

Subject Name
Software Engineering

Topic : Requirement Engineering

Department of Computer Engineering & Applications

Requirement Engineering

Requirement Engineering is mainly focused on discovering -

- what is to be developed?
- How it should be developed?

Main aspects are:

- What does the customer want?
- What does the customer require in order to use the system?
- What will be the software's impact on user be?



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with **A** Grade by **NAAC**

What requirement process ensure?

It ensures that our software must meet the user expectations and ending up with a high quality software.

It's a critical stage of the software process as errors at this stage will reflect later on the next stages, which definitely will cause you a higher costs.

What should we do when we are going to develop a software?

To develop the software system we should have clear understanding of Software system.

To achieve this we need to continuous communication with customers to gather all requirements.

Requirement Engineering Process

It is a four step process, which includes:

**Requirement
Elicitation**

**Requirements
Analysis**

**Requirement
Documentations**

**Requirements
Review**

Requirement Elicitation

It is the process to find out the requirements for an intended software system by communicating with client, end users, system users and others who have a stake in the software system development.

Requirement Analysis

- It is a process to determine the expectation of user.
- Requirement must be detailed and relevant
- These requirements are also called functional specification of a software.

Requirement Documentations

- Generally we can say that software requirement document is a description of features and functionalities of targeted application/software
- Software requirement documents specify that what the software/application will do.

Requirement Review

- It is a review process to ensure that all requirements are identified properly.
- It is a process in which both parties involve.

Types of requirements

Known Requirements

Something on which the stakeholder believes to be implemented

Unknown Requirements

Forgotten by stakeholders as they are not needed right now.

Undreamed Requirements

Stakeholder may not be able to think due to lack of domain knowledge

Feasibility study

What is feasibility study?

It is a analysis that accounts all the factors, i.e economic, technical, legal and others with are necessary for the completion of a project successfully.

Feasibility study focuses on

- Is the concept of project is workable
- Should we proceed with the proposed project idea
- To find out the estimated cost and schedule of project
- Identification of all the major risks
- Stability of requirements
- Identification of new possibilities through investigative process

References

1:K.K. Aggarwal &Yogesh Singh

2:<https://www.tutorialspoint.com/>

Software Requirement Specification

It is a description of a software system to be developed.

A **software requirements specification** (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements.

The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system.

The software requirement specification document consistent of all necessary requirements required for project development.

To develop the software system we should have clear understanding of Software system.

To achieve this we need to continuous communication with customers to gather all requirements.



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with

A

Grade by **NAAC**

Requirement Elicitation

Requirements Elicitation is the process to find out the requirements for an intended software system by communicating with client, end users, system users and others who have a stake in the software system development.



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with

A

Grade by **NAAC**

Requirement Elicitation Process

**Requirement
gathering**

**Requirement
Organization**

**Negotiation
& Discussion**

**Requirement
Specification**

Requirement Elicitation Process

■ Requirements gathering

The developers discuss with the client and end users and know their expectations from the software.

■ Organizing Requirements

The developers prioritize and arrange the requirements in order of importance, urgency and convenience.

Requirement Elicitation Process(cont..)

■ Negotiation & discussion

If requirements are ambiguous or there are some conflicts in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders. Requirements may then be prioritized and reasonably compromised

■ Requirement Specification

Software requirement specification(SRS)is a detailed description of a software system with its functional and non functional requirements.

Functional Requirements

- It specifies the behavior of a system.
- What software system will do.

Non-Functional Requirements

- It is related to the quality of a system.
- How software system will do.



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with

A

Grade by **NAAC**

Non-Functional Requirements(Contd..)

Generally Non-functional requirements fall into many areas some of them are as given below.

Efficiency

Usability

Availability

Portability

Flexibility

Maintainability

Requirement Elicitation Techniques

Interviews

- Interviews are strong medium to collect requirements. Organization may conduct several types of interviews
 - **Structured (closed) interviews**
 - Where every single information to gather is decided in advance
 - Follow pattern and matter of discussion.
 - **Non-structured (open) interviews**
 - Where information to gather is not decided in advance These are more flexible and less biased.

Interview(cont..)

- Oral interviews
- Written interviews
- One-to-one interviews

which are held between two persons across the table.

- Group interviews

Held between groups of participants.

Help to uncover any missing requirement as numerous people are involved.

Brainstorming

- It is a group technique
- So many peoples are involved in this activity so lots of new ideas are shared
- Every idea/view is documented properly so that every one would be able to see these ideas.

Requirement Elicitation Techniques

Brainstorming

- A detailed report of all is prepared which consist list of requirements and their priority.
- As it's a group activity so there are chances of group conflict and bias. To handle such circumstances a highly trained facilitator is required.

Requirement Elicitation Techniques

FAST

- It stands for **Facilitated Application Specification Technique**.
- Due team oriented nature of this approach it is somewhere similar brainstorming.
- Main motive of this approach is to fill the gap between the customer and developer.
- Each participants has to prepare his/her list.
- All different lists are combined and redundant lists are removed.
- Sub teams are made for mini specifications.

Requirement Elicitation Techniques

QFD

- It stands for **Quality Function Deployment**.
- Main priority of this technique is customer satisfaction.
- It is obvious ,if customer requirements are fulfilled then customer will be satisfied.

Requirement Elicitation Techniques

QFD(Cont..)

There may be various types of customer requirements.

Normal Requirement

In it all the requirements of software (which is to be developed) are discussed with customer.

Expected requirements

Requirements which are not explicitly stated by customer.(e.g:Security from unauthorized access)



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Accredited with



Grade by **NAAC**

Requirement Elicitation Techniques

QFD(Cont..)

Exciting Requirement

These are beyond the expectation of user.
In it developer adds some unexpected features to make customer more satisfied.

References

1:K.K. Aggarwal &Yogesh Singh

2:<https://www.tutorialspoint.com/>

Use case diagram

- Use case diagrams are graphical representations
- Use case diagram is the primary form of system/software requirements for a new software program underdeveloped.
- Use cases specify the expected behavior (what), and not the exact method of making it happen (how).
- Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram).
- A key concept of use case modeling is that it helps us design a system from the end user's perspective.
- It is an effective technique for communicating system behavior in the user's terms.

Use case diagram(cont..)

- Use case diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:
- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts

Components of use case approach

Actor

An actor or external agent, lies outside the system model, but interacts with it in some way.

Actor-----> Person/Machine/Information System

Actor may be distinguished as primary actor and secondary actor

Primary Actor: Primary actor is one having a goal requiring the assistance of the system.

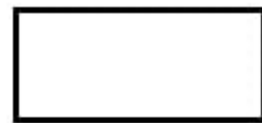
Secondary Actor: Secondary actor is one from which system needs assistance.

Actor

Actor has responsibility toward the system (inputs) and actor has expectations from the system (outputs).

Actors can be a human user, some internal applications, or may be some external applications.

