

❖ SYSTEM ANALYST:-

❖ WHO IS SYSTEM ANALYST?

- Individual who perform the system investigation as distinct from those merely involved in the detailed computer programming are called "system Analyst".
- The system analyst is a key person analyzing the business, identifying opportunities for improvement, and designing information system to implement these ideas.
- The pure definition of a Systems Analyst is a programmer or consultant who designs and manages the development of business applications. Typically, systems analysts are more involved in design issues than in day-to-day coding.
- A System analyst is a person who conducts a study, identifies activities and objectives and determines a procedure to achieve the objective.
- The Main objective of a system analyst is to provide right type of information, in right quantity, in right time and right cost to management or for the end user.

❖ ROLE OF A SYSTEM ANALYST?

- The role of an analyst is to help organization understand the challenges before them to make this transition and to ensure that the need and expectation of the client are represented correctly in the final solution.
- System analyst is the brain for data processing. Following are roles of System Analyst:-
 1. System Analyst- an agent of change
 2. System Analyst- a motivator
 3. System Analyst- an organizer
 4. System Analyst- an architect
 5. System Analyst- an intelligent salesperson

[1] System Analyst – an agent of change:-

- A system analyst works towards the future. Future is uncertain and different.
- Change is the only thing which is permanent and the system analyst has to prepare a vehicle to work in that changing environment.
- The greatest hurdle for him is that people resist change. To overcome this he has to secure user acceptance through user participation in the design and implementation of the system.
- He is the creator of new environments.

[2] System Analyst – a motivator:-

- Acceptance cannot be forced down the throats of system users.
- Proper identification of right personnel and exacting feeding of right motivating factor can go a long way in making a system successful.
- A good motivator has to be a good psychologist.

[3] System Analyst – an organizer:-

- A system is the systems analysts conceptual child. Hence he has to be clear about all activities of the system.
- The sequence of activities, their purpose and their consequences must be clear to him. He is responsible for the execution of all activities and event and hence that of the system.
- The role of the organizer includes that of puzzle solver whenever problems arise. He is also an evaluator of his system.

[4] System Analyst – an architect:-

- A system analyst must have a fairly good idea of his final system at the raw materials stage itself.
- He prepares the blue print, modifies, improves, and provides aesthetic value to his product.
- For example a system analyst may bring in a better changed environment for the users.

[5] System Analyst – an intelligent salesperson:-

- A good systems analyst is one who can sell a refrigerator to an Eskimo.
- System selling is harder than that because the system analyst has to sell it to a user, who knows the existing system in and out.
- To sell his system he should be a good communicator and genuinely interested in understanding the real needs of the user.
- System selling takes place at all stages of design and later at all levels of the organization.

❖ SYSTEM DEVELOPMENT STRATEGY:-

- A system development methodology is an orderly and integrated collection of various methods, tools and notation used for system development.
- List out system development strategies are:-
 - [A] System Development Life Cycle (SDLC)
 - [B] Structured Analysis Development Strategy (SADS)
 - [C] System Prototype Method. (SPM)

[A] System Development Life Cycle (SDLC):-

- The systems development life cycle (SDLC) is a systematic approach to solving business problems.
- The basic idea of SDLC is that there is a well defined process by which a system is conceived, developed and implemented.
- System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle also means software development life cycle.
- We mention here a six stage SDLC which is performed in

- [1] Preliminary Investigation
- [2] Determination of system requirement
- [3] Design of system
- [4] Development of software
- [5] System Testing
- [6] Implementation and Evaluation

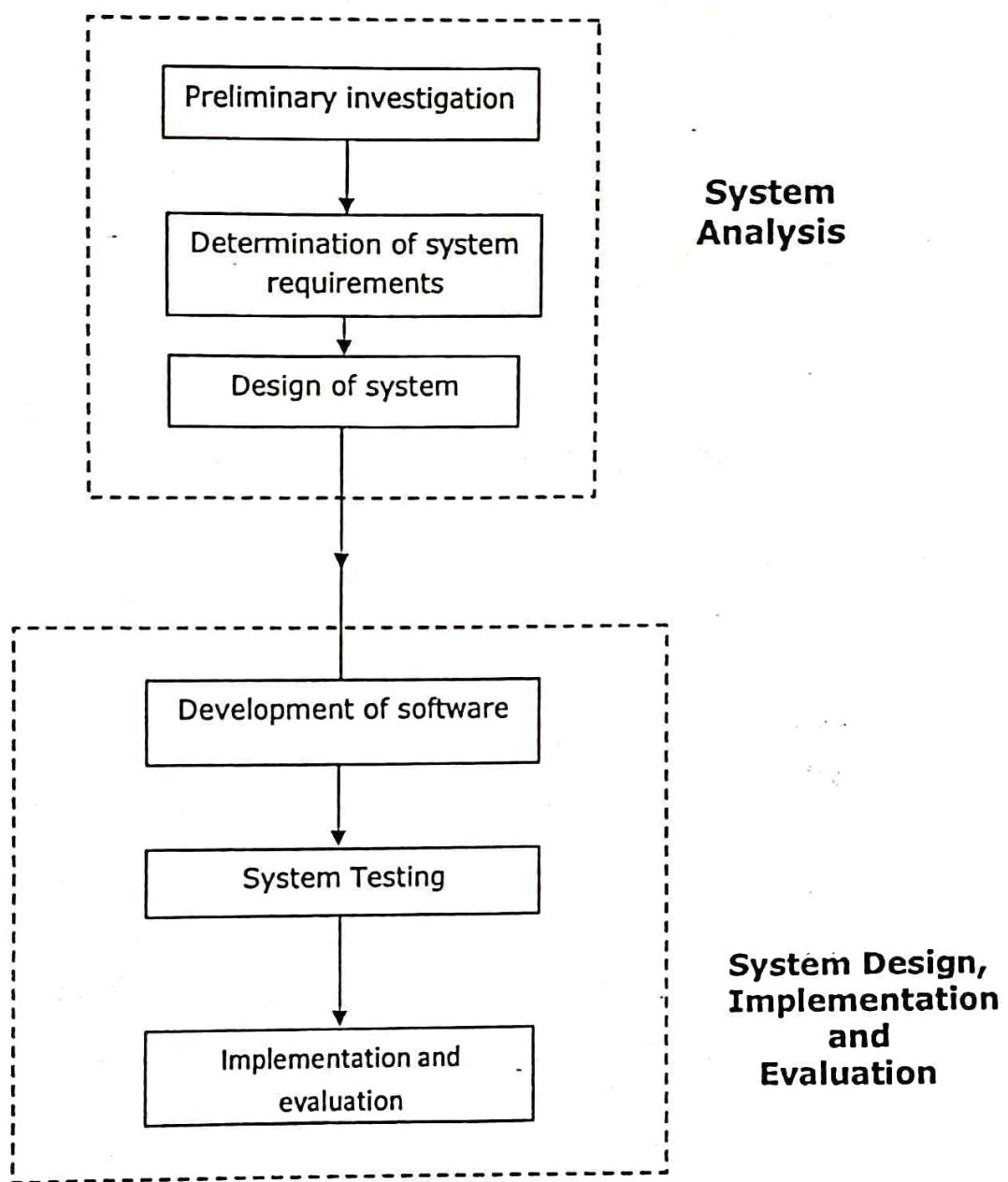


Figure: - System Development Life Cycle

[1] Preliminary Investigation:-

- Before taking any decision ,gathering information is called preliminary investigation
- When that request is made, the first system activity, the preliminary investigation,begins.This activity has three parts:-
 - [a] Request clarification
 - [b] Feasibility Study

[B] Structured Analysis Development Strategy

- Structured Analysis Development method (strategy) helps the analyst in better understanding such systems.
- It does that by partitioning the system into components and constructing a model of the system.
- There are two types of Structured Analysis Development Method :-
 1. Data Flow Diagram
 2. Data Dictionary.

[1] DATA FLOW DIAGRAM:-

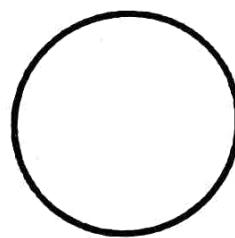
- Data flow Diagram (DFD) is a graphical aid for defining system inputs, processes and outputs. It represents flow of data through the system.
- The DFDs are graphics representation of data movement, process and files (data store) used in support of an information system.
- A graphical tools used to describe and analyze the movement of data through a system including the processes, stores of data and delays in the system
- A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).
- DFDs serve two purposes:
 1. Provide a graphics tool which can be used by the analyst to explain his understanding of the system to the user.
 2. They can be readily converted into a structured chart which can be used in design.

• Symbols used in DFDs:-

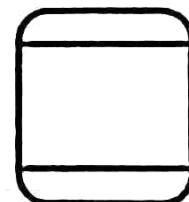
- Most DFD use four types of symbols which represent system component such as:
 1. Process
 2. External Entity
 3. DATA Flow
 4. Data Store.
- The use of specific items associated with each element depends on whether Yourdon or Gane & Sarson approach is used.

