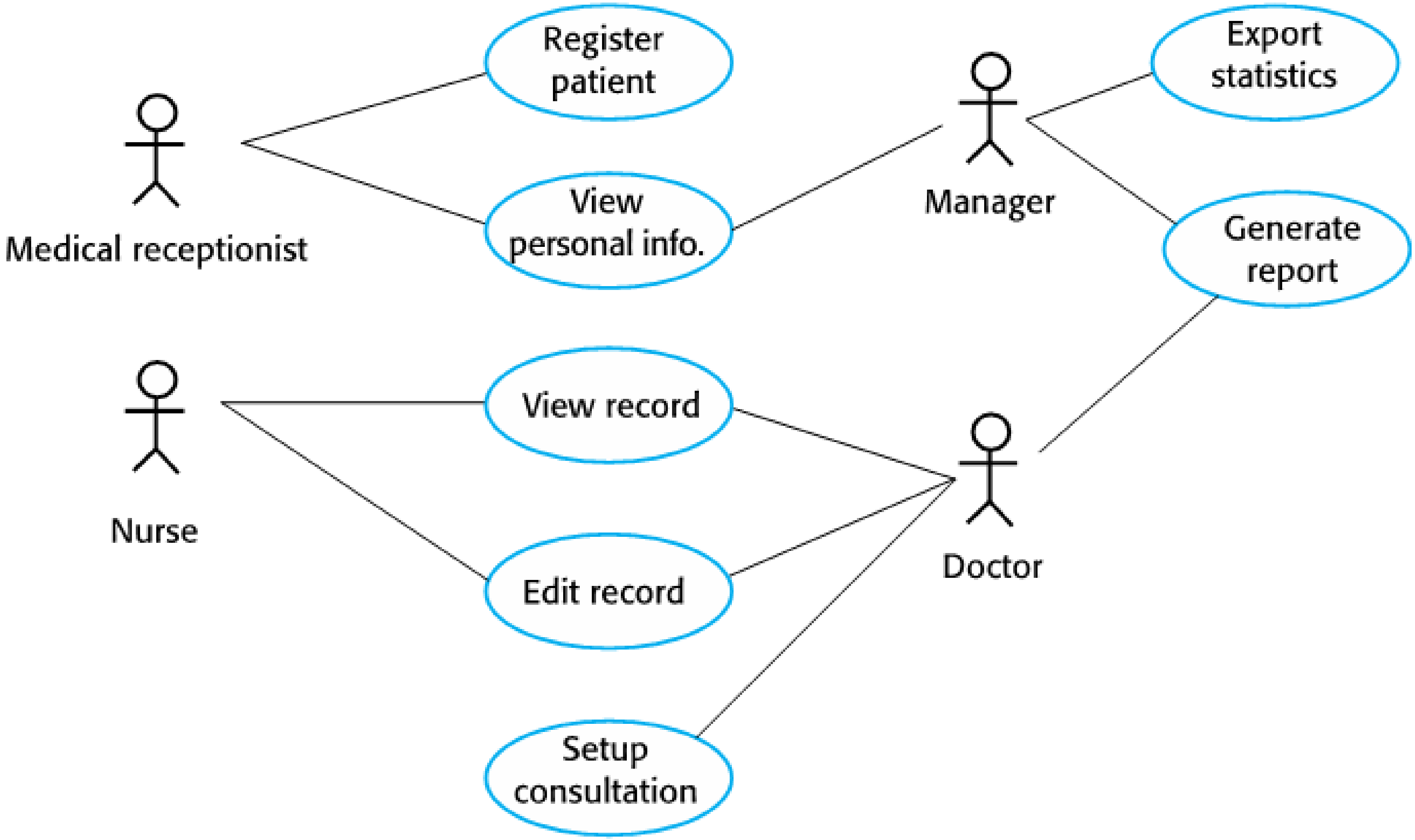


Use Case

Actor

Use Case Diagrams

- Example Mentcare System:



Use Cases

Use Cases

- A Use Case is an alternative way to document requirements

- Emerged from the waterfall-like world
- Often captures more detail
- NB: Useful in agile development when more detail is needed!