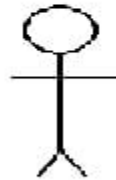
Notations



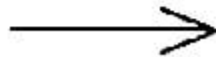
System



use case



Actor

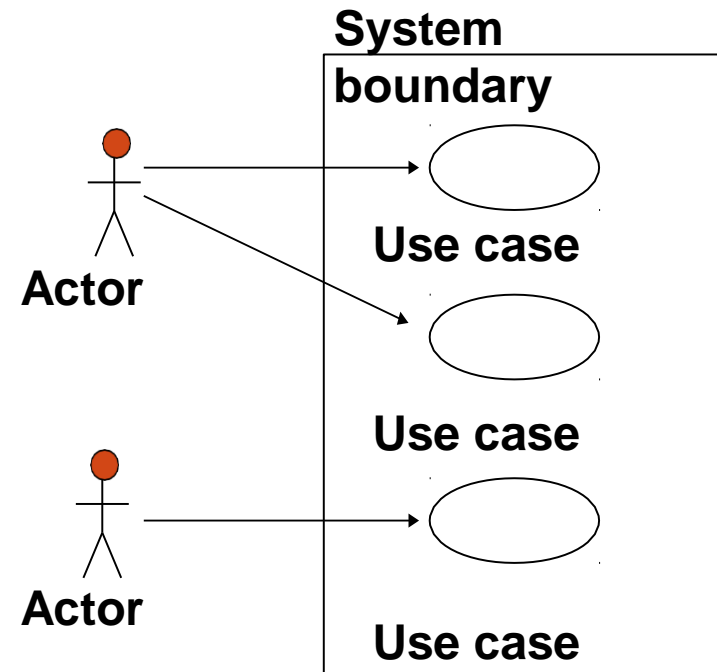


Relationship

Components of use case diagram

- **Actor** : Who interacts with the system
- **Use case** : Functionality and services provided by the system
- **Relationship** : Relation between actor and use cases
- **System boundary**

- Actors appear outside the rectangle.
- Rectangular box represent the System
- Use cases within rectangle providing functionality.
- Relationship association is a solid line between actor & use cases.



Components of use case approach(cont..)

Use Case

A use case is initiated by a user with a particular goal in mind, and completes successfully when the goal is satisfied.

It describes the sequence of interactions between actors and the system necessary to deliver the services that satisfied the goal.

So, we can say that

A use case captures who(actor) does what (interacting) with the system, for what purpose(goal),without dealing with system internals.

Components of use case approach(cont..)

Communication link

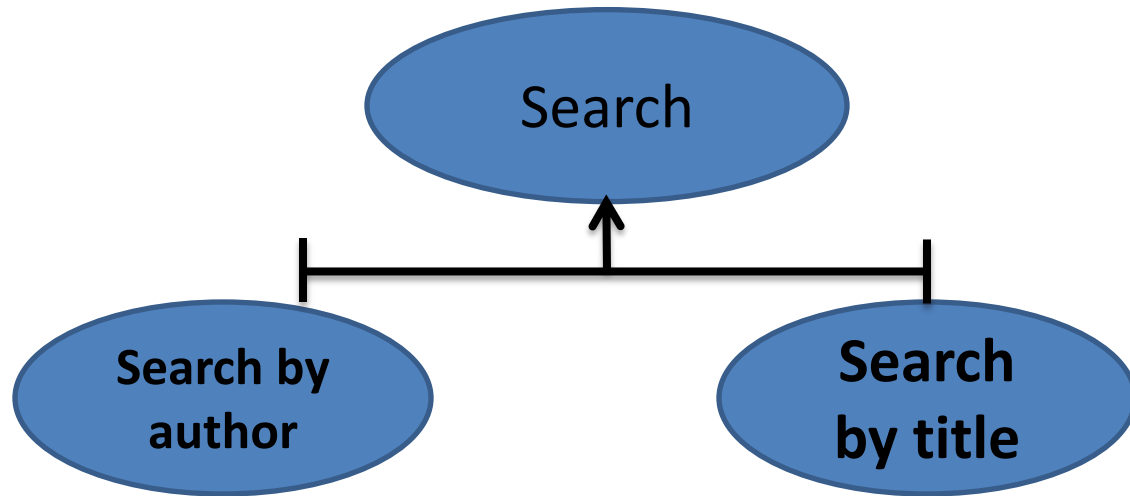
The participation of an actor in a use case is shown by connecting an actor to a use case by a solid link.

Boundary of the system

The system boundary is an entire system.