[1] Process:-

- Here flow of data is transformed.
- People, procedures or devices that use or produce (transform) data.
- Processes that handles the data.
- The process symbol is define flow of data is transformed from input to output.
- Example:-verify credits, update inventory file.



Yourdon

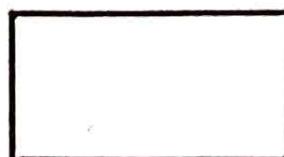


Gane & Sarson

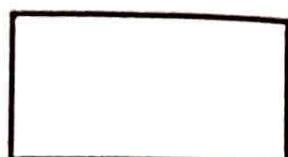
[2] External Entity:-

- The External Entity symbols is used to source or destination of data which is external to the system.

- External Source or destination of data, which may be people, programs organization or other entities, interact with the system but are outside its boundary.
- Example: - Supplier, customer etc...



Yourdon



Gane & Sarson

[3] A Data Flow:-

- It is a packet of data.
- Data move in specific direction from an origin to a destination in the form of a document, letter, telephone call or virtually any other medium.
- A data flow shows the flow of information from its source to its destination. A data flow is represented by a line, with arrowheads showing the direction of flow.



Yourdon

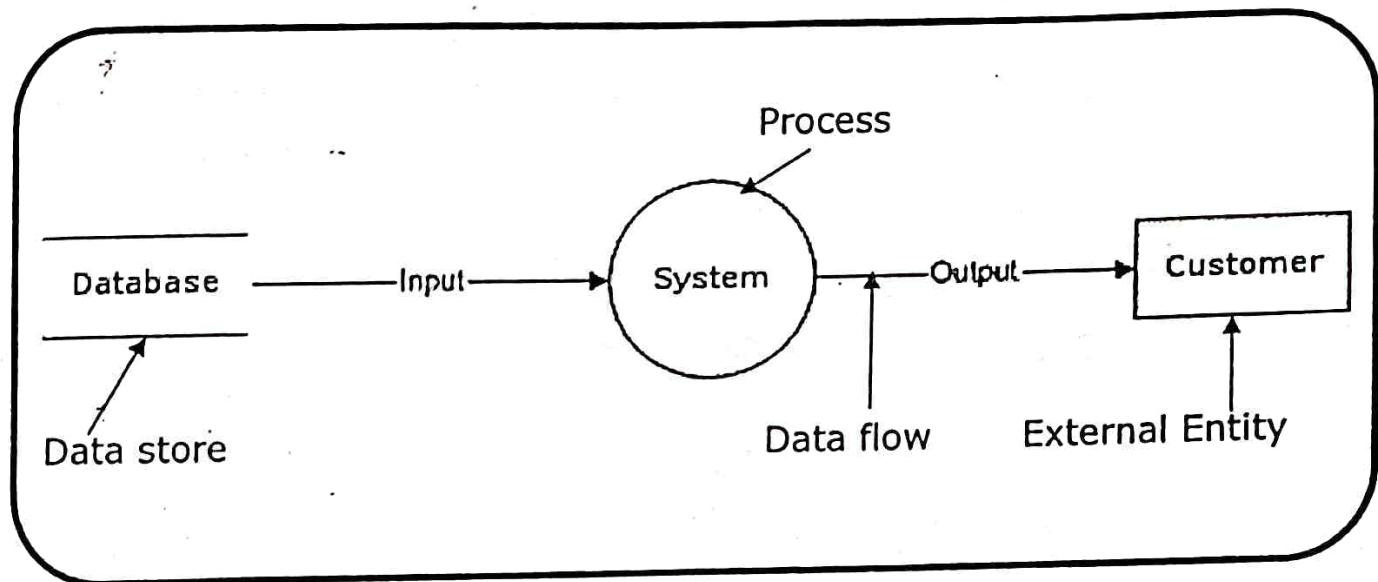


Gane & Sarson

[4] A Data Store:-

- Data store symbols are used for storing data.
- Here data are stored or referenced by process in the system.
- Any stored data but with no reference to the physical method of storing.
- Data stores, within the system (diary, filing cabinet or computer file)
- Example:-Inventory Master File, Customer master file.

Yourdon

Gane & Sarson

Example of DFD

NOTE -1:-These symbols are not standardized .Different books use different symbols with minor variations.

NOTE -2:-The process represented by a circle is also known as "BUBBLE" (sometimes DFDs are referred as "BUBBLE DIAGRAMS").

❖FACT FINDING(Gathering) TECHNIQUE

- The specific methods analysts use for collecting data about requirements are called "**Fact Finding**" techniques.

OR

- Fact finding technique is the formal process of using research, interviews, questionnaires, and other technique to collect information about system, requirement and preference.
- The analyst does not know the working process of the user for which ,he is going to develop information system .The analyst uses specific method for collecting data about requirements are **called fact fining techniques**.
- It is also called information gathering or data collection.
- These techniques are used in system analysis, design or even during implementation stage.
- The different fact finding techniques (**METHOD**) employed are:-

- [1] Interviewing
- [2] Questionnaires
- [3] Record review or Record inspection
- [4] Observation.

[1] Interviewing:-

- This technique is used to collect information from **individuals** or from **groups**.
- Analyst selects the people who are related with the system for the interview. In this method the analyst sits face to face with the people and records their responses.

- Interviews provide analysts with opportunities for gathering information from respondents who have been chosen for their knowledge of the system under study.
- It is an invaluable technique to gather qualitative information, opinions, policies, suggestions, underlying problems, descriptions of activities & problems, etc.
- This method of fact-finding can be especially helpful for gathering information from individuals who do not communicate effectively in writing or who may not have the time to complete questionnaires.
- Interviews allow analysts to discover areas of misunderstanding, unrealistic expectations & even indications of resistance to proposed system.

- **Types Of Interview :-**

- [I] Structured Interview
- [II] Unstructured Interview.

[I] Structured Interview:-

- Structured interview is more formal interview where fixed questions are asked.
- The questions are standardized having a predefined format.

Advantages:-

- ◆ Ensure uniform working of questions for all respondents.
- ◆ Easy to administer and evaluate.
- ◆ Result in shorter interview.

Disadvantages:-

- ◆ Cost of preparation is high in it.

- ◆ Respondents may not accept high level of structure and mechanical posing of questions.

[II] Unstructured Interview:-

- Unstructured interviews which are of questions and answers format are appropriate when analyst want to get the general information about a system.

Advantages:-

- ◆ Here interviewer has greater flexibility in wording questions to suit respondents.
- ◆ Interviewers can pursue areas that arise spontaneously during the interview.
- ◆ This format encourages respondents to share their feelings, ideas and beliefs.

Disadvantages:-

- ◆ Interviewers may introduce their biases in questions or reporting results.
- ◆ Extraneous information may be gathered.
- ◆ Analysis and interpretation of results may be lengthy.
- ◆ Take extra time to collect essential facts.

[2] Questionnaires:-

- Questionnaires may be used as a supplement to interviews. More people can be reached & answers can be corroborated.
- The use of questionnaires allows analysts to collect information about various aspects of a system from a large number of persons.

- The use of standardized question formats can yield more reliable data than other fact-finding techniques.
 - It is the technique used to extract information from number of people. This method can be adopted and used only by an skillful analyst. The Questionnaire consists of series of questions framed together in logical manner.
 - This method is very useful for attaining information from people who are concerned with the usage of the system and who are living in different countries. The questionnaire can be mailed or send to people by post. This is the cheapest source of fact finding.
- **Types of Questionnaires :**
- [I] Open Response (ended) Questionnaires
 - [II] Closed – Response (ended) Questionnaires.

[I] Open Response (ended) Questionnaires:-

- Analysts often use Open-Ended questionnaires to learn about feelings, opinions & general experiences or to explore a process or problem.
- The questionnaires can have open ended question like – What are the major & minor problems in the existing system.

[I] Closed Response (ended) Questionnaires:-

- Closed questionnaires control the frame of reference by presenting respondents with specific responses from which to choose. This format is appropriate for eliciting factual information.
- A Close Ended questionnaire will have fixed responses like – What is the average value of invoice in your department?

- 1) Less than Rs. 3000/-
- 2) Rs. 3000/- to Rs. 5000/-
- 3) Rs. 5000/- to Rs. 10000/-
- 4) Rs. 10000/- & above.

[3] Record Review OR Record Inspection:-

- Many kinds of records and reports can provide analysts with valuable information about organization and operations. In record review analysts **examine** information that has been recorded about the system and users. This process at the beginning of the study.
- Records include written policy manuals, regulations and standards operating, procedures used by most organization as a guide for managers and most employees.
- This Record Review can take place at the beginning of the system study or later in the study for comparing actual operations with what the records indicate.
- **Records may include :**
 - Written policy manuals.
 - Rules & regulations.
 - Standard operating procedures used in the organization.
 - Forms & documents.
- **The following questions may be useful in analysis of forms :**
 - Who uses these forms?
 - Do they include all the necessary information?
 - How readable & easy to follow is the form?
- They can help analysts understand the system by familiarizing them with what operations must be supported & with formal relations within the organization.