- Probably won't use in your CS471/CS481 projects unless your sponsor requests them

Use Case Example (Part 1)

- Name:
  - Create New Account
- Description:
  - A new user creates a new account
- Actors:
  - New User
- Preconditions:
  - An account with this name must not already exist

Use Case Example (Part 2)

- Main Sequence:

1. New User clicks Create New Account
2. Server displays the Create New Account dialog
3. New User enters their eMail address in the User Name field
4. New User enters their chosen password in the Password field
5. New User enters their chosen password in the Confirm field
6. New User clicks Create
7. Server creates a time-stamped, Unverified status entry in UserAccount table
8. Server constructs a VerifiedAddress URL
9. Server sends VerifiedAddress URL via eMail to the User Name
10. New User receives eMail and clicks on VerifiedAddress URL
11. Server changes UserAccount entry’s status to Verified
12. Server sends Welcome Message eMail to User Name

Use Case Example (Part 3)

- Alternate Sequence: Invalid eMail Address:
  1. Every hour, the Server reviews all the entries in the UserAccount table
  2. The Server deletes any Unverified entry older than 24 hours

Use Case Terminology: User

- Usually a person

- Can also be an external software system interacting with our product

Use Case Terminology: Actor

- A role played by a user of the product

- Similar to a role in a User Story (e.g., role-goal-benefit)

- Any given user may play more than one role

- Multiple users may play the same role

Use Case Terminology: Scenario

- A sequence of steps documenting an interaction between an actor and our product

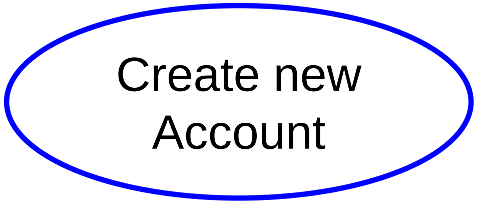
- Each step describes:
  - who does what? and
  - how the product responds?

Use Case Terminology: Use Case

- An interaction by one or more actors with the product
- Described in detail by a set of 1..N *scenarios* having a shared user goal
- Always has a *main scenario (or main sequence)*, informally the *happy path*
- *Other scenarios* are known as *extensions*, variations of the main *scenario*

Reviewing the Example…

Use Case Example (Part 1)



Use case representation in the Use Case Diagram

Use Case Example (Part 1)

- Name:
  - Create New Account
- Description:
  - A new user creates a new account
- Actors:
  - New User
- Preconditions:
  - An account with this name must not already exist

Actual Use Case Details

Create new Account

Use case representation in the Use Case Diagram

Use Case Example (Part 1)

- Name:
  - Create New Account
- Description:
  - A new user creates a new account
- Actors:
  - New User
- Preconditions:
  - An account with this name must not already exist

Use Case Example (Part 1)

- Name:
  - Create New Account
- Description:
  - A new user creates a new account
- Actors:
  - New User

Role played by a user

- Preconditions:
  - An account with this name must not already exist

Use Case Example (Part 1)

- Name:
  - Create New Account
- Description:
  - A new user creates a new account
- Actors:
  - New User

Role played by a user

- Preconditions:
  - An account with this name must not already exist

Could reference other use cases…

Use Case Example (Part 2)

- Main Sequence:
  1. New User clicks Create New Account
  2. Server displays the Create New Account dialog
  3. New User enters their eMail address in the User Name field
  4. New User enters their chosen password in the Password field
  5. New User enters their chosen password in the Confirm field
  6. New User clicks Create
  7. Server creates a time-stamped, Unverified status entry in UserAccount table
  8. Server constructs a VerifiedAddress URL
  9. Server sends VerifiedAddress URL via eMail to the User Name
  10. New User receives eMail and clicks on VerifiedAddress URL
  11. Server changes UserAccount entry's status to Verified
  12. Server sends Welcome Message eMail to User Name

Use Case Example (Part 2)

- Main Sequence:

Interaction between actors and our system

1. New Use clicks Cre
2. Serve plays the
3. Ne r enters their eMail address in the User Name field
4. hosen password in the Password field
5. chosen password in the Confirm field
6. New User clicks Create
7. Server creates a time-stamped, Unverified status entry in UserAccount table
8. Server constructs a VerifiedAddress URL
9. Server sends VerifiedAddress URL via eMail to the User Name
10. New User receives eMail and clicks on VerifiedAddress URL
11. Server changes UserAccount entry's status to Verified
12. Server sends Welcome Message eMail to User Name

"happy path"

Use Case Example (Part 3)

- Alternate Sequence: Invalid eMail Address:
  1. Every hour, the Server reviews all the entries in the UserAccount table
  2. The Server deletes any Unverified entry older than 24 hours

Use Case Example (Part 3)

- Alternate Sequence: Invalid eMail Address:

  1. Every hour, the Server reviews all the entries in the UserAccount table
  2. The Server deletes any Unverified entry older than 24 hours

Variation of main scenario