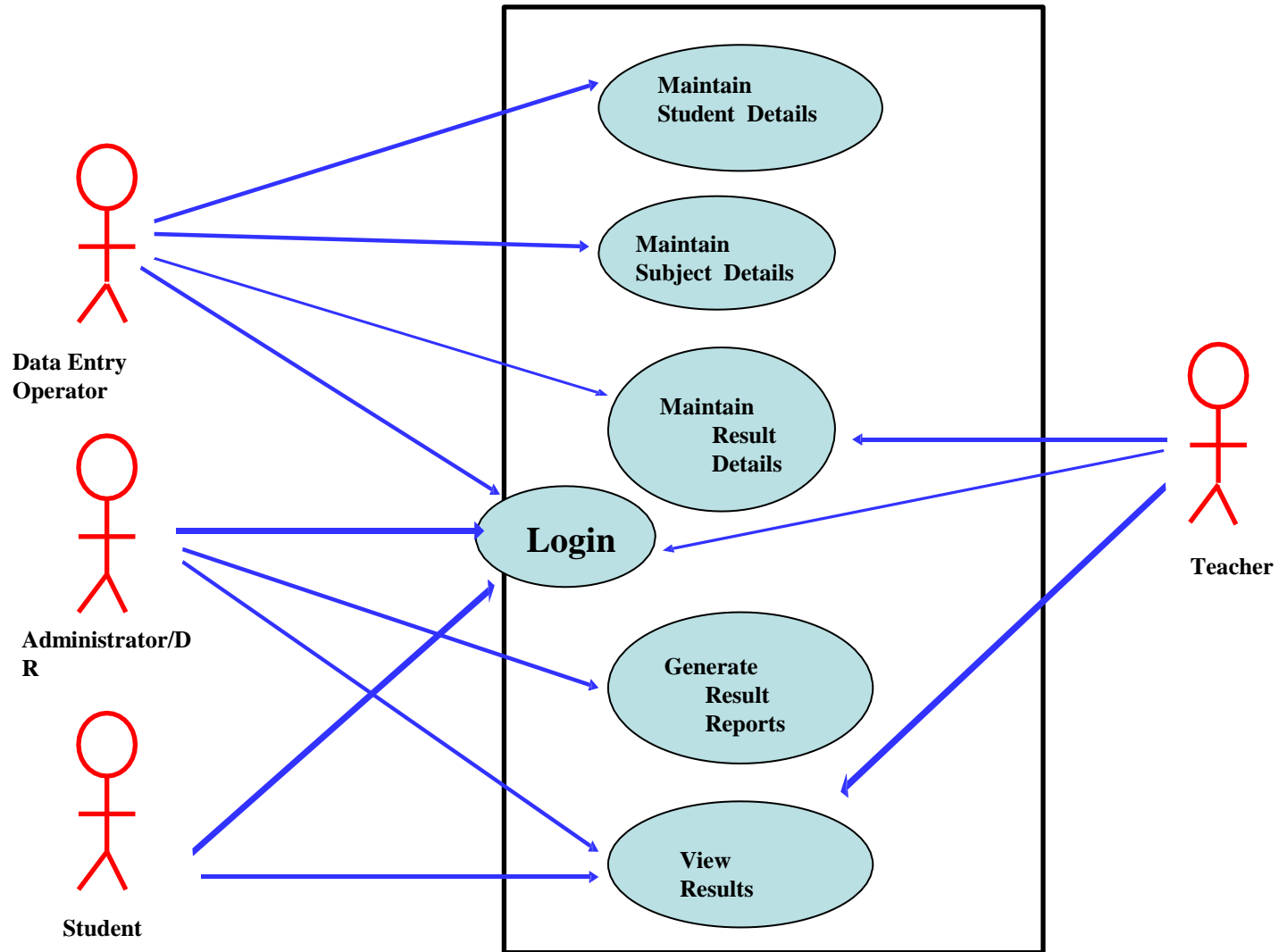
Use Case Example

Generalization Relationship

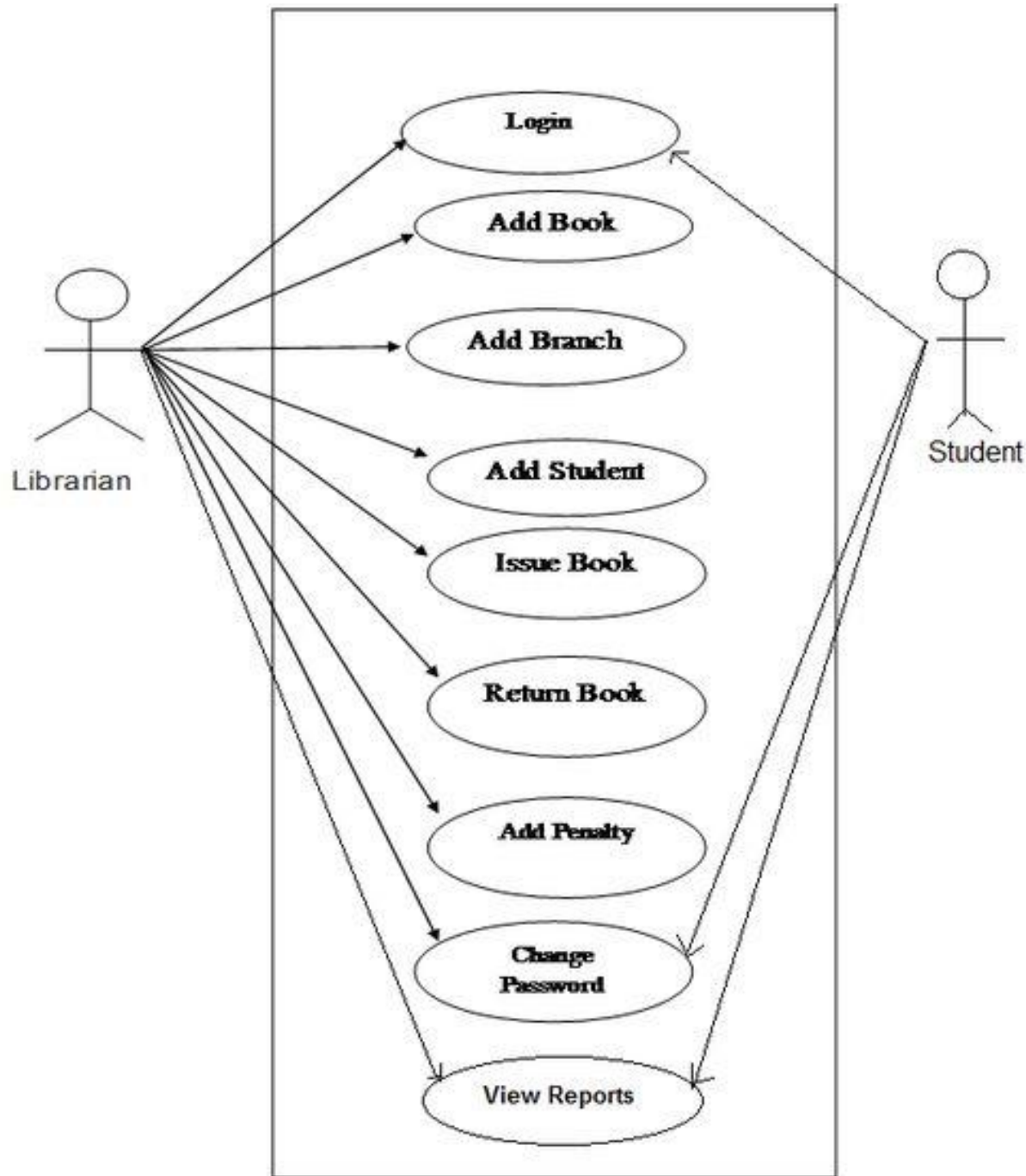


A generalization relationship means that a child use case inherits the behavior and meaning of the parent use case.

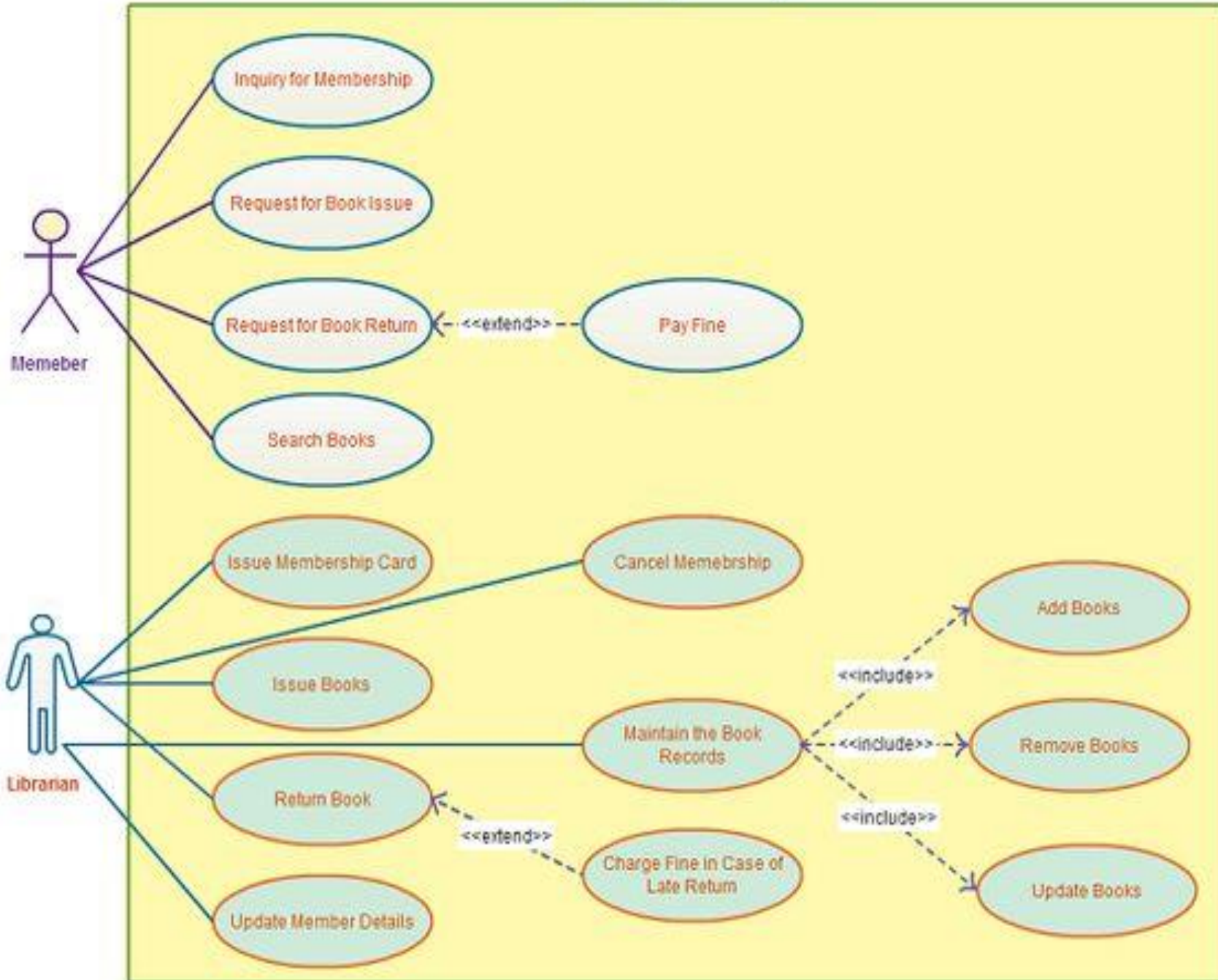
Use case diagram for Result Management System



Use case diagram for Library Management System



Library Management System



How to Identify Actor

Who uses the system?