[4] Observation:-

- Observation allows analysts to gain information they cannot obtain by any other fact finding method. Through observation analysts can obtain first hand information about how activities are carried out.
- This method is most useful when analysts need to actually observe how documents are handled, how processes are carried out and whether specified steps are actually followed.
- On site Observation provides close view of the working of the real system. He can observe people, objects, documents & occurrences of events.
- Observation can look for :
 - Operational inefficiencies.
 - Alternate routes & procedures.
 - Interruptions in the normal flow of work.
 - The usage of files & documents.
 - Informal communication channels etc.

❖ FORM DESIGN:-

- Form is very important means of communication in many organizations the forms are used for both input & output. Form is a physical carrier of data as well as information. The purpose of form is to convey or collect information quickly & completely.
- A form is a group of controls that the user interacts with and sends the result to a specific file as designed by an application developer.
- The form design activity will be meaningful if it involves users and system analyst.
- Form is a tool with a message. It is the physical carrier of data-of information. It is either an authority for action or a request for action.

❖ Types Of Form Design:-

- A printed form is generally classified by what it does in the system.
- There are three primary **Types of Form Design OR classifications of form design:-**
 1. Action form
 2. Records Memory
 3. Reports

Form type	Characteristics	Example
[1] Action Authorities. Confirmation.	1.Orders, Instructs, Amendments, order 2.Achieve results 3.Movable	- Purchase order,P.O -Q.C Test report, Application form. Material Requisition, sales slip, Claim issue slip, gate issue slip and Gate pass
[2] Record - Memory	1.Data generally used for reference 2.Shows historical data 3.Serves as control on certain details	-Inward/outward register, Cheques. - Policies. - Stocks ledger, purchase Record.
[3] Report	1.summary of jobs/project 2.Normally used by manager with authority to effect change 3.Used as a basis for decision making	- States report - Balance sheet profit & loss -Sales analysis, Trial balance

- The forms can also be classified as input or output forms depending on their function in a system environment.

~~10~~ ♦ Basic Principles of Form Design:-

1. The form should be as simple as possible to understand & compile. The languages used should be understandable to the user.
2. Entries should flow in a logical sequence.
3. Layout & form sizes should be standardized as far as is practicable. This will reduce the difficulties encountered in handling the forms.
4. Ease of use: - The format of the form should be such that it can be filled up easily & quickly without missing any information.
5. Duplicating of data should be avoided wherever possible
6. Combine related form when feasible.
7. Too many operations on one form to be operated by different people will be confusing & time consuming requesting in information bottle-necks.

◆ Data Capture Objectives:-

- If the data input is bad then output will be worse. This calls for clear data captures objectives such as:
1. Reduction in the volume of input to the extent possible.
 2. Lesser manual preparation.
 3. An input design which will ease the work of the person engaged in input preparation.
 4. Minimizing the number of steps practicable in the data capturing process.

✓ [B] Data Validation

- The objective of a data validation system is to detect errors at the earliest possible stage before costly activities are performed on invalid data.
- Data validation is the process of ensuring that a program operates on clean, correct and useful data. It uses routines, often called "validation rules" or "check routines", that check for correctness, meaningfulness, and security of data that are input to the system.
- Some data validation is done by way of manual verification in data capture stage itself. In spite of this, still there may be incorrect batches of input data, missing data, duplicated data and incorrect file records etc.
- This invalid data is checked manually for low casting errors. After correcting these errors the data is again subjected to above data validation process of accurate input.

◆ Validation Checks:-

- Data validation checks that the data are valid and sensible/reasonable before they are processed.
- There are various categories of checks which can be applied to data during a validation run.
- There are two types of validation checks in Input design
 - [I] Field checks
 - [II] Transaction checks

[I] Field checks:-

- include the following field check are:-
 - a) Limit check
 - b) Picture check
 - c) Valid code check
 - d) Check digit
 - e) Arithmetic check
 - f) Cross check

a) Limit check:-

- May be applied to each field (data item) of a record to ensure that contents lie within predefined size.
- Unlike range checks, data are checked for one limit only, upper OR lower, e.g., data should not be greater than 2 ($<=2$).

b) Picture check:-

- May be applied to each field to detect entry of incorrect characters in the field.

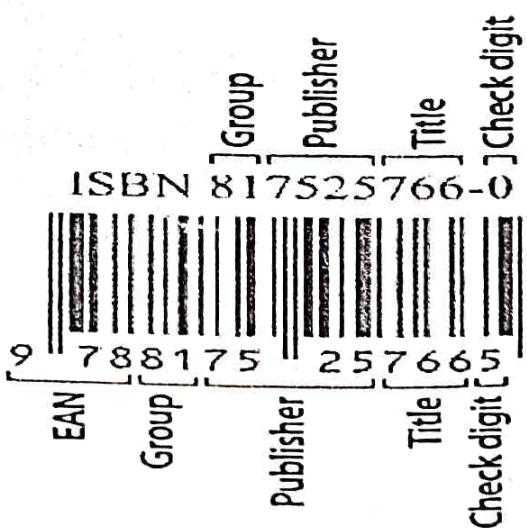
- **Example:-** PIC of EMPLY-NO is AAA 9999. An EMPLY-NO PRD24N6 would be rejected as there is a letter in the sixth position and this should be only numeric.

c) Valid code check:-

- To validate input against predefined transaction codes. These predefined codes may either be embedded in the programs or stored in files.
- **For Example:-** Contents of EMPLY-CATEGORY field A, B, C, D, E are only valid and any other letter coming in that field will be rejected.

d) Check digit:-

- It is used to detect transposition errors when recording "key" fields.
- Used for numerical data. An extra digit is added to a number which is calculated from the digits. The computer checks this calculation when data are entered. For example the last digit of an ISBN for a book is a check digit calculated modulus 10



e) Arithmetic check:-

- Arithmetic checks are used to ensure the validity of the results by performing arithmetic operation in different ways.

f) Cross check:-

- May be applied to verify field appearing in different files to verify that result fully.

[II] Transaction checks:-

- include the following Transaction check are:-

- a) Sequence checks
- b) Format completeness
- c) Redundant data check
- d) Combination checks
- e) Probability checks
- f) Pass words
- g) Checks
- h) Batch total
- i) Hash total

a) Sequence checks:-

- Are applied to detect any missing transaction (e.g. off serially numbered vouchers.)
- This is exercised to detect any employee being missed out from payment of salary.

b) Format completeness:-

- Are used to check the presence and position of all fields in a transaction.

c) Redundant data checks:-

- Are employed to check the validity of codes with reference to description.

d) Combination checks:-

- May be applied on various fields of a file.
- This may be exercised on various fields of a file.
- For Example:-to check the amount written in figures and in words

e) Probability checks:-

- Are used to avoid unnecessary rejection of data.

f) Pass words:-

- May be exercised to check entry of data by unauthorized persons in on line system.

g) Checks:-

- May be incorporated to ensure that transaction pertains to the current period.

h) Batch total:-

- Can be used to ensure that transaction have been transcribed correctly. A total of some common component of a batch of data so as to enable a control to be maintained over the validity of data.

- Checks for missing records. Numerical fields may be added together for all records in a batch. The batch total is entered and the computer checks that the total is correct, e.g., add the 'Total Cost' field of a number of transactions together.

i) Hash total:-

- A control total i.e. the sum of values in a particular field or record area of a file, to ensure that transaction have been transmitted currently.
- This is just a batch total done on one or more numeric fields which appears in every record. This is a meaningless total, e.g., add the Telephone Numbers together for a number of Customers.

❖ OUTPUT DESIGN

- The output generally refers to the results and information that are generated by the system. One of the most important features of an information system from the point of view of users is the output it produces.
- The output generally display the result & information that are acquire by the system. If any system the output is legally depends on input data
- If the output is of poor quality, the whole system is in peril because the users will then avoid using it. Hence, the design of output assumes greater importance. In any system the output is largely dependent on input.
- One of the most important features of an information system for users is the output it produces. Without quality output, the entire system may appear to be so unnecessary that users will avoid using it, possibly causing it to fail.

❖ Design Principle Of Output

➤ A system designer should try to incorporate the following design principles for output:-

- [1] Principle of starting with output.
- [2] Principle of acceptability of reports.
- [3] Principle of timely output.
- [4] Principle of enhancing the decision making process.
- [5] Principle of practicing "Management by Exception".
- [6] Principle of duplication reduction in reports.
- [7] Principle of simplicity in reports.

[1] Principle of starting with output:-

- In any system the output is largely dependent on input. However, no system can be designed properly without known as "Principle of starting with output". This means that organization output needs should be considered first before devising appropriate methods and procedures, database, planned input and effective internal control

[2] Principle of acceptability of reports:-

- The end user has to accept these outputs since they are the people who will be using it for their desired purposes. Hence it is ideal to have their participation in the output design phase for the greater success of the system.

[3] Principle of timely output:-

- A stitch in time saves nine. This is very true for information output. An output in time can help to make better decisions.

[4] Principle of enhancing the decision making process:-

- After all, the systems are designed for the manager and other personnel to make better decisions. Naturally the output report must be prepared keeping this principle constantly in mind.

