

Software Requirements

Inst. Nguyễn Minh Huy

Contents



- Requirement Concepts.
- Requirement Gathering.
- Requirement Specification.
- Requirement Validation.

Contents



- **Requirement Concepts.**
- Requirement Gathering.
- Requirement Specification.
- Requirement Validation.



■ What is requirement?

■ Software requirements:

- Description of what software should do:
 - Features.
 - Interactions.
 - Constraints.
- To satisfy stakeholders' needs.

■ Stakeholders:

- Users.
- Client.
- Related systems.
- Development team.
- ➔ Factors affect requirements.

Requirement Concepts



■ Needs vs. Requirements:

- Needs: stakeholders' expectations.

- Raw goals.

- Requirements: software can do.

- Realized goals.

Stakeholders' needs	Software requirements
Là giám đốc, tôi muốn biết doanh thu bán hàng trong tháng, quý, năm.	Thống kê doanh thu bán hàng
Là độc giả, tôi muốn tìm sách theo tên tác giả và năm xuất bản.	Tra cứu sách
Là người dùng, tôi muốn thời gian chờ xử lý ở mỗi tác vụ không quá 1 phút.	Xử lý và phản hồi nhanh
Là người dùng đã đăng ký, tôi muốn mật khẩu đăng nhập của tôi không thể dễ dàng dò được.	Bảo mật mật khẩu

Requirement Concepts



■ Requirement Classification:

■ Functional requirements:

- Users can use.
- Interactions between users and system.
- ➔ Software features.

■ Non-functional requirements:

- No interactions.
- ➔ Constraints on software.

Requirements	Classification
Thống kê doanh thu bán hàng	Functional
Tra cứu sách	Functional
Xử lý và phản hồi nhanh	Non-functional
Bảo mật mật khẩu	Non-functional



■ Requirement Abstraction Levels:

■ User requirements:

- General form of requirements → used for customers.
- Software features.

■ System requirements:

- Requirements in details → used for development.
- System interactions.

User-level	Development-level
Tra cứu sách (Functional)	<ul style="list-style-type: none">- B1: độc giả cung cấp tên tác giả, năm xuất bản.- B2: độc giả ra lệnh tra cứu.- B3: hệ thống thông báo chờ.- B4: hệ thống liệt kê danh sách tìm được theo thứ tự tên sách.
Bảo mật mật khẩu (Non-functional)	Mật khẩu người dùng dài tối thiểu 8 ký tự, có ít nhất 1 chữ số, được mã hóa khi lưu trữ trong CSDL.



■ Non-functional Requirements:

■ Quality constraints:

- User-friendly, performance, ...
- Availability, scalability, security, fault-tolerance, ...

■ Business constraints:

- Business process.
- Industry standard.

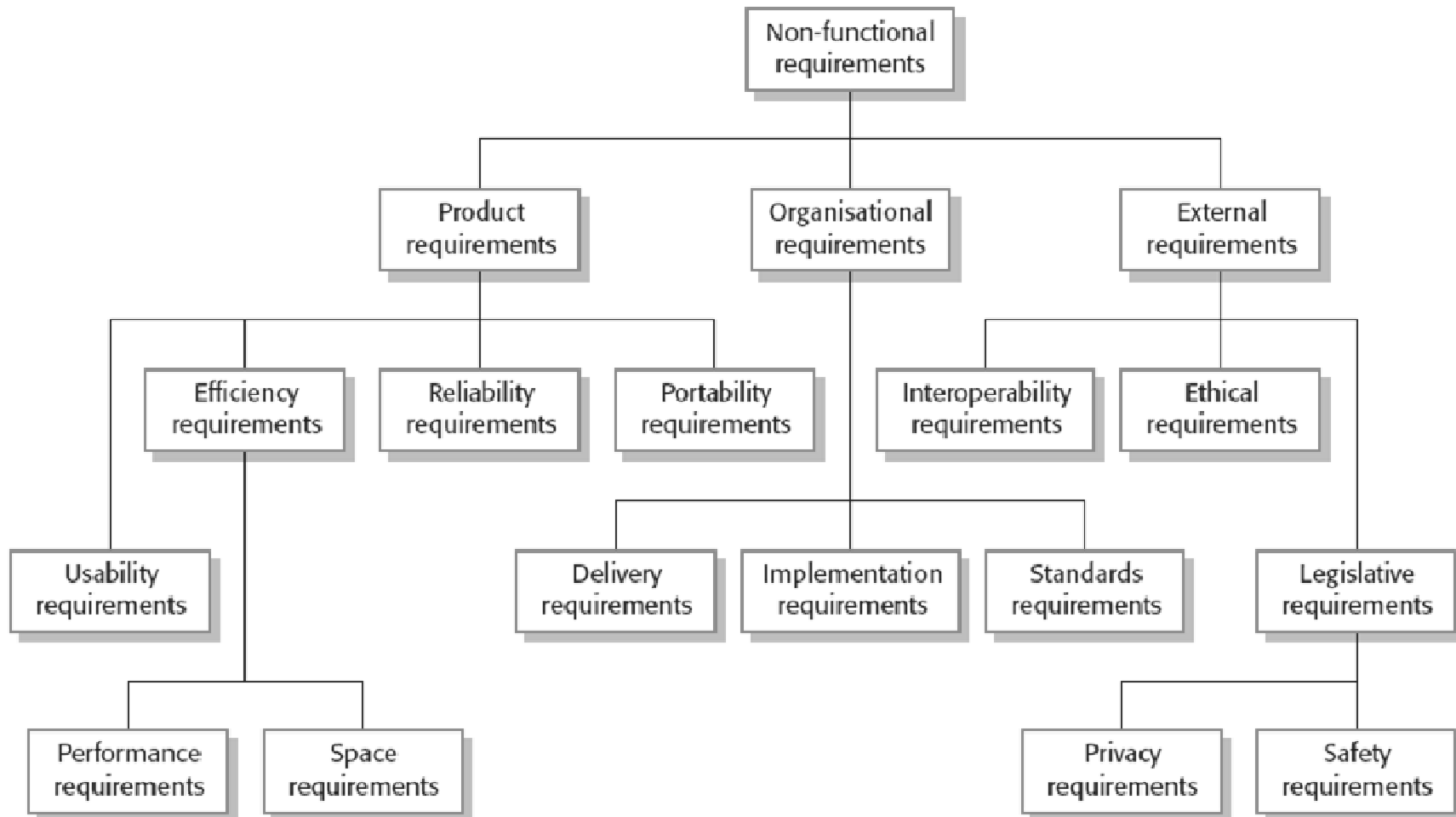
■ Environment constraints:

- Network, devices.
- Interactions with other systems.

Requirement Concepts



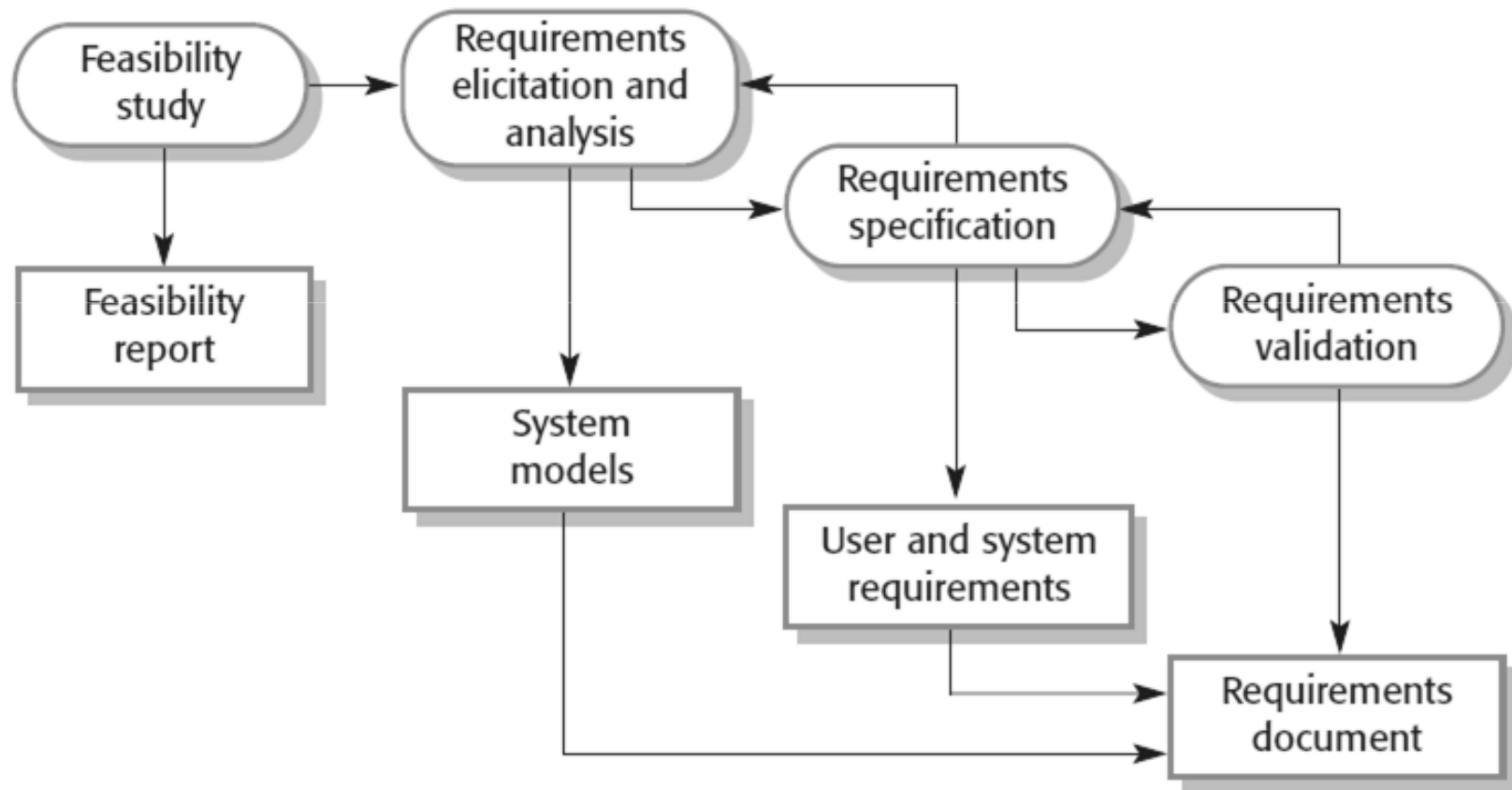
■ Non-function Requirements:



Requirement Concepts



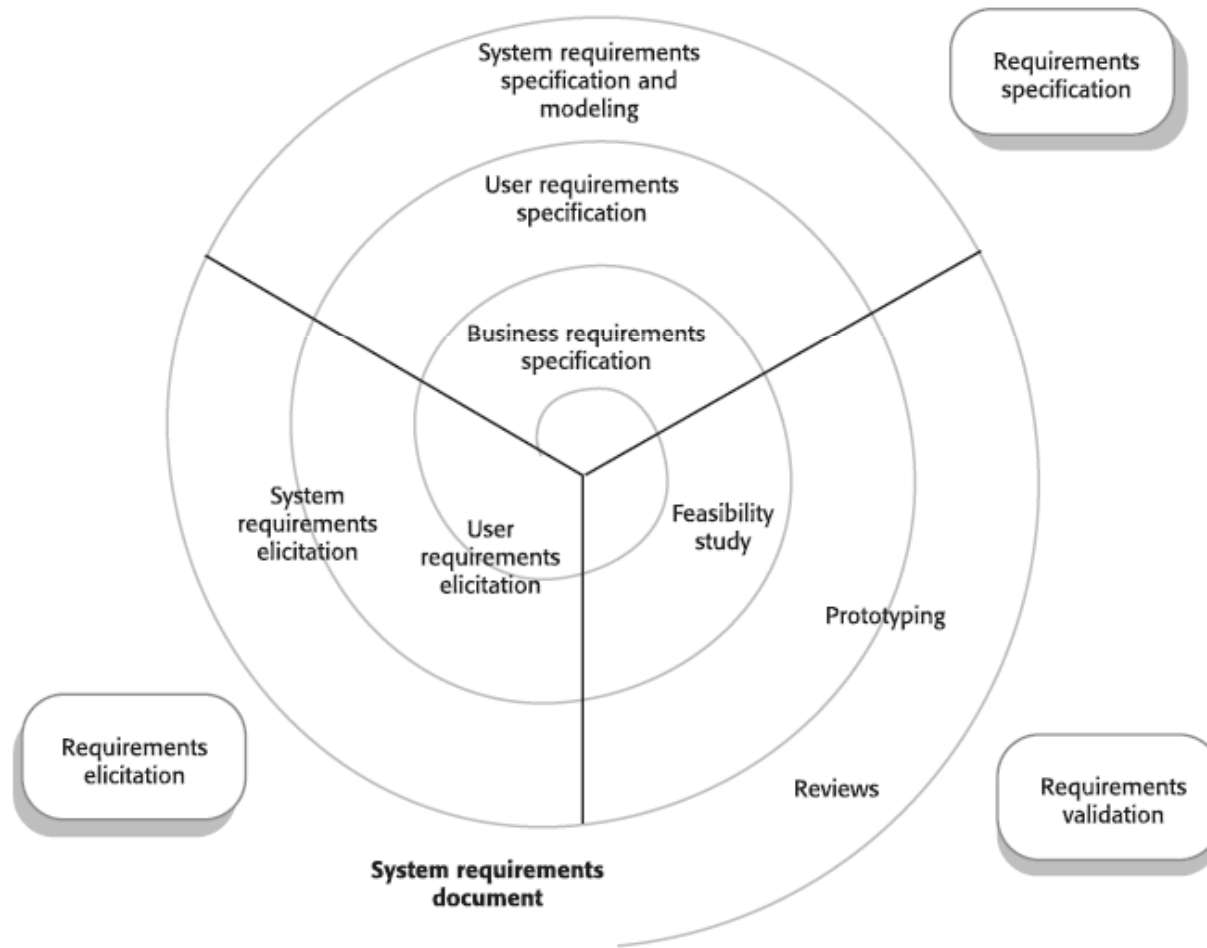
■ Requirements Analysis:



Requirement Concepts



■ Requirements Analysis:



Contents



- Requirement Concepts.
- **Requirement Gathering.**
- Requirement Specification.
- Requirement Validation.



■ Concepts:

■ Transform needs → requirements:

- Discover.
- Make clear.
- Advise.

■ Deliverables:

- Stakeholders list.
- Users requirements (functional & non-functional)..

■ Methods:

- Interviewing.
- Observing.
- JRD meeting.



■ Quality of Requirements:

■ Correctness:

- Avoid ambiguous words.
- Explain business terms and process.

■ Full-fill:

- Working with all stakeholders.
- Make clear non-functional requirements.

■ Consistency:

- No conflicts.
- Group related requirements.
- Negotiate with all stakeholders.



■ Interviewing:

- Discover user requirements directly.

■ Types:

- Direct: in-person, phone, conference.
- Indirect: question form, email.

■ Questions:

- Closed question: limited answers.
- Open question: unlimited answers.

■ Limitations:

- Time & schedule (client).
- Hard to control.
- Hard to find out details.



■ Observing:

■ Goals:

- Discover requirements indirectly.
- Find out business process & problems.

■ Types:

- Visit business places.
- Join a business process.
- Internship.

■ Limitations:

- Time & schedule (development team).
- Interfere client business.



■ JRD Meeting:

- Joint Requirement Development.

- Goals:

- Make clear.
- Find out conflicts and negotiation.

- Organization:

- Meeting with all stakeholders.
- BA control the meeting.
- Iterate each requirement.

- Limitations:

- Schedule (all stakeholders).

Contents



- Requirement Concepts.
- Requirement Gathering.
- **Requirement Specification.**
- Requirement Validation.



■ Concepts:

■ Organize requirements:

- Breakdown users requirements.
- Show relationship amongst requirements.

■ Describe system requirements:

- Functional: interactions of each features.
- Non-functional: constraints.

■ A good specification must have:

- Context.
- Interactions.
- Constraints.
- Testable.

Requirement Specification



■ Specification Methods:

■ Natural language:

➤ Easy to write, but wordy & ambiguous.

➔ Follow form.

Insulin Pump/Control Software/SRS/3.3.2	
Function	Compute insulin dose: Safe sugar level
Description	Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units
Inputs	Current sugar reading (r2), the previous two readings (r0 and r1)
Source	Current sugar reading from sensor. Other readings from memory.
Outputs	CompDose—the dose in insulin to be delivered
Destination	Main control loop
Action: CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.	
Requires	Two previous readings so that the rate of change of sugar level can be computed.
Pre-condition	The insulin reservoir contains at least the maximum allowed single dose of insulin.
Post-condition	r0 is replaced by r1 then r1 is replaced by r2
Side effects	None

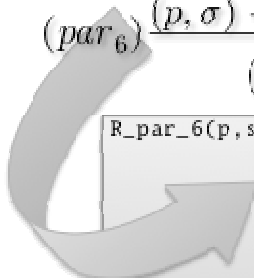


■ Specification Methods:

■ Formal specification:

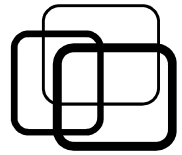
- Mathematical equations & programming languages.
- Pros:
 - Clear & correct.
 - Auto-verification.
 - Auto-generation.
- Cons:
 - Complex.
 - Limit.

$$(par_6) \frac{(p, \sigma) \xrightarrow{m} (p', \sigma'), (q, \sigma) \xrightarrow{n} \checkmark}{(p || q, \sigma) \xrightarrow{m|n} (p', \sigma')}$$



```
R_par_6(p,s) = if( is_par(p) , { a: ActionTransition |  
    isOrderedActionList(ac(a))  
    &&  
    exists a1, a2: ActionTransition.  
    a1 in R( pi_1(p), s)  
    && pi_t(a1) == checkmark  
    && a2 in R( pi_2(p), s)  
    && pi_t(a2) != checkmark  
    && OrderAction(ac(a1) ++ ac(a2)) == ac(a)  
    && pi_t(a2) == (pi_t(a))  
    && sigma'(a) == sigma'(a2)  
}, {});
```

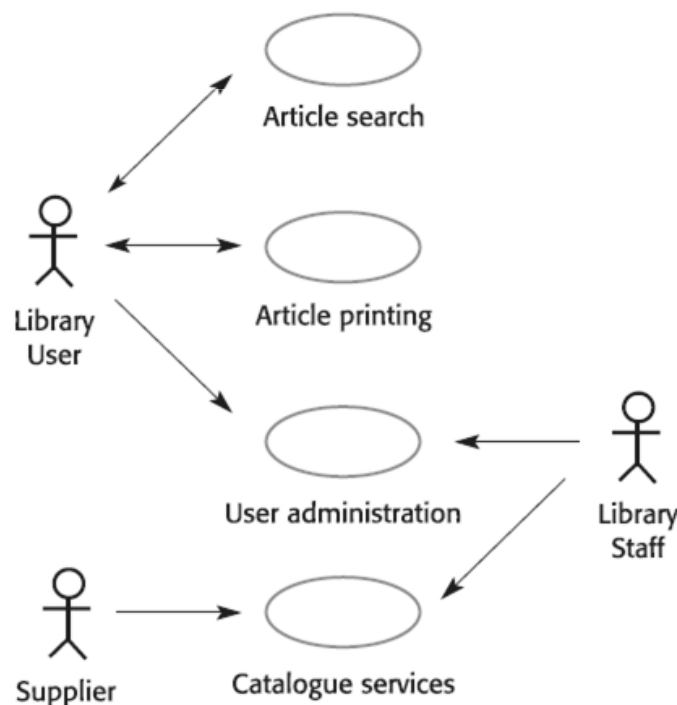
Requirement Specification



■ Specification Methods:

■ Models:

- Diagram + Form spec.
- ➔ Use Case Analysis.



Initial assumption: The user has logged on to the LIBSYS system and has located the journal containing the copy of the article.

Normal: The user selects the article to be copied. The system prompts the user to provide subscriber information for the journal or to indicate a method of payment for the article. Payment can be made by credit card or by quoting an organisational account number.

The user is then asked to fill in a copyright form that maintains details of the transaction and submit it to the LIBSYS system.

The copyright form is checked and, if it is approved, the PDF version of the article is downloaded to the LIBSYS working area on the user's computer and the user is informed that it is available. The user is asked to select a printer and a copy of the article is printed. If the article has been flagged as 'print-only' it is deleted from the user's system once the user has confirmed that printing is complete.

What can go wrong: The user may fail to fill in the copyright form correctly. In this case, the form should be re-presented to the user for correction. If the resubmitted form is still incorrect, then the user's request for the article is rejected.

The payment may be rejected by the system, in which case the user's request for the article is rejected.

The article download may fail, causing the system to retry until successful or the user terminates the session.

It may not be possible to print the article. If the article is not flagged as 'print-only' it is held in the LIBSYS workspace. Otherwise, the article is deleted and the user's account credited with the cost of the article.

Other activities: Simultaneous downloads of other articles.

System state on completion: User is logged on. The downloaded article has been deleted from LIBSYS workspace if it has been flagged as print-only.



■ Use Case Analysis:

■ Goals:

- Describe requirements in details.
- Describe interactions of features.

■ Ideas:

- Software ~ Movie/Play.
- Stakeholders ~ Actors/Actresses.
- Features ~ Scenes.
- Interactions ~ Screenplay/Scripts.



■ Use Case Analysis:

Scene	Actor	Screenplay
Login	Reader/ Librarian	Login successfully scenario - System: request username and password. - User: provide username and password. - User: submit login. - System: validate username and password. - System: redirect user to main screen.
		Login failed scenario <i>At step 4 of success scenario:</i> - System: validate username and password. - System: report error and request user re-login.

Contents



- Requirement Concepts.
- Requirement Gathering.
- Requirement Specification.
- **Requirement Validation.**

Requirement Validation



■ Importance of Validation:

- Understand the right problem:
 - Cause 80% of failed projects.
- Reduce hidden defects pass to next phase.
 - 1 analysis defect ~ 100 implementation defects.

■ Validation Methods:

- JRD meeting.
- Prototype:
 - Sketches & Dummies.
 - Demo & simulations.

STEP 1: Make an Observation

① LOCATION

Pick up: ☒ Airport name or code: ☐ Airport name:

Return: ☐ Airport name or code:

② DATE

Pick up:

Return:

③ Car type: ☒ Yes ☐ No

SUBMIT



■ Software Requirement Specifications:

■ “Student Management System”.

■ Roles:

➤ Customer:

- Identify stakeholders.
- For each stakeholder, list at least 2 user needs.

➤ Requirement Gathering:

- Transform user needs into user requirements.

➤ Requirement Specification:

- Choose 2 user requirements.
- Create 2 system requirements by use case analysis.

➤ Requirement Validation:

- Draw prototypes for 1 system requirements.