Who installs the system?

Who starts up the system?

Who maintains the system?

Who shuts down the system?

Who gets information from this system?

Who provides information to the system?

How to Identify Use Cases?

What functions will the actor want from the system?

Does the system store information? What actors will create, read, update or delete this information?

Does the system need to notify an actor about changes in the internal state?

Are there any external events the system must know about? What actor informs the system of those events?

Representation

A case is often represented in a plain text or a diagram. Due to the simplicity of the use case diagram, it is considered to be optional by any organization

Example

Here we will discuss the case for 'Login' to a 'School Management System'.

Use Case Name ----- Login

Use case Description----- A user login to System to access the functionality of the system.

Actors-----Parents, Students, Teacher, Admin

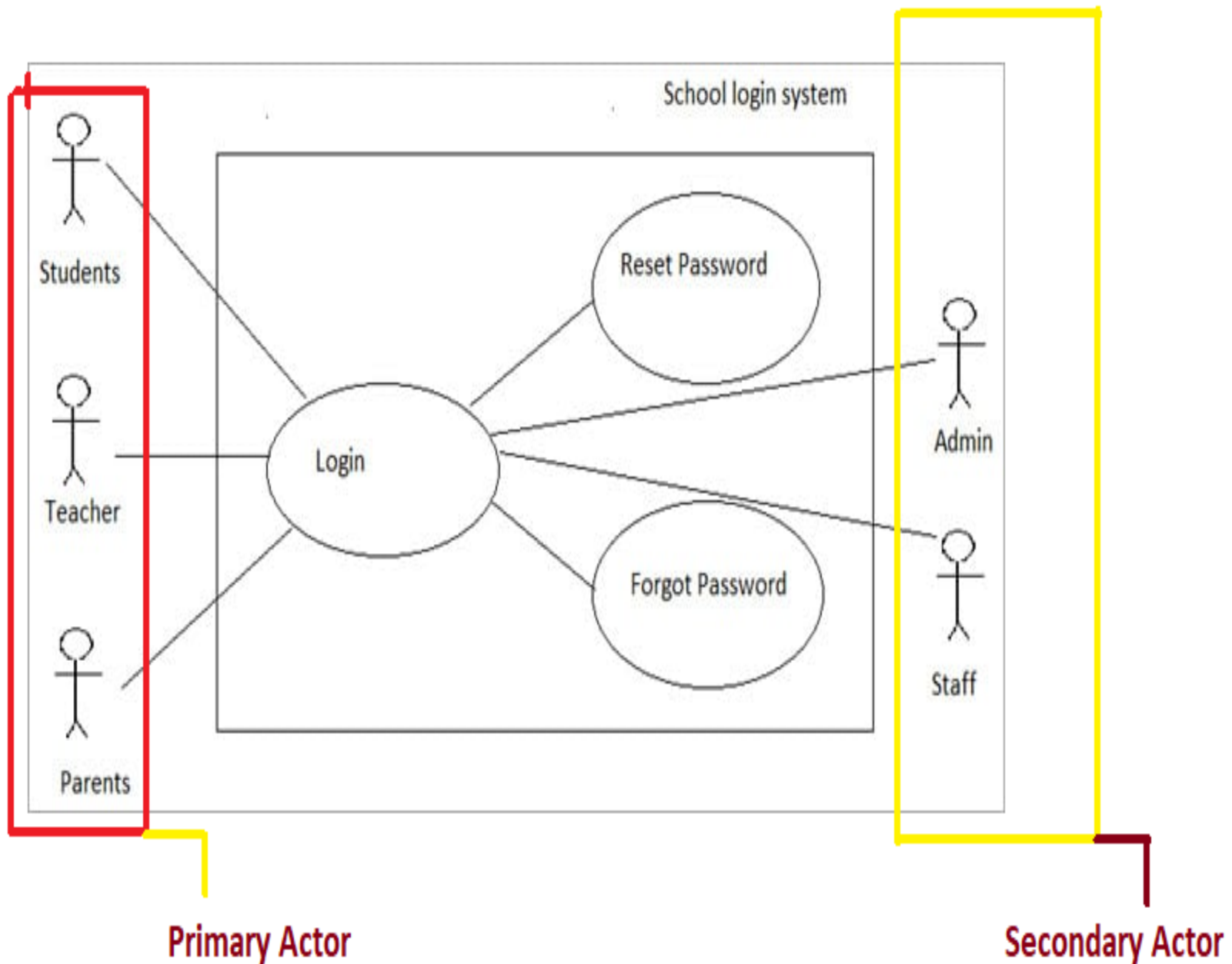
Pre-Condition-----System must be connected to the network.

Post –Condition----- After a successful login a notification mail is sent to the User mail id

Use Case Testing

It comes under the Functional Black Box testing technique. As it is a black box testing, there won't be any inspection of the codes.

It ensures if the path used by the user is working as intended or not. It makes sure that the user can accomplish the task successfully.