[5] Principle of practicing "Management by Exception":-

- Management controls through completed tasks. The report should be designed not only for what has happened but for what deviations were there from actual plans. Significant deviations as and when it occurs be brought to the notice of the management through exception reports of the system.

[6] Principle of duplication reduction in reports:-

- Duplicate or unnecessary information in the reports should be minimized to the extent possible. This automatically reduces the cost of processing.

[7] Principle of simplicity in reports:-

- Reports should be concise, simple and self explanatory.

◆ Output Objectives

- Before designing output the objectives of each output must be clear. Without objective the system is not a properly work as per system requirement
- Simply because the output is very attractive or it has used the latest computer technology, Output cannot be regarded as 'good'.
- It must accomplish one or more the following objectives:-

[1] Convey information about

- a) Past Activities :-Personnel File, Vendor History
- b) Current Status :-Inventory On Hand, Cash On Hand
- c) Future Projection:-Sales or Cost of manufacturing a new item

[2] Confirm an action :- Complete Task.

[3] Trigger an action :- Rush purchase, Market Loss

[4] Signal events :-Through exception reporting or report on lurking opportunities etc

* Computer Aided System Tools OR CASE TOOLS

- CASE stands for "Computer Aided Software Engineering" OR "Computer Assisted System Engineering".
- Tools means a device that when used properly will improve the performance of a task is called tool.
- Computer-aided software engineering (CASE) is the scientific application of a set of tools and methods to a software system which is meant to result in high-quality, defect-free, and maintainable software products. It also refers to methods for the development of information systems together with automated tools that can be used in the software development process
- The CASE generally used to communicate between information.
- The long term objective of CASE tools is automate key aspects of the entire systems development process from beginning to end.
- Because CASE Tools are general-purpose tools, i.e., not limited to any specific application area such as manufacturing control, investment portfolio analysis or accounts management, the challenge of fully automating the S/W generation process is substantial.
- The CASE tools are called general purpose tools because a no limit to any specific application area.
- The most case system heavily on the editing and modify mode in system analysis.

❖ CASE Component

➤ The CASE tools are generally including five components.

- [A] Diagramming Tools
- [B] Centralized Information Repository.
- [C] Interface Generator.
- [D] Code generator.
- [E] Management tools.

[A] Diagramming tools:-

- Diagramming Tools support Analysis & Documentation of application requirements.
- They include the capabilities to produce data flow diagrams, data structure diagrams, and program structure charts.
- The Diagramming tool is high level tools that are establish for support of the structure analysis methodology.
- They support the capability to draw diagrams and charts and to store the details internally.
- The diagramming tools is provide the facilities for change and update the activity about system analysis.
- Example:-Diagramming tools include produce DFD,Data structure, structure chart, use case diagram,E R diagram.

[B] Centralized Information Repository:-

- The capture, analysis, processing and distribution of all systems information is aided by a centralized information repository or data dictionary.

- The dictionary contains the details of system components, such as data items, data flows, and processes and also includes information describing the volume and frequency of each activity.
- They are designed so that the information is easily accessible, they also include built in controls and safeguards that preserve the accuracy and consistency of system details.
- It provides central information tools to provide the facility for handle (management) the hold system by single authorize person.
- Centralize information is provide the number of advantage.
 - Authorize level.
 - Process validation(control)
 - Easy to testing.

[C] Interface Generator:-

- System interfaces are the means by which users interact with an application, both to enter information and data and to receive information.
- Interface generators provide the capability to prepare mock-ups and prototypes of user interfaces.
- They support the rapid creation of demonstration system menus, presentations screens, and report layouts.
- The interface is an important element for development the system application.

[D] Code generator:-

- Code Generators automatically generates program and database definition code directly from the models stored in the Information Repository.
- They incorporate methods that allow the conversion of system specifications into executable source code.
- Thus the best generation will produce approximately 75 percent of the source code for an application. Others should be written by hand.
- The code Generator is a best technique for to design the code.

[E] Management tools:-

- CASE systems also assist project managers in maintaining efficiency and effectiveness throughout the application development process.
- They can also assist development managers in the analysis and design activities and the allocation of resources to different project activities.
- In the case too the important parts is to management the all system by using controlling tools.
- Some CASE management tools allow project managers to specify custom elements. For example they can select the graphic symbols they want to describe processes, people, departments, etc.

- Each structured flowchart is shown on a single sheet of paper
- When designing a structured flowchart the systems analyst specifies the logic in a top-down fashion.
- There is a single exit from the process.
- The analyst begins with a major process & introduces other symbols to subdivide the process.
- Each process is named, but, if the name is not underlined, it is reference to some other diagram or description.

[2] HIPO Diagrams (Hierarchical Input Process Output):-

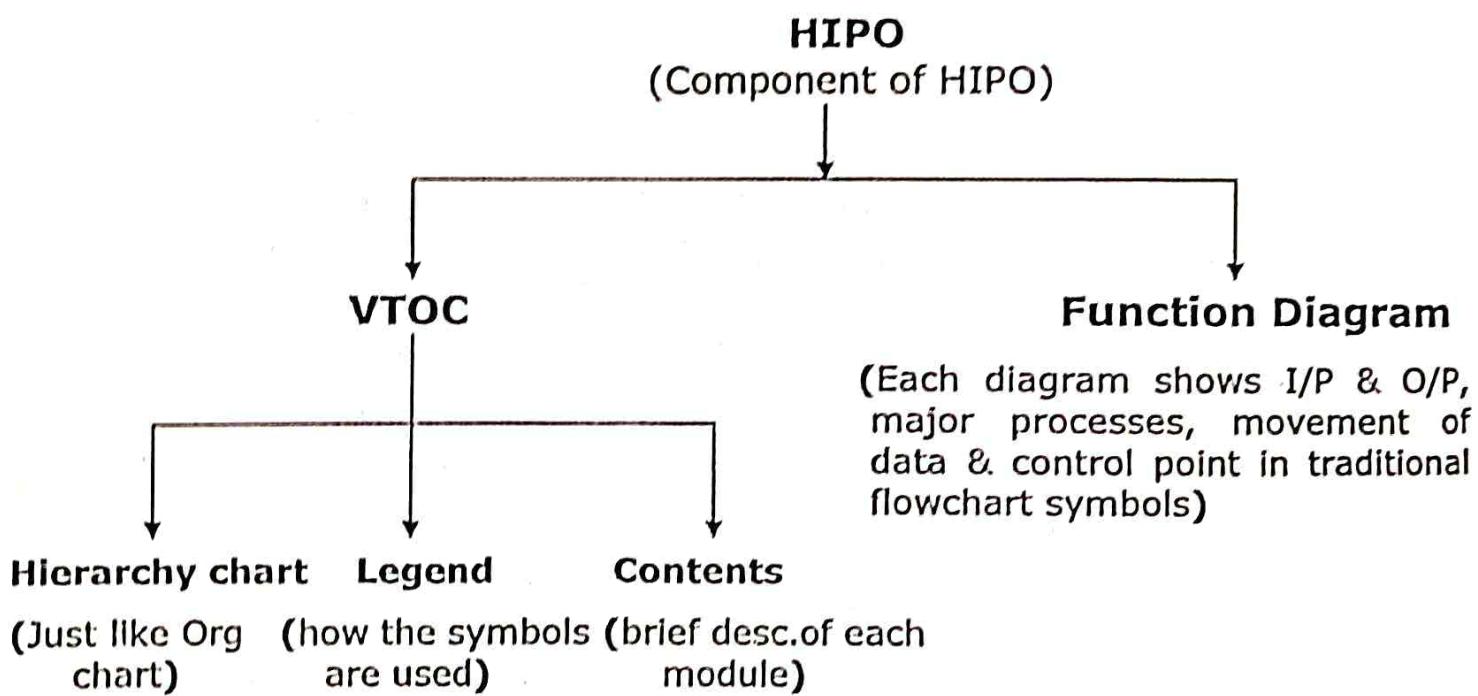
- HIPO is commonly used method for developing system s/w.
- A HIPO model consists of a hierarchy chart that graphically represents the program's control structure and a set of IPO (Input-Process-Output) charts that describe the inputs to, the outputs from, and the functions (or processes) performed by each module on the hierarchy chart.
- HIPO Diagram are graphic description of the system.
- An acronym for **Hierarchical Input Process Output**, this method was developed by IBM for its large, complex operating systems.
- The HIPO Technique is tools for planning and documenting a computer program.
- HIPO diagrams are graphic, rather than narrative description of the system.

