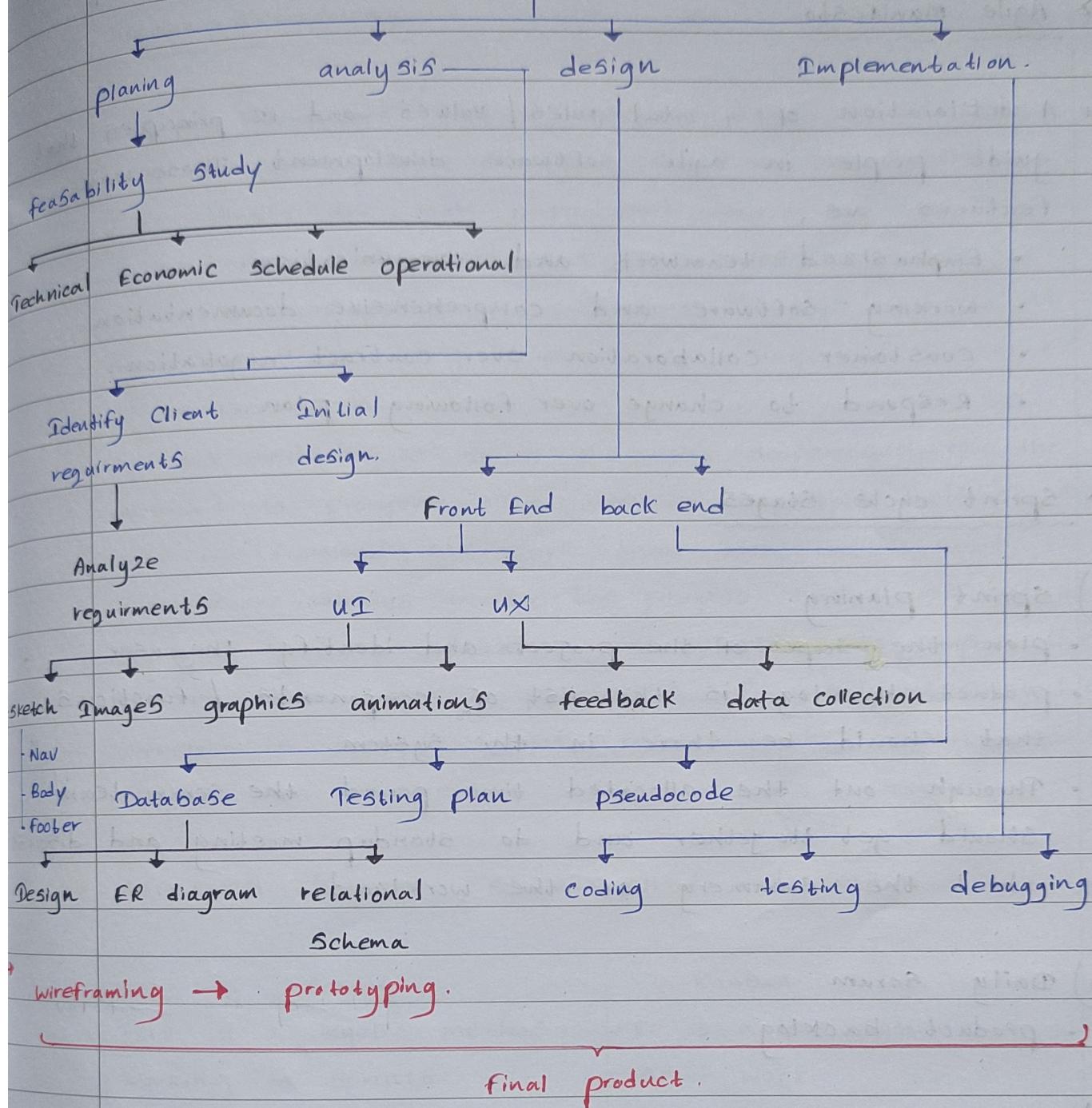


Example for work Breakdown Structure (WBS)