Test Case Example

Test Cases

Steps

Expected Result

1

Enter Student Name

User can enter
Student name

2

Enter Student ID

User can enter
Student ID

3

Click on View Mark

System displays
Student Marksfor
'Show Student
Marks' case

Subject Name

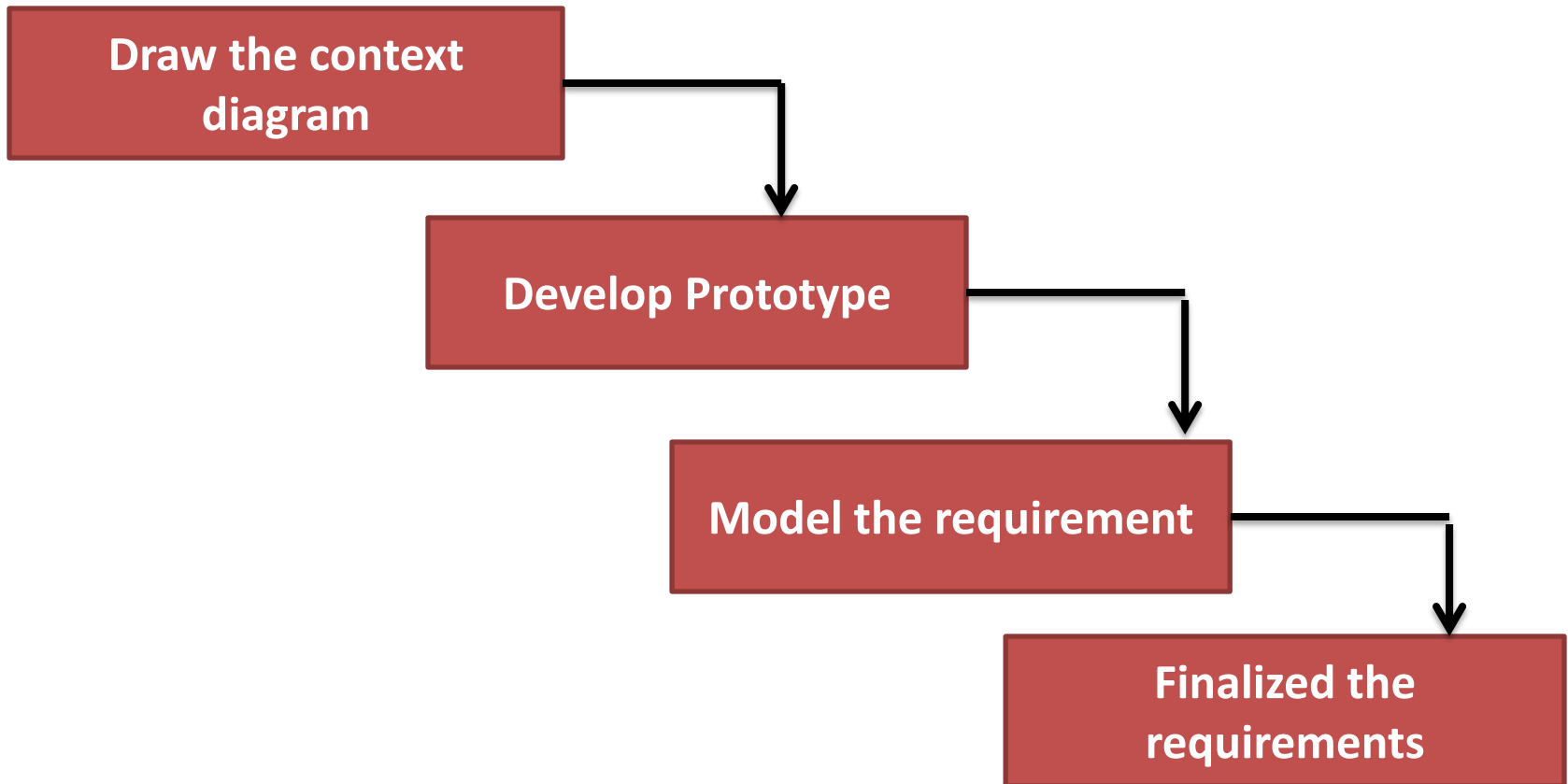
Software Engineering

Topic: Data Flow Diagrams



GLA
UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)

Requirement Analysis Steps



Requirement Analysis Steps(cont..)

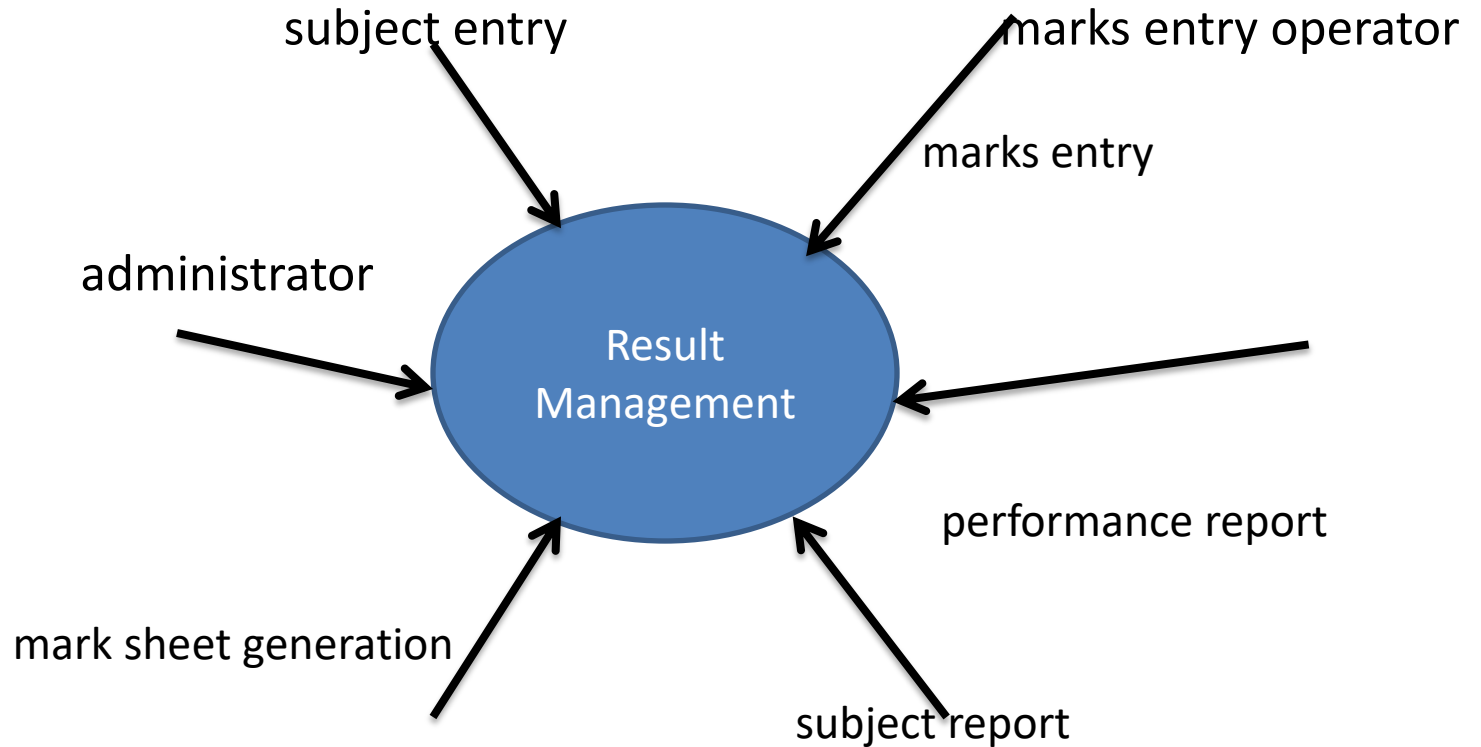
Draw the context diagram

It is a simple diagram.

It defines interface and boundaries of system with external environment.

It identifies the entities outside the proposed system, which interact with the system

Context diagram....



Develop Prototype

One effective way to find out the customer's expectation is to construct a prototype.

Customer's feedback can be used to modify the prototype until the customer is satisfied

In case where developers and users are not sure about some of the elements, a prototype may help both the parties to take a final decision.

Model the requirement

This process generally consists of various graphical representations of the functions, data entities, external entities, and the relationships between them.

Finalized the requirements

After framing the requirements, we will have a better understanding of the system behavior.

The inconsistencies and ambiguities have been identified and corrected.

The flow of data amongst various modules has been analyzed.