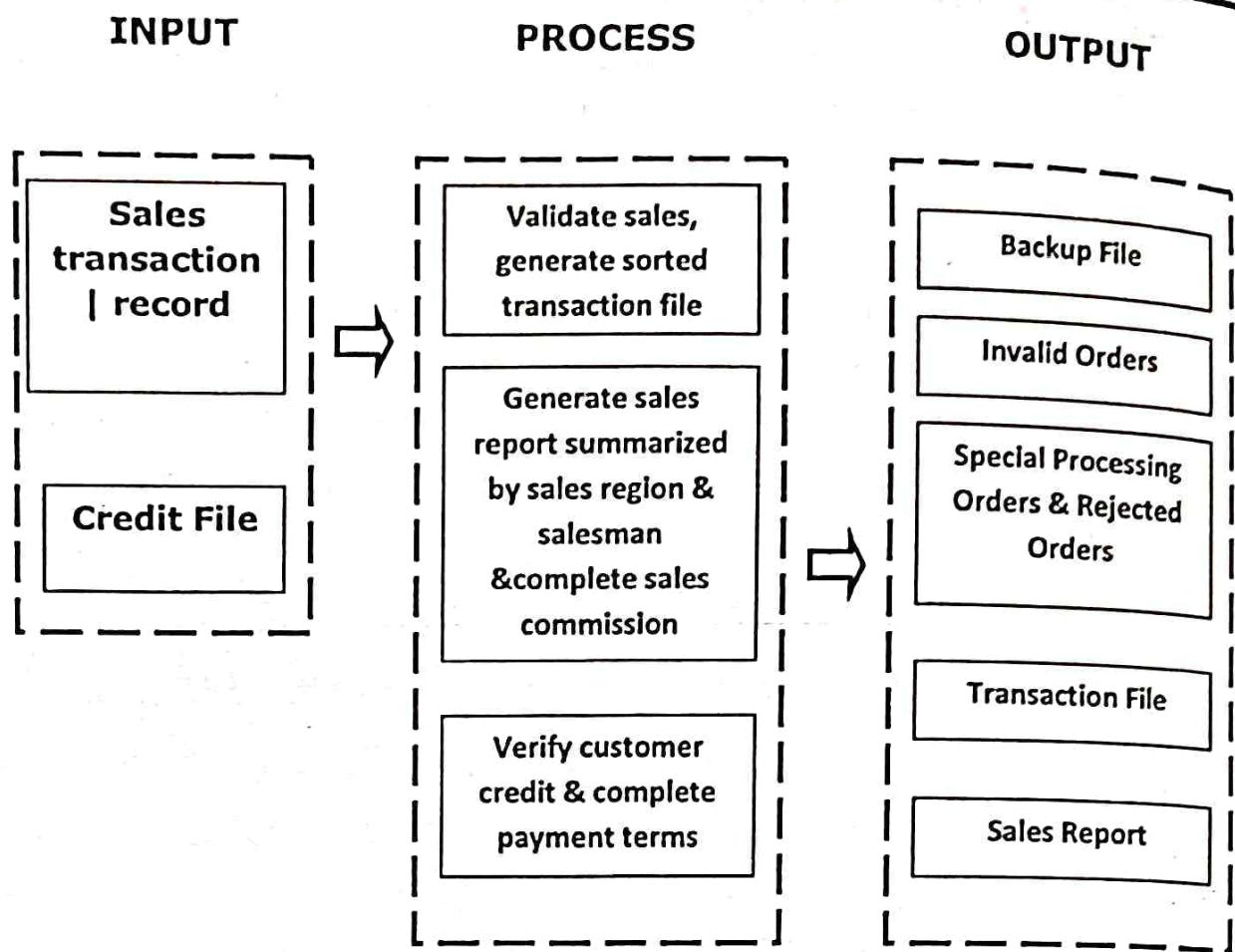
- HIPO is a popular system analysis design aid and documentation technique representing the module of a system as a hierarchy and for documenting each module.
- They assist the analyst in answering three guiding questions:-
 - What does the system or module do? (Asked when designing the system)
 - How does it do it? (Asked when reviewing the code for testing or maintenance)
 - What are the inputs and outputs? (Asked when reviewing the code for testing or maintenance).

➤ A Basic Components of HIPO:-

[I] Visual Table of Contents (VTOC)

[II] Functional Diagrams.





➤ Advantages of HIPO :

- HIPO allows a program or a system to be easily understood.
- HIPO is design, development & documentation tool.
- HIPO packages provide a common base for education & communication.
- HIPO has less duplication on information & more information can be obtained in a glance.
- It is a top-down approach with successive levels going for greater details.

~~MOST IMP~~

LEVELS of TESTING

- Except for small programs, systems should not be tested as a single.
- Large systems are built out of subsystems, which are built out of modules, which in turn composed of procedures & functions.
- The testing process should proceed in stages where testing is carried out incrementally.
- The most widely used testing process consists of five stages as shown in the figure below :

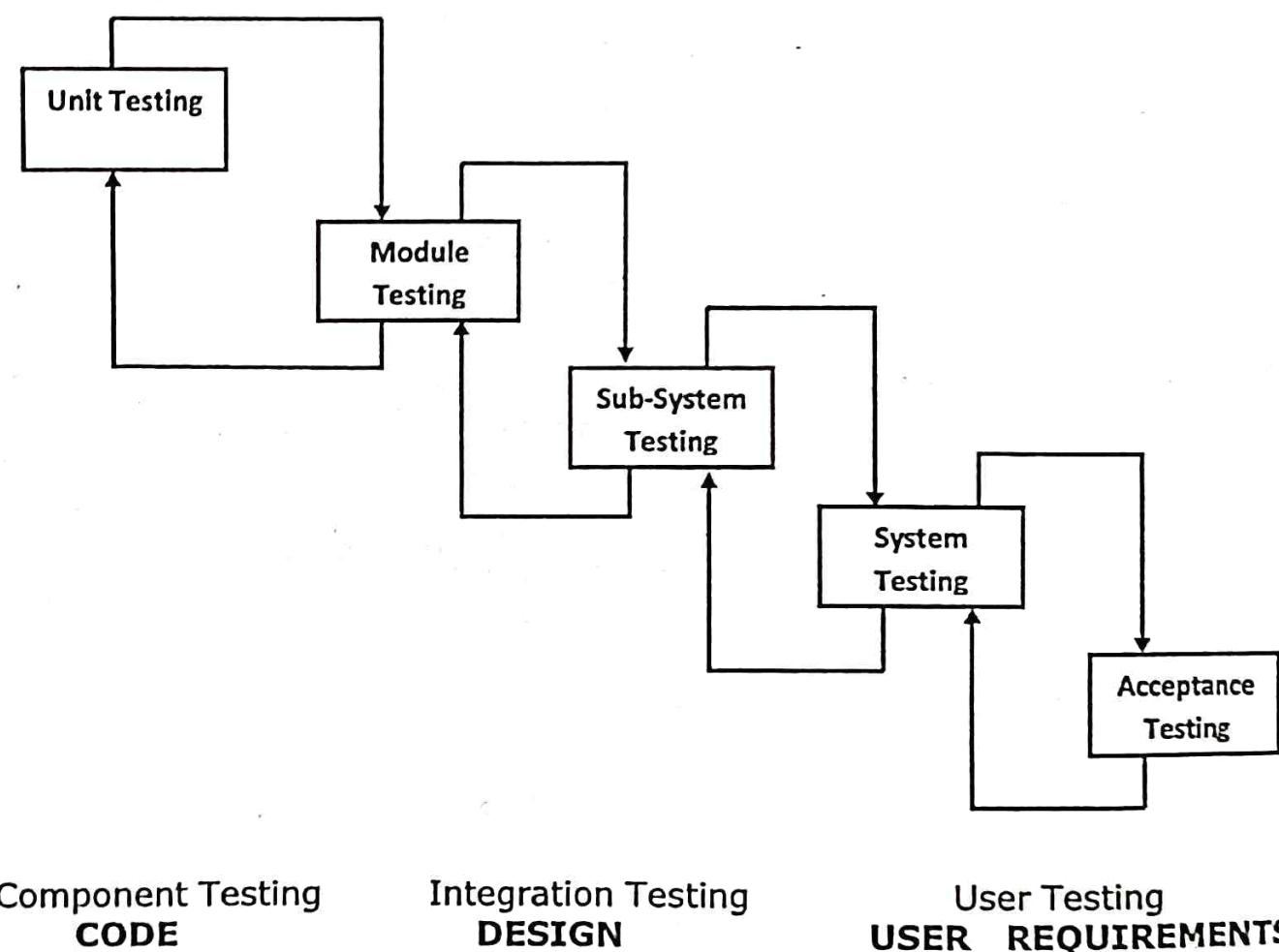


Figure show:-Testing Process

➤ The process is an iterative one with information being fed back from later stages to earlier parts of the process.

➤ **The stages in the testing process are :**

- [1] Unit Testing
- [2] Module Testing
- [3] Sub System Testing
- [4] System Testing
- [5] Acceptance Testing.

Explain level of testing in details:-

[1] Unit Testing:-

- Unit testing is a method of software testing by which individual units of source code are tested to determine if they are fit for use .A unit is the smallest testable part of an application.
- Testing of individual software component or modules. Typically done by the programmer and not by tester, as it requires detailed knowledge of the internal program design and code.
- Unit testing is also known as program testing.
- In unit testing the analyst tests the programs making up a system.
- A unit is smallest testable piece of software:-
 - Can be compiled, Linked, Loded.
 - Normally done by programmer.
 - Test cases written after coding
 - Example:-function/procedures, classes, interface.