Website development project

from 17.03.2023



Name

STID

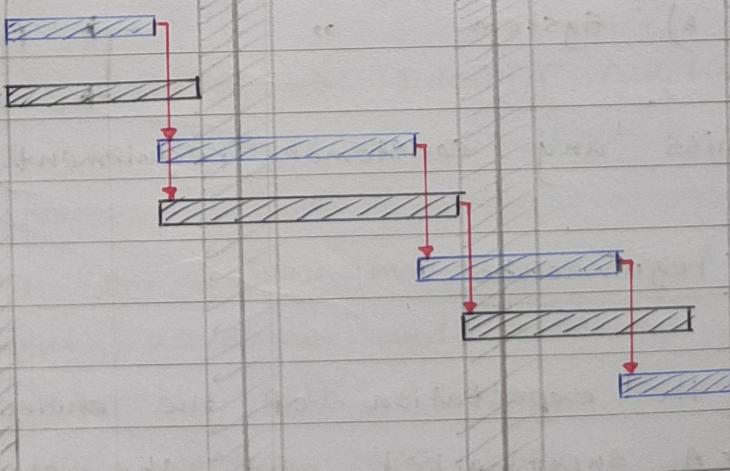
S.C.S. Sandanayake	25002
G.D.P. De Zoysa	25076
G.L. Sandeepa	25619
A.R. Jayawickrama	26136
G.D. Jinasena	25947
K.B.R. Jayaweera	26088

> Gantt chart

- used for managing the time of a project. It can be use in any project to show the time plan with activities and dependancies.
- we create a gantt chart before starting a project.
- A type of bar chart that illustrates a project schedule.
- main columns in the table are ID, Task name, predecessors (dependancies) and duration. other section is chart (bar chart)
- Each task name has an ID to identify it.
- predecessor is the task or ID number that needed to be completed before start the present task.
- In gant chart we consider weekend dates as working days and represent in bar chart.
- Because , we need to identify the end date of the project and project should be end according the deadline.
- but weekends are ~~not~~ consider as holidays in IT firm.

> Exercise

ID	A	P	D	May 1	May 8	May 15	May 22						
	S	M	T	W	T	F	S	S	M	T	W	F	S
1	a	-	4										
2	b	-	5										
3	c	2,3	5										
4	d	1,2	6										
5	e	2,3	3										
6	f	4	4										
7	g	5	3										
8	End	6,7											



> Requirement Determination

- To perform a comprehensive analysis about the system we need to do three main things.
 - 1) collect data
 - 2) Analyse the data
 - 3) coming up with an initial plan , design for the product.
- At the end of this phase we give system proposal and the deliverable.

1) Gathering the requirements.

- Requirements are the expected functions that should be included in the application.
 - mainly collecting the needs and ideas of the client .
 - Know his expectations beforehand to give him / her a product that will satisfy them.
 - There are 3 main requirement types to be considered.
 - 1) Business requirement
 - 2) User "
 - 3) System "
 - Never miss any customer requirement .
- functional requirements
→ non-functional "

1) business requirements

- This is the expectation of the entire organization .
ex:- * A supermarket and the owner needs a system to manage their activities . And they plan to increase the profit and sales . This is a business requirement .
- * Reach large customer base .

2) User requirements.

- user requirements are each and every individual expectation.
ex:- * cashier in a supermarket need to generate sales report at the end of the day.
* waiting to login to the system. (Log In)

3) System requirements.

- user is expecting something and how the system can provide.
- facilitating user requirement is called as a System requirement.

i) functional requirements.

* functionalities or activities that a system can do (features / behaviour) is called functional reg.

ii) non-functional requirements.

* characteristics of the system. called non-functional.

> Flow,

Gather user requirements → Identify functional requirements
→ convert it to a system as their expectation.

ex:- user Log In (have many functional reg.).

- Input username and password
- verify it with existing database.
- finally confirm the user
- Should display message either successful or invalid.

Summary .

- user req. and functional req. go hand in hand . Each user requirement is co-relate to functional requirement.
- non-functional requirements are characteristics .
ex:- • Images
• speed of the system
• Languages in the system.
• colours and design
• Security standards need to be maintain ...
- Business requirements are important it directly affects on the success of the product or service in business
- The workload needed to be minimum than before after implementing a system that client needed.
- when writing the functional requirements you need to write both function & how it will perform
- functional requirements can be information and data gathering needs.
- If we take a numbewise comparison,
 - 1) business reg. → lowest
 - 2) user reg. → 2nd highest
 - 3) System reg. → highest number of requirements.
- Exercise (slide)

business	user	functional	non-functional
1, 3, 13 15, 14, 16 17	8, 18, 19 7, 20, 9	11, 12, 4 5	2, 10, 6

> Requirement elicitation (gathering) techniques.

- There are many requirement gathering techniques but only the most common ones will be discuss.

1) Interview method.

- discussion between 2 persons. Interviewer and client (Interviewee).
- ASK questions and get details from the Client.
- Interview types → 1) face to face
2) Remote
- Question types
 - 1) close ended → has direct answers and only one clear answer is expected.
 - 2) open ended → have long answers with many information and description

2) questionnaire method.

- List of questions will be distributed.
- There is a time limit for a session.
- can't guarantee whether all the participants answer or not. that is the main limitation.
- Difficult to compare the answers.
- Answers depend on the way the participants understand the question.
- Validity of the answers can be less cause, there can be anonymous answers.
- Don't make lengthy questionnaires, try to make it short and include all necessary points.
- Can apply for large group of people.

for best practice,

- Don't include all the questions in one pic, responders won't willing to answer all.
- Don't add important questions at the end.
- Don't give all the acronyms (shorten forms) at the begining.
- categorize the questions.
- designed it properly.
- Don't make any lengthy questions.

3) observation method.

- Have to go to the particular place physically.
- Time consuming method.
- need to put lots of effort.
- won't be able to watch / observe future plans and historical data.
- Realistic nature of people can't be observe.
- won't be able to identify the weakness of the current process. cause they are giving 100%.

4) Document review method.

- collecting data from existing document.
- ex:- bills, monthly reports, annual reports
- difficult to collect future oriented data.
- No way of clarification. your doubts.
- Time consuming.
- Applied as combination of methods. " Interview + document

5) JAD method (Joint application development)

- you are conducting a discussion same as interview method.
- participating as a group for discussion.
- group of people provide their feedback.
 - 1) clients
 - 2) technical team
 - 3) JAD facilitator.
- JAD facilitator provides required infrastructure and location.

ex:- white screen, projector, white board,
group support server.

- All of the participants can see each other.
- have short time period.
- can validate with the team.
- you can reach large group of people when using this method.

questions.

- 1) create a list of advantages and disadvantages of requirement collection methods.
- 2) compare and contrast the features of these methods.

exercise.

- 1) List down user categories of daraz?
 - 2) Select 'buyer' category
 - 3) Identify 5 user requirements of above category
 - 4) pick 1 and listdown functional requirements.
-
- 1) buyer , Sellar , Admin-user , Delivery partners , Bankers

3) user requirements

1. Login
2. Sign up
3. place order
4. return and refund
5. provide feedback
6. make payments
7. view products
8. Trackers

4) functional requirements,

- Login →
1. use phone number or e-mail to login
 2. or FB or google account.
 3. password to login
 4. password recovery using phone number / mail

> Requirement analysis.

- Requirement analysis means evaluate the requirements to mapping user requirements, flow.
- And also check whether there is any dependencies.
Ex:- If we want to log into a system there should have sign-up option.

- And also should identify the flow of a system it means the behaviour or overall flow of the system.
- To do a requirement analysis we should do requirement modeling.
- To perform requirement modeling we can use design diagrams.
- To analyse we can use following 2 methods,
 - 1) use case analysis →
 - These are technical parts (High number of Information).
 - 2) user stories →
 - These use to communicate with clients there are not much technical staff.
- Based on number of information provide we can devide two types of use case,
 - 1) basic (Limited Info)
 - 2) Advanced (more technical stuff)

User Stories	use case.
<ul style="list-style-type: none"> • Short descriptions • requirements who and why • General guidance • No technical details 	<ul style="list-style-type: none"> • Short or lenght description • use flow or interaction • In-depth guidance • Detailed to write.

> Basic usecase.

Example: using previous example we can create basic usecase.

- 1) Name :- Login
- 2) Brief description :- user perform login to access their account
- 3) Actors :- Buyer, Seller
- 4) Basic flow :-
 - 1) Enter phone number
 - 2) Enter password
 - 3) Click login button
- 5) Alternative flow :-
 - 1) login using FB or google
(extensions)
 - 2) Recovering the password
- 6) pre-conditions :- he should be a registered user
(user should sign-up to the system)
- 7) Post conditions :- user should successfully login to the account. (get access)

- Scope of the work will depend on the team members.
- use case or 1 component → sprint in agile method
'Scrum method'

➤ Explain basic usecase terms.

- 1) Name :- • user case name
- 2) Brief description :- • giving an idea to the other parties of the requirement function.
- 3) Actors :- • who are going to access this
• user categories that going to use the particular usecase.
• which parties will need this function.
- 4) Basic flow :- • The main activity flow the actor has to perform to achieve the objective.
- 5) Alternative flow :- • Extended activity flow need to do
• Extensions.
- 6) Pre-conditions :- • pre conditions are the pre requirements the actor should fulfill to perform the requirements or user case.
- 7) Post conditions :- • The state the user reach after performing the user-case or the result

- Exercise 01.

* write a description for a usecase where the user search for item in an online sales site.

- 1) Name :- Search item
- 2) Description :- To lookup items available on the website.
- 3) Actors :- customers / buyers
- 4) Basic flow :-
 - 1) Login to website
 - 2) Enter item name in search bar
 - 3) Click search bar.
- 5) Alternative flow :-
 - 1) use filters
- 6) pre-condition :- * item should be available in the system
- 7) post-condition :- * customer get searched item successfully

- Exercise 02.

* making a doctor appointment.

- 1) Name :- make appointment
- 2) Description :- make reservation for a doctor
- 3) Actors :- Receptionists
- 4) Basic flow :-
 - 1) Access to the system
 - 2) Enter patient details
 - 3) check availability
 - 4) make reservation (date and time)
- 5) Alternative flow :-
 - 1) can cancel the appointment
 - 2) can change the new item slot
- 6) pre-condition :- * Receptionist should be a registered to the system.
* Doctor details should be available in system
- 7) post-condition :- * successfully making the appointment

➤ use case modeling

- design diagram show the layout of the system
- Behavioral diagram shows how the system works.

➤ unified modeling language (UML)

- UML is the standard set of model constructs and notations.
- This can be used if you are following OOP concepts.
- It gives you collection of diagrams.

UML diagrams.

Behavioral

- This shows the functionality of the system.
- represent dynamic relationship among the objects.

- use-case diagram
- Activity diagram
- Sequence diagram

Static (Structure)

- This use to show the structure of the system.
- for represent data and static relationship.

- class diagrams
- state transition diagrams

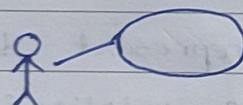
- we can show different angles and perspectives using UML diagrams
- It shows the functionality, processes, who is performing, dependencies and etc....
- It represent dynamic relationships.

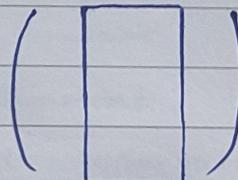
1) use-case diagram.

- A use-case describes how an external user triggers an event to which the system must respond.
- A use-case diagram summarizes all of the use cases together in one picture.
- use case describes events triggered by an actor.

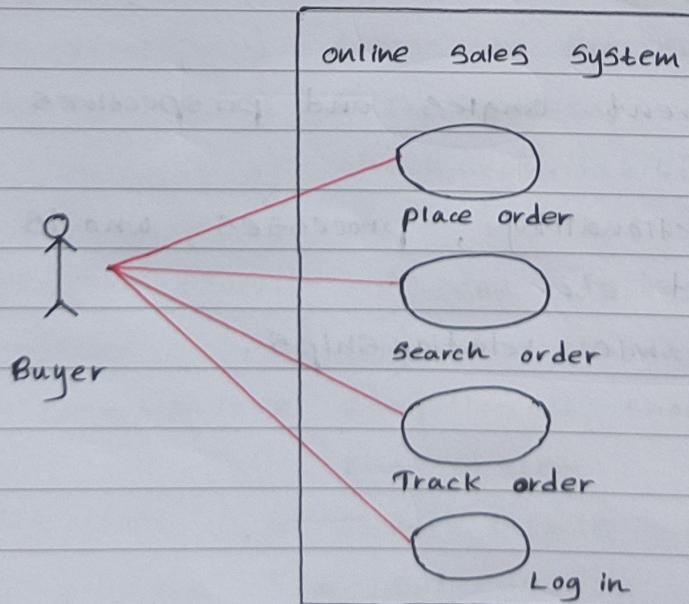
Symbols.

- 1) Actor () → This shows the user role
- 2) use case () → Shows the interaction or event
- 3) Association () → The relationship between use case and actor.

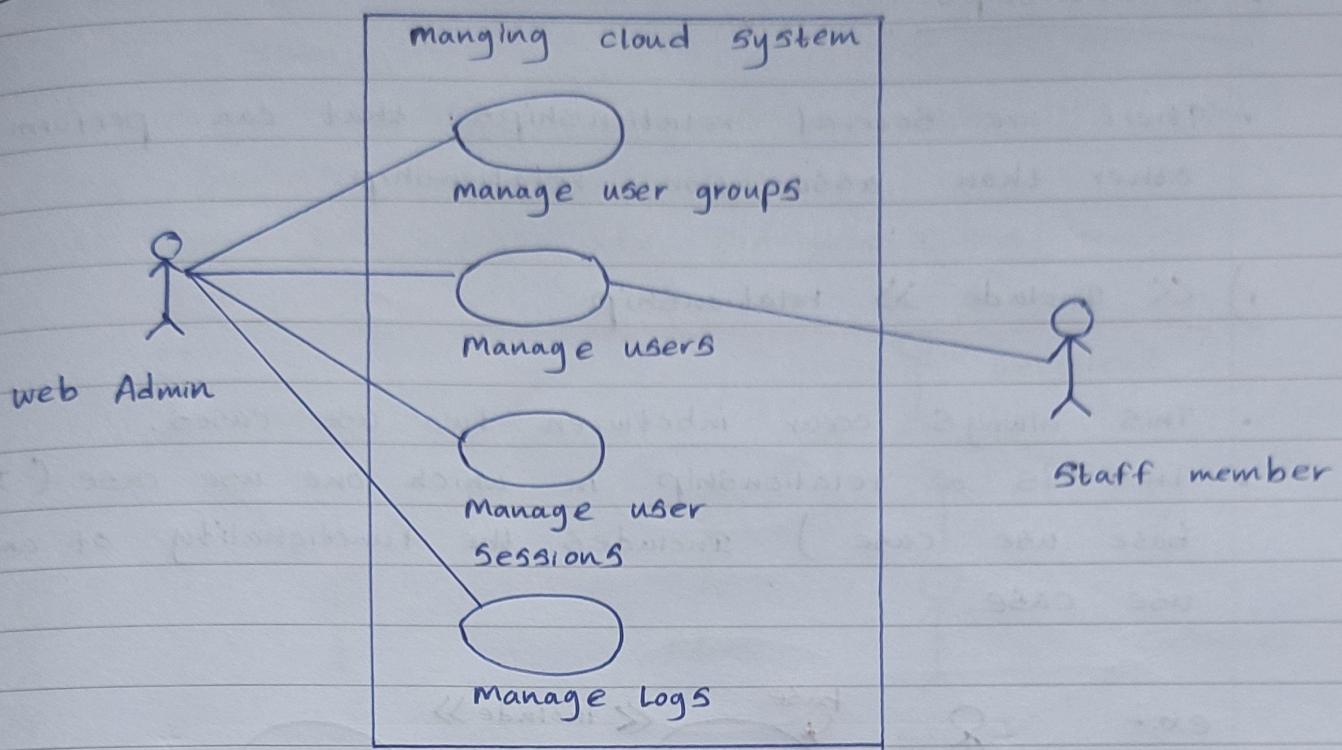


- 4) System boundary () → Subject
→ Shows the boundary of the system.

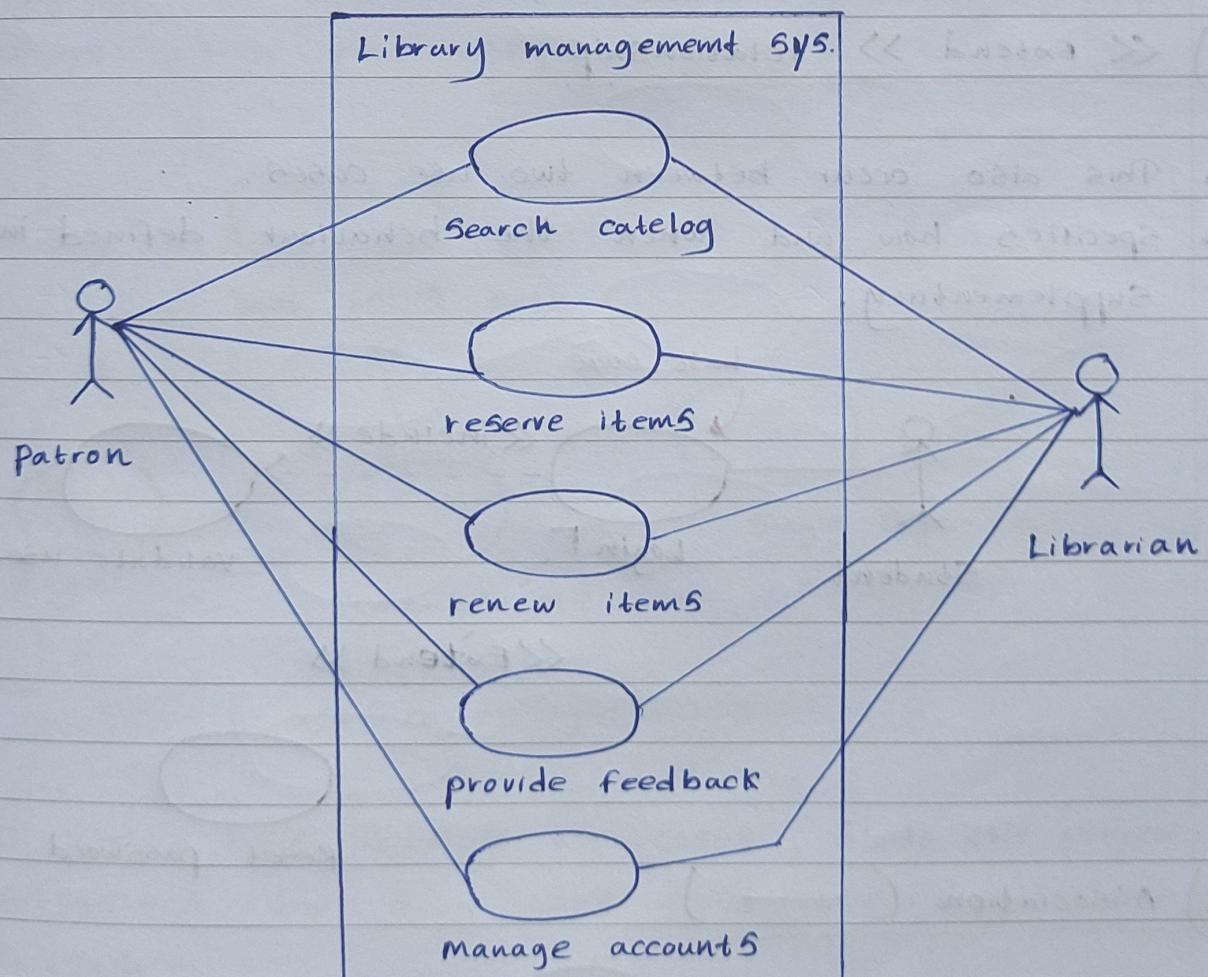
ex:-



Exercise 01



Exercise 02.

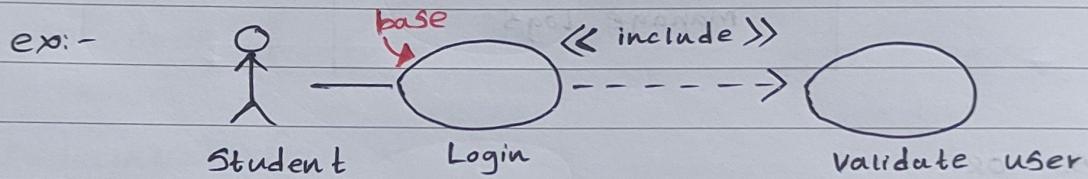


> Relationships.

- There are several relationships that can perform other than association relationship.

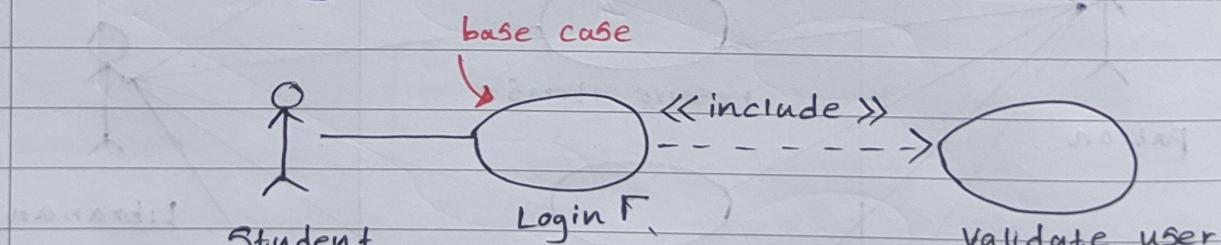
1) « Include » relationship

- This always occurs in between two use cases.
- This is a relationship in which one use case (the base use case) includes the functionality of another use case.



2) « Extend » relationship

- This also occurs between two use cases.
- Specifies how and when the behaviour defined in usually supplementary.

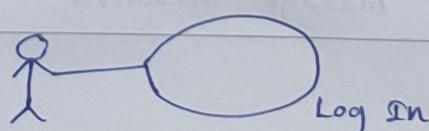


<< Extend >>



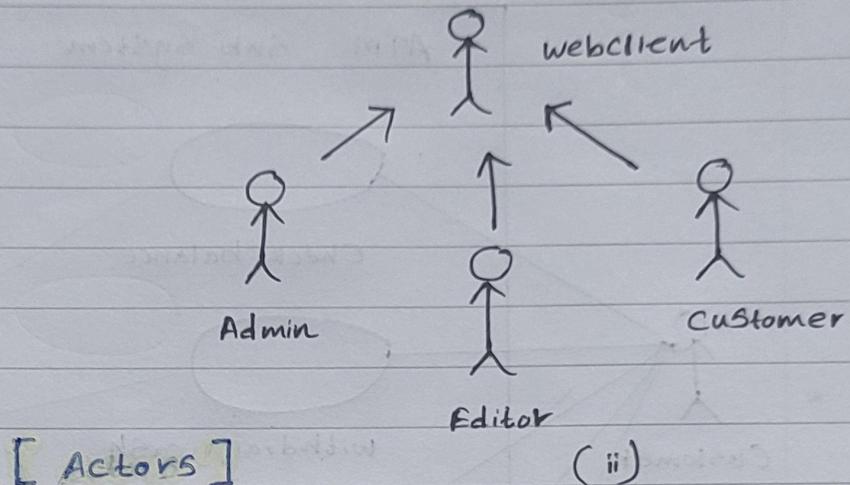
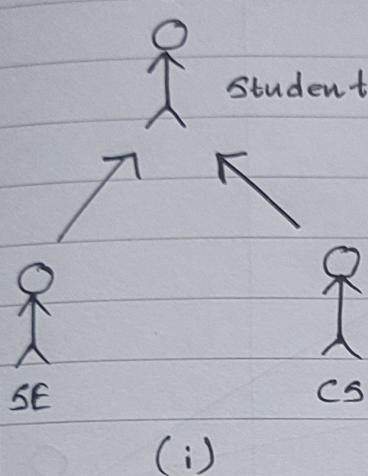
Reset password

3) Association (—)

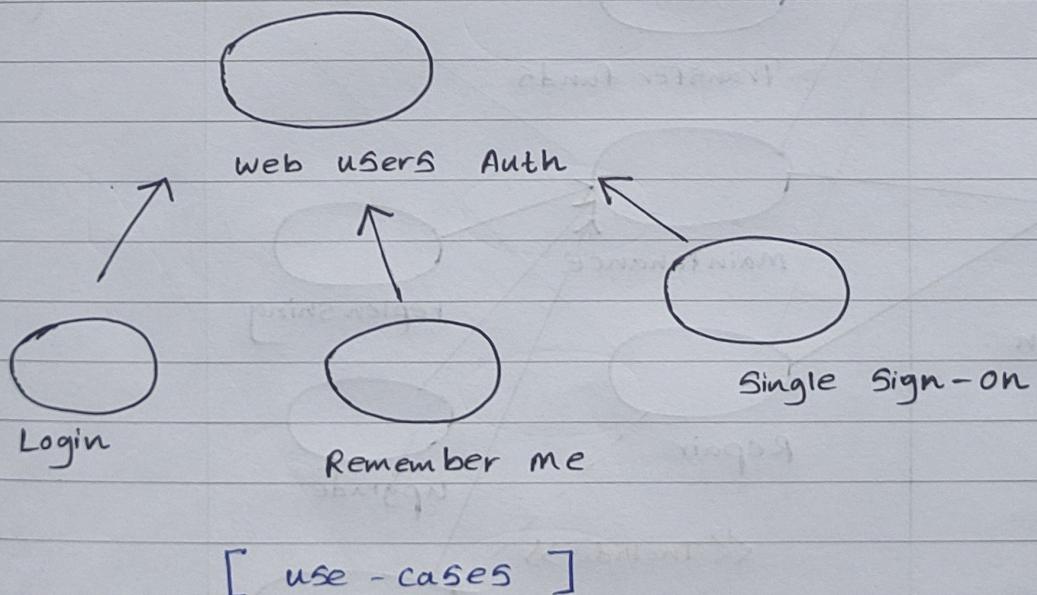


g) generalization (specialized behaviour)

- Abstract or concrete actors and specialize them using generalization relationship.
- Some can be applied to the use cases.



- Abstract or concrete use-cases specialized using generalization relationship.



* If project have multiple use cases , divide the project into Subsystems and draw each use-case diagram to each Subsystem.

- Exercise

- An automated teller machine (ATM) in a banking system.

Actors → customer , Bank actor , technician