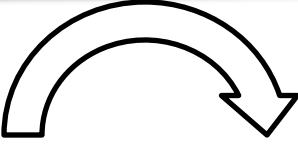
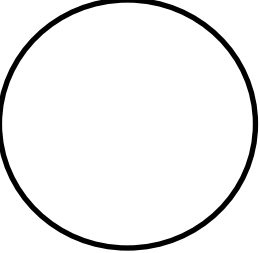

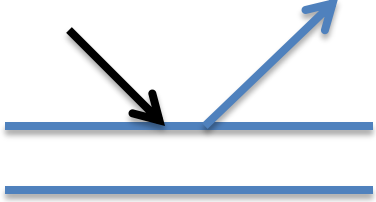
Elicitation and analyze activities have provided better insight into the system.

After that we finalize the analyzed requirements, and the next step is to prepare documentation in proper way.

Data Flow Diagram

- ❖ It shows the flow of data through system.
- ❖ All names should be unique
- ❖ It shows how the data enter and leaves the system and where data is stored.
- ❖ The objective of DFD is to show the scope and boundaries of a system as whole.
- ❖ It is also called data flow graph or bubble chart.

Standard symbols of Data flow diagram

Symbol	Name	Function
	Data Flow	It is used to connect processes with each other
	Process	Perform transformation of its input data to output data
	Source or sink	A source of system inputs or sink of system outputs
	Data Store	A repository of data a arrow head indicate net input and net output of data

Levels in Data Flow Diagrams (DFD)

0-level DFDM

- It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows.
- Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs.

0-level DFDM(continue)

This process may be repeated at as many levels as necessary until the program at hand is well understood.

It is essential to preserve the number of inputs and outputs between levels, this concept is called leveling by DeMacro. Thus, if bubble "A" has two inputs x_1 and x_2 and one output y , then the expanded DFD, that represents "A" should have exactly two external inputs and one external output as shown in fig:

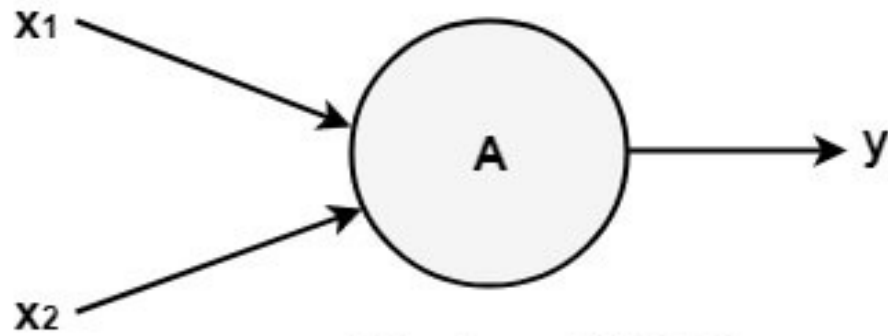


Fig: Level-0 DFD.

The Level-0 DFD, also called context diagram.

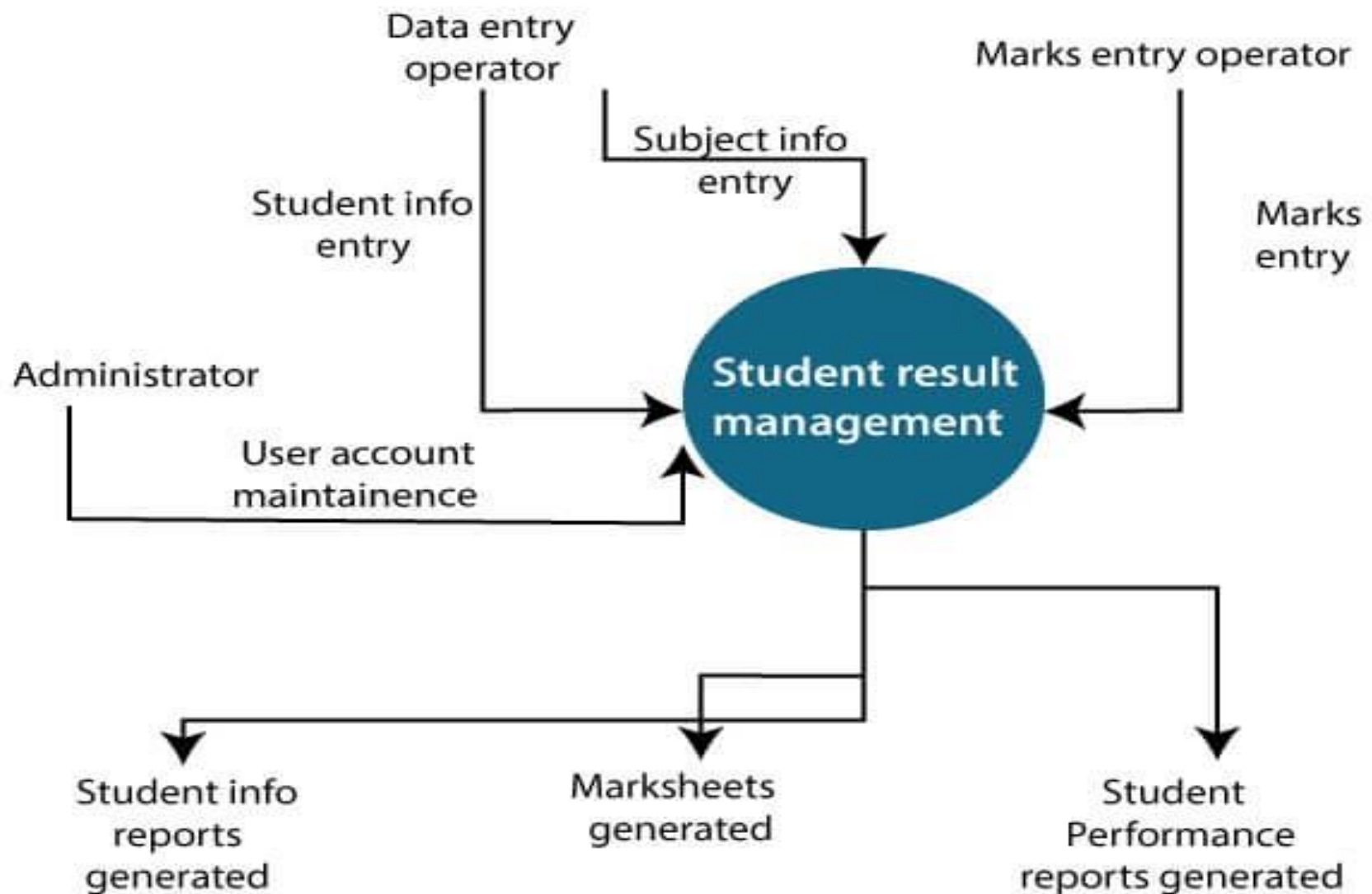
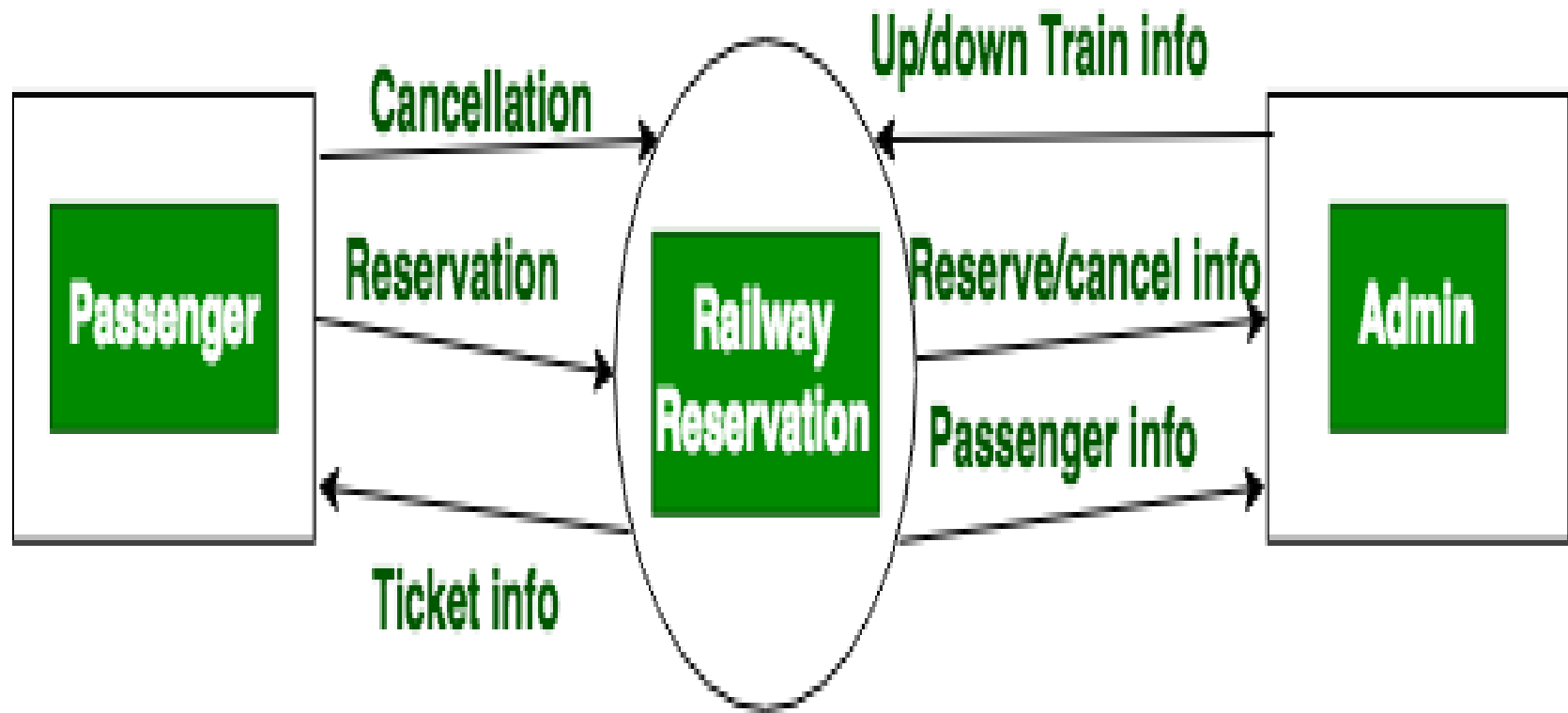