- In a large system, many modules at different levels are needed.
- Unit testing focuses first on the modules, independently of one another, to locate errors.
- This enables the tester to detect errors in coding & logic that are contained within that module alone.
- **For Example:** a hotel information system consists of modules to handle reservation; guest checking and checkout, restaurant, room service & miscellaneous charges; accounts receivable billing.
- For each, it provides the ability to enter, change or retrieve data & respond to inquiries or print reports.
- Unit testing can be performed from the bottom up, starting with the smallest & lowest-level modules & proceeding one at a time.

[2] Module Testing:-

- Module testing is concerned with the testing of the smallest pieces of software for which separate specification exists.
- A module is a collection of dependent components such as an object class, an abstract data type or some looser collection of procedures & functions.
- A module encapsulates related components so that it can be tested without other system modules.
- A module test is performed with the knowledge of the module internal in mind.

[3] Sub System Testing:-

- The sub-system test process should therefore concentrate on the detection of interface error by rigorously exercising these interfaces.
- This phase involves testing collection of modules, which have been integrated into sub systems.
- Sub systems may be independently designed & implemented.

[4] System Testing:-

- Entire system is tested as per the requirement, Black box type testing that is based on overall requirement specification ,cover all combined part of a system.
- System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.
- The sub systems are integrated to make the entire system.
- The testing process is concerned with finding errors.
- It tests to find discrepancies between the system & its original objective, current specifications & systems documentation.

[5] Acceptance Testing:-

- At last the system is delivered to the user for Acceptance testing.

- Acceptance test are specified by the customer and analyst to test that the overall system is functioning as required (do developer builds the right system).
- Software developers often distinguish acceptance testing by the system provider from acceptance testing by the customer (the user or client) prior to accepting transfer of ownership.
- **There are TWO types of Acceptance Testing:-**
 - [I] Alpha Testing
 - [II] Beta Testing.

[I] Alpha Testing:-

→ Alpha testing is the final test in the testing process before the system is accepted for operational use.

→ It involves testing the system with data supplied by system procurer rather than simulated data developed as part of the testing process.

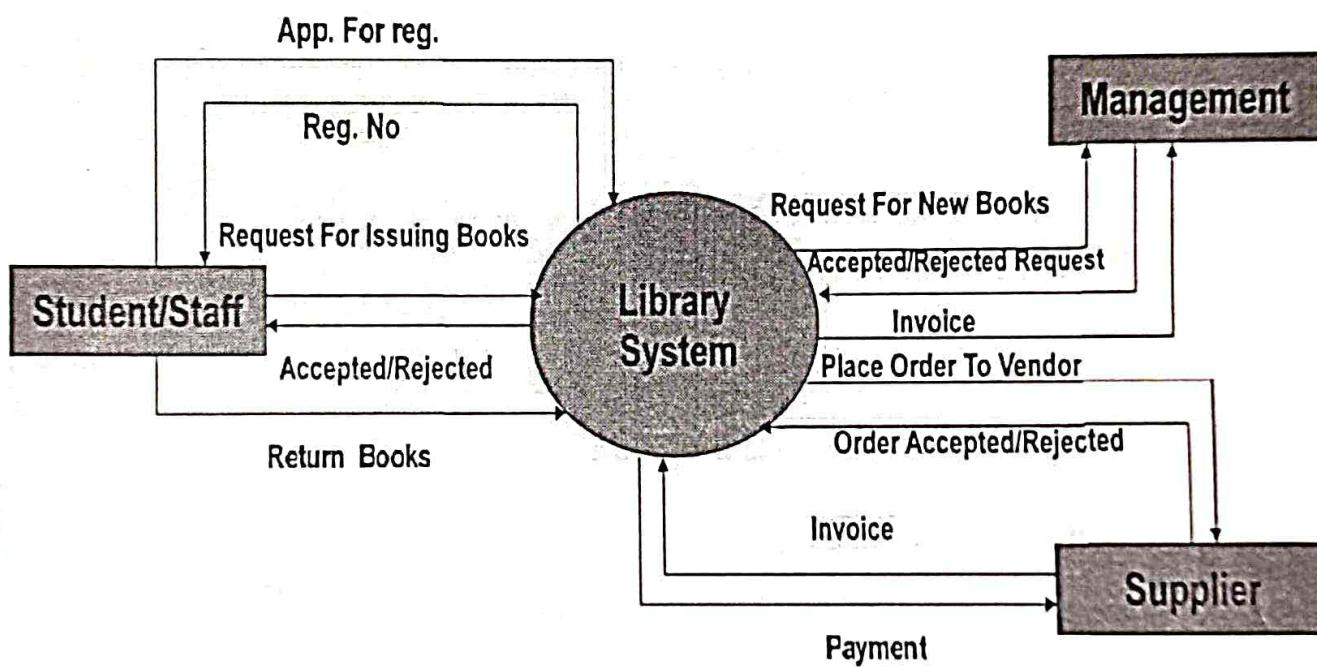
→ It often reveals errors & omissions in the system definition.

[II] Beta Testing:-

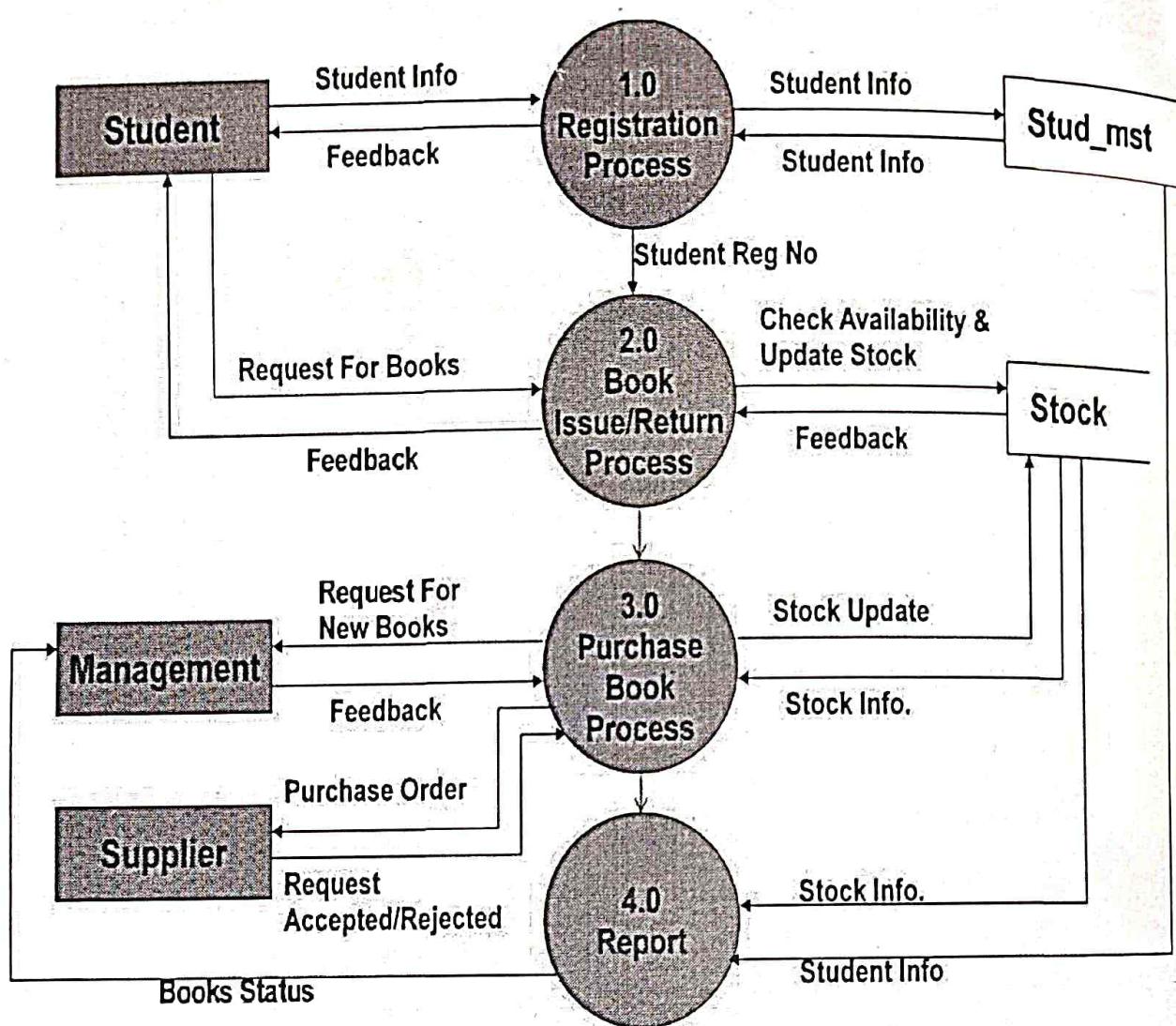
→ When a system is to be tested as a software product a testing process called beta testing is often used.