Fig: Level-0 DFD of result management system



0-LEVEL DFD

Online Shopping System

Order

Delivery

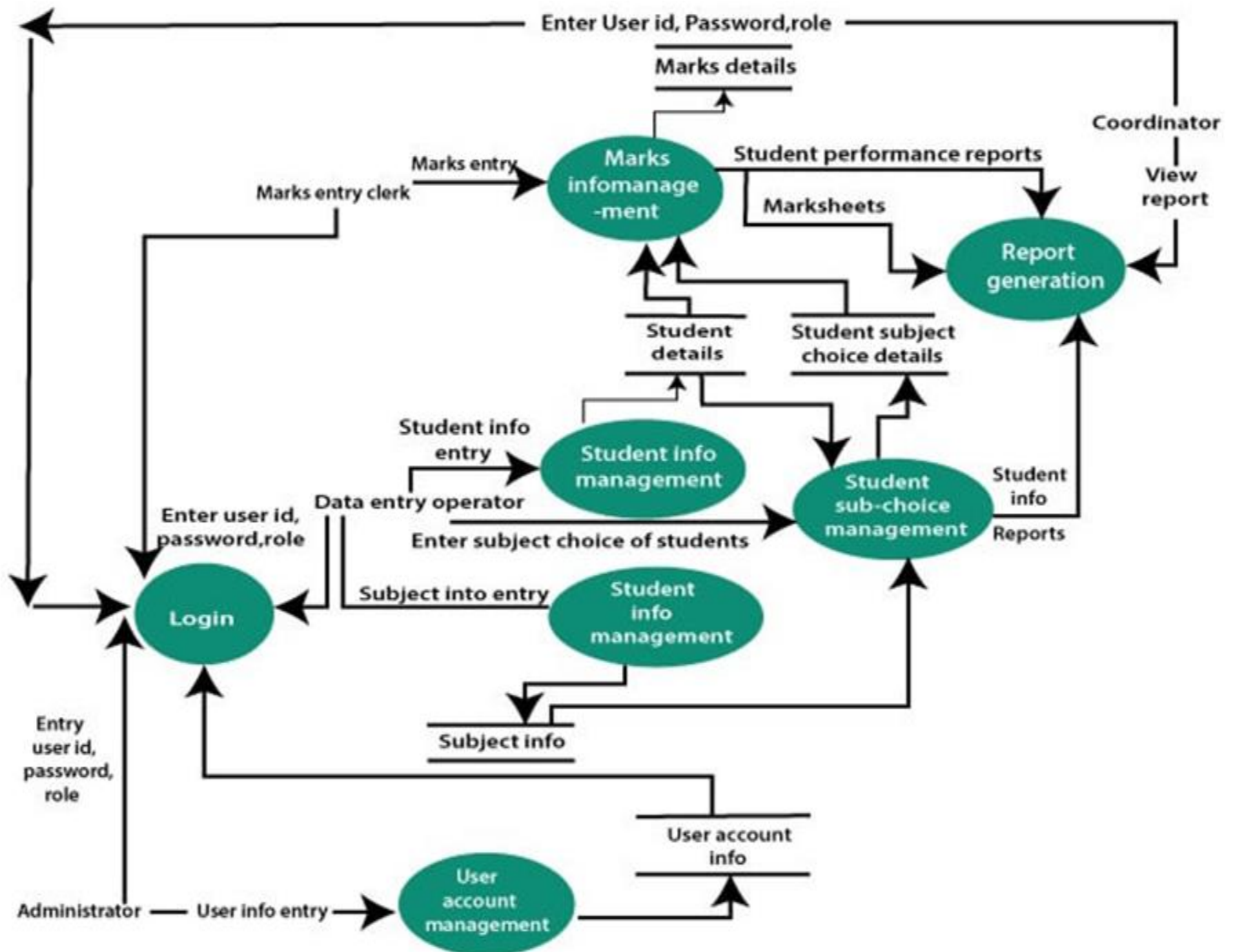
Customers

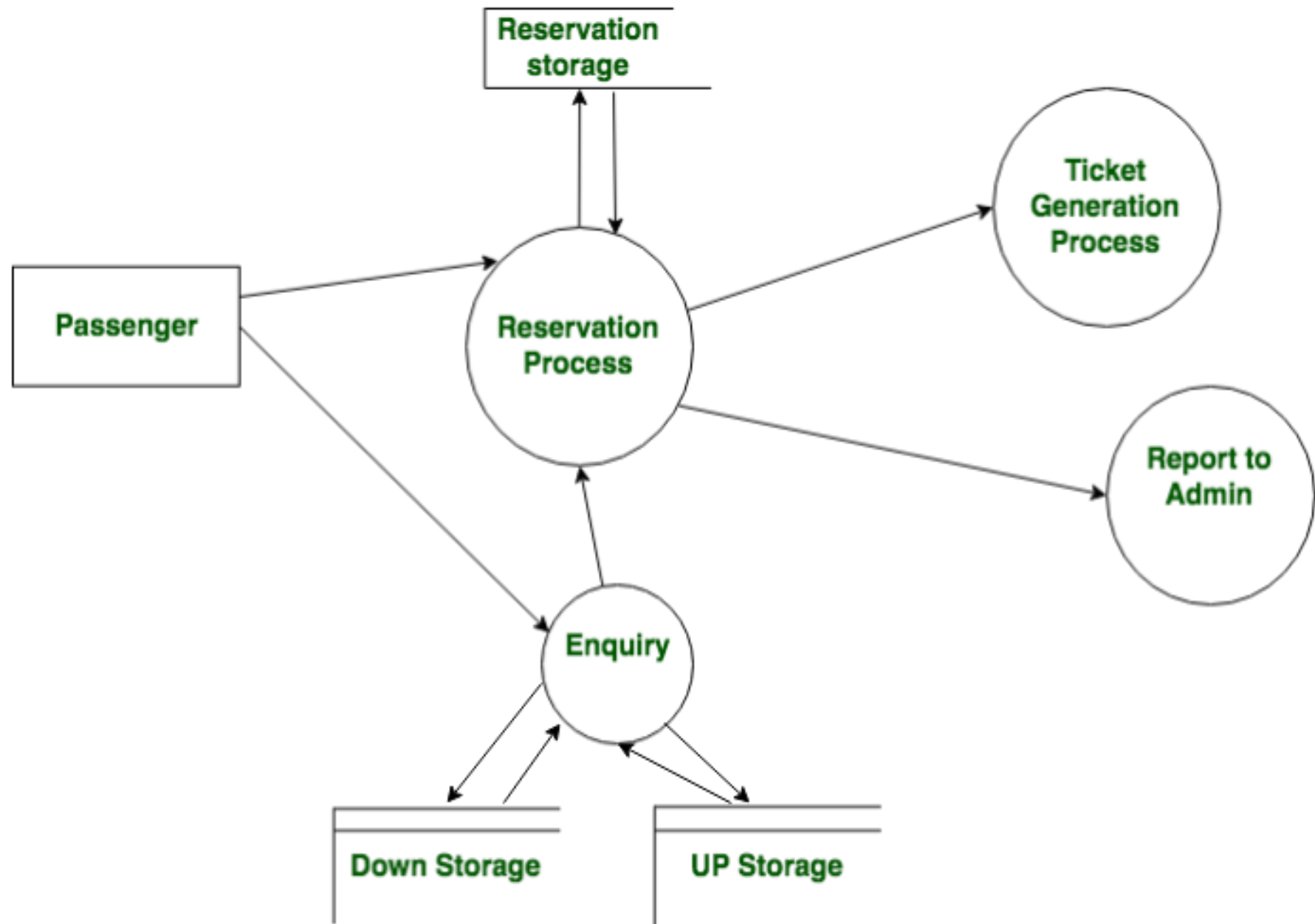


```
graph TD; C[Customers] -- Order --> OS[Online Shopping System]; OS -- Delivery --> C;
```

1-level DFD

- In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level,