LIBRARY MANAGEMENT SYSTEM

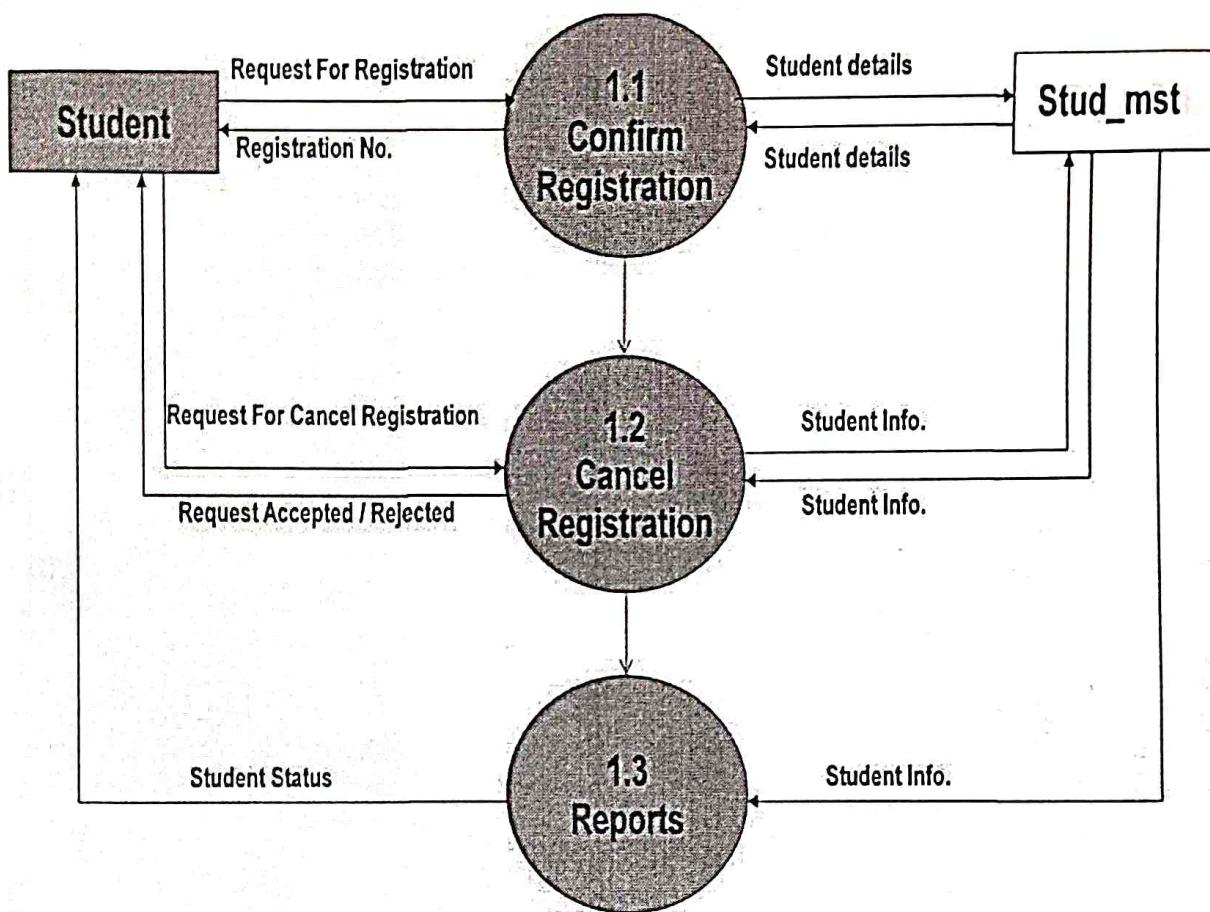
— Context Level DFD For Library System



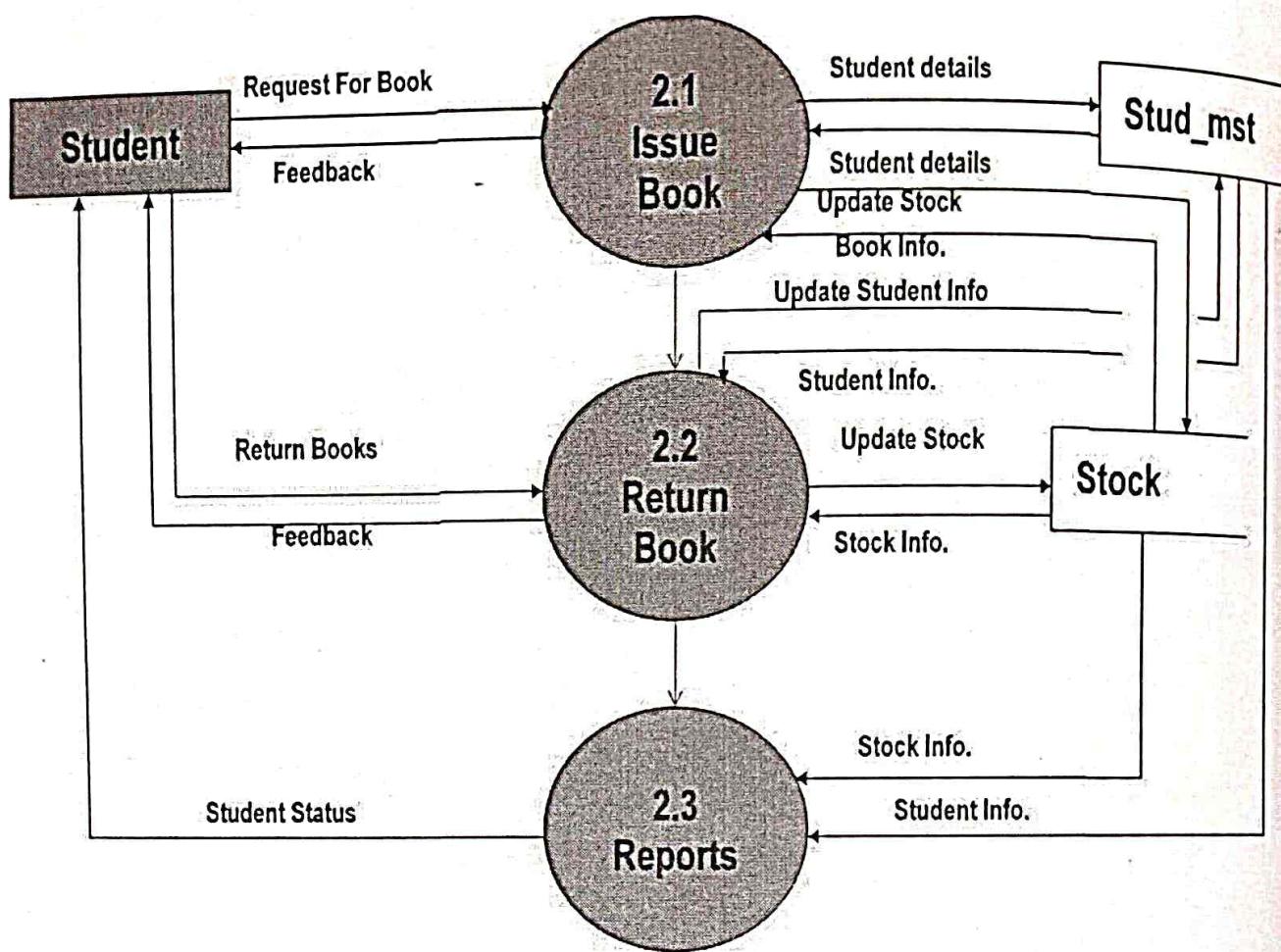
First Level DFD For Library System



Second Level DFD For Process 1.0(Registration)

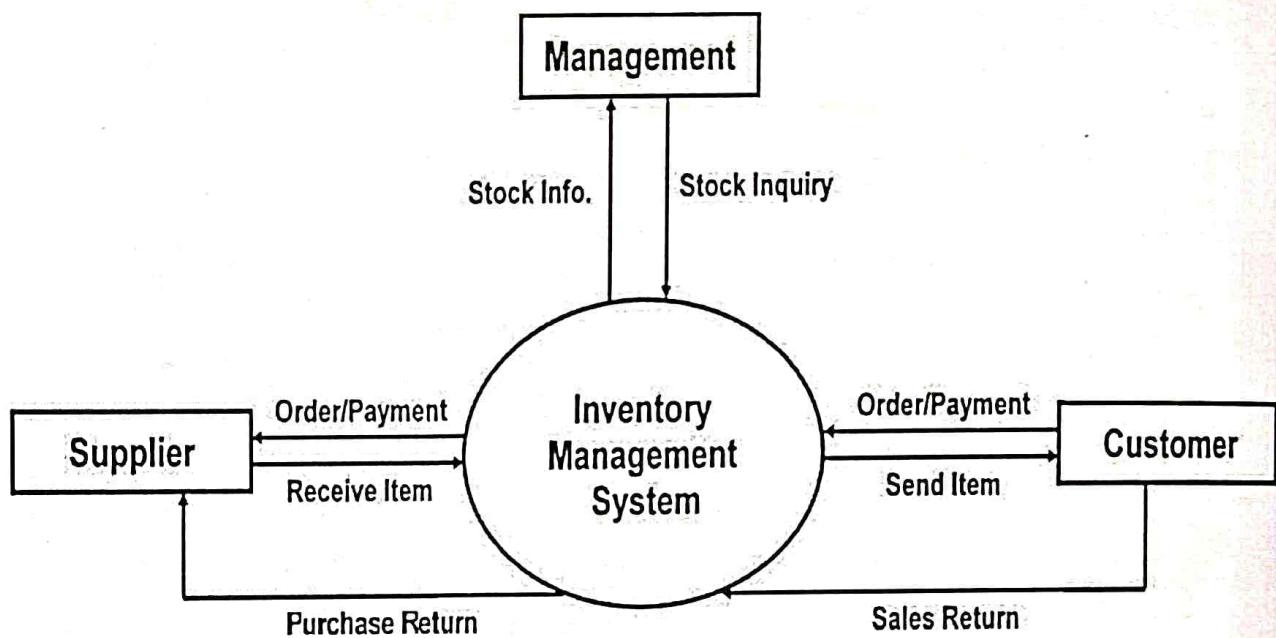


Second Level DFD For Process 2.0 (Book Issue/Return)

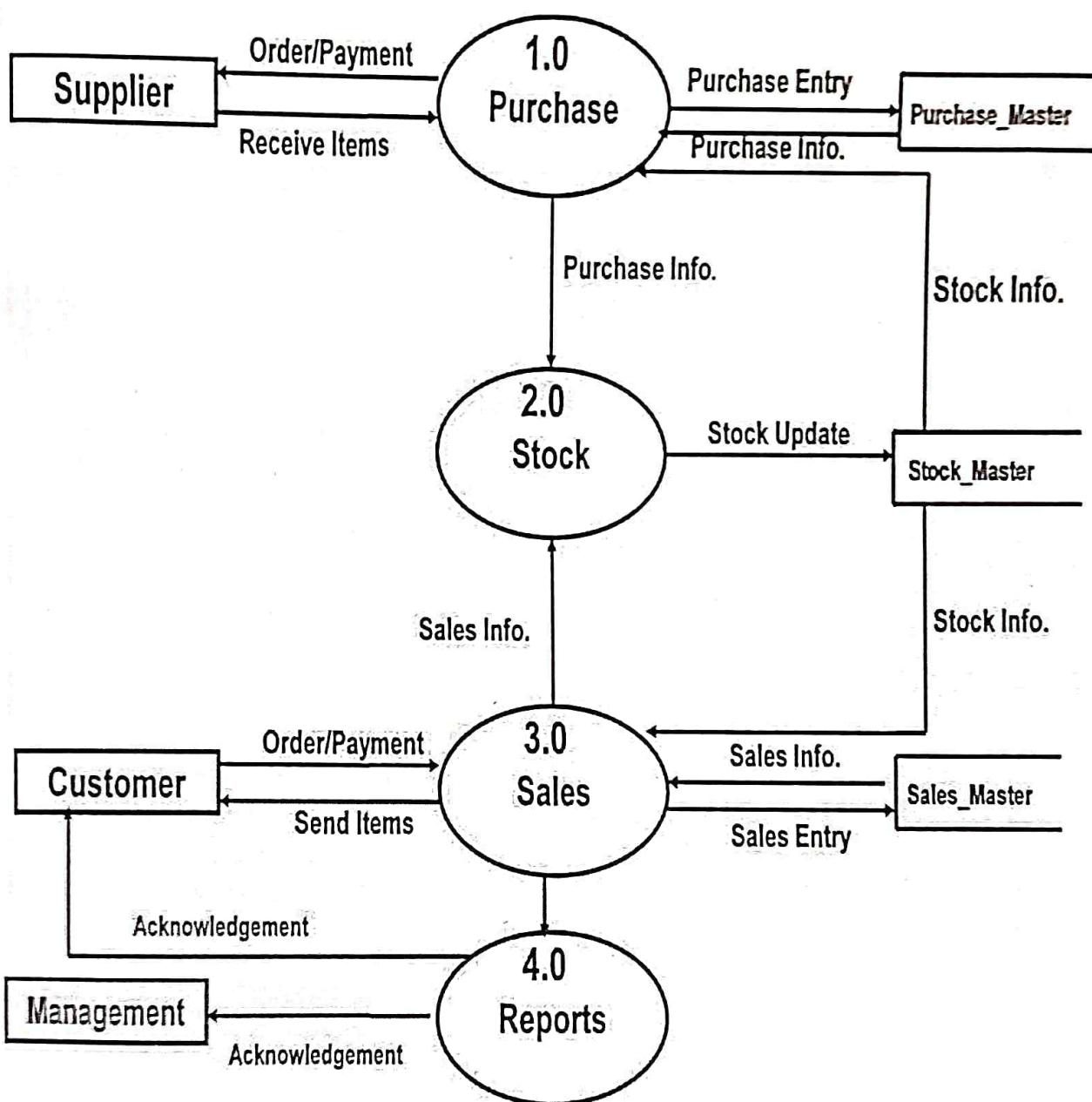


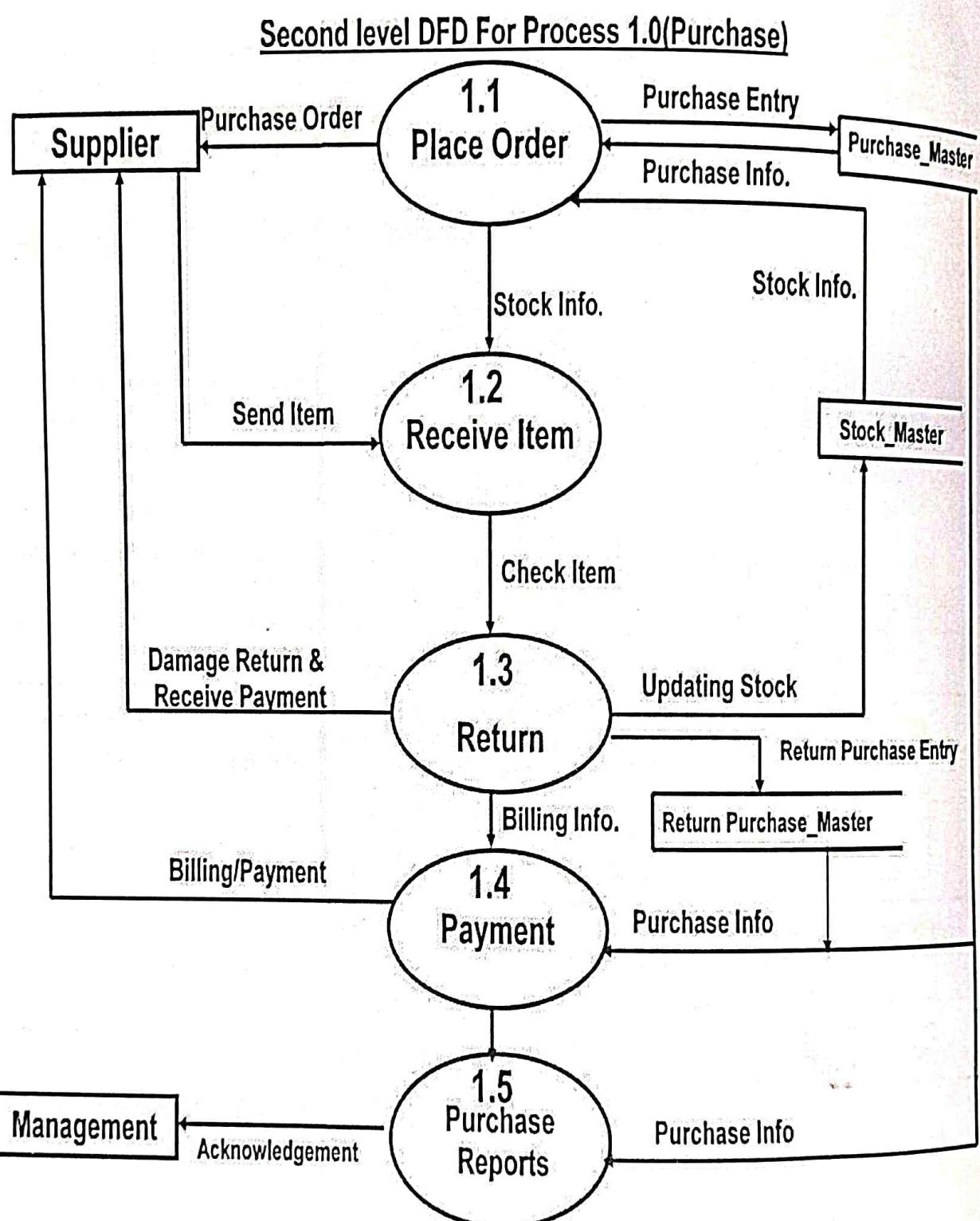
INVENTORY MANAGEMENT SYSTEM

Context Or 0 (Zero) Level DFD For Inventory Management System



First Level DFD For Inventory Management System





Second level DFD For Process 3.0(Sales)