- we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into sub processes.



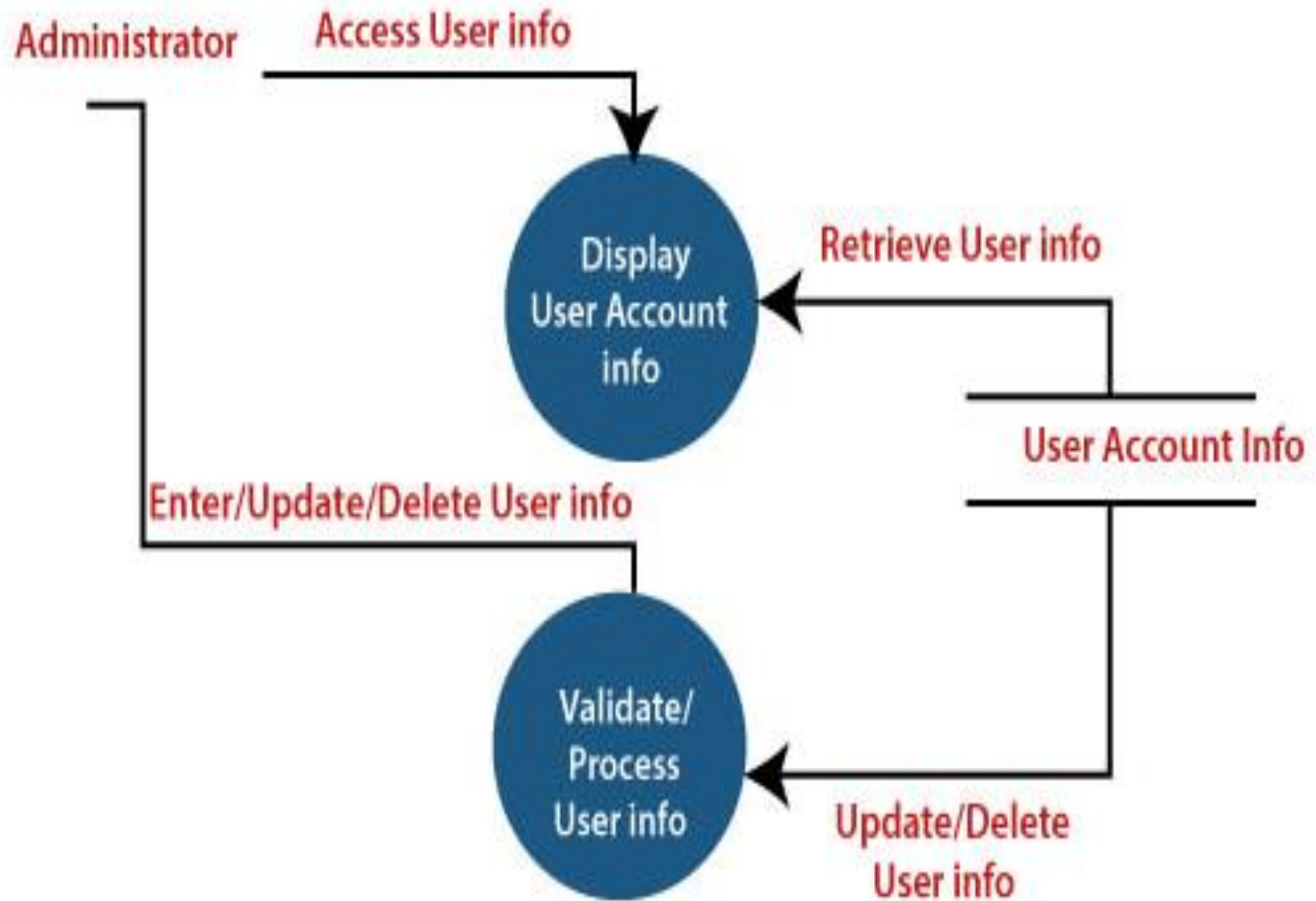


1-LEVEL DFD

2-Level DFD

- 2-level DFD goes one process deeper into parts of 1-level DFD. It can be used to project or record the specific/necessary detail about the system's functioning.

User Account Maintenance



Login

The level 2 DFD of this process is given below:

