# Systems Analysis And Design

**MCS-014**

***For***

**Master In Computer Applications [MCA]**

***By***

**Dinesh Verma**

MCA, BCA, 'A' & 'O' Level, SCJP(USA), JP

**Dr. A.K. Saini**

Ph.D, M.PHIL., MCA, M.Sc., MBA


***Useful For***

IGNOU, KSOU (Karnataka), Bihar University (Muzaffarpur), Nalanda University, Jamia Millia Islamia, Vardhman Mahaveer Open University (Kota), Uttarakhand Open University, Kurukshetra University, Himachal Pradesh University, Seva Sadan's College of Education (Maharashtra), Lalit Narayan Mithila University, Andhra University, Pt. Sunderlal Sharma (Open) University (Bilaspur), Annamalai University, Bangalore University, Bharathiar University, Bharathidasan University, Centre for distance and open learning, Kakatiya University (Andhra Pradesh), KOU (Rajasthan), MPBOU (MP), MDU (Haryana), Punjab University, Tamilnadu Open University, Sri Padmavati Mahila Visvavidyalayam (Andhra Pradesh), Sri Venkateswara University (Andhra Pradesh), UCSDE (Kerala), University of Jammu, YCMOU, Rajasthan University, UPRTOU, Kalyani University, Banaras Hindu University (BHU) and all other Indian Universities.

Closer to Nature  We use Recycled Paper

GPH BOOK®

**GULLYBABA PUBLISHING HOUSE PVT. LTD.**

ISO 9001 & ISO 14001 CERTIFIED CO.

Published by:

**GullyBaba Publishing House Pvt. Ltd.**

**Regd. Office:**
2525/193, 1st Floor, Onkar Nagar-A,
Tri Nagar, Delhi-110035
(From Kanhaiya Nagar Metro Station Towards Old Bus Stand)
011-27387998, 27384836, 27385249
+919350849407

**Branch Office:**
1A/2A, 20, Hari Sadan,
Ansari Road, Daryaganj,
New Delhi-110002
Ph. 011-45794768

**E-mail:** hello@gullybaba.com, **Website**:GullyBaba.com, GPHbook.com

**New Edition**

**Price:** ₹149/-

**ISBN:** 978-81-89086-45-9

# Preface

This book is mainly targeted for MCA (New Course) exam of Systems Analysis And Design. It has been introduced in market after seeing the huge demand of ready to grasp material for exams with high level of quality, and its un-availability in market. We the GullyBaba Publishing House took a step ahead to publish the quality material focusing on exams at the same time giving you indepth knowledge about the subject.

GPH Book is the pioneer effort that provides a unique methodology so as to perform better in exams. If your goal is to attain higher grade use this powerful study tool independently or along with your text.

*On the Web :* ***www.gullybaba.com*** *is the vital resource for your exams acting as catalyst to boost up your preparation. Now you can access us on the net through* ***www.doctorignou.com, www.doeacconline.com, www.gphindia.com, and www.gullybabapublishinghouse.com.***

***Feedback about the book can be sent at*** **mcs014writers@gullybaba.com.**

***Acknowledgments :***

*We appreciate the staff and facility support provided by GullyBaba Publishing House. In particular, we appreciate the encouragement and professional advises received from Mr. Ajay Saini, Mr. Tarun Sharma, H. Faheem Ahmed and many more.*

*Also we would like to thank typesetting and designing team Mrs. Bhawna Verma, Mr. Shravan Vats, Mr. Mukesh.*

*We gratefully acknowledges the significant contributions of Mr. Mahesh Chand, Mrs. Bimla Devi, Mrs. Bhawna Verma and our experts in bringing out this publication.*

*New Delhi*

# Contents



# Question Papers

# Chapter-1

# Introduction To SAD

## INTRODUCTION

This chapter introduces the focus on fundamental of systems and various approaches for development of information systems. The study of "SYSTEMS" is not a new or a recent endeavour. People freely talk of different types of systems in their day-to-day life.

**Q1. Define the term system. Also Explain various types of Systems ?**
**A system is** an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal, where every component perform a particular task.
**Types of Systems :** Systems have been classified in different ways. Common classifications are:
**(i)** Physical or abstract systems
**(ii)** Open or closed systems
**(iii)** Deterministic or probabilistic systems
**(iv)** Man-made information systems

**(i) Physical or Abstract Systems**
Physical systems are tangible entities that may be static or dynamic in operation. Abstract systems are conceptual or non-physical entities which may be as straightforward as formulas of relationships among sets of variables or models - the abstract conceptualization of physical situations.

**(ii) Open or Closed Systems**
An open system continually interacts with its environments. It receives inputs from and delivers output to the outside. An information system belongs to this category, since it must adapt to the changing demands of the user. In contrast, a closed system is isolated from environmental influences. In reality completely closed systems are rare.

**(iii) Deterministic or Probabilistic Systems**
A deterministic system is one in which the occurrence of all events is perfectly predictable. If we get the description of the system state at a particular time,

the next state can be easily predicted. An example of such a system is a numerically controlled machine tool. Probabilistic system is one in which the occurrence of events cannot be perfectly predicted. An example of such a system is a warehouse and its contents.

**(iv) Man-made Information Systems**

It is generally believed that information reduces uncertainty about a state or event. For example, information that the Foreign Direct Investment has doubled in India during 2004-2005 reduces the uncertainty that a share market will go down. An information system is the basis for interaction between the user and the analyst. It determines the nature of relationship among decision makers. In fact, it may be viewed as a decision centre for personnel at all levels. From this basis, an information system may be defined as a set of devices, procedures and operating systems designed around user-based criteria to produce information and communicate it to the user for planning, control and performance. Many practitioners fail to recognise that a business has several information systems; each is designed for a specific purpose. The major information systems are:

- formal information systems
- informal information systems
- computer based information system

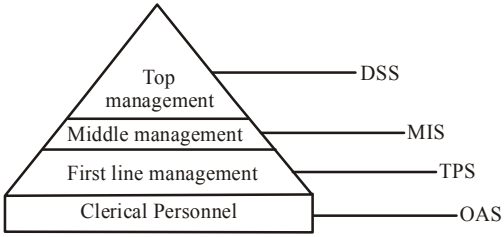
*A Formal information system is based on the organisation represented by the organisation chart. The chart is a map of positions and their authority relationships, indicated by boxes and connected by straight lines. It is concerned with the pattern of authority, communication and work flow.

*An Informal information system is an employee-based system designed to meet personnel and vocational needs and to help in the solution of work-related problems. It also funnels information upward through indirect channels. In this way, it is considered to be a useful system because it works within the framework of the business and its stated policies.

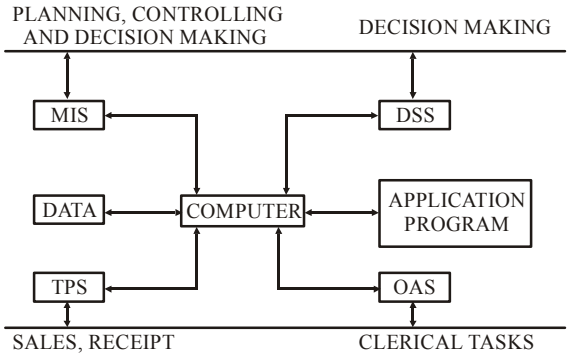
* Third category of information system depends mainly on the computer for handling business applications. Systems analysts develop several different types of information systems to meet a variety of business needs. There is a class of systems known collectively as Computer Based Information Systems. As we have different types of transportation systems such as highway system, railway system and airline system, computer based information systems are of too many types. They are classified as:

• Transaction Processing Systems (**TPS**)
• Management Information Systems (**MIS**)
• Decision Support Systems (**DSS**)
• Office Automation Systems (**OAS**).

The figure shows the organisation chart of computer based information system (CBIS) and the hierarchical view of CBIS.




**The Hierarchical View of CBIS**




**CBIS in an Organisational Context**

**Q2. Differentiate the following :**
**(a) Physical and Abstract systems**
**(b) Open and Closed systems**
**(c) Deterministic and Probabilistic systems**

**(a) Physical or Abstract Systems**

Physical systems are tangible entities that may be static or dynamic in operation. Abstract systems are conceptual or non-physical entities which may be as straightforward as formulas of relationships among sets of variables or models - the abstract conceptualization of physical situations.

**(b) Open or Closed Systems**

An open system continually interacts with its environments. It receives inputs from and delivers output to the outside. An information system belongs to this category, since it must adapt to the changing demands of the user. In contrast, a closed system is isolated from environmental influences. In reality completely closed systems are rare.

**(c) Deterministic or Probabilistic Systems**

A deterministic system is one in which the occurrence of all events is perfectly predictable. If we get the description of the system state at a particular time, the next state can be easily predicted. An example of such a system is a numerically controlled machine tool. Whereas, on the other side in Probabilistic system is one in which the occurrence of events cannot be perfectly predicted. An example of such a system is a warehouse and its contents.

**Q3. Write brief note on the characteristics of system.**

**Following are the characteristics of system :**

**a. Organisation:**

Organisation implies structure and order. It is the arrangement of components that helps to achieve objectives. In the design of a business system, for example, the hierarchical relationships starting with the president on top and leading downward to the blue-collar workers represents the organisation structure.

**b. Interaction:**

Interaction refers to the procedure in which each component functions with other components of the system. In an organisation, for example, purchasing must interact With production, advertising with sales and payroll with personnel. In a computer system also, the central processing unit must interact with other units id solve a problem.

**c. Interdependence:**

Interdependence means that component of the organisation or computer system depend on one another. They are coordinated and linked together in a planned way to achieve an objective.

**d. Integration:**

Integration is concerned with how a system is tied together. It is more than sharing a physical part or locations. It means that parts of the system work together within the system even though each part performs a unique function.

**e. Central Objective:**
Central objective is the last characteristic of a system. Objectives may be real or stated. Although a stated objective may be the real objective, it is quite common that organisation may set one objective and operate to achieve another.

**Q4. What are the various approaches for development of Information Systems. Explain JAD.**
**Various approaches for development of Information Systems are :**
**Model Driven :** It emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.

**Accelerated Approach :** A prototyping approach emphasizes the construction of model of a system. Designing and building a scaled-down but functional version of the desired system is known as Prototyping. A prototype is a working system that is developed to test assumptions and ideas about the new system. It consists of working software that accepts input, perform calculations, produces printed or display information or perform meaningful activities.

**Joint Application Development :** It is defined as a structured approach in which users, managers and analysis work together for several days in a series of intensive meetings to specify or review system requirements. In this approach, requirements are identified and design details are finalized.

**Q5. List the fundamental principles of SDLC.**
The fundamental principles of SDLC are :
**1.** Management and users should be involved because they can explain the problem that is to be taken for design and development in accurate manner.
**2.** A problem solving approach should be followed.
**3.** Phases and activities should be established.
**4.** Some standards should be maintained for consistent development of System.
**5.** Development of Information System should be considered as capital Investment.
**6.** Divide and conquer approach should be adopted. It is one approach of making complex problem easier.
**7.** For making a system successful, it should be designed for change and growth.

# Chapter-2

# System Analyst-A Profession

This chapter describes the role of the analyst in system development and the interface maintained between the analyst and the user. Designing and implementing systems to suit organisational needs are the functions of the system analyst. The role of a system analyst designs an information system is the same as an architect of a house. He/She plays a major role in seeing business benefit from computer technology. The analyst is a person with unique skills. The job of an analyst is not confined to data processing such as, because as it deals heavily with people's procedures and technology. Commonsense a structured framework, and a discipline approach to solving problems are a part of analysis. Three groups of people are involved in developing information systems for organisations. They are managers, users and computer programmers who implement the system. Many individuals in business can benefit by learning the functions and techniques used by the analyst. Users, managers, accountants and auditors must understand how the analyst relates to them.

The role of the analyst has been emerging with changing technology. System analyst plays an impart role in the system development. It facilitates the development of information system and computer application. The system analyst must need to interact with each of the existing person or slake hold or in the system. System analyst typically identify and validates the business problems and needs of system owners and users. For the system designers, and builders, the analyst ensure that the technical solutions fulfill the business needs and then integrates the technical solution into business.

In simple terms, system analysts are people who understand both business and computing. They study business problems and opportunities and then transform business and information requirements into the computer based information system that are implemented by various technical specialists including computer programmer.

**Q1. What is Systems Analyst?**

When *information technology* is used, the analyst is responsible for the efficient capture of data from its business source, the flow of that data to the computer, the processing and storage of the data by the computer, and the flow of usual

and timely information back to the business and its people.
A **systems analyst** studies the problems and needs of an organization to determine how people, data, processes, communications, and information technology can best accomplish improvements for the business.

**Q2. Discuss the role of a Systems Analyst, in a software development organisation.**
**Following are the roles of a System Analyst :**
**As a Architect :** The analyst's role as an architect is liaison between the user's logical design requirements and the detailed physical system design. As architect the analyst also creates a detailed physical design of candidate systems. A systems analyst makes the design of information system architecture on the basis of end user requirements. This design becomes the blue print for the programmers.

**As a Psychologist :** As the system development is performed around people. The analyst plays the role of psychologist in the way s/he reaches people, interprets their thoughts, assesses their behavior and draws conclusions from these interactions. Psychologist plays a major role during the phase of fact finding.

**As a Motivator :** System acceptance is achieved through user participation in its development, effective user training and proper motivation to use the system. The analyst's role as a motivator becomes obvious during the first few weeks after implementation and during times when turnover results in new people being trained to work with the candidate system.

**As a Intermediary :** While implementation the system developed, the analyst tries to appease all parties involved. Diplomacy in dealing with people can improve acceptance of the system. The analyst's goal is to have the support of all the users. S/he represents their thinking and tries to achieve their goals through computerization.

**As a Change Agent :** The analyst may be viewed as an agent of change. As it is important that the user accepts change. For user acceptance, analysts prefer user participations during design and implementation. Analyst carefully plans, monitors and implements change into the user domain because people inherently resist changes. In the role of a change agent, Systems Analyst may use different approaches to introduce changes to the user organization.

**As a Investigator and Monitor :** S(he) may investigate the existing system

to find the reasons for it's failure. The role of an investigator is to extract the problems from existing systems and create information structure that uncovers previously unknown trends that may have a direct impact on organization. The role of a Monitor is to undertake and successfully complete a project. In this role, analysts must monitor programs in relation to time, cost and quality.
All these multiple roles helps the analyst to handle the problem in a logical way, and pay attention to details.

**Q3. What are the duties and attributes of an effective system Analyst? Explain them.**
**#: What are the attributes of a good System Analyst ? List and Explain them.**
Following are the duties and attributes of system analyst :
**DUTIES**
**(a)** The first and perhaps most difficult task of systems analyst is problem definition. Business problems are quite difficult to define. It is also true, that problems cannot be solved until they are precisely and clearly defined.
**(b)** Initially a systems analyst does not know how to solve a specific problem. He must consult with managers, users and other data processing professionals in defining problems and developing solutions. He uses various methods for data gathering to get the correct solution of a problem.
**(c)** Having gathered the data relating to a problem, the systems analyst analyses them and thinks of plan to solve it. He may not come up personally with the best way of solving a problem but pulls together other people's ideas and refines them until a workable solution is achieved.
**(d)** Systems analysts coordinate the process of developing solutions. Since many problems have number of solutions, the systems analyst must evaluate the merit of such proposes solution before recommending one to the management.
**(e)** Systems analysts are often referred to as planners. A key part of the systems analyst's job is to develop a plan to meet the management's objectives.
**(f)** When the plan has been accepted, systems analyst is responsible for designing it so that management's goal could be achieved. Systems design is a time consuming, complex and precise task.
**(g )** Systems must be thoroughly tested. The systems analyst often coordinates the testing procedures and helps, in deciding whether or not the new system is meeting standards established in the planning phase.

**ATTRIBUTES:**
**(a)** Knowledge of Business functions: A systems analyst must know the environment in which he or she works. He must be aware of the peculiarities of management and the users at his installation and realize how they react to

systems analyst. A working knowledge of accounting and marketing principles is a must since so many systems are built around these two areas. He must be familiar with his company's product and services and management's policies in areas concerning him.

**(b)** Knowledge of people: Since a systems analyst works with others so closely, he or she must understand their needs and what motivates them to develop systems properly.

**(c)** Knowledge of Data processing principles: Most systems today are computer based. The systems analyst must be fully aware about the potential and limitations of computers.

**(d)** Ability to communicate: As a coordinator, a systems analyst must communicate properly with people of different levels within an organisation. Systems analyst must listen carefully to what others say and integrate the thoughts of others into the systems development process.

**(e)** Flexibility: Systems analysts must be flexible in their thinking since they often do not get-their own way. Different sections in an organisation have conflicting needs and most systems are the result of compromise. The analyst's goal is to produce the system that will be the best for his organisation. This requires an open mind and flexibility in his ideas.

**(f)** An analytical mind: It takes an unusual person to see through problems facing an organisation and develop solutions that will work. Systems analysts often find themselves with more data than they can cope with. It requires an analytical mind to select pertinent data and concentrate on them in defining problems and forming solutions.

**(g)** Well educated with sharp mind: Systems analysts are called upon to work with people at all levels virtually in every aspect of business. They must know how to work with all of them and gain their confidence. Analysts must have sharp mind to learn quickly how people do their jobs.

**Q4. Explain the role of a System Analyst. Also mention what kind of qualities s(he) should possess.**

Refer to Q2 & Q3

**Q5. List the qualifications of system analyst.**

**1.** Technical knowledge.
**2.** Planning, organizing, scheduling.
**3.** Maintaining customer relation.
**4.** Providing supervision and leadership.
**5.** Training others.
**6.** Documentation.
**7.** Maintaining communication.

**8.** Assessing customer needs and providing recommendation.
**9.** Job commitment and effort.
**10.** Debugging.
**11.** Program, system modification, development.
**12.** Conducting presentations.

**Q6. Shown by diagram the relationship between a system analyst's skills and system development life cycle.**

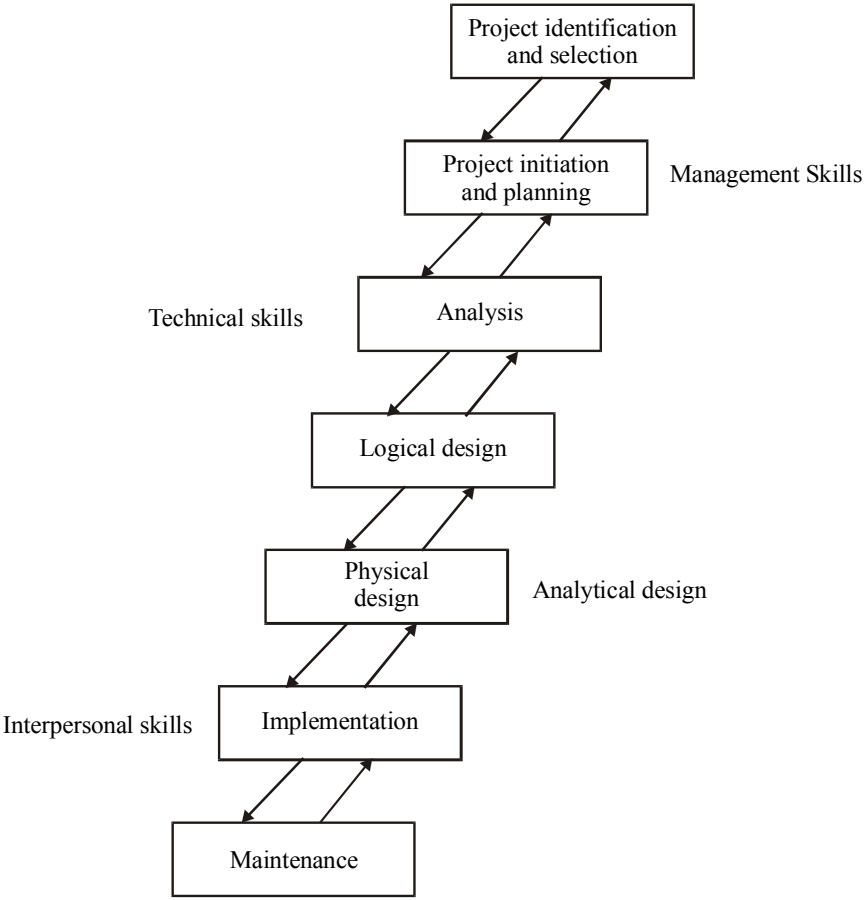



**Diagram showing Relationships between system analyst's skills and system development life cycle**

**Q6. Are excellent programmers necessarily excellent systems analysts? Justify your answer.**

An excellent programmer is not necessarily an excellent system analyst. A programmer is given clear specification (often in writing) and designs efficient and maintainable programs. She/he need not have good communication skills and inter-personal relations. She/he need not have knowledge of functionality of organizations. In other words, a programmer works with clear specifications, whereas an analyst has to arrive at clear specifications from unclear requirements.

**Q7. Why should a systems analyst be able to communicate well ?**

He has to understand the users' requirements mostly by interviewing them and thus he has to ask the right questions, listen carefully and summarize the list of conversation. He should also be able to present and explain orally to the users, the system designed by him and clarify doubts they may have after the oral presentation or explanation. His main job is to interact with the management, users and the programmers, so it is obvious that he must have good communication skills.

**Q8. Give an example of a situation which would demand greater interpersonal skills than technical skills from a systems analyst and vice-versa.**

**#: Explain about Interpersonal skills and Technical Skills.**

The system analyst must possess various skills to effectively carry out the job. The skills (attributes) to be possessed by the system analysts are broadly divided in to following categories. These are

**1. Interpersonal Skills**

**2. Technical skills**

**The Interpersonal skills in context with the system are as follows:**

• **Communication :** The system analyst should have a very good communication skill. He is supposed to interact with the other person of the organisation. Communication means listening others, feeling what other says and coming to one another.

• **Understanding:** Identifying the problems and assessing their solutions with the consideration and constraints proposed by the organisation.

• **Teaching:** Educating the people to understand computer, its user how it works, what more and more can be obtained through the computer.

• **Selling:** Selling ideas and promoting innovations in problem solving using computers.

• **Creativity:** Helping users model ideas into concrete plans and developing candidate systems to match user requirements.

• **Project management:** Scheduling, performing well under time constraints,

coordinating team efforts and managing costs and expenditure.
• **Dynamic interface:**
• Questioning attitude and inquiring mind.
• Knowledge of basic of the computer and business functions.

**Technical Skills:** In order to develop computer based information systems, systems analyst must understand information technologies, their potentials and their limitations. In general, a systems analyst should be familiar of technologies such as :
Microcomputers, workstations, minicomputers, and mainframe computers,
Programming languages,
Operating systems, both for PC's and networks.
Database and File management system,
Data communication standards and software for local and wide area networks,
System development tools and environments (such as forms & report generators and graphical user interface design tools), and
Decision support systems and data analysis tools.

**Situation where greater interpersonal skills are required than technical skills from a system analyst are following :**
• During analysis, there is greater need for interpersonal skills – working with the user to determine requirements and transact them into design criteria.
• During design, it requires highly technical procedures and methodologies.
• During implementation, analyst / user interface and user participation is more emphasized.
• During program construction, loading and testing are carried out with some user participation.

**Situation where greater technical skills are required than interpersonal skills from a system analyst are following :**
• During design, when input, output screen design, module design etc.
• During implementation, when system developed is implemented at the client site
• During system testing, where it is worked out that which hardware / supporting software configuration will be the best to run the proposed system
• During maintenance, also technical skills are needed to make the system error free or update according to the current changed requirements.

# Chapter-3

# Process Of System Development

This chapter introduces the system development process used to develop information systems. System development is not a hit-or-miss process! As with any product, information systems must be carefully developed. Successful systems development is governed by fundamental, underlying principles. System development revolves around a life-cycle that begins with the recognition of user needs. The life-cycle is not a procedure that deals with hardware and software. It is building computer-based systems to help the user operate a business or make decisions effectively and manage enterprise successfully. A systems development process is a set of activities, methods, best practices, deliverables and automated tools the user or developer (stakeholders) use to develop and maintain information systems and software.

**Q1. Briefly describe the phases of a system development life cycle through a diagram. [DEC05, Q1(a)]**

System Development Life Cycle, or SDLC, is the process used by a systems analyst to develop an information system, including requirements, validation, training, and user ownership through investigation, analysis, design, implementation and maintenance. SDLC is also known as information systems development or application development. An SDLC should result in a high quality system that meets or exceeds customer expectations, within time and cost estimates, works effectively and efficiently in the current and planned Information Technology infrastructure, and is cheap to maintain and cost-effective to enhance. SDLC is a systems approach to problem solving and is made up of several phases, each comprised of multiple steps which are shown in the figure below :

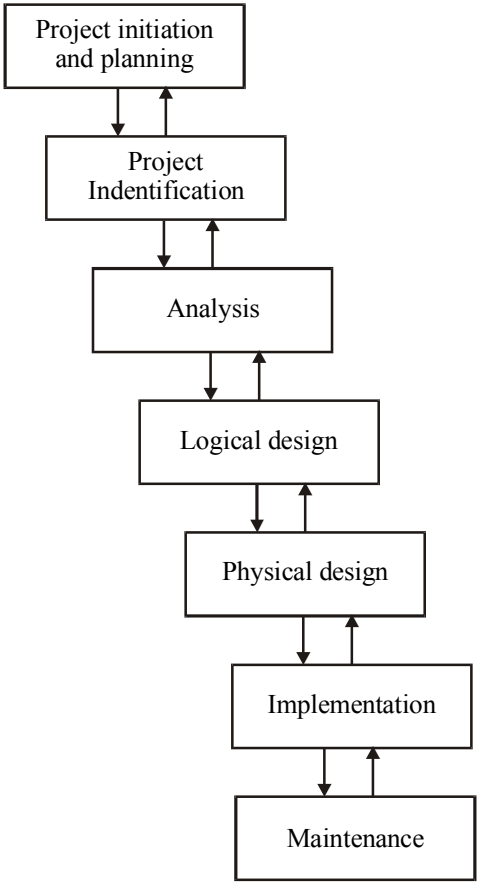



**Figure : System development life cycle**

**(i) Project Identification and Selection :** The first phase in the SDLC is called project identification and selection. In this phase, the user identifies the need for a new or improved system.

During project identification and selection, an organization determines whether or not resources should be devoted to the development or enhancement of each information system under consideration. The outcome of the project identification and selection process is a determination of which systems development projects should be undertaken by the organization at least in terms of an initial study.

**(ii) Project initiation and Planning :** The second phase is project initiation and planning. The problems that are identified should be investigated and a

decision to implement the information system or not for the organization should be taken.

**(iii) Analysis :** It has several sub-phases. The first is requirements determination. In this sub-phase, analysts work with users to determine the expectations of users from the proposed system. This sub-phase usually involves a careful study of current systems, manual or computerized that might be replaced or enhanced as part of this project. Next, the requirements are studied and structured in accordance with their inter-relationships and eliminate any redundancies. Third, alternative initial design is generated to match the requirements. Then, these alternatives are compared to determine which alternative best meets the requirement in terms of cost and labour to commit to development process.
In this phase, feasibility study of the proposed system is also performed.

**Various types of feasibilities are:**
- Technical feasibility
- Economic feasibility
- Behavioural feasibility
- Operational feasibility
- Legal feasibility
- Time feasibility.

If the proposed system is not feasible to develop, it is rejected at this very step. The output of the analysis phase is a description of (but not are detailed design for) the alternative solution recommended by the analysis team. Once, the recommendation is accepted by those with funding authority, you can begin to make plans to acquire any hardware and system software necessary to build or operate the system proposed.

**(iv) Logical Design :** After analysis phase is complete, design of the system begins. The design consists of logical and physical design of the system. Logical design concentrates on the business aspects of the system. Logical design is not tied to any specific hardware and systems software platform. Theoretically, the system could be implemented on any hardware and systems software. The idea is to make sure that the system functions as intended.

**(v) Physical design :** In physical design, the logical design is turned into physical or technical specifications. For example, you must convert diagrams that map the origin, flow, and processing of data in a system into a structured systems design that can then be broken down into smaller and smaller units known as modules for conversion to instruction written in a programming

language. The physical system specifications are turned over the programmers as the first part of the implementation phase.

**(vi) Implementation :** Application is installed or loaded, on existing or new hardware and users are introduced to new system and trained. During implementation, you turn system specification into working system that is tested and put into use. Implementation includes coding, testing and installation. During coding, programmers write programs that make up the system. During testing, programmers and analysts test the individual programs and the entire system in order to find and correct errors. During installation, the new system becomes a part of the daily activities of the organization. There are ways in which installation of the system can be done :

(i) **Direct conversion:** In this type of conversion, the software is directly installed at user's site.

(ii) **Parallel conversion:** In this type of conversion, both the old and new systems are run in parallel for some time. After monitoring the new system for a reasonable period of time and if it is performing well, then, the new system is implemented replacing the old one.

(iii) **Phased conversion:** In this types of conversion, the system is installed module by module.

Implementation activities also include initial user support such as the finalization of documentation, training programs, and ongoing user assistance.

**(vii) Maintenance :** This is the final phase in SDLC. When a system is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions. Also, the organization's requirements with respect to the system change with time. During maintenance, programmers make the changes that users ask for and modify the system to reflect and support changing business conditions. These changes are necessary to keep the system running and useful.

**Q2. Define SRS. Explain the seven characteristics of SRS.**

**[JUNE05, Q4(b)]**

**System requirement specification(SRS)** is a set of complete and precisely stated properties along with the constraints of the system that the software must satisfy. A well designed software requirements specification establishes boundaries and solutions of system to develop useful software. All tasks, however minute, should not be underestimated and must form part of the documentation.

**Following are the seven characteristics of SRS :**

**1.** All the requirements must be stated unambiguously. Every requirement stated has only one interpretation. Every characteristic of the final product must be described using a single and unique term.

**2.** It must be verifiable. The requirements are verified by system tester during system testing. So, all the requirements stated must be verifiable to know conformity to the requirements. No requirement should conflict with any other requirements.

**3.** The requirements should be realistic and achievable with current technology. There is no point in specifying requirements which are unrealisable using existing hardware and software technology.

**4.** It should be complete. The definition should include all function and constraints intended by the system user.

**5.** It should be modifiable. The structure and style of the SRS are such that any necessary changes to the requirements can be made easily, completely and consistently.

**6.** SRS should not only addresses the explicit requirement but also implicit requirements that may come up during the maintenance phase of the software. It must be usable during operation and maintenance phase. The SRS must address the needs of the operation and maintenance phase, including the eventual replacement of the software.

**7.** It should be traceable to other requirements and related documents. The origin of each requirement must be clear. The SRS should facilitate the referencing of each requirement for future development or enhancement of documentation.

**Q3. What are the products of SDLC phases ?**

**Following are the products of SDLC phases :**

Project Identification and Selection : Priorities for systems and project, architecture for data, networks, hardware and Information System Management are the result of the associated system.

**Project Initiation and Planning :** Detailed work plan for project, specification of system scope and high system requirements, assignment of team members and other resources.

**Analysis :** Description of current system, need to enhance or replace current system, explanation of alternative systems and justification of alternatives.

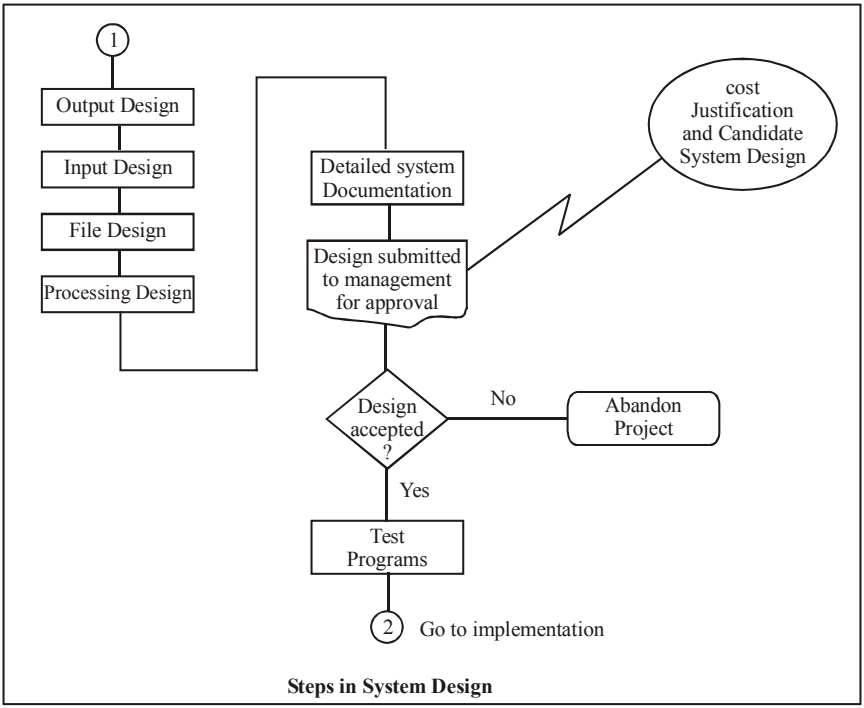
**Logical Design :** Functional and detailed specification of all system elements (data, process, input and output).

**Physical design :** Technical, detailed specifications of all system elements, i.e. programs, files, network, system software etc. and acquisition plan for new technology.

**Implementation :** Code, documentation, training programs and support capabilities.

**Maintenance :** New version of software with associated updates of documents, training and support.

**Q4. Show diagrammatic representation of steps in System Design.**




**Steps in System Design**

# Chapter-4

# Introduction To Documentation Of Systems

Documentation forms the foundation for successful development of a project and more importantly provide guidance for software maintenance task. Documentation establishes design and performance criteria for different phases of the software project. Effective and accurate documentation is very necessary for the success of any system. Documentation of a system describes scope and purpose of document, project objectives, performance issues, management and technical constraints, project estimates, project risks, Risk management, schedule project resources, staff organisation, tracking and control mechanisms etc. as well as any precondition and assumption about the state of the project. Documentation becomes part of each step of system development throughout the process of system development even before the documentation starts officially. During the process of development, study report and other necessary information is documented to help people involved in system development to understand the process. There are many kinds of documents namely analysis document, design, interface, internal program and user-oriented documentation. Documentation should not be seen as a overhead for system development rather, as an investment for the development of a high quality software.

**Q1. Define the term 'Documentation' according to ISO/ IEC 12207 : 1995. Also define SRS, and explain the structure of a typical SRS document. [DEC05, Q2(a)]**

The ISO standard ISO/IEC 12207:1995 describes documentation "as a supporting activity to record information produced by a system development life cycle process".

**System requirement specification(SRS)** is a set of complete and precisely stated properties along with the constraints of the system that the software must satisfy. A well designed software requirements specification establishes boundaries and solutions of system to develop useful software. All tasks, however minute, should not be underestimated and must form part of the documentation.

**Following is the Structure of a Typical SRS Document :**

**1. Introduction**

* System reference and business objectives of the document.
* Goals and objectives of the software, describing it in the context of the computer-based system.
* The scope of the document.

**2. Informative description about the system**

* Information flow representation.
* Information content and structure representation.
* Description of sub-systems and System interface.
* A detailed description of the problems that the software must solve.
* Details of information flow, content, and structure are documented.
* Hardware, software, and user interfaces are described for external system.

**3. Functional Description of the system**

* Functional description.
* Restrictions/limitations.
* Performance requirements.
* Design constraints.
* Diagrams to represent the overall structure of the software graphically

**4. Tests and validation criteria**

* Performance limitation, if any.
* Expected software response.
* It is essential that time and attention be given to this section.

**5. Glossary**

* Definitions of all technical or software-specific terms used in the document

**6. Bibliography**

* List and reference of all documents that relate to the software.
* Supplementary information to the specification.

**Q2. What is the need of documentation ?**

The ISO standard ISO/IEC 12207:1995 describes documentation 'as a supporting activity to record information produced by a system development life cycle process'.

**Documentation is needed because :**

**1.** It is a means for transfer of knowledge and details about description of the system.

**2.** It is to communicate among different teams of the software project;
**3.** It is to help corporate audits and other requirements of the organization.
**4.** It is to meet regulatory demand;
**5.** It is needed for IT infrastructure management and maintenance; and
**6.** It is needed for migration to a new software platform.

**Q3. List the various stages of the process of documentation.**
Various stages of the process of documentation are the following :
**(i)** Acquire source material for documentation.
**(ii)** Planning documentation.
**(iii)** Review documentation plan.
**(iv)** Develop document as per the plan.
**(v)** Test the usability of documentation.
**(vi)** Reproduce and distribute document.
**(vii)** Maintain the document.

**Q4. Describe the importance and characteristics of a good documentation in the systems development process. What elements should be included in a good documentation package?**
**# List and explain the essential elements that comprise a documentation package.**
**# Why is documentation needed? What are the elements that comprise a documentation package? Explain these elements briefly.**
**Importance or Need of a good Documentation :**
Documentation improves overall operation in addition to management and audit control. Preparation of documentation is quite important as it depicts what the system is supposed to be and how it should perform its functions. It illustrates both technically and economically how a system would better serve the objectives and goals of the company.
It also serves the following purposes:
**(i)** Reviews the progress or development of an application software.
**(ii)** Communicates facts about system to users.
**(iii)** Communicates between personnel working on a development project.
**(iv)** Provides necessary guidelines to allow correction or revision of a system or its computer programs.
**(v)** Documentation serves as a focal point from which the analysts' design can be assessed and as a standard to be utilised as a reference once the system is implemented.
**(vi)** Assists in the reconstruction of the system in case it is destroyed.
**(vii)** It helps the management to determine if the new design achieves the objectives of the company within the established constraints and if it is justifiable

from a cost standpoint.
**(viii)** Provides operating instruction to users and operating staff.

**Characteristics of a good documentation :**
**a. Availability :** It should be accessible to those for whom it is intended.
**b. Objectivity :** It must be clearly defined in a language that is easily understood.
**c. Cross-referable :** It should be possible to refer to other documents.
**d. Easy to maintain :** When the system gets modified, it should be easy to update the documentation.
**e. Completeness :** It should contain everything needed, so that those who have gone through it carefully can understand the system.

**Elements that comprise a documentation package :**
**(i)** Cover letter
**(ii)** Table of contents
**(iii)** Cost of the proposed system and of its alternatives
**(iv)** Test brochures
**(v)** File specification
**(vi)** Program specification
**(vii)** Narrative
**(viii)** Flow charts
**(i) Cover letter:** The cover letter is a correspondence primarily to management, that describes the benefits of the new design and that generally helps in selling the system. It should be kept in mind that unless the documentation is approved, the new system will never be implemented and the analysts' work will be of no use. Thus the analyst must try to convince the management that the new design presented is feasible and appropriate to satisfy the objectives of the company. The cover letter should describe the purpose and function of the new system clearly. It should be written in concise language to facilitate executive understanding, without requiring complete familiarity with the intricacies of the system.
**(ii) Table of Contents:** The inclusion of a table of contents is an absolute necessity. Pages in the documentation package must be numbered and cross referenced in this table of contents.
**(iii) Cost of the proposed system and of its alternatives:** The details of cost must be shown as part of documentation.
**(iv) Test Brochures:** The analyst should describe the operations and procedures (including test data) that will be employed to test the new system, once it is approved.
**(v) File Specification:** Each file within the formal design must he described

with regard to:
- Purpose
- Programs that will use the file
- Volume
- Frequency of use
- Source from which the file is obtained
- Description of fields
- Layout and samples

**(vi) Program Specification:** At this point, the analyst must segment the new design so that each unit will have separate program, assuming that the design is itself approved by the management.

**(vii) Narrative:** With the narrative, we begin the detailed formulation of the new system.

**(viii) Flowcharts:** Each subsystem within the analyst's formal design should be explained with the help of flowchart.

**Q5. List at least five major types of documentation. Explain any one of them in detail.**

**Following are the five major types of documentation :**

**1.** Program Documentation
**2.** Operation Documentation
**3.** User Documentation
**4.** Management Documentation
**5.** Systems Documentation

**Management Documentation :**

Documents used by corporate management to gauge performance of their organisation is called management documentation. This serves following purpose :

**1.** Evaluate progress on system development
**2.** Monitor existing systems
**3.** Understand the objective and methods of new and existing systems.

**Q6. What do you think are three common mistakes that novice analysts make in developing documentation?**

Three common mistakes that novice analysts make in documentation :

**(i)** Reviews or Modification change request are not properly and adequately documented.

**(ii)** Faulty operating instructions.

**(iii)** No proper synchronization between different stages of system development life cycle e.g. initial study to requirement specification to system design are

not properly linked this is poor documentation.
**(iv)** Nominal change in specification or design often go unnoticed and thus missing in documentation.

**Q7. List the five major types of documentation and explain them.**
There are five major types of documentation listed below :
**i.** Program Documentation
**ii.** Operation Documentation
**iii.** User Documentation
**iv.** Management Documentation
**v.** Systems Documentation

**(i) Program Documentation**
Before a program is developed, the systems analyst should provide the programmer with the required documentation. The logic in some programs is best described by a flowchart. Sometimes, decision tables are most appropriate for explaining the logic of a program. Programmers should insist on proper documentation before starting a job.
Four items constitute normal documentation required for each program.

- Copying in final form of all input/output documents affecting the program.
- Statement of standards for coding structures and input/output layouts.
- Clarification of the program's interface with other related programs.
- General flowchart or decision table.

The programmer's responsibility in documentation is to provide information to enable future programmers to make necessary changes. A company can never think that a programmer assigned to a specific program will be available in two years, when some modifications to that program are required. For continuity of information a company must insist on complete and meaningful documentation. Typically a documentation folder is provided for each program which contains all the input/output forms associated with the program, a detailed flowchart or decision table for the program use a set of operator and user instructions. Maintaining this type of documentation is costly and time consuming.

**(ii) Operations Documentation**
A well designed system may run for a long time with little or no assistance from the systems department. This can happen only when the system has been documented in a proper way. For smooth running of the system, the console operator must have complete knowledge about the job. Providing the computer centre with a set of operating instructions will not serve the pur-

pose. The instructions must be in a form readily accessible to the console operator and written in simple and understandable style. A systems analyst must thoroughly discuss all the requirements of new jobs with the operations staff before the job can be properly transferred.

The run book is collection of operator instructions for each program at an installation and typically contains:

i. Narrative, describing the run

ii. Listing of the programmed error conditions

iii. Detailed information for running the job, including:

- input/output forms to be used
- anticipated problem areas and how to handle them
- detailed description of file assignment of each input/output device
- disposition of data files after completing the job
- general block diagram of the programming logic
- restart procedures

**(iii) User Documentation**

Systems users require proper documentation to prepare a developing system and to smoothly carry out existing ones. To meet this requirement, each system should have a manual that spells everything the users must know to do their job correctly.

The manual should supply the following information.

- General flowchart of the system
- Assignment of responsibility for specific tasks
- Standards for work flow, including target dates and deadlines for specific tasks
- Simple input and output documents
- Detailed procedures
- Anticipated exceptions and instructions on how to handle them
- Accuracy standards for data in the system

Properly prepared manual which is always available can provide the information needed by the user. Supervising staff in user areas must understand the overall picture in each system just as staff members must understand the details of their function. This requires documentation, in the form of charts, graphs and illustrations, so that the supervising staff have a clear grasp of their department's role in the total system.

**(iv) Management Documentation**

The documentation required by corporate management differs quite a lot from that required by users. The systems designer must know the requirements of the management and provide documentation to enable management to per-

form three functions:
**1.** Evaluate progress on systems development
**2.** Monitor existing systems
**3.** Understand the objectives and methods of new and existing systems
Management is primarily interested to know in general the system's overall objectives and basic operations.

**(v) Systems Documentation**
Each phase in the systems development cycle is accompanied by appropriate documentation. The systems request, even if it is initially mark verbally, eventually must be written. It is desirable for the client and a systems analyst to work jointly in writing the request since each can contribute knowledge the other does not have. The written system's request is merely a statement of the user's problem.
In documenting the results of its deliberations, the selection committee must specify the following:
**a.** The objectives of the impending feasibility study
**b.** The extent of the authority of the feasibility team

**Q8. Write different ways of modelling databases.**
There are many different ways of modelling databases :
- Hierarchical model
- Network model
- Relational model
- Entity-Relationship model
- Semantic-Object model
- Object-oriented model

and many different graphical techniques for representing the models. You could rigorously apply the modelling techniques to any given database but a wide-spread practice is to combine the techniques. We tend to use a graphical modelling technique which borrows bits from various models.

**Q9. Explain in detail E-R Diagram. Also write details process of making ERD.**
We use symbols and notions from the ERM or ERD to model relations and their associations and the symbols and techniques are influenced by notions borrowed from the network model. We am going to start with the ERM (Entity-Relationship Model).

**Entity - Relationship Model**

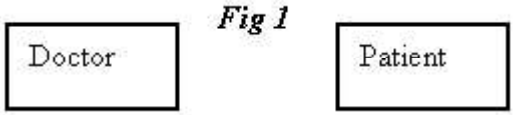



*Fig 1*

The ERM is based on entity sets and the relationships between the sets of entities. Figure 1 shows two entity sets, Doctor and patient. These could represent flat-files or tables or just the contents of a filing cabinet. Whatever the means of storage you know, from your own experience, that a doctor has many patients and that a patient has many doctors. There is some practical relationship between doctors and patients.

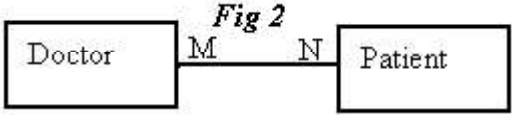



*Fig 2*

We can show the relationship between the entity sets: a doctor has many patients and a patient has many doctors. M and N are symbols which represent number.


*Fig 3*

In the ERM the relationship is shown as a diamond, and the entity sets are shown as rectangles. The entity sets are tables, the diamonds express some kind of relationship like: *has many, has one, has zero or one, may have*.

If figure 3 represents a medical practice then given a specific doctor, say Dr Verma we can safely assume that Dr Verma has many patients. Likewise a given patient, say Mrs Sharma might, over time, see many doctors in the practice.

You can see from this simple model that We are making some assumptions about the medical practice we are modelling. We are assuming that it has more than one doctor, that the doctors will see many patients and that the patients will not always be treated by the same doctor. It need not be this

way. In the case of a specialist practice it may be that all patients have only one doctor.

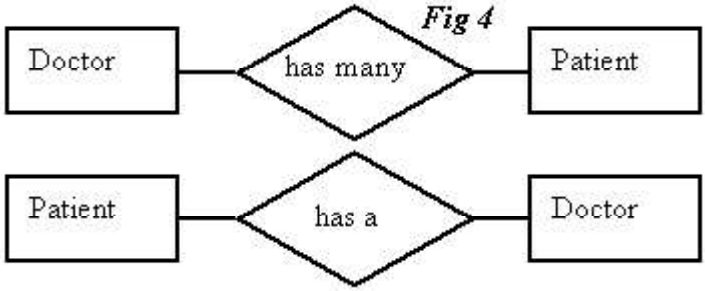
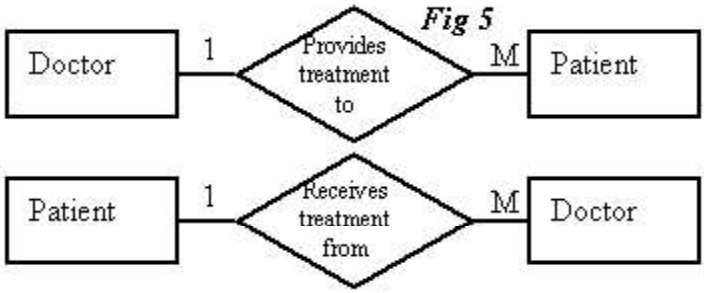



Figure 4 shows that one doctor has many patients and that one patient has one doctor. We can say that this model represents a practice with a single doctor and many patients.




In the model we are developing we assume that the medical practice is the more usual general practice. You can see that we have changed the relationship to show something more practical - doctors treat patients. The practice has a number of doctors, the doctors treat many patients and the patients see are treated by many doctors.

If we look at the doctor and patient tables without the assistance of the treatment relationship we can't tell which doctor treated which patient on any given occasion. In complex entity-relationships like the many-to-many relationship between doctors and patients we need to resolve the confusions. How do we tell which doctor treated which patient when?

**Relationships as relations**

Obviously the treatment the doctor provides and the treatment the patient receives are the same thing. Like doctors and patients the treatment is an entity itself. The difference is that the treatment entity expresses the relationship between the doctors and the patients. It is a record of a point at which a doctor and a patient interact at on specific occasion. Since it is a record we can use it resolve the confusions.

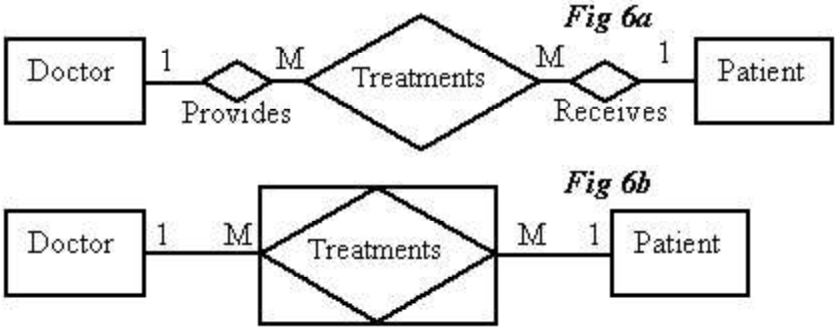
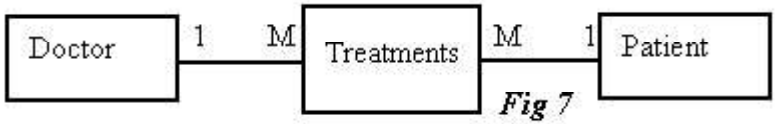



Figure 6a shows the model in all its detail. Figure 6b simplifies this by showing both the entity and the relationships in a single symbol. This is known as a composite entity or an intersection entity.

Figure 6b is a pure Entity-Relationship model or diagram (ERM or ERD). It depicts three relations: Doctors, Patients and Treatments. We use some simplifications, based on network database modelling, which are shown in Figures 7 and 8. These aren't really standard or recommended practice but do simplify the modelling process once you understand what the particular ERD represents. In brief, there is less drawing to do. For most of your work you should use the ERM.




A simplified model using numbers and symbols.

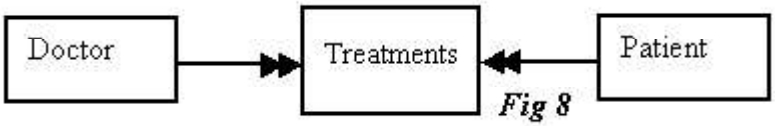



*Fig 8*

A simplified model using arrowheads.

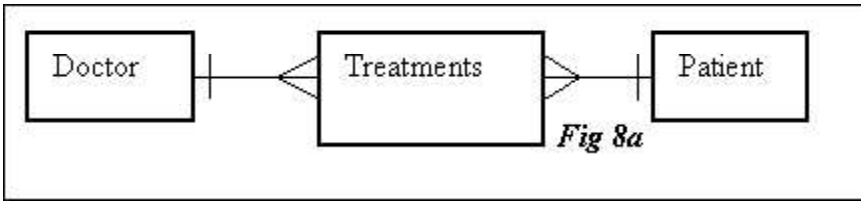



*Fig 8a*

A simplified model using crowsfeet.

**The Entity-Relationship Model and the Relational Model.**

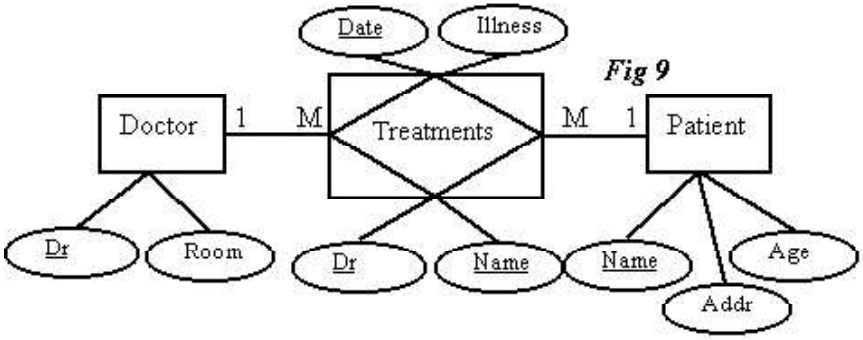
In an earlier lesson we said of the relational model that:
a relational data base is a collection of relations, there need not be any explicit links between the files and that a relation was a two-dimensional table. You were also introduced to the intension (a way of stating a relation's structure) of a relation, for example :

**Customer(Name, Address, Town, Balance)**

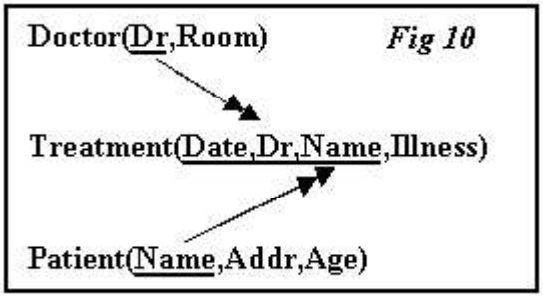
The relational model (RM) depicts the database in a verbal form (the intension), the ERM depicts the database in a graphical form (the ERD). Both models are depicting the same thing, a database or a part of a database. We should know enough now to compare, in a very limited way, the RM and the ERM.

The ERM is particularly useful for modelling the relationships between entities but it doesn't tell us much about the structure of the entities. The RM is particularly good for detailing the structure of the entities but doesn't tell us much about the relationships between the entities.

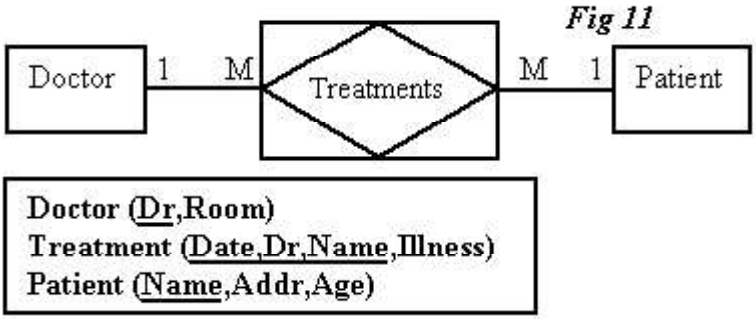
If we combined both models then we would have a modelling technique that gave us the best of both worlds. A graphical model that showed the relationships between the entities and a verbal model that showed the internal structures of the entities.




One way to do this would be to extend the ERD to include the attributes of each entity, as shown here in figure 9. A problem here is that the ERD would soon become very complex. In this example I have shown on a few of the likely attributes of each entity and already the ERD is starting to get unwieldy.

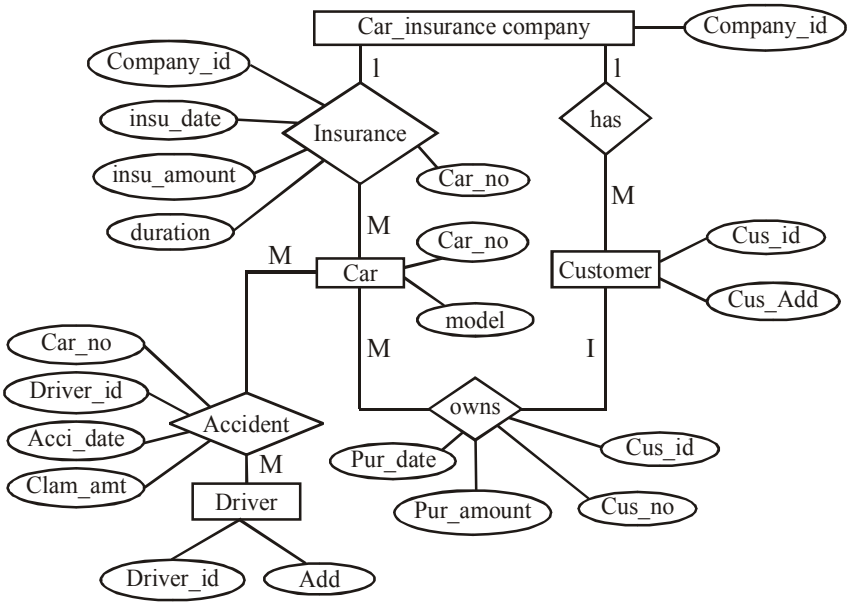



We could use the intension as the basis of a diagram and add directed lines which show the relationships. Figure 10 shows the three intensions and, a one-to-many relationship for doctor-treatment and patient-treatment relationships. This has a problem too since it loses the clarity of the ERD.



We find that the best way for me is to use both the ERD and the intension but to keep them separate as shown in figure 11.

**Q10. Construct an E-R diagram for a car insurance company that has a set of customers, each of whom owns one or more cars. Each car has associated with it zero or any number of accidents.**




**entity set CAR_INSURANCE COMPANY**

company_id : string ; (**PK**)

**entity set CAR**

car_no : string ; (**PK**)
mode : string ;

**entity set CUSTOMER**

cus_id : string ; (*unique indentification)
cus_add : string ;
company_id : string ;

**entity set DRIVER**

driver_id : string ; (*unique identification)
Add : string ;

**relationship set OWNS**

cus_id : string ;
car_no : string ;
pur_date : date ;
pur_amount : integer ;

**relationship set ACCIDENT**

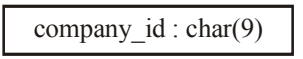
car_no : string ;
dri_id : string ;
acc_date : date ;
dam_amt ; integer ;

**relationship set INSURANCE**

car_no : string ;
company_id : string ;
insu_date : date ;
duration : date ;
insu_amount : integer ;

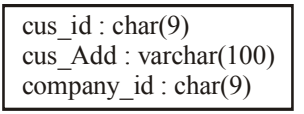
**Date structure :**

**(1) Company**



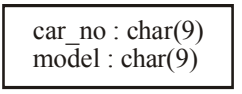

PK(Primary Key) ⇒ company_id

**(2) Customer**



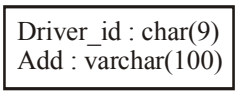

PK(Primary Key) ⇒ cus_id
FK(Foreign Key) ⇒ company_id reference (company)

**(3) Car**



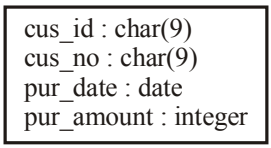

PK ⇒ car_no

**(4) Driver**



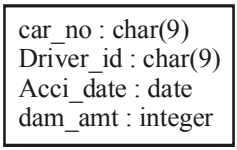

PK ⇒ Driver_id

**(5) Owns**



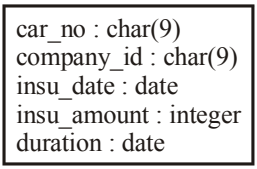

FK ⇒ company_id references
(company)
FK ⇒ car_no references
(car)

**(6) Accident**




FK ⇒ car_no references
(car)
FK ⇒ Driver_id references
(Driver)

**(7) Insurance**




FK ⇒ company_id references
(company)
FK ⇒ car_no references
(car)

**Q11. What is a data dictionary ? Specify at least 4 symbols and 4 rules that govern the construction of data dictionary entries.**

**Data Dictionary :** It is a powerful tool that is extensively used in system analysis. DDs as they are called provide a detailed reference to every data item - the different names by which the item is represented, in different program modules, different data structures used to represent the item in different modules, the modules where the data item is generated, where it is stored and destroyed. In essence it provides a quick snapshot of every data item used by the information system. Needless to say it is extremely detailed and very useful for consistency checks, system modification and completeness checking.

**Four symbols and four rules that govern the construction of data dictionary entries :**

**Four Symbols :**

| | | |
|---|---|---|
| (i) | = | Equivalent to |
| (ii) | + | And |
| (iii) | [ ] | Either/or |
| (iv) | ( ) | Optional entry |

**Four rules :**

**(i)** Words should be defined to stand for what they mean and not the variable names by which they may be described in the program; use FIRST-NAME not XYZ or NAME06. Capitalization of words helps them to stand out and may be of assistance.

**(ii)** Each word must be unique; we cannot have two definitions of the same client name.

**(iii)** Aliases, or synonyms, are allowed when two or more entries show the same meaning; a vendor number may also be called a customer number. However, aliases should be used only when absolutely necessary.

**(iv)** Self-defining words should not be decomposed.

**Q12. Write some advantages of Data Dictionaries. Explain why data dictionary is used.**

Following are the advantages of using Data Dictionaries :

**1.** Manage the detail in large systems

**2.** Communicate a common meaning for all system elements

**3.** Document the features of the system

**4.** Facilitate analysis of the details in order to evaluate characteristics and determine where system changes should be made

**5.** Locate errors and omissions in the system

**Uses of Data Dictionary :**

**Manage Detail :**

Large systems have huge volumes of data flowing through them in the form of documents, reports, and even conversations (but even small systems will seem to carry large quantities of data). Similarly, many different activities take place that use existing data or create new details.

**Communicate Meaning :**

Data dictionaries assist in ensuring common meanings for system elements and activities. Data dictionaries record additional details about the data flow in a system so that all persons involved can quickly look up the description of data flows, data stores, or processes.

**Document System Features :**

Documenting the features of an information system is the third reason for using data dictionary systems. Features include the parts or components and the characteristics that distinguish each. Having to formally describe system features will produce a more complete understanding. And once the features have been articulated and recorded, all participants in the project will have common source for information about the system.

**Facilitate Analysis :**

The fourth advantage of data dictionaries is to determine whether new features are needed in a system or whether changes of any type are in order. What questions would you ask and what information would you want to have available for examination?

- **Nature of Transactions**

The business activities that will be carried on while using the system, including the data needed to accept, authenticate, and process each activity.

- **Inquiries**

Requests for the retrieval of information or processing to generate a specific response.

- **Output and Report Generation**

Results of system processing presented to users in an acceptable form.

- **Files and Databases**

Details of transactions and master records of concern to the organization

- **System Capacity**

The ability of the system to accept, process, and store transaction and data.

**Locate Errors and Omissions :**

Having information in a dictionary about these system characteristics-transactions, inquires, data and capacity-will tell you a great deal about a system and allow you to evaluate it. But you need to know that the information itself is complete and accurate.

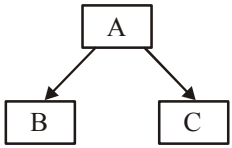
**Q13. Define Structure Chart.**

It is a graphic representation of the control logic of processing functions or modules representing a system.

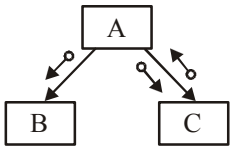
**Elements of structure chart are :**

**(a) Module →** The module is represented by rectangle with a name


**(b) Connection →** Represented by arrow which links the two module.




**(c) Couple →** couple represented by arrow with circular tail. It represent data items moved from one module to another.




**Q14. What is user manual ? List the categories of usual manual.**

User Manual is a document, which is completed at the end of the software development process.

**Types of User Documentation :** Users of the system are not of the same category and their requirements vary widely. In order to cater to the need of different class of user, different types of user documentation are required.

Following are the various categories of manuals:

**1.** Introductory manual : How to get started with the system?
**2.** Functional description : Describes functionality of the system.
**3.** Reference manual : Details about the system facility.
**4.** System administrator guide : How to operate and maintain the system?
**5.** Installation document : How to install the system?

**Q15. Write the components of software user documentation.**

**Following are the components of software user documentation :**

**1.** Identification data (e.g., Title Page or First Page)
**2.** Table of contents
**3.** Table of illustration used in documentation
**4.** Introduction
**5.** Information for use of the documentation such as description of software etc.
**6.** Concept of operations
**7.** Procedures
**8.** Information on software commands
**9.** Error messages and problem resolution
**10.** Glossary (to make the reader acquainted with unfamiliar terms)
**11.** Related information sources
**12.** Navigational features
**13.** Index
**14.** Search capability (for electronic type documentation)

# Chapter-5

# Process Of System Planning

The most critical phase of managing system projects is "PLANNING". To launch a system investigation we need a master plan detailing the steps to be taken, the people to be questioned, and the outcome expected. The initial investigation has the objective of determining whether the user's request has potential merit. When initial investigation is completed, the user receives a system proposal summarizing the recommendation of analyst. Today's systems analysts to be successful in defining system requirement they must be skilled in effective methods for gathering information-fact finding. Fact finding is used across the entire development cycle, but it is extremely critical in the requirement analysis phase.

**Q1. What is fact finding technique?**

**Fact Finding Techniques**

Fact finding is the formal process of using research, interview, meeting, questionnaires, sampling and other techniques to collect information about systems, requirements and preferences. It is also called information gathering or data collection. Fact finding techniques are an essential skill for all systems analysts. It is a classical set of technique used to collect information about system problems, opportunities, solution, requirement and priorities. It is one of the best requirement discovery techniques. Tools, such as data, process and object modules, will eventually be used to document fact, and conclusion will be drawn from facts. But if you can't collect the facts, you can't use these tools. Fact-finding skills must be learned and practised!

**Q2. Write advantages and disadvantages of various fact finding techniques.**

**There are many fact finding techniques below are the advantages and disadvantages of various fact finding techniques available :**

**Interviews :** Personal interview is a **well known** recognized and most important fact finding technique, where the systems analyst gathers information from an individual through face to face interaction. Interviews are taken to find the facts, verify facts, clarify facts, get the customer involved, identify the system requirements and know all options. The systems analyst generally conducts the interview. To conduct interview, the interviewer must have

personality which helps him/her to be social with strangers or different types of people. Always and for all situations, interviews are not appropriate fact finding methods. It has both impacts of advantages and disadvantages, which are mentioned below :

**Advantages :**

**1.** Interviews permit the systems analyst to obtain a better clarity of the problem due to feedback from the interviewees.

**2.** Interviews allow the systems analyst to get individual's views and get the specific problem work wise and operation wise.

**3.** Interviews apprise the systems analyst to understand the user requirements and to know the problems faced by the user with the current system.

**4.** In the process of interviews, the interviewer has time and scope to motivate the interviewee to respond freely and openly.

**5.** It is an effective technique to gather information about complex existing systems.

**Disadvantages :**

**1.** Success of interviews, in most of the cases, depends on the systems analyst's interpersonal relationship skills.

**2.** Interviews are very time consuming.

**3.** Sometimes, interviews may be impractical due to the location or venue of interviewees.

**Site Visits :** The engineers of the development organization are visiting the sites. Generally, the systems analysts visit sites to get first hand information of the working of the system. In this technique, systems analyst watches the activities of different staff members to learn about the system. When there is confusion about the validity of the data collected from other sources, the systems analyst uses the method of site visits. The main objective of the site visit is to examine the existing system closely and record the activities of the system. Following are the advantages and disadvantages :

**Advantages :**

**1.** Site visit is inexpensive when compared to other fact finding techniques.

**2.** The process of recording facts, site visits is highly reliable.

**3.** Sometimes, site visits take place to clear doubts and check the validity of the data.

**4.** The system analyst can easily understand the complex processes in the organization.

**5.** In this technique, systems analyst will be able to see the processes in the organization at the first hand.

**Disadvantages :**

**1.** There may be scheduling problems for the system analysts when the activities take place during odd hours.

**2.** Site visits are done during a specific period and during that period, complexities existing in the system may not be experienced.

**3.** Due to interruptions in the task being observed, the information that is collected may be inaccurate.

**4.** People generally feel uncomfortable when being watched; they may be unwillingly perform their work differently when being observed.

**5.** Sometimes, people may be more careful to adopt the exact procedure which they do not typically follow.

**Questionnaires :** Questionnaires are special purpose documents that allow the analyst to collect information and opinion from respondents. By using questionnaires, it is possible to collect responses or opinion from a large number of people. This is the only way to get response from a large audience. Following are the advantages and disadvantages :

**Advantages :**

**1. Customers** can complete it at their convenience.

**2.** It is an inexpensive means of collecting the data from a large group of individuals.

**3.** Responses can be tabulated and analyzed quickly.

**4.** Proper formulation and interaction with respondents leads to unbiased response from the customers.

**5.** It requires less skill and experience to administer questionnaire.

**Disadvantages :**

**1.** There is no guarantee that the respondents will answer all the questions.

**2.** Sometimes, the individual may misunderstand the question. In that situation the analyst may not get correct answer.

**3.** Sometimes, the number of respondents is low.

**Q3. List three problems that an analyst faces when observing an existing system for fact finding.**

The three problems that the analyst faces while observing an existing system are as follows:

**(i)** Some competitive organization may not want to share their experiences.

**(ii)** Organizations which are to be visited for observing the system may be too large or too small for accurate comparisons and some organizations may be unwilling to waste employee time demonstrating their system.

**(iii)** Although hardware and software vendors supply very useful information

for observing an existing system but they should be reviewed carefully as vendors are only interested in promoting their product.

**Q4. Suppose that you were asked to develop an Information System relating to programmes offered by a university. How will you conduct structured interviews for finding facts which will be used to develop this system? List the questions that will be asked by you to the concerned staff of the university.**

**UNIVERSITY PROGRAMMES INFORMATION SYSTEM**

System Analyst will be conducting the structured interview.
Interviewee : Sanjay
Interviewer : Sapna
Duties Controller
Date Month dd, 20xx
As attendant at the "May I Help You" window, he sees that several obvious problems exist:

**QUESTIONNAIRE**

A questionnaire for staff of university will be sent. System Analyst wanted to determine the nature of complaints and gather additional study facts to design the system

**(UNIVERSITY)**

**(INFORMATION SYSTEM RELATING TO PROGRAMMES QUESTIONNAIRE)**

*Dear Staff* :- As you know, we are enjoying the high repute of university among the students and able to capture a large amount of International Students. To serve the increasing quantity of students in a better way we need your help by answering the following questions:-

1. You learn of coming programs in time to plan to attend
   Strongly Disagree 1 2 3 4 5 6 7 8 9 10 Strongly Agree
2. You always get updated course information
   Strongly Disagree 1 2 3 4 5 6 7 8 9 10 Strongly Agree
3. You can search for a particular course related information easily.
   Strongly Disagree 1 2 3 4 5 6 7 8 9 10 Strongly Agree
4. You are able to get desired information efficiently & without hassle
   Strongly Disagree 1 2 3 4 5 6 7 8 9 10 Strongly Agree

Comments:
The questionnaire will be asked to fill by atleast 200 students from different

regions to understand them better. Later on the basis of which the system under study will be planned.

**Q5. Explain all the inter-related types of feasibility studies.**

**Feasibility Study :** It is an investigation to determine whether a proposed project, system, etc would be desirable, cost effective, etc to develop or put into operation. Or The analysis of a problem to determine if it can be solved effectively. The operational (will it work?), economical (costs and benefits) and technical (can it be built?) aspects are part of the study. Results of the study determine whether the solution should be implemented.

**Following are the seven inter-related types of feasibility explained :**

**(i) Technical Feasibility :** Technical feasibility is concerned with the availability of hardware and software required for the development of the system, to see compatibility and maturity of the technology proposed to be used and to see the availability of the required technical manpower to develop the system. These three issues are addressed during this study.

Is the proposed technology proven and practical? At this stage, the analyst has to see or identify the proposed technology, its maturity, its ability or scope of solving the problem. If the technology is mature, it it has large customer base, it will be preferable to use as large customer base already exists and problems that stem from its usage may be less when compared to other technologies which don't have a significant customer base. Some companies want to use the state of art new technology irrespective of the size of customer base.

The next question is: does the firm possess the necessary technology it needs. Here, we have to ensure that the required technology is practical, and available. Now does it have the required hardware, and software? For example, we need ERP software, and hardware which can support ERP. Now, if our answer is no for either of the questions, then the possibility of acquiring the technology should be explored.

The last issue related to technical feasibility is the availability of technical expertise. It may be difficult to find skilled manpower.

**(ii) Operational Feasibility :** Operational feasibility is all about problems that may arise during operations. There are two aspects related with this issue:

* What is the probability that the solution developed may not be put to use or may not work?
* What is the inclination of the management and end users towards the solution?

Though, there is very least possibility of management being averse to the

solution, there is a significant probability that the end users may not be interested in using the solution due to lack of training, insight etc. Other issues related with operational feasibility are timely and accurate information, proper services, efficiency.

**(iii) Economic Feasibility :** It is the measure of cost effectiveness of the project. The economic feasibility is nothing but judging whether the possible benefit of solving the problems is worthwile or not. At the feasibility study level, it is impossible to estimate the cost because customer's requirements and alternative solutions have not been identified at this stage. However, when the specific requirements and solutions have been identified, the analyst weighs the cost and benefits of all solutions, this is called "cost benefit analysis". A project which is expensive when compared to the savings that can be made from its usage, then this project may be treated as economically infeasible. Knowing this is the objective of this feasibility study.

**(iv) Legal Feasibility :** Legal feasibility studies issues arising out of the need to the development of the system. The possible consideration might include copyright law, labour law, antitrust legislation, foreign trade, regulation, etc. Contractual obligation may include the number of users who will be able to use the software. Legal feasibility plays a major role in formulating contracts between vendors and users. If the ownership of the code is not given to the user, it will be difficult to install it without proper permission to other systems. Another important legal aspect is that whenever an IT company and the user company do not belong to the same country then the tax laws, foreign currency transfer regulations, etc., have to be taken care of.

**(v) Social feasibility :**Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project being accepted by the group directly affected by the proposed system change.

**(vi) Management feasibility**

It is a determination of whether a proposed project will be acceptable to management. If management does not accept a project or gives a negligible support to it, the analyst will tend to view the project as a non-feasible one.

**(vii) Time feasibility**

Time feasibility is a determination of whether a proposed project can be implemented fully within a stipulated time frame. If a project takes too much time it is likely to be rejected.

**Q6. Explain the two different types of finding the requirements of a system.**

**[JUNE05, Q1(b)]**

**Following are the two different types of finding the requirements of a system :**

**(i) Joint Application Development :** It is a process in which group meetings are held to analyze the problem and define the requirements of the desired system. In the process of Joint Application Development (JAD), each participant is expected to attend and actively participate. The group includes: sponsor, the facilitator, the user manager and IT staff. When JAD technique is used to find the requirements, it is known as Joint Requirements Planning. Participants of Joint Application Development are, **Sponsor** is a person in top management. The sponsor plays a vital role in the process of JAD. S/he works closely with JAD leader to plan the session by identifying individuals from the user community.

**Facilitator** is a single individual who plays an important role as leader. S/he leads all the session that held for system development. This implies S/he must have good communication skill, negotiation sell, ability to remove group conflicts, possess good knowledge of business, has strong organizing power, quick and impartial decision making capability. The facilitator plans session for JAD, conducts the session and follows the decision of the session. **Representatives of the Clients** will also attend the JAD session. They are chosen by the project sponsor based on their knowledge of the business system. The role of the representatives of the clients is to communicate the business rules and the requirements of the desired system. **Scribe** records proceedings of the meeting. **IT staff** such as programmer also participate in the session. IT staff listen and take notes regarding issues and requirements mentioned by the clients and analysts.

JAD session spans from 3-7 days, but in special cases, it may continue up to two weeks. Success of JAD session depends upon proper planning. For a successful JAD session, all the participants should be informed about the schedule of the session before hand and they should come prepared. The analyst must work closely with the sponsor to determine the scope of the issues that will be discussed in JAD session.

There are three steps that are to be followed for a successful JAD session:

- Selection of a location for JAD session.
- Selection of JAD participants.
- Preparation of agenda items for JAD session.

JAD sessions are usually held in a location different from the work places. Preparation of the agenda is the key for the success of JAD session.

JAD has some Disadvantages also which are as follows :
· Since it is a meeting of many people, there may not be sufficient time for everyone to speak.
· Only a few people may dominate the discussion. So, the outcome of the meeting will be the view of those who spoke most during the meeting.
· The problem with such meetings is that some people are afraid to speak out for fear that they may be criticised.

**(ii) Prototyping :** In software development, a prototype is a rudimentary working model of a product or information system, usually built for demonstration purposes or as part of the development process. In the systems development life cycle (SDLC) Prototyping Model, a basic version of the system is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed.
- In prototype-based programming, a prototype is an original object; new objects are created by copying the prototype.
- In hardware design, a prototype is a "hand-built" model that represents a manufactured (easily replicable) product sufficiently for designers to visualize and test the design.
It is the process in which we produce an original type, form, or instance serving as a basis or standard for later stages; or an original, full-scale, and usually working model of a new product or new version of an existing product; or an early, typical example.
This model is useful for determining requirements for the software to be built in the following situations:
**1.** Requirements are not clear.
**2.** For any complex systems, prototypes are more useful.
**3.** In the cases where communication problems exist between customer and analyst, this model is useful.
**4.** Tools and data are readily available for building the working system.

There are some disadvantages of the prototype model :
**1.** In case of prototyping, formal documentation is
**2.** Usually, prototypes are stand alone systems. Building prototypes is difficult in cases where data has to be shared.
**3.** Important issues, such as security and validation, are not given importance

**Q7. Define the term Feasibility Study. Describe at least four types of feasibility analyses. [DEC05, Q3(a)]**
**Feasibility Study :** It is an investigation to determine whether a proposed

project, system, etc would be desirable, cost effective, etc to develop or put into operation. Or The analysis of a problem to determine if it can be solved effectively. The operational (will it work?), economical (costs and benefits) and technical (can it be built?) aspects are part of the study. Results of the study determine whether the solution should be implemented.

**Following are the four types of feasibility explained :**

**(i) Technical Feasibility :** Technical feasibility is concerned with the availability of hardware and software required for the development of the system, to see compatibility and maturity of the technology proposed to be used and to see the availability of the required technical manpower to develop the system. These three issues are addressed during this study.

Is the proposed technology proven and practical? At this stage, the analyst has to see or identify the proposed technology, its maturity, its ability or scope of solving the problem. If the technology is mature, it it has large customer base, it will be preferable to use as large customer base already exists and problems that stem from its usage may be less when compared to other technologies which don't have a significant customer base. Some companies want to use the state of art new technology irrespective of the size of customer base.

The next question is: does the firm possess the necessary technology it needs. Here, we have to ensure that the required technology is practical, and available. Now does it have the required hardware, and software? For example, we need ERP software, and hardware which can support ERP. Now, if our answer is no for either of the questions, then the possibility of acquiring the technology should be explored.

The last issue related to technical feasibility is the availability of technical expertise. It may be difficult to find skilled manpower.

**(ii) Operational Feasibility :** Operational feasibility is all about problems that may arise during operations. There are two aspects related with this issue:

* What is the probability that the solution developed may not be put to use or may not work?
* What is the inclination of the management and end users towards the solution?

Though, there is very least possibility of management being averse to the solution, there is a significant probability that the end users may not be interested in using the solution due to lack of training, insight etc. Other issues related with operational feasibility are timely and accurate information, proper services, efficiency..

**(iii) Economic Feasibility :** It is the measure of cost effectiveness of the project. The economic feasibility is nothing but judging whether the possible

benefit of solving the problems is worthwile or not. At the feasibility study level, it is impossible to estimate the cost because customer's requirements and alternative solutions have not been identified at this stage. However, when the specific requirements and solutions have been identified, the analyst weighs the cost and benefits of all solutions, this is called "cost benefit analysis". A project which is expensive when compared to the savings that can be made from its usage, then this project may be treated as economically infeasible. Knowing this is the objective of this feasibility study.

**(iv) Legal Feasibility :** Legal feasibility studies issues arising out of the need to the development of the system. The possible consideration might include copyright law, labour law, antitrust legislation, foreign trade, regulation, etc. Contractual obligation may include the number of users who will be able to use the software. Legal feasibility plays a major role in formulating contracts between vendors and users. If the ownership of the code is not given to the user, it will be difficult to install it without proper permission to other systems. Another important legal aspect is that whenever an IT company and the user company do not belong to the same country then the tax laws, foreign currency transfer regulations, etc., have to be taken care of.

**Q8. How can one define cost-benefit analysis?**

Since cost plays quite an important role in deciding the new system, it must be identified and estimated properly. Costs vary by type and consist of various distinct elements. Benefits are also of different type and can be grouped on the basis of advantages they provide to the management.

**Cost-benefit analysis can be defined as :**

**(i)** A method by which we find and estimate the value of the gross benefits of a new system specification.

**(ii)** A method by which we find and determine the increased operating costs associated with the above mentioned gross benefits.

**(iii)** The subtraction of these operating costs from the associated gross benefits to arrive at net benefits.

**(iv)** A method by which we find and estimate the monetary value of the development costs that produce the above mentioned benefits.

**(v)** Those methods by which we show the time-phased relationship between net benefits and development costs as they relate to cash flow, payback on investment, and time-in-process taking(or not taking) into operation factors such as inflation etc. In short, the calculation of actual net benefit as cash flowback over time.

**Q9. Why is cost benefit analysis important in detailed analysis phase**

Cost Benefit analysis is important in detailed analysis phase to guide/help in the selection process i.e. the analyst selects those alternatives which are more workable and studies them further and make decisions which alternative should be selected. Therefore, analyst needs to be familiar with the cost and benefit categories and evaluation of various methods before a final selection can be made.

Since cost plays quite an important role in deciding the new system, it must be identified and estimated properly. Costs vary by type and consist of various distinct elements. Benefits are also of different type and can be grouped on the basis of advantages they provide to the management.

**Q10. Distinguish between :**

**(i) Opportunity and Sunk cost**

**(ii) Direct and Indirect cost**

**(iii) Fixed and Tangible cost**

**(i) Opportunity and Sunk cost**

**Opportunity :** These are the costs omitted selecting one alternative over another. They do not show in the organizations accounts and therefore are not easy to identify. These opportunity costs are not early understood.

**For example** Bad debt whose exact amount is not so immediate.

**Sunk Cost :** Some cost are constant, regardless of how well a system is used. There are sunk costs. Once encountered they will not reoccur.

**For example** Straight line deprecation of hardware employee salaries and insurance.

**(ii) Direct and Indirect cost**

**Direct :** Direct costs are those with which a dollar figure can be directly associated in project. They are applied directly to the operation for example The purchase of a box of diskettes for $5 is a direct cost because we can associate the diskettes with the dollar expended.

**Indirect Cost :** Indirect costs are the results of operations that are not associated with a given system or activity. They are often referred to as overhead. For example Insurance maintenance, protection of the computer center.

**(iii) Fixed and Tangible cost**

**Tangible Cost :** Tangible cost is an outlay of cash for a specification or activity. They are usually shown as disbursements on the books. The purchase of hardware or software personnel, training and employee salaries are example of tangible costs. They are readily identified and measured.

**Fixed cost :** They are constant and do not change i.e. and non-recurring e.g. Straight line depreciation of hardware, exempt employee salaries and insurance.

**Q11. What finding techniques would you use in Deciding on the design of a new application form for IGNOU Courses?**

The fact finding techniques to decide a new application form for IGNOU are as follows :

**1) Review of written documentation :** All documentation on data carriers (forms, records reports, manuals etc.) is organized and evaluated. For this, analyst needs to find out that how the listing IGNOU forms are filled out & how useful they are to the user, what changes need to be made and how easy they are to read.

**2) On-Site Observations :** Here analyst has to observe the physical layout of IGNOU forms & the work flow which decides that (a) what behaviour can be observed from available forms that can or cannot be described in other ways? (b) what date can be obtained more easily?

**3) Conducting personal interview and the questionnaire :** This is to make the study of the reliability and validity of the present forms available: After these steps the IGNOU forms may be designed maintaining the following rules:

**a)** The material should be arranged in a logical sequence so that it is easy to fill and to understand. It should also be seen that document filling is sufficiently clear so that doubles and queries are avoided.

**b)** The most important thing while designing new IGNOU form is that the questions to be asked must be put using simple language arranged in the expected sequence for easy comprehension.

**c)** The IGNOU form must carry necessary instructions properly placed, for its correct and effective use. Notes and instructions should be put where they are read before entries are made. Preliminary general notes at the head of the form and specific ones near the point at which the relevant entries are to be made should be there. Overleaf notes should be avoided since it make reference to entry difficult.

**d)** The IGNOU form must be designed to facilitates the correct, efficient and economic use of data.

**e)** The form should be designed with full consideration of the methods of storing or housing during its inactive life. The paper should be of good GSM.

**f)** For designing the IGNOU forms fulfilling the above requirement, proper persons must be consulted for options, suggestions and an authorization.

**Q12. Fact-finding techniques To decide on design of a ballot paper that can automatically be scanned for counting vote.**

**1) Interview with the voter**
**2) Interview with the counter**
**3) Record review**
**4) Observation**

**1) Interview with the voter →**
i) To determine whether he is aware of the correct place to put the seal, on the paper.
ii) To determine whether he is aware of the fact that only one of the candidates can be voted
iii) To determine if the size or space of the ballot paper be increased or decreased to meet his convenience

**2) Interview with the counter →**
i) To know on what basis is a vote counted as valid or invalid.
ii) What kinds of errors are encountered in the votes
iii) What is the amount of acceptance for a machine-counting.

**3) Report review →**
i) To get the facts & figures of number of votes polled, valid votes, disqualified votes, speed of manual counting etc.
ii) To get information about the system rules and constrains as per which the votes could be considered as valid.
iii) To get information about the prescribed standards to be met by the counting system.

**4) Observation:**
i) To obtain a direct and first-hand information of the voting styles, patterns and manners of voters.
ii) Place where they are less informed or commit errors
iii) To obtain first-hand information about how the activities in the entire system are carried out.
iv) To gain information about handling of votes, inconvenience or problems caused due to size of ballot paper or printing styles.
v) Cases where information is deemed ambiguous on the ballot paper.
vi) To review the general short coming in the entire process.
(More prints can be added based upon a person's analysis of the system).

# Chapter-6

# Modular And Structured Design

**Q1. Define System Analysis and Design. Also write difference between both.**

**# Explain briefly System Analysis and System Design ?**

**Systems analysis** is the study of a business problem domain to recommend improvements and specify the business requirements for the solution, independent of any technology that can or will be used to implement a solution to that problem.

**Systems design** is the specification or construction of a technical, computer based solution for the business requirements identified in a systems analysis.

Remember : Increasingly, the design takes the form of working prototype.

Some of you will routinely work with systems analysts. The rest of you will be customers of systems analysts who will try to help you solve your business and industrial problems by creating and improving your access to the data and information needed to do your job.

**Q2. List out various parts of system design process. Explain them briefly.**

The computer system design process is an exercise of specifying "how" the system will work. It is an iterative process which is based on "What" the system will do as shown in the feasibility report.

Mainly, there are five parts of system design processes. These are :

**(i) Output design** :- The starting point of the design process is the proper knowledge of system requirements which will normally be converted in term of output.

**(ii) Input design :-** Once the output requirements have been finalised the next step is to find out what data need to be made available to the system to produce the desired outputs. The basic documents in which these data are available need to be identified. If necessary, these documents may have to be revised or near documents may have to be introduced.

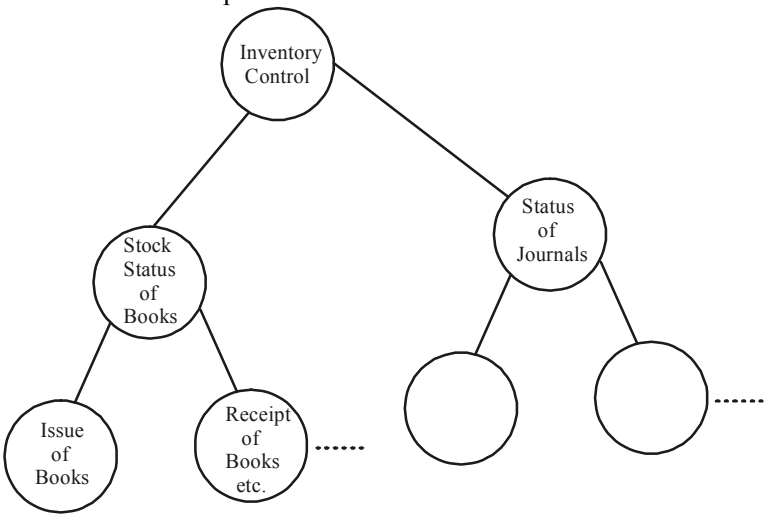
**(iii) File Designing :-** Once the input data is captured in the system, these may have to be preserved either for a short or long period. These data will

generally be stored in files in a logical manner. The designer will have to devise the techniques of storing and retrieving data from these files.

**(iv) Procedure design :-** This step involve specifications of how processing will be performed. In this there are two aspects.
(i) Computer procedure AND (ii) Non-computer procedure
The computer procedure will specify what functions will be carried out on computer. What will be different programs and in what sequence the program will be run. The non-computer procedure will specify the manual procedures for feeding input data, receiving outputs etc.

**(v) Control design :-** This indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc. This will ensure that system is functioning as per plan.
Generally, all of these (five) steps are interdependent and some of them may have to be used together and traversed many times until a satisfactory design is prepared. The system design process is, therefore, an iterative process where decision made or changed at one step will have a "triple effect" on other steps.

**Q3. Define Modularisation.**
In structured design a program is segmented into small, independent modules. These are arranged in a hierarchy that approximates a model of the business area and is organised in a top-down manner with details shown at the bottom. Thus, in structured design, we try to minimise the complexity of the problem and make it manageable by sub-dividing it into smaller segments which is called modularisation or decomposition. Shown below:



## Decomposition - A Framework

**Q4. Define Modularity. Write at least five goals of a good design. Also, write at least four guidelines for achieving the goals mentioned.** **[JUNE02, Q2(a)]**

**# Explain the six goals of design. Based on the goals, give a set of guidelines for arriving at good design.** **[DEC05, Q3(b)]**

**Modularity** is the property of computer programs that measures the extent to which they have been composed out of separate parts called modules.

Programs that have many direct interrelationships between any two random parts of the program code are less modular than programs where those relationships occur mainly at well-defined interfaces between modules. Modular programming techniques are those which increase modularity.

**Following are the five goals of design :**

A system which can be read easily, code easily, then it possesses fine design. Any design which achieves the goals given below can be termed as good design:

**1.** The design of the system should be module based. It means there are modules which together make up the system and the organization of these modules is hierarchical.

**2.** Each module controls the functions of a suitable number of subordinate modules at the next hierarchical level.

**3.** One of the important features of good design is that the modules, which make up the system don't communicate intensively. The communication should be kept at minimum level. The reason for this imposition is that modules should be independent of each other to the maximum extent possible which means, "one module's functionality should not be dependent on the internal functions of other module".

**4.** The size of module should be appropriate as required for the features it should possess like being relatively independent of other modules etc.

**5.** The coding of modules should be generic. It enables the system to use the module as frequently as possible.

**Following are the four guidelines for achieving the goals above mentioned :**

**1.** A superordinate module should control not more than seven subordinate modules. Of course, this guideline is not strict and varies from system to system.

**2.** Usually, a module if of not more than hundred lines. It may be a minimum of fifty lines. But, these sizes are not to be strictly followed and they may vary from system to system. It is possible here that lesser the lines of code it will be easier to read.

**3.** A module should not perform more than one function. There should be no line in the code of the module, which is concerned with a function that is not the objective of that particular module. One easy check for this conformance is that the module's function should be describable easily in few words. This is called cohesion.

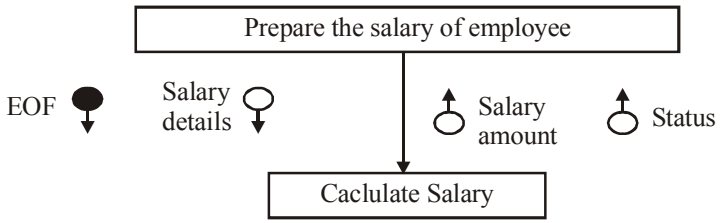
**4.** Modules at the lower level of the design are called by more than one superordinate module.

**Q5. Explain the five types of coupling with an example for each.**
**[JUNE05, Q4(a)]**

**Coupling :** The degree to which components depend on one another. There are two types of coupling, "tight" and "loose". Loose coupling is desirable for good software engineering but tight coupling may be necessary for maximum performance. Coupling is increased when the data exchanged between components becomes larger or more complex.

**Following are the five types of coupling with example :**

**(i) Data Coupling :** In this type of coupling, the communication between the modules is through passing of data as parameters. The other alternative in this type of coupling is the use of flags. So, one module will not be and need not be aware of the internal structure of the module with which it is communication.




Consider the Figure above. Prepare the salary of employee is the superordinate module. Calculate Salary is the subordinate module. It is coupled with Calculate Salary. But, Prepare the salary of employee need not be aware of the internal structure of Calculate Salary. Calculate Salary needs to know the data being passed to it for doing the requisite task and the data that has to be returned by it to the superordinate module.

**(ii) Stamp Coupling :** The mechanism of communication in Stamp Coupling is achieved by passing data structures. Alternatively, records consisting of requisite data are sent. The problem with this type of coupling is that any changes in the data structure will lead to a chain reaction and all the modules that use this

data structure have to be change. Sending data as stated in the technique of data coupling is ideal. Stamp Coupling increases the dependency levels among the modules. All the modules, which are using the same data structure, should be aware of the same data structure.

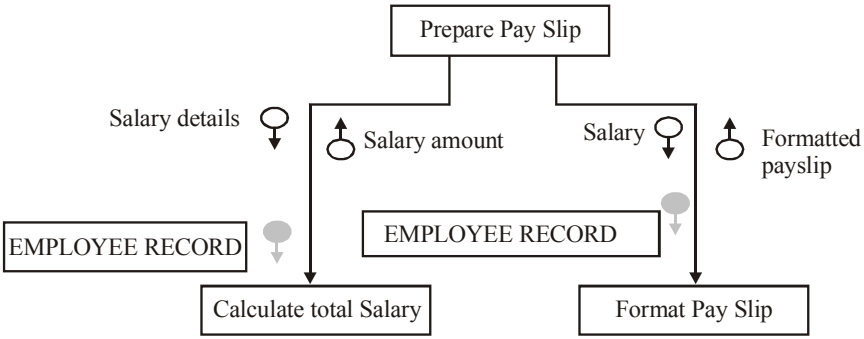



Figure above demonstrates Stamp coupling. Obviously, the entire Employee record will contain more data than data required by the Calculate Total Salary module. The process Format Payslip then uses data structure Employee record. Once again Employee record contains too much data for this process. It would be better for both superordinate, subordinate modules and for the system as a whole if only the relevant data elements were passed instead of the entire record.

**(iii) Control Coupling :** Here the superordinate module communicates with subordinate module by passing control information. The control information conveys the functions to the subordinate module that are to be performed by it. In this type of coupling, interdependence between the superordinate and subordinate modules is high as the superordinate module should definitely know the internal functions of the subordinate module to invoke it for a particular task.

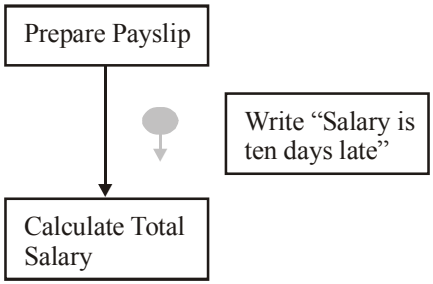



Figure above depicts an example of Control coupling. The signal that control

information is being passed is that the label of the flag starts with the verb Prepare Payslip. It is to be noted that, in some cases, control information may be passed from the subordinate module to superordinate module. But, this rarely occurs.

**(iv) Common Coupling :** Here the global data areas are used by the multiple modules. All modules which use this data area will be accessing the data in that area that is present there at that point of time. If any module performs invalid operation on the data in the global data area then the data area holds the resultant wrong value. But, this wrong value will be used by the module which subsequently accesses this global data area resulting in further invalid processing. In this type of coupling, interdependence among the modules is very high as they are sharing the data area and any wrong doing by any of the modules on this global area is going to impact the processing of all the subsequent modules which use the data in this global data area.

**(v) Content Coupling :** This coupling is used when none of the above are possible. The major drawback of this technique is that one module can access the data inside another module and alter it. Also, it is possible to change the code of one module by another module. This is the technique of coupling in which independence among the modules is not even slightly visible. Fortunately, most of the High level languages don't support content coupling.

All the coupling techniques that are given above rate from top to bottom in terms of priority. Data coupling should be most sought after technique of coupling followed by stamp coupling and then control coupling followed by common coupling and lastly content coupling.

**Q6. How do coupling and cohesion measure the strength of Module Design?**

**#: Define Coupling and Cohesion.**

A fundamental principal of structured design is that program should be decomposed into smaller manageable units called modules. Each module should be independent to each other.

**Coupling** refers the level of dependency that exists between modules. During the process Data flow diagram this coupling checks can be performed. Two main types of coupling exist.

**1. Loose coupling**

**2. Tight coupling**

We may consider different types of coupling.
a. Data Coupling
b. Stamp Coupling
c. Control Coupling
d. Common Coupling
e. Content Coupling

**Cohesion** refers to the degree to which a module's instructions are functionally related. Thus, highly cohesive module contain instruction that collectively work together to solve the specific task. Program that are made up of highly cohesive modules are easy to understand, modify and maintain. There are seven types of cohesion.
a. Functional Cohesion
b. Sequential Cohesion
c. Communication Cohesion
d. Procedural Cohesion
e. Temporal Cohesion
f. Logical Cohesion
g. Coincidental Cohesion
The cohesive nature of a program can be ascertained from the Data Flow Diagram.

**Strength of Module Design :** Modules should have low coupling. Low coupling minimizes the "ripple effect" where changes in one module causes errors in other modules. While a module should be highly cohesive. As Modules whose elements are strongly and genuinely related to each other are desired.

**Cohesion**
Cohesion is a measure of the strength of association of elements within a module it is recommended that element should be highly cohesive or strongly interrelated. A cohesive module performs a single task within a software procedure, requiring littler interaction with procedures being performed in other parts of a program. Simply, a cohesive module should (ideally) do just one thing. A simple way to examine the cohesiveness of a module is to check each instruction in it. Every instruction is related to the performance of a single task. The performance of single task leads to a larger degree of portability and we can directly plug the module in an application which requires the performance of the task. Since the module performing single task, it will be loosely coupled and accept data from a superordinate module executes the required function and return the result. There are seven major levels of cohesion:
(i) Function cohesion

(ii) Sequential cohesion
(iii) Communicational cohesion
(iv) Procedural cohesion
(v) Temporal cohesion
(vi) Logical cohesion
(vii) Coincidental cohesion

**(i) Functional Cohesion:** Functional cohesion means that the module performs a single identifiable function and in the best type of cohesion. A module is said to be functionally cohesive if every instruction in the module is related to a single task. The name of the module will usually indicate whether it is functionally cohesive or not. For example, generate employee's pay slip, print grade cards etc., are name of modules that perform single function.

**(ii) Sequential Cohesion:** Sequential cohesion means that data produced in an earlier part of the module will be used in a later part of the same module i.e. in this type of cohesion, it is difficult to know whether the modulate is performing single function or not. However, to accomplish this task the module is simulated and instruction wise simulation is examined. Then we can conclude that module is sequentially cohesive. In other words, the concept of sequential cohesion is similar to pipeline processing. Sequencing of instructions plays a major role in the cohesiveness of module.
Example of sequential cohesion is

Produce purchase order
↓
Prepare shipping order
↓
Update inventory
↓
Update accounts

To the above set of instructions, purchase order is the initial input. Produce purchase order is the input to the second instruction when the shipping order is prepared. This output will serve as an input to the third instruction to update inventory and this will serve as inputs to update accounts. So in this way a sequence of instructions executed.

**(iii) Communicational Cohesion:** In communicational cohesion, the elements in the module process data items in the same film, but not necessarily in any specific order. In other words, communicational cohesion means
-Grouping of sequence of operations that operation that same data in the same

in the same module.
The user of the model may want to use a subset of the operation.

Thus in communicational cohesion all processing elements concentrate on one area of a data structure. Without having any restriction on the sequencing pattern of the instruction so in this type of cohesion, ordering of instruction is irrelevant. The most important thing is that, input data for each instruction is same, it is a moderate level of cohesion.

**(iv) Procedural Cohesion:** In procedural cohesion, the processing elements of a module are related and must be executed in a specific order. This is moderate level of cohesion. High cohesion is characterized by a module that performs one distinct procedural task. The execution of instruction in the module which is procedurally cohesion usually leads to the calls to other modules.

**For example:**
Collect the study material from university. Check the enrolment number on the hall ticket. Attend the compelling classes at the study center and submit the project and assignment at the study centre.
In the above example, instructions are not related to each-other in terms of sequence. However, each instruction is separate in functionality and leads to the execution of instruction in other module.

**(v) Temporal Cohesion:** In temporal cohesion, elements are grouped under one module because they are time related i.e. in other words, when a module contains tasks that are related by the fact that all must be executed with the same span of time, the module exhibits temporal cohesion.

Consider the following example:
Delete duplicate data from statement file
Reindex statement file
Back up of statement file

i.e. all the above stated operations has nothing in common except that they are end of the day activities.

**(vi) Logical Cohesion:** A module that performs tasks that are related logically (i.e. a module that produces all output regardless of type) is logically cohesive. In other words, in logical cohesion, a module performs somewhat related operations one of which is selected by a control flag that is passed to the module. A logically cohesive module consists of instructions in the form of sets. So, execution takes place in the term of set of instructions rather than individual instructions.

**For example:**
Combining different I/O operations in one module and passing an argument to choose which operation will be used.
It is called logical because the control flow of the modules is the only thing that ties the operations in the module together. This is bad type of cohesion, it is very difficult to maintain logically cohesive module.

**(vii) Coincidental Cohesion:** A module that performs a set of instructions task that relate to each-other loosely, if at all, is termed as coincidental cohesion. In this type of cohesion, there is no relationship between the instructions. This is the low type of cohesion. The reason for placing such type of totally unrelated instructions may be to save time from programming, to fix errors in the existing modules.

An example of *" low cohesion"* or coincidental cohesion:
Consider a module that performs error processing for an engineering analysis package. The module is called when computed data exceed pre-specified bounds, it performs the following tasks:

**(1)** Compute supplementary data based on original computed data.
**(2)** Produces an error report on the user's work stations
**(3)** Performs follow up calculations requested by the user
**(4)** Update a data base.
**(5)** Enable menu selection for subsequent processing.

Although the proceeding tasks are loosely related, each as an independent functional entity that might best be performed as a separate module. Combining the functions into single module can only serve to increase the likelihood of error propagation when a modification is made to one of the processing tasks stated above.
Coincidental cohesion is the work type cohesion among all the other types.
The order of priority of cohesion is functional cohesion, sequential cohesion, communicational cohesion, procedural cohesion, temporal cohesion, logical cohesion and coincidental cohesion.

***Diagram Showing Bad and Good Modularisation***

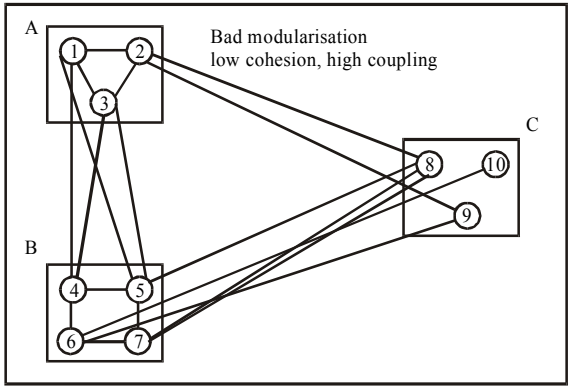



**Illustration of Bad Modularisation**

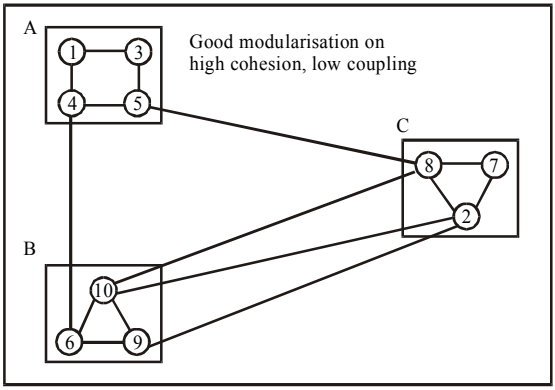



**Illustration of Good Modularisation.**

# Chapter-7

# System Design And Modeling

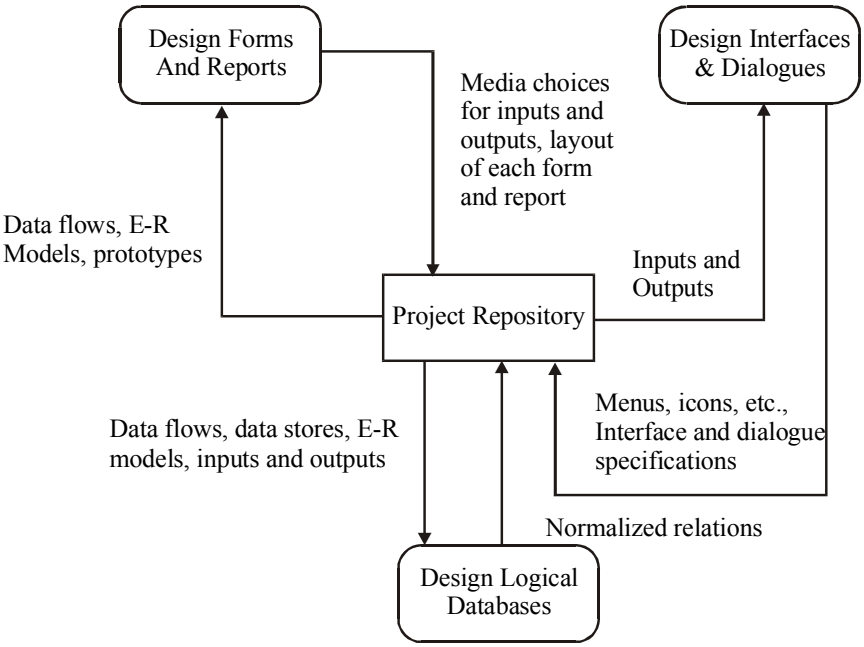
System design is a process through which requirements are translated into representation of software. Preliminary design is concerned with the transformation of requirements in to data and software architecture. Detail design focuses on refinements to the architectural representation that lead to detailed data structure and algorithmic representation for the software. In addition to data, architectural, and procedural design, many modern applications have a distinct interface design activity. Interface design establishes the layout and interactions between for human-machine interaction. System analysts convert the system specifications into logical and then physical specification. He/She designs the all aspect of system from input and output screens to reports, database and processes. Different modeling tools are used to design the database. Computer processes are designed by using pseudo English (structured English notation), decision trees and decision tables. Logical system design does not depend on any specific hardware/software platform. Basically, logical design considers the business aspects of the system. Physical design is obtained from logical design. During the physical design, the decision regarding the coding language database management system hardware platform, operating system, and networking environment being taken by the developing team. Final output of design phase is handled over to the programmers and system designer's for coding.

**Logical Design :** It is the phase of system development life cycle in which system analyst and user develops concrete understanding of the operation of the system. Various steps involved in the logical design are shown in the following figure. It includes the following steps:

**1.** Designing forms (hard copy and computer displays) and reports, which describe how data will appear to users in system inputs and outputs;

**2.** Designing interfaces and dialogues, which describe the pattern of interaction between users and software and

**3.** Designing logical databases, which describe a standard structure for the database of a system that is easy to implement in a variety of database technologies.




In Here, in the logical design, all functional features of the system chosen for the development in analysis are described independently of any computer platform. Logical design is tightly linked to previous system development phases, especially analysis. The three sub phases mention in the figure are not necessarily sequential.

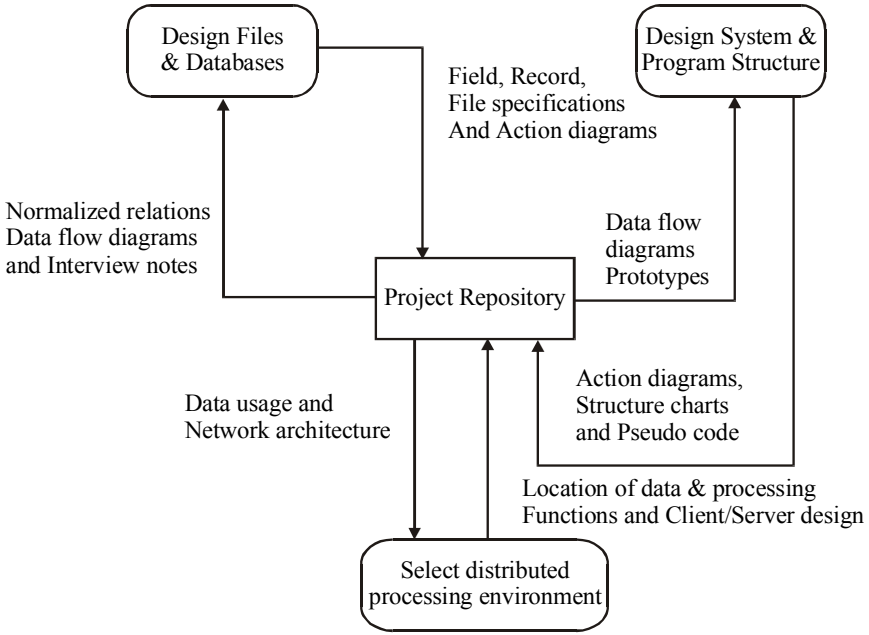
**The steps to be followed during physical design are given below:**

**1.** Designing physical files and databases – describes how data will be stored and accessed in secondary computer memory and how the quality of data will be ensured.

**2.** Designing system and program structure – describes the various programs and program modules that correspond to data flow diagrams and other documentation developed in earlier phases of lifecycle.

**3.** Designing distributed processing strategies – describes how your system will make data and processing available to users on computer networks within the capabilities of existing computer networks.

Various steps involved in physical design are depicted in the figure next:




**Q1. Define a DFD. Write the conventions that govern the construction of DFDs.**

**Data Flow Diagram (DFD)** is a powerful diagram that can be used to document the information flow. It also presents itself to be broken down in top-down fashion. At the top level, data flows are represented at very abstract aggregate level. Each component of the data flow is further broken down to different levels, so that at each level we have just a few entities to concentrate on. DFD have developed a representation scheme to represent data store (storage & retrieval of data), processes (where some changes are made to the system) and entities (the player in the game) and the actual information flows.

**The following seven conventions govern the construction of DFDs :**

**1.** Arrows should not cross each other.

**2.** Squares, circles, and files must bear names.

**3.** Decomposed data flows must be balanced (all data flows on the decomposed diagram must reflect flows in the original diagram).

**4.** No two data flows, squares, or circles can have the same name.

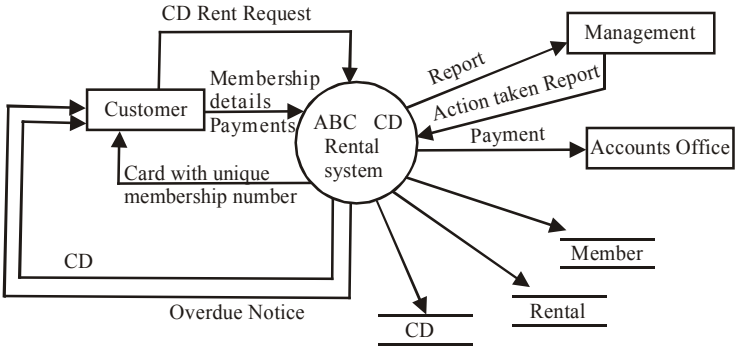
**5.** Draw all data flows around the outside of the diagram.

**6.** Choose meaningful names for data flows, processes, and data stores. Use

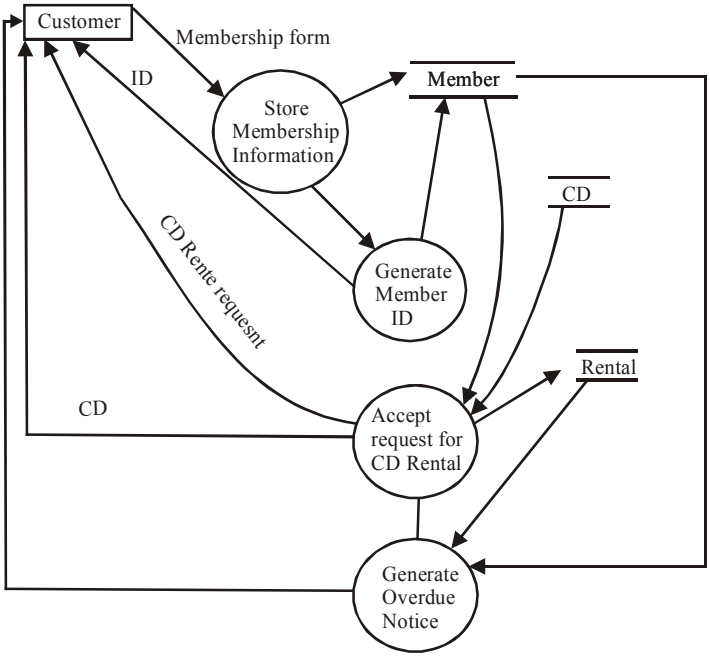
strong verbs followed by nouns.
**7.** Control information such as record counts, passwords, and validation requirements are not pertinent to a data-flow diagram.
If too many events seem to be occurring at a given point, an analyst can decompose a data conversion (circle). The new data conversions form a parent-child relationship with the original data conversion: the child circle.

**Difference between DFD and Decision Table : -**
**1.** DFD shows Flow of Data from one process to another while decision table never shows any flow of data.
**2.** DFD does not show how many conditions can be in the system into consideration while decision table shows all the conditions and actions for those in the system into consideration.
**3.** The primary strength of DFD is its ability to represent data flows. It may be used at high or low levels of analysis and provides good system documentation. However the tool weakly shows input and output detail. The user often finds it confusing initially.
Decision tables are best suited for dealing with complex branching routines. A decision table is perhaps the most useful tool for communicating problem details to the user.

**Q2. ABC is a CD Rental Library System. To become a member, and application form is completed by the customer. The form is then submitted to the Rental Library System. The customer receives a membership card, each card with a unique membership number. Only a member is allowed to borrow a CD. The system collects the payment and transfers it to the Accounts office. The details of each member are stored in 'Member' data store. For each rented CD, the system maintains a separate data store, 'Rental', which contains details, CD number (unique), rented date, membership number of the customer how rented the CDE, etc. The third data store is 'CD' which is the CD's inventory. The Rental Library System sends overdue notice to the members who have not returned the CD on due date. The system sends a management report to the management at the end of every month, which in return sends back the 'Action taken' Report.**
**Prepare a Context Diagram and Level 1 DFD for the above CD Library System.**




Context Level DFD




Level 1 DFD

**Q3. Write about Decision table.**

**Decision tables** are a precise yet compact way to model complicated logic. Decision tables, like if-then-else and switch-case statements, associate conditions with actions to perform. But, unlike the control structures found in

traditional programming languages, decision tables can associate many independent conditions with several actions in an elegant way.

Structure

Decision tables are typically divided into four quadrants, as shown below.

**The four quadrants**

| | |
|---|---|
| Conditions | Condition alternatives |
| Actions | Action entries |

Each condition corresponds to a variable, relation or predicate whose possible values are listed among the condition alternatives. Each action is a procedure or operation to perform, and the entries specify whether (or in what order) the action is to be performed for the set of condition alternatives the entry corresponds to. Many decision tables include in their condition alternatives the **don't care** symbol, a hyphen. Using don't cares can simplify decision tables, especially when a given condition has little influence on the actions to be performed. In some cases, entire conditions thought to be important initially are found to be irrelevant when none of the alternatives influence which actions are performed.

Aside from the basic four quadrant structure, decision tables vary widely in the way the condition alternatives and action entries are represented. Some decision tables use simple true/false values to represent the alternatives to a condition (akin to if-then-else), other tables may use numbered alternatives (akin to switch-case), and some tables even use fuzzy logic or probabilistic representations for condition alternatives. In a similar way, action entries can simply represent whether an action is to be performed (check the actions to perform), or in more advanced decision tables, the sequencing of actions to perform (number the actions to perform).

**Example**

The limited-entry decision table is the simplest to describe. The condition alternatives are simple boolean values, and the action entries are check-marks, representing which of the actions in a given column are to be performed.

A technical support company writes a decision table to diagnose printer problems based upon symptoms described to them over the phone from their clients.

**Printer troubleshooter**

| Printer troubleshooter | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Conditions | Printer does not print | Y | Y | Y | Y | N | N | N | N |
| | A red light is flashing | Y | Y | N | N | Y | Y | N | N |
| | Printer is unrecognized | Y | N | Y | N | Y | N | Y | N |
| Actions | Check the power cable | | | X | | | | | |
| | Check the printer-computer cable | X | | X | | | | | |
| | Ensure printer software is installed | X | | X | | X | | X | |
| | Check/replace ink | X | X | | | X | X | | |
| | Check for paper jam | | X | | X | | | | |

Of course, this is just a simple example (and it does not necessarily correspond to the reality of printer troubleshooting), but even so, it is possible to see how decision tables can scale to several conditions with many possibilities.

**Software engineering benefits**

Decision tables make it easy to observe that all possible conditions are accounted for. In the example above, every possible combination of the three conditions is given. In decision tables, when conditions are omitted, it is obvious even at a glance that logic is missing. Compare this to traditional control structures, where it is not easy to notice gaps in program logic with a mere glance — sometimes it is difficult to follow which conditions correspond to which actions!

Just as decision tables make it easy to audit control logic, decision tables demand that a programmer think of all possible conditions. With traditional control structures, it is easy to forget about corner cases, especially when the else statement is optional. Since logic is so important to programming, decision tables are an excellent tool for designing control logic. In one incredible anecdote, after a failed 6 man-year attempt to describe program logic for a file maintenance system using flow charts, four people solved the problem using decision tables in just four weeks. Choosing the right tool for the problem is fundamental.

**Q4. Write advantages of Decision Tree.**

Amongst other data mining methods, decision trees is the method that has

several advantages:

• **Decision trees are simple to understand and interpret.** People are able to understand decision tree models after a brief explanation.

• **Data preparation for a decision tree is basic or unnecessary.** Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed.

• **Is able to handle both nominal and categorical data.** Other techniques are usually specialised in analysing datasets that have only one type of variable. Ex: relation rules can be only used with nominal variables while neural networks can be used only with numerical variables.

• **Is a white box model.** If a given situation is observable in a model the explanation for the condition is easily explained by boolean logic. An example of a black box model is an artificial neural network since the explanation for the results is excessively complex to be comprehended.

• **It is possible to validate a model using statistical tests.** That makes it possible to account for the reliability of the model.

• **Is robust, perform well with large data in a short time.** Large amounts of data can be analysed using personal computers in a time short enough to enable stakeholders to take decisions based on its analysis.

**Q5. What is meant by 'Process Specification Tool'? Describe any two such tools. [DEC05, Q1(b)]**

The **process specification** is the description of what's happening inside each bottom-level, primitive bubble in a dataflow diagram. The purpose of a process specification is quite straightforward: it defines what must be done in order to transform inputs into outputs. It is a detailed description of the user's business policy that each bubble carries out.

There is a variety of tools that we can use to produce a process specification: decision tables, structured English, pre/post conditions, flowcharts, Nassi-Shneiderman diagrams, and so on. While most systems analysts favor structured English, you should remember that **any method can be used, as long as it satisfies two crucial requirements:**

· The process specification must be expressed in a form that can be verified by the user and the systems analyst. It is precisely for this reason that we avoid narrative English as a specification tool: it is notoriously ambiguous, especially when describing alternative actions (decisions) and repetitive actions (loops). By its nature, it also tends to cause great confusion when expressing compound Boolean conditions (i.e., combinations of the Boolean operators AND, OR, and NOT).

· The process specification must be expressed in a form that can be effectively communicated to the various audiences involved. While it will typically

be the systems analyst who writes the process specification, it will usually be a diverse audience of users, managers, auditors, quality assurance personnel, and others who must read the process specification.

**Following are the two Process Specification tools described :**

**(i) STRUCTURED ENGLISH**

Structured English, as the name implies, is "English with structure." That is, it is a subset of the full English language with some major restrictions on the kind of sentences that can be used and the manner in which sentences can be put together. It is also known by such names as PDL (for program design language) and PSL (for problem statement language or problem specification language). Its purpose is to strike a reasonable balance between the precision of a formal programming language and the casual informality and readability of the English language.

A sentence in structured English may consist of an algebraic equation, **for example**,

X = (Y*Z)/(Q+14)

or a simple imperative sentence consisting of a verb and an object. Note that this sentence does not have the semicolon that terminates a programming statement in many different programming languages; it may or may not terminate with a period (“.”), depending on your taste in such things. Also, note that sentences describing computations can be prefixed with the verbs COMPUTE, ADD, SET, and so on; thus, we could have written the above example as

COMPUTE X = (Y*Z)/(Q+14)

and we can have structured English computations like the following ones:

SET TAX-RATE TO 13

ADD 3 TO X

MULTIPLY UNIT-PRICE BY QUANTITY

Verbs should be chosen from a small set of action-oriented verbs such as:

GET (or ACCEPT or READ)

PUT (or DISPLAY or WRITE)

FIND (or SEARCH or LOCATE)

ADD

SUBTRACT

MULTIPLY

DIVIDE

COMPUTE

DELETE

FIND

VALIDATE

MOVE
REPLACE
SET
SORT

Finally, structured English allows sentences to be combined in a few limited ways; these are taken from the familiar structured programming constructs.

· The IF-THEN-ELSE construct is used to describe alternative sentences that are to be carried out based on the result of a binary decision. The IF-THEN-ELSE construct can take either of the following two forms:

IF condition-1
sentence-1
ENDIF

or

IF condition-1
sentence-1
ELSE
sentence-2
ENDIF

Thus, the systems analyst may write:

IF customer lives in New York
add customer to MARKETING-PROSPECTS
ENDIF

or

IF customer-age greater than 65
set billing-rate to senior-citizen-rate
ELSE
set billing-rate to normal-rate
ENDIF

Like the above there are many other constructs like CASE / DO CASE, DO WHILE.

**(ii) DECISION TREE**

We will use an example to explain decision trees:

There is Mr. Dinesh; is the manager of a famous golf club. Sadly, he is having some trouble with his customers attendance. There are days that everyone wants to play golf and the staff of the club is not enough for them; on some other days for no apparent reason, no one plays golf; and the club has a high slack of employees.

Dinesh's objective is to optimise the staff availability by trying to predict when people will play golf using. To accomplish that he needs to understand the reason people decide to play and if there is any explanation for that.

So during two weeks he has been recording the:

Outlook, whether it was sunny, clouded or raining. The temperature in degrees Fahrenheit. The Relative Humidity in percent. Whether it was windy or not.

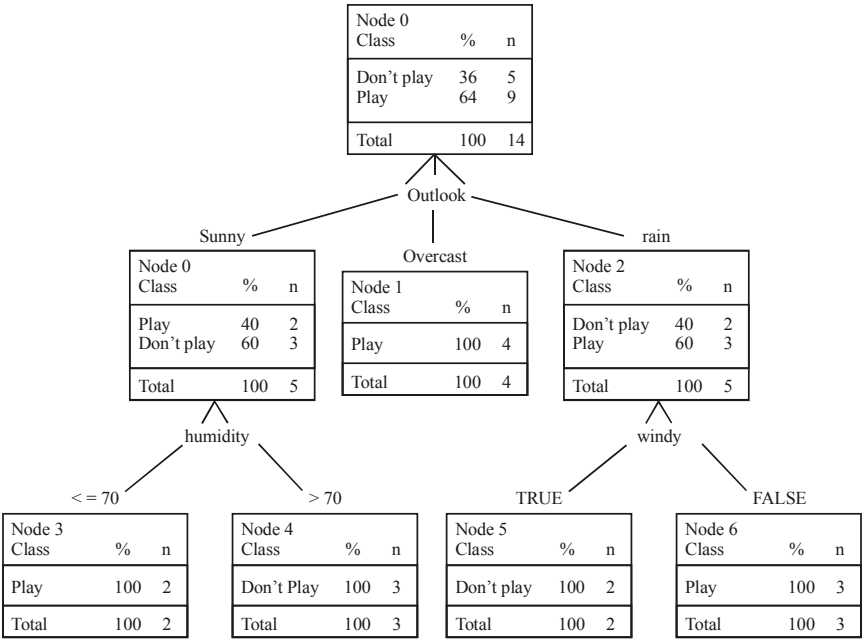
And of course if people attended to the Golf club on that day. He ended with this dataset containing 14 rows and 5 columns.

**Play golf dataset**

| **Independent variable** | | | | **Dep. var** |
|---|---|---|---|---|
| **OUTLOOK** | **TEMPERATURE** | **HUMIDITY** | **WINDY** | **PLAY** |
| Sunny | 85 | 85 | FALSE | Don't Play |
| sunny | 80 | 90 | TRUE | Don't Play |
| overcast | 83 | 78 | FALSE | Play |
| rain | 70 | 96 | FALSE | Play |
| rain | 68 | 80 | FALSE | Play |
| rain | 65 | 70 | TRUE | Don't Play |
| overcast | 64 | 65 | TRUE | Play |
| sunny | 72 | 95 | FALSE | Don't'Play |
| suny | 69 | 70 | FALSE | Play |
| rain | 75 | 80 | FALSE | Play |
| sunny | 75 | 70 | TRUE | Play |
| overcast | 72 | 90 | TRUE | Play |
| overcast | 81 | 75 | FALSE | Play |
| rain | 71 | 80 | TRUE | Don't Play |

A decision tree model is then proposed to solve Dinesh's problem.

Below is the Decision Tree for the given problem.




A decision tree is a model of the data that encodes the distribution of the class label (again the Y) in terms of the predictor attributes.

It is a directed, acyclic graph in form of a tree. The top node represents all the data. The classification tree algorithm finds out that the best way to explain the dependent variable, play, is by using the variable Outlook. Using the categories of the variable outlook three different groups were found:

The group that plays golf when is sunny, the group that plays when is clouded and surprisingly we realise that when it is raining some people do play golf!

Our 1st conclusion: if the outlook is overcast people always play golf and there are some fanatical people that play golf even in the rain.

Then again we divide the sunny group in two groups. We realise that customers don't like to play golf if the humidity is higher than seventy percent.

Finally we divide the rain category in two and find out that customers will not play golf if it is windy.

And here is the short solution of the problem given by the classification tree software. Dinesh; dismiss most of the staff on days that are sunny and humid or on rainy days that are windy because almost no one is going to play golf on those days. On the other days when a lot of people will play golf, you can hire some temporary staff to help you on the job.

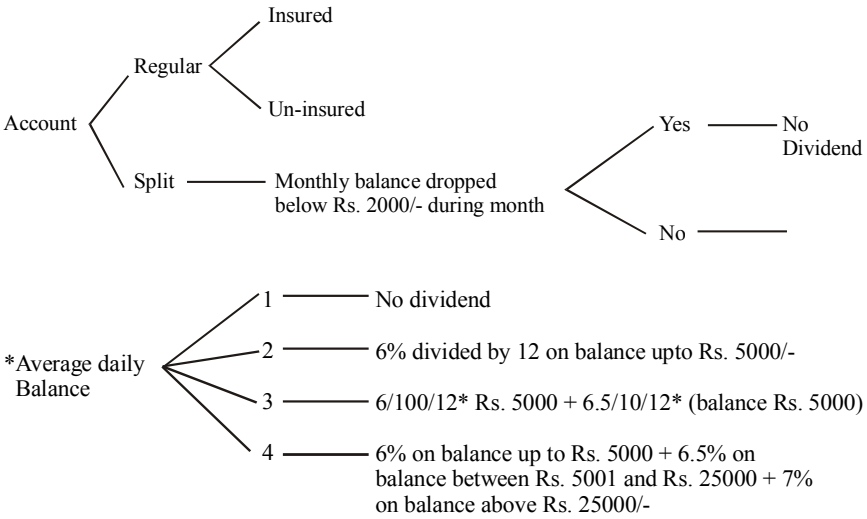
The conclusion is that decision tree helped us turn a complex data representation into a much easier structure.

**Q6. Draw a decision table for following policy statement:**
**"A bank offers two types of savings accounts, regular rate and split rate. The regular rate account pays dividends on the account balance at the end of each quarter. Funds withdrawn during the quarter earn no dividends. There is no minimum balance on the regular account. Regular rate account may be insured. Insured account gets 5.75 percent annual interest. Uninsured regular rate accounts get 6.00 percent annual interest.**

**For split rate accounts, dividends are paid monthly on the average daily balance for that month. Daily balances go up and down in accordance with deposits and withdrawals. The average daily balance is determined by adding each days closing balance and dividing this sum by the number of days in the month.**

**If the balance dropped below Rs.2,000/- during the month no dividend is paid. So, if the average daily balance is less than Rs.2,000/-, then no dividend is paid. Otherwise, if the average daily balance is Rs.2,000/- or more, then an interest of 6% per annum is paid on the first Rs.5,000/-, 6.5% on the next Rs.20,000/- and 7% on funds over Rs.25,000/-. There is no insurance on split rate.**

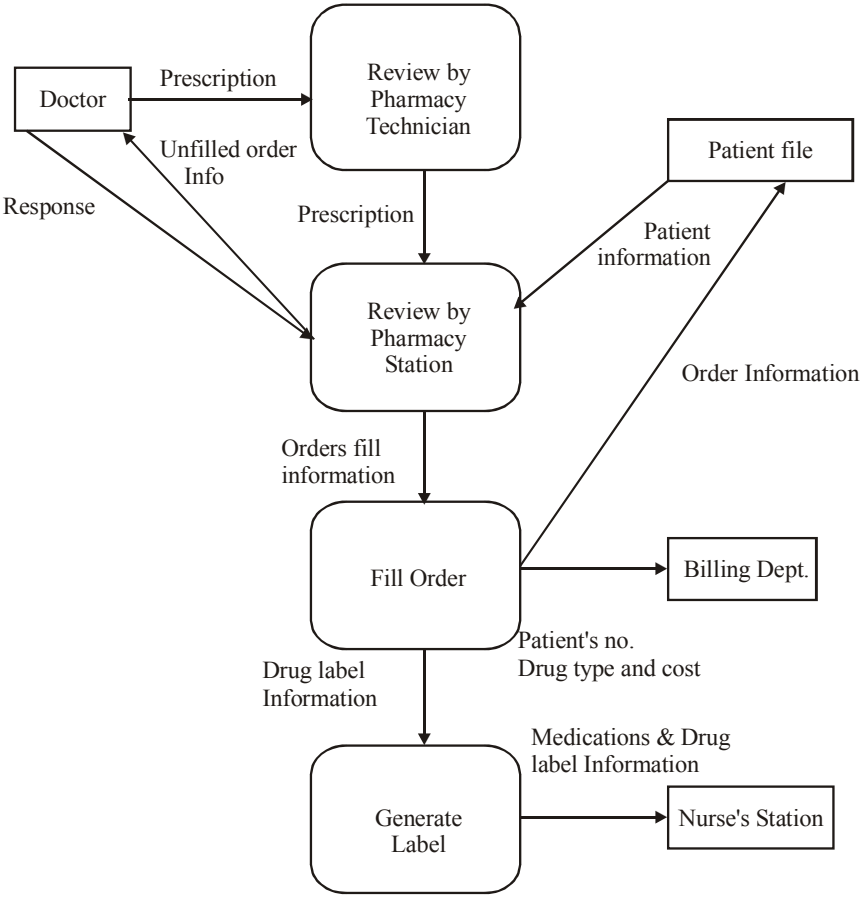
| | Process Name | Dividend rate Rules | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Conditions | Account type | R | R | S | S | S | S | S |
| | Insurance | Y | N | -- | -- | N | N | N |
| | Balance dropped below Rs. 2000/- during month | -- | -- | Y | N | N | N | N |
| | Average daily balance | -- | -- | -- | 1 | 2 | 3 | 4 |
| Actions | Pay no dividend | | | X | X | | | |
| | 5.75% divided by 4 quarterly dividend on entire balance. | X | | | | | | |
| | 6.000% divided by 4 quarters | | X | | | | | |
| | 6.000% divided by 12 monthly dividend on balance up to Rs. 5000/- | | | | | X | X | X |
| | 6.500% divided by 12 monthly dividend on balance between Rs. 5001/- to Rs. 20000/- | | | | | | X | X |
| | 7.000% divided by 12 monthly dividend on a balance of above Rs. 25000/- | | | | | | | X |




**Q7. Develop a context level DFD and First level DFD for the hospital pharmacy system describe in the following case study : "The pharmacy at Sanjeevni Hospital fills medical prescriptions for all patients and distributes these medications to the nurse stations responsible for the patient's care. Medical prescriptions are written by the doctors and sent to the pharmacy. A pharmacy technician reviews the prescriptions and sends them to the appropriate pharmacy station. At each station, a pharmacist reviews the order, checks the patient file to determine the appropriateness of the prescription and fills the order. If the pharmacist does not fill the order, the prescribing doctor is contacted to discuss the situation. In this case the order may ultimately be filed or the doctor may write another prescription, depending on the outcome of the discussion. Once filled, a prescription level is generated listing the patient's name, the drug type and dosage, an expiration date and any special instruction. The level is placed on the drug container and the order is sent to the appropriate nurse station. The patient's admission number, the drug type, and the cost of the prescription are then sending to the billing department".**

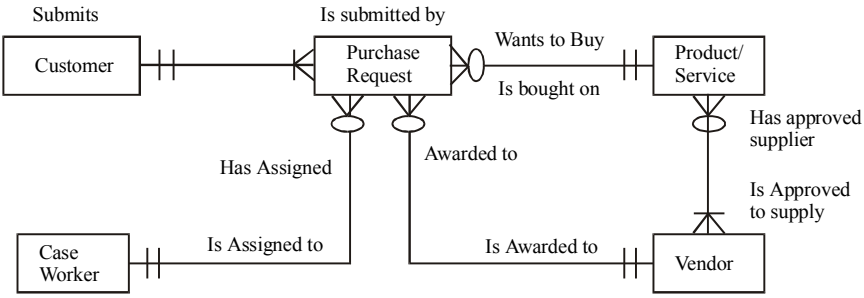
Following is the Zero Level DFD for the given problem :


Following is the First Level DFD for the given problem :




**Q8. Draw an E-R diagram for the following case study:**
**"In a purchasing department at one company, each purchase request is assigned to a 'case worker' within the purchase department. This case-worker follows the purchase request through the entire purchasing process and acts as the sole contact person with the person or unit buying the goods or services. The purchasing department refers to its fellow**

**employees buying goods and services as "customers". The purchasing process is such that purchase request must go to vendors. The product or service can simply be bought from any approved vendor but the purchase request must still be approved by the purchasing department."**



# Chapter-8

# Forms And Report Design

This chapter describes the input-output design of graphical user interface. The input user interface is through **Forms** and output through **Reports** that are associated with the system. Forms and Reports are the key ingredients for a successful system. The logical phase within the system involves the designing of Forms and Reports. A Form is nothing but a box that contains control objects, Forms are used to collect data from the user for the system and Reports to deliver the relevant information to the user. Forms can be used for a variety of purposes. e.g. to enter data into the table; to create a switch-board to open other Forms and Reports and to create a custom dialog box to accept user input and they carry out an action on the output. We can tabulate the deliverables produced during the process of designing forms and reports.

**Q1. What is form and report?**

**Forms :**

As a paper form which is used to fill out information with a pen or pencil, a Form in database design identifies the data we count to collect. Form can be used for variety of purposes, to create data entry form to enter data into table. A form represent a window's or dialog box's visual appearance. A Form can have its own set of properties.

Examples of Forms are, Business transaction forms, Electronic spread-sheets, ATM transaction etc.

Importance of Forms

— A Form provides a user friendly look.
— A Form provides a convenient way to enter data.
— A Form presents data with special fonts and other graphical effects.
— A Form offers the most convenient way to add, delete, find etc. records in the data base.
— Entry field in a form can be designed to present a list of valid values from which user can easily pick and use.
— A Form can be easily customized to the user requirements.

**Reports :**

A report is an effective and efficient way to present the stored data in the database in a printing format. We can have control over the size and

appearance of every thing on a report i.e. A report is the information that is organized and formatted to fit the required specification. It is a passive document that contains only predefined data and is used for output and reading. Reports can be printed on paper or may be saved in a file to view later. Reports are the most visible component of a working information system.
Example of reports, invoices, list of students in a course, weekly sales, collection summary, payment schedule etc.

**Q2. List the importance of forms and reports. Also write difference between them.**

**Forms**

As a paper form which is used to fill out information with a pen or pencil, a Form in database design identifies the data we count to collect. Form can be used for variety of purposes, to create data entry form to enter data into table. A form represent a window's or dialog box's visual appearance. A Form can have its own set of properties.

Examples of Forms are, Business transaction forms, Electronic spreadsheets, ATM transaction etc.

Importance of Forms

— A Form provides a user friendly look.
— A Form provides a convenient way to enter data.
— A Form presents data with special fonts and other graphical effects.
— A Form offers the most convenient way to add, delete, find etc. records in the data base.
— Entry field in a form can be designed to present a list of valid values from which user can easily pick and use.
— A Form can be easily customized to the user requirements.

**Reports**

A report is an effective and efficient way to present the stored data in the database in a printing format. We can have control over the size and appearance of every thing on a report i.e. A report is the information that is organized and formatted to fit the required specification. It is a passive document that contains only predefined data and is used for output and reading. Reports can be printed on paper or may be saved in a file to view later. Reports are the most visible component of a working information system.
Example of reports, invoices, list of students in a course, weekly sales, collection summary, payment schedule etc.

**Q3. Explain the process of designing the forms and reports.**

**[JUNE02, Q1(c)]**

Good quality business processes deliver the right information to the right people

in the right format and at the right time. The design of forms and reports concentrates on this goal.

Designing of forms and reports is a user-focused activity where prototyping approach is followed. There are some useful questions related to the creation of all forms and reports such as "who, what, when, where and how" which must be answered in order to design effective forms and reports.

WHO Understanding who the actual users are, their skills and abilities, their education level, business background, etc., will greatly enhance the ability to create effective design.

WHAT We need to have a clear understanding of what is the purpose of the form or report and what task will the users be performing and what information is required so as to successfully complete the given task.

WHEN Knowing when exactly the form or report is needed and used will help to set up time limits so that the form or report can be made available to the users within that time frame.

HOW How many people will be using this form or report, i.e., if the form or report is to be used by a single person, then it will be simple in design but if a large number of people are going to use it, then the design will have to go through a more extensive requirements collection and usability assessment process.

After having answered all the above questions, we would have collected all the initial requirements. The next step is to refine this information into an initial prototype. Structuring and refining the requirements are completed without interacting with the end users, although we may need to occasionally contact users in order to clarify some issues that might have been overlooked during analysis. Once the initial prototype is ready, we should ask the users to review and evaluate the prototype. After the review, the design may be accepted by the users. Or at times the users may ask for certain changes to be made. In case changes are to be made then the construction-evaluation-refinement cycle will have to be repeated until the design is accepted.

The next step in the Design process is to Design, Validate and Test the outputs using some combination of the following tools:

**a)** Layout tools (Ex.: Hand sketches, printer/display layout charts or CASE)

**b)** Prototyping tools (Ex.: Spreadsheet, PC, DBMS, 4GL)

**c)** Code generating tools (Ex.: Report writer )

The initial prototype may be constructed in numerous environments. For example, a CASE tool or the standard development tools that are used within the organization be used. Usually, initial prototype are mock screens that can

be produced using word processor, computer graphics design package, or electronic spreadsheet or for reports using specialized software like Crystal Reports. Mock screens are not the working modules or systems.

**Q4. List types of information. Also define them.**

**Types of Information**

Information can be classified according to their distribution, inside or outside the organisation and the people who need and use them are as follows;

**1. Internal information**
**2. External information**
**3. Turn around information.**

**1. Internal Information :** It is the information that is collected, generated or consumed within an organisation. This type of information is intended only for the internal system, owners and users of within the organisation. Internal information can consists of day to day business operations, management decision of reports summarizing the daily activities within the organisation.

**2. External Information :** External information or outputs are intended for customers, suppliers, partners and regulatory agencies. They usually conclude on business transactions.

Examples of external outputs are invoices, account statements, pay checks, courses schedule, airline tickets, boarding passes etc.

**3. Turn around Outputs or Information :** Turn around outputs are those external outputs that eventually reenter the system as input. The document that consists of such information is called turn around document. These documents delivered to an external customer as an output, but return (in part or whole) as an internal to provide a new information as an input to the system. For example, warranty cards or acknowledgment slip.

**Q5. Explain at least five criteria for form design and report design.**
**[DEC05, Q4(b)]**

**Five criteria for form design are** (i) Organization, (ii) Consistency, (iii) Completeness, (iv) Flexible entry, and (v) Economy, which are followed to achieve the goal of good form design.

**(i) Organization :** The different parts of a form must be arranged in a proper order with visual separation between the parts. Balancing of different information on the form should be done according to the sequence of entry, frequency of use, function and significances of that particular data. The first

data available, the most important data and the data that is going to be used most frequently should always be placed in the beginning of the form. If there are groups of data of the like information, they should be placed together just as Name+Age+Address+Contact Number. Grouping of information will help the user to understand which section of the form they are completing.

**(ii) Consistency :** Forms must also be consistent with related forms and with other forms in the organization. If the forms are consistent, then it will be easy for the users to learn how to fill them. Consistent forms reduce errors and data capture costs.

**(iii) Completeness :** The form should gather all the necessary data at the source so that there is no need to transcribe data to other forms. This reduces the major source of errors.

**(iv) Economy :** The total cost of design, printing, data entry, etc., must be minimized. Most of the times, it is required to increase one cost to reduce another. Usually, handling costs are much more than the cost of designing and printing. Having spent more resources on design and printing often reduces the cost of data capture and keying.

**(v) Flexible Entry :** It should be possible to enter data by hand or with a typewriter. Mostly, both kinds of entries occur.

**Five criteria for report design are** (i) Relevance, (ii) Accuracy, (iii) Clarity, (iv) Timeliness, and (v) Cost, which are followed to achieve the goal of good report design.
**(i) Relevance :** Only the information that is relevant to the purpose of the report should be present in the report. This is a selection process, i.e., all the relevant information should be included and all the irrelevant information or data should be excluded. Only required information should be printed or displayed.

**(ii) Accuracy :** Accuracy is very important because if the data are inaccurate, then the main purpose of the report which is to provide accurate information to the user will not be accomplished. Incomplete data are also inaccurate.

**(iii) Clarity :** The information that is present on the report should be clear and understandable. The information present should be balanced on the report, the display should not be too crowded and not too spread out. Sufficient margins and spacing throughout the output will enhance readability. Desired information

must be easy to locate. Comparisons, ratios, percentages, exception flags and graphs should be used whenever needed.

**(iv) Cost :** Every report has two costs. First is the cost of preparation, which consists of analysis, design, computation and distribution. Second is the cost of reading the report and locating germane parts of it. Often the cost of reading the report is forgotten during the calculation of costs. The reading cost can be significantly reduced only if the appropriate information is presented clearly on the report. The total cost should always be less than the expected benefits. Only then the report should be prepared.

**(v) Timeliness :** Reports must be prepared and ready for use in time. Most reports provide information, which is used to make decisions. Hence, this information must reach the recipients while the information is pertinent to transactions or decisions. Information is of very little use if it arrives after the decision are made. So, it must be time bounded.

**Q6. List the various advantages of using Forms and Reports.**
Some of the advantages of using the forms are mentioned below:
**1.** Forms provide an easy way to view data.
**2.** Using forms, data can be entered efficiently.
**3.** We can present data in attractive format using forms.
**4.** Forms offer convenient layout for entering, changing and viewing records present in the database.
The advantages of the reports are:
**1.** Reports organize and present data in groups.
**2.** With reports, we can calculate running totals, group totals, grand totals etc.
**3.** We can present data in attractive format.
**4.** Within reports we can include sub-forms, sub-reports and graphs.

**Q7. What are the merits and demerits of using colors when designing system outputs?**
**Merits** of using colours:
**1.** Soothes or strikes the eye;
**2.** Emphasize the logical organization of information;
**3.** Draws attention in warnings;

**Demerits** of using colours :
**1.** Resolution may degrade with different displays;
**2.** Colours fidelity may degrade on different displays and
**3.** Printing or conversion to other media may not easily translate.

# Chapter-9

# Physical File Design And Database Design

In this chapter we will study the various steps involved in the design of a database system of an organization. It begins with the identification of a problem area in the information handling capability of an organization. The feasibility of using a database system to resolve these problems is studied and where it is found that such approach is required, the design process starts. The processing requirements for the database is gathered and preliminary design is undertaken. The preliminary design is independent of any DBMS or its data model. The decision of which computing system and DBMS system is made upon the completion of preliminary and the identification of the processing, storage and performance needs. Once the decision is made on these aspects, the final design begins. It consists of mapping the preliminary design into the model of the selected DBMS and implement the various schemes in the DDL (Data Definition Language) of the DBMS. The physical database design is completed and the internal schema is designed. The application programs and the transaction is coded and the system is integrated and evaluated. At the end of evaluation phase the system is made available for performance evaluation/testing. A good database design helps efficient data storage and retrieval.

**Introduction to Database Design**

Database design is an iterative process. Over a period of time, a number of design methodology have been developed. A database can be thought of as a set of logically related files organized to facilitate access by one or more application program and to minimize data redundancy.

**Flat Files versus Database**

Traditionally the term file has been used to refer to the folder that holds all the related material in ordered storage for quick reference. Consider an example of flat file, which stores the pin code of vendor's address. So any change in the field size will enforce changes in the entire program which uses the data related to the vendor's address and if the file being used by more than one application the problem expands its area.

The traditional/flat file oriented approach to information processing has a separate master file and its own set of personal files for each application. — Traditional file approach significantly lacking sharing of data across various applications.

**1. Program/data dependency :** if a data field is to be added to a master file, all such programs that access the master file would have to be changed to all for this new field.

**2. Data redundancy :** the same piece of information may be stored in two or more files.

**3. Lack of flexibility:** As there is a strong coupling between data and program, most of the data/information can be retrieved in a way it has been programmed.

**Database approach:** In this approach a database is available to many users for different purposes. The content of database is obtained by combining data from all sources in an organization.

The following are the advantages of DBMS over flat files:

- Reduction *of Redundancies:* Centralized control of data by the DBA avoids unnecessary duplication of data and effectively reduces total amount of data storage required.
- Sharing of data-Allows sharing of data across any number of program applications or users.
- *Data integrity:* Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity.
- *Data security:* Proper data access procedure can be implemented for various types of data and operations by the DBA.
- *Conflict resolution:* DBA can choose the best file structure and access method to get optimal performance for the response critical applications.
- *Data independence:* Provides physical and logical data independence.

**Steps in Database Design**

**(1) Analysis :** It is a process of collection of requirements and specification details about a system so as to create a conceptual data model independent of the target database technology. This step basically results an E-R data model.

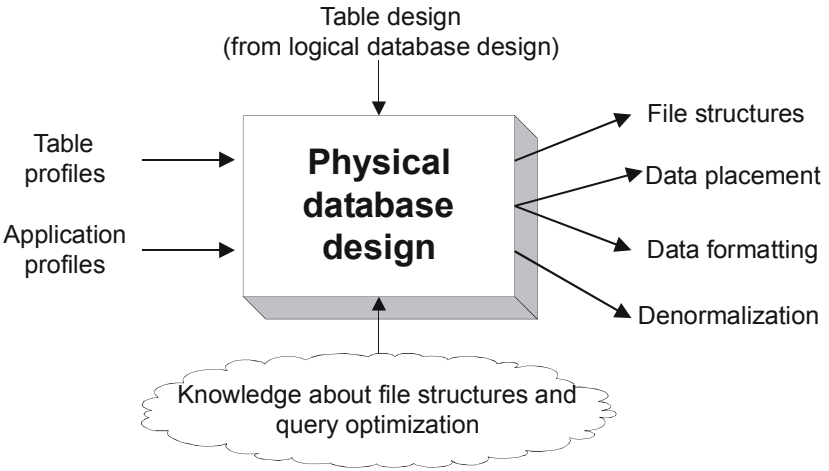
**(2) Design :** It is a process of creating a logical data model dependent on the target technology (i.e. hierarchical, network or relational) but not to a specific database implementation (e.g. Oracle, SQL server SYBASE, BD2).

**(3) Implementation :** It is a process of creating the physical model or scheme for one specific data base management system (such as Oracle, MS SQL, DB2).

This results in optimized physical design model.

**Q1. What are the inputs to physical database design? Also, write at least five guidelines for database design. [DEC05, Q1(c)]**
**#: Explain the five steps of Database Design with the help of a figure. [JUNE05, Q5(a)]**

Physical database design consists of a number of different inputs and outputs as depicted in this figure.




The starting point is the table design from the logical database design phase. The table and application profiles (Inputs) are used specifically for physical database design.
The most important outputs are decisions about file structures and data placement.
Knowledge about file structures and query optimization is in the environment of physical database design rather than being an input.
Physical database design is better characterized as a series of decision-making processes rather than one large process.

**Inputs of Physical Database Design :**
Physical database design requires inputs specified in sufficient detail.
Table profiles and application profiles are important and sometimes difficult-to-define inputs. Inputs specified without enough detail can lead to poor decisions in physical database design and query optimization. A table profile summarizes a table as a whole, the columns within a table, and the relationships between tables. Application profiles summarize the queries, forms, and reports that access a database.

**Following are the Guidelines for Database Design :**

* ensure that the data stored in the files (database tables) are atomic. Data stored in the atomic form can be combined later to generate data in specific form;
* every table must have a primary key which identifies each record in the table distinctly. Descriptive and meaningful name is to be used while naming a field in the table (For example, use product_id instead of ID);
* use single column primary key whenever possible. As most of the join operations are made on primary key and composite primary keys make the operation slower;
* use numeric key whenever possible;
* use primary key name as foreign key for better readability;
* avoid allowing null values to go into the columns that have discrete range of possible values; and
* avoid multiple tables with similar structure when one table is sufficient.

**Q2. Define the term Record. Write at least five activities that will enable the queries to run faster. Explain the process of Denormalization of Tables. [JUNE05, Q2(b)]**

**A Record** is a collection of fields. Records are common to both databases and files. Records are collection of fields in a predefined format. The design of physical record involves putting the collection of fields in a single logical unit so that the fields are stored in adjacent locations for better storage and retrieval.

**Following are the five activities that will enable the queries to run faster :**

**1.** Combining tables to avoid joins

**2.** Vertical partitioning refers to placing different columns of a table into separate files by repeating the primary key in each of the files.

**3.** Horizontal partitioning refers to placing different rows of a table into separate files. For example, in an order table order pertaining to different regions can be kept in a separate table for efficient retrieval of records.

**4.** Data replication refers to the same data being stored in multiple places in the database.

**5.** Record partitioning refers to a combination of both horizontal and vertical partitioning as in distributed database processing

**Denormalization** is the process of transforming normalized relations into unnormalized physical record specifications. The motivation behind de-normalization is poor performance of normalized table. The following may be of

use for denormalization.

**Below is the Process of Denormalization :**
**(i)** Select the dominant process based on frequency of execution and frequency of data access;
**(ii)** Define the join table for dominant process;
**(iii)** Evaluate the cost of query, updates and storage for the schema. Consider possibility of renormalization to avoid table joins.

**Q3. List the various types of Files and File organisations used.**
**Design of Physical Files** – A file is a collection of identical record type occurrences pertaining to an entity set and is labeled to identify the entity set. The storage method of files in memory depends upon the operating system. As there is operating system that allows the files to split into pieces. Following are the types of files:
**1.** Work file
**2.** Report file
**3.** Audit file
**4.** Back up file
**5.** Transaction file
**6.** Table file
**7.** Master file
**8.** Archive or archival file
**9.** Library file
**10.** Dump file

**Work file** – It is a temporarily created file for holding intermediate result of data processing e.g. a stored list of teachers, customers, students etc.

**Report file** – These are collected contents of individual output reports or document produced by the system. Report files are produced by the system, but printing may not be available for all the files.

**Audit file** – This type of file does not store business data but a log related to transaction e.g. data and time of access, modification etc.

**Back up file** – It is a copy of master, transaction or table file. It helps in retrieving data if anything happens to the original.

**Transaction file** – It is a temporary file used for two purposes. First of all, it is used to accumulate data about events they occur (items bought, sold, dele-

tion, updation etc.). Secondly, it helps in updating in the master files into reflect the result of current transactions. The term transaction refers to any business that affects the organization about which data is captured.

**Table file** – It is a special type of master file, which is included in many systems to meet the specific requirements where data must be referenced repeatedly.

**Master file** – It is a permanent file. Most file design activities concentrate here. Master files are considered very significant because they contain all the essential data or records needed by a business. For example, a file containing the information about customers, students, bank account holders etc..

**Archive or archival file** – These files are copies made for long term storage of data that may be required at much later date. Usually, archival files are stored away from the computer centre, so that they can not be easily retrieved for use.

**Library file** – It generally contains the application programs, utility programs and system software packages.

**Dump file** – It is a copy of computer held data at a particular point of time.

**File Organizations** – Files are organized to ensure that records are available for processing. Before a file is created the application to which the file will be used must be carefully examined. There are different types of file organizations depending on the organizations of records in the disk and the secondary storage. Decision on the specific file organization must ensure that its application leads to the following:
**1.** Efficient use of disk space
**2.** Reduce disk access time
**3.** Fast retrieval of records

**Serial File Organization** – A serial file is created by placing the record as it is created. It leaves no gap between the records stored on the disk. The packing density is 100 percent in this case. Examples of serial file organizations are – dump file, print file, log file and transaction files.

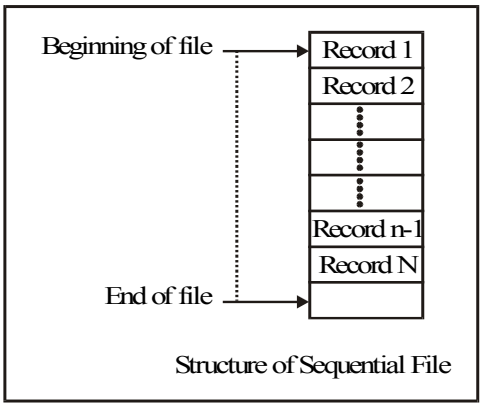
**Sequential File Organisation** – It is the simplest method to store and retrieve data from a file. In a sequential file records are created consecutively when the file is created and must be accessed consecutively when the file is later used for input. In a sequentially organized file records are maintained in

the logical sequence of their primary key values. The processing of sequential file conceptually simple, but inefficient for random access. However, if access to the file, is strictly sequential, a sequential file is suitable.

A sequential file could be stored on a sequential storage device such as magnetic tape.

Advantages offered by the sequential file is the case of access to the next record, simplicity of organization and the absence of auxiliary data structured.

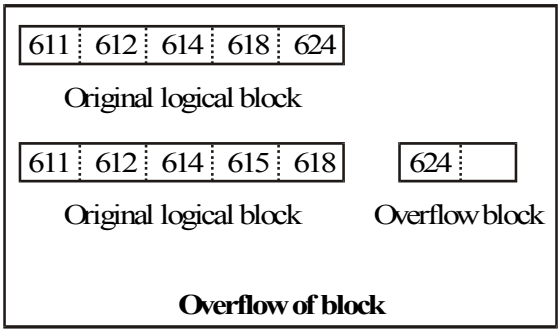
Disadvantages – To locate a particular record, the program starts search from the beginning of the file till the matching primary key is found i.e. larger record access time. Deletion of record may cause wastages of space and adding a new record needs rewriting of the file. This type of file organization is suitable for master files.



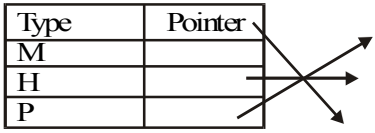
Structure of Sequential File

**Indexed Sequential File Organization –** An index is a set of (key, address) pairs. Indexing associates a set of objects to a set of orderable quantities to provide a mechanism for faster access. Indexes created from a sequential set of primary keys are referred to as index sequential.

A sequential (or sorted on primary keys) file that is indexed is called an index sequential file. The index provides random access to records, while the sequential nature of the file provides easy access to the subsequent records as well as sequential processing. An additional feature of this file system is the overflow area. This feature provides additional space for record addition without necessitating the creation of new file.

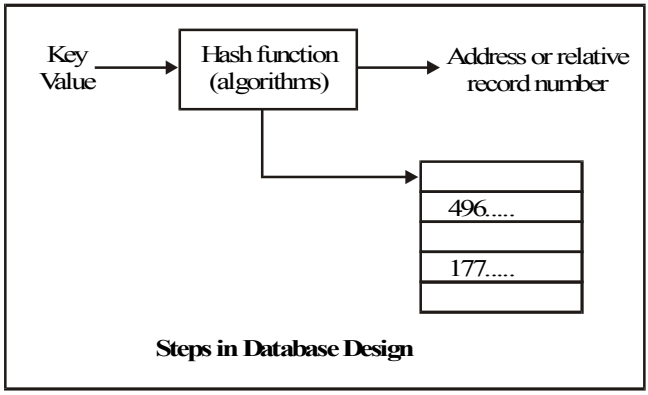



| Item-code | Item-name | Type | Price |
|---|---|---|---|
| 245 | Processor | P | 820 |
| 246 | Hard Disk | H | 55 |
| 247 | Memory | M | 12 |




The above figure depicts the indexed sequential file organization.

**Hashed File Organization** – In this organization, the key value is mapped directly to the storage location i.e. records are physically ordered according to hash function. The method of direct mapping is by performing arithmetic manipulation of the key value is called hashing process.

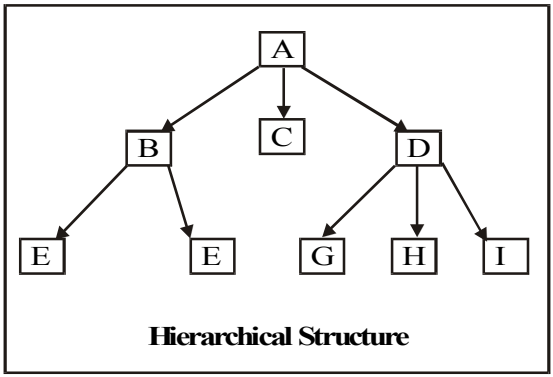


**Design of Database** – The robustness and performance majority depends on the design of database. It is similar to the pillars of a multistory building. Poor database design may lead to degraded performance of the software. In case of real time system it may cause disasters. Following are the steps to keep in mind while designing a Database:

**1.** Selection of database architecture
**2.** Designing database scheme
**3.** Selecting indexes
**4.** Estimation of the database capacity

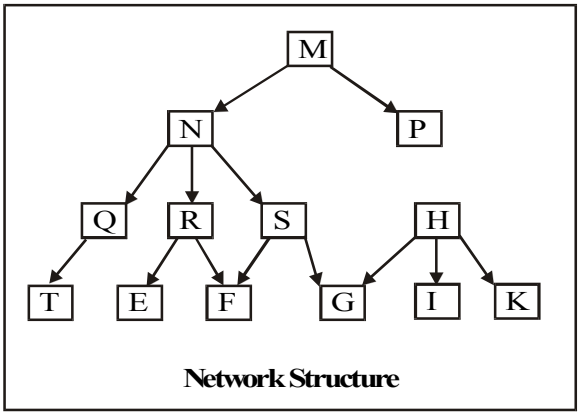
**Selection of Database Architecture** - This is the root of database design for any information system. Before we decide the target DBMS we must consider its implementation. Databases relate data sets in one of the following models:

**Hierarchical Model** – In hierarchical database structure, the stored data get more and more detailed as one branches further and further out on a tree. In other words, it links records in tree data structure such that each record has only one 'parent' from which it emanates.




Hierarchical Structure

**Network Model** – The network related data sets are similar to hierarchical ones, except that a node may have more than one parent. Thus hierarchical DBMS is a subset of network DBMS. A neural network is an example of network model.




**Network Structure**

**Relation Model** – Relational data structure organizes the data in terms of two dimensional tables. It is the most commonly used database model. Relational data sets order data in a table of rows and columns and differ from hierarchical or network counterparts. In a relational database management system files are represented by two dimensional tables, each of which is called a 'relation'. Reference keys (primary, foreign keys etc.) join different tables together. Indexes provide fast access to specific record in the database, for example MYSQL, DB2 etc.

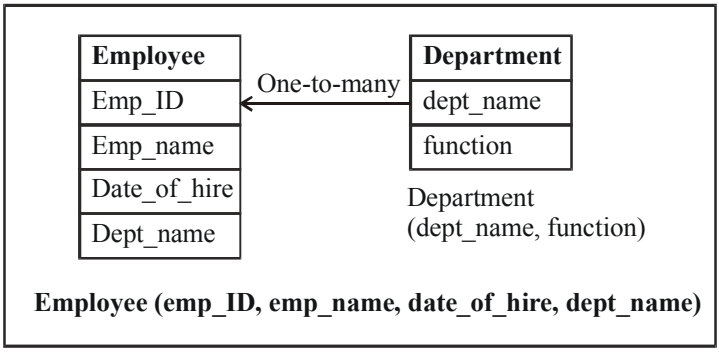
**Object-Oriented Model** – Object-Oriented Database management is based on object-oriented technology. In this database, data is stored as objects and can be interpreted by using the methods specified by its class.
In database design, several views of data must be considered, as the logical view is what the data look like, regardless of how they are stored.
The physical view is the blueprint of database i.e. the way data exist in physical storage. It deals with how data is stored, accessed or related to other data in storage.

**Designing Database Schema** – The schema is the view of data, the overall logical data structure which is held by the DBMS. A database schema is also a set of relation schemas.

The additional case is required while designing the database schema. Important guidelines are
1. Each attribute should be implemented as a field.

2. Each entity should be implemented as a data base table.
3. Appropriate foreign key
4. Each table must have a primary key and an index based on the key.

**For example :** Following figure depicts the database schema for employee and its department.

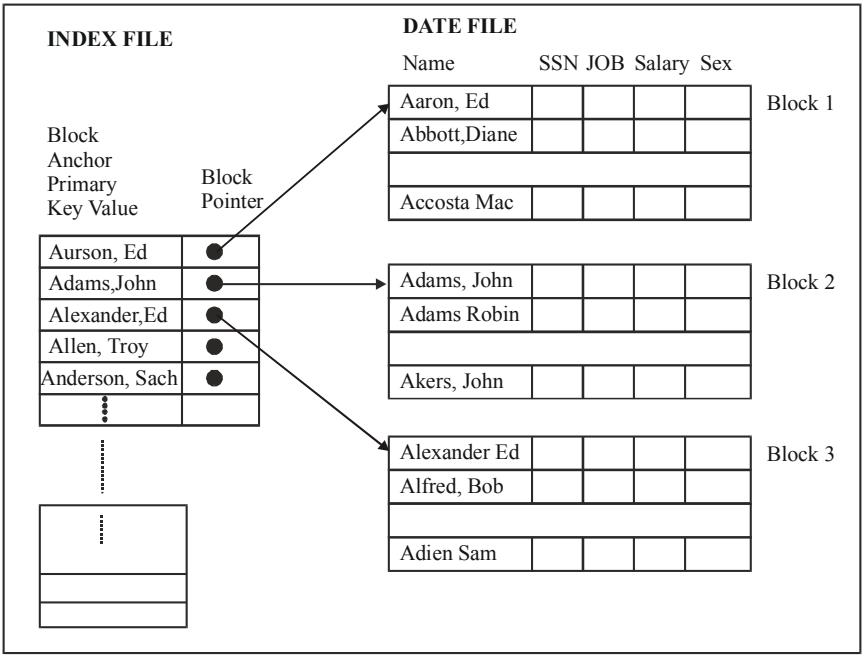



A database schema defines database in terms of tables, keys, indexes and constraints.

The various objects of a relational schema are – Fields, Tables, Records, Database and Constraints.

**Selecting Indexes** – In the process of designing database one must choose based on a single field (primary key) or multiple fields. The section of index directly affect the performance of inserting, deleting and finding records. While indexes can be used generally on tables primarily used for query purpose with rare necessity to update records, they should be used judiciously in tables that support transaction processing which involves large insertion, updation and deletion operation. The amount of time needed to maintain the indexes in database table increases with the number of rows stored.

When an index is created on the table, DBMS, allocated a separate storage area to store the index structure. As a database table can have one or more indexes associated with it.



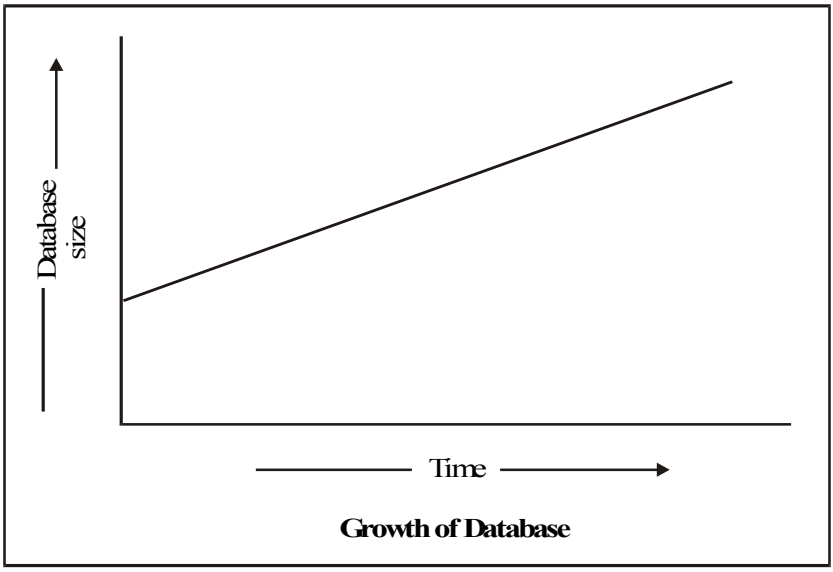

The index table contains a sorted test of primary key (or ID) values. Indexes improve significantly the performance of query. However, many indexes on a table slow down the speed of some inserts, updates and deletes executed on database tables. Indexes also take additional disk space as they are stored separately.

**Estimating Capacity of the Database** – Database capacity or space required by the database for storage is calculated by the database administrator.

**1.** For a given table, calculate the record size by adding the sizes of various fields in it.
**2.** Total number of records that will be present at a particular period of time.

Table size = Record size x Number of records in the table.

However, database size is the sum of sizes of all tables in that database. While designing a database future growth of database should be kept in mind. Most of the business database have a linear growth rate.




**Growth of Database**

# Chapter-10

# CASE Tools For System Development

CASE technologies have a wide range of topics that surround software engineering methods and project management procedures. Computer Aided Software Engineering (CASE) is changing the IT Industry's approach of traditional process of software system development. However, without the use of any CASE tool. The SDLC seems to be flexible, but time consuming and expensive. Keeping in view the various limitations of SDLC process, the computer aided software engineering have emerged to help the organisation in software system development. CASE are basically software packages which provide the engineer an ability to automate manual activation of system development of lifecycle. It forms the building blocks of a workshop of software development. The CASE tools are helpful in many ways such as business systems planning, project management, support, system analysis and design, user interface design, data box design, programming etc.. Today, where quick product delivery is the trend in industry it could be the most challenging task for the software developing organizations, CASE tools provide an absolute helping hand to assist organization to manage and automate the software development process. It is observed that some capabilities are found in every modern software tool. CASE tool have the capabilities to automatically design the requirement analysis, feasibility study, user interface, E-R model and even can auto generate codes etc. Keeping in view of these advantages and demand for software it is very difficult to imagine the life of a programme without CASE tools.

**What is CASE?**

CASE is a Computer-aided Application Software aimed to support one or more software engineering activities in developing, managing and maintaining software. Ideally CASE tool need to support all the activities of software engineering starting from requirement analysis to designing, coding, testing and quality assurance, documentation and implementation. But in reality, CASE tools support are activity or a group of related activities.

Now, software development activities getting more complex and relatively un-

manageable, there has been the need for software engineering tools to provide automated or semi-automated support to help the developers to accomplish the task within the specified time limit. Initially the primary focus was on program support tools such as translators, compilers, assemblers and macro processors etc. With the outgrowth of hardware computer becomes more powerful and the software has grown larger and more complex to run on them, the supporting tool need to expand further to meet the demand.

Use of computer has become the necessity for various activities of any organisation. This leads to the large/medium software projects development process.

A software development effort is highly significant to design appropriate solutions to the problems, test and implement the solutions and finally documenting the solution in order to help developers a wide range of support methodology and tools has begun to emerge.

**Use of CASE Tools by Organizations** – Following are some of the ways in which CASE tools are used :

**1. Single design methodology** – CASE tools facilitate an organisation to standardize the software development process. It helps formatted development and software integration as most of the standardized software follow a common/formal methodology.
**2. Testing** – Testing of developed software is essential. The CASE tools for testing case and improve testing process by automatic testing of the total system with error reporting and automatic recovery from error.
**3. Rapid, quick/fast application development** – It improves the quality and speed of system development.
**4. Reduction of maintenance cost and an increase in productivity** – Maintenance of developed software is required for error correction and enhancements. It has been found that the highest costs are associated with this phase, it is this stage that the CASE tools address bests. As CASE tools automate the various manual activation of software development that results a better productivity of the development team.
**5. Project Management** – Project Management spans the entire life-cycle of a project. It improves project management activities such as skill inventory, network and bar chart, drawing, project costing etc.
**6. Documentation** – Once the design is complete, the code needs to be produced. Most software development organizations spend a substantial amount of time in developing documents and in many cases the documentation process

itself is quite efficient. Organizations spend as much as 20 to 20% of all its software development effort on documentation. Then CASE tool provide an important opportunity to improve the quality and uniformity of documentation at various stage of SDLC.

**Role of CASE Tools** – CASE Tools play a major role in the following activities :

**1.** Project Management
**2.** Schema Generation
**3.** Business System Planning
**4.** Project Scheduling
**5.** Data Dictionary
**6.** Interface Design and Development
**7.** Code Generation
**8.** Reverse Engineering
**9.** Creation of meta-data
**10.** Object-Oriented analysis and design
**11.** Version Control
**12.** Cost estimation
**13.** Documentation
**14.** Prototyping
**15.** Software Testing
**16.** Maintenance

CASE tools has played a significant role in improving the software development process speed, quality and overall productivity. An ideal CASE tool should support all the different phases of SDLC. However, all the aspects of software engineering are not supported by the today tools available in the market. Most of them are good for data modeling, object oriented design and programming, documentation, testing and maintenance of software is moderately supported by these tools.

**Advantages/Benefits of using CASE** – By using CASE tools you can better understand:

1. Corporate and departmental mechanisms and responsibilities
2. The goals of the company and its department
3. The influence of operations in achieving these goals
4. The timeliness and sequence of operations
5. Factors influencing operations and goal achievement
6. Allocation of resources in support of operations. The effect of external in-

fluences on the organization.
7. Problem facing the organization
8. The importance of information relative to the success of the organization.

- Coordinated and integrated development environment – CASE tools provide unique user-interface standards for the analysts and developer. Reducing time and tedious activities of code generation of application interface.

- Guidance in software development – It provides a common platform for all the engineers involved in the development.

- Consistency between the data modeling and documentation – CASE tools automatically generate the documentation leading to consistency between model and documentation.

**Disadvantages of CASE Tools :**

- Project Management – There can be many project problems which are not amenable to automation. Thus CASE tool implementation fails.
- Complex functionality of an organization.

**Components of CASE –** Automation of an activity practically or fully depend on the quality of CASE tools that are used. Most of the CASE tools govern a working model or prototype to make development process easier and faster.

**Types of CASE Tools –** The following are the various type of CASE Tools:

**1. Project Planning and Management Tools** – Tools of this category are available to trace requirements from the original customer request for proposal (REP) to the software development. By using CASE tools, the project manager can create useful estimates of effort, cost, duration of a software project, define a work breakdown structure (WBS) and plan a workable project schedule, and track projects on a continuing basis. The tools can also be used by the manager to collect metrics that will ultimately provide initial software development productivity and product quality.

**2. Analysis and Design Tools** – These tools enable the developer to generate a model of the system to be built and ensures the system requirements. Study being done correctly during the analysis phase in the development process. Analysis and design tools help in the generation of the model and in the evaluation of the model's quality. These tools perform the consistency and validation checking on the model and help to remove errors before they propagate into the design.

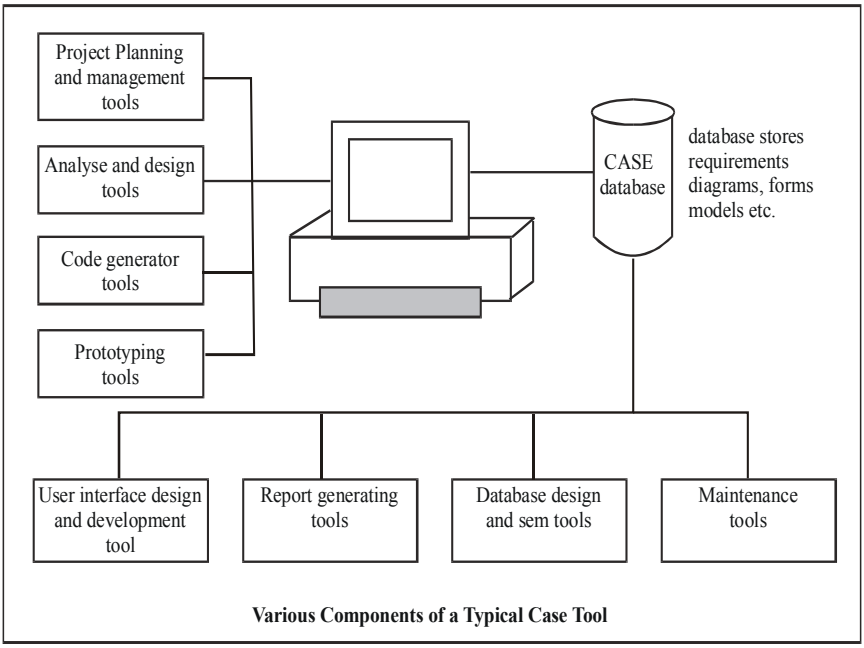
**3. Code Generation Tools/Coding Tools** – Code generation tools translates a system description into an operational program and also help to verify the correctness of the system specification so that the resulting output will confirm to the user requirements.

**4. User Interface Design and Development Tool** – It has been observed that between 50 to 80% of all code generated for interactive applications is generated to manage and implement the user interface design. However, interface design and development tools are actually a tool kit of program components that enable the fast/rapid on screen creation of user interfaces that confirm to the interfacing standard that has been adopted for the software.

**5. Maintenance Tools** –CASE tools for software maintenance take source code as input and generate graphical structured analysis and design models and other design information. These tools help to analyze program syntax, generate a control flow diagram and automatic generate a structured program. CASE tools are very useful to modify a live database systems.

**6. Report Generating Tools** – These tools enable the software engineers to generate reports based on the system requirement specifications.

All CASE tools are based on prototyping, which is widely used in software engineering paradigm. Prototyping is very useful in defining the user requirements. Prototyping tools can create realistic screen images that can be used to illustrate system function and behaviour to the customer.




**Various Components of a Typical Case Tool**

Classification of CASE tools – CASE tools can be broadly classified into four categories:

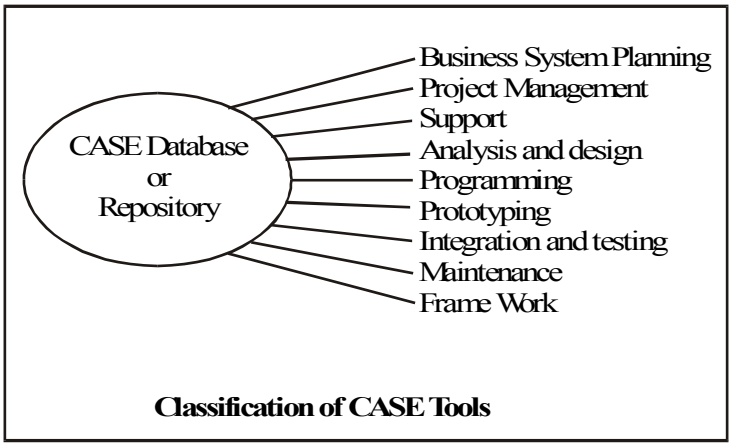
**1. Front-end CASE Tools** – These tools deal with the high level design, specification and analysis of software and requirements. These tools may include computer-aided diagramming tools, oriented towards a particular programming design methodology.

**2. Back-end CASE Tools** – These tools deal with the detailed design, coding, assembly and testing of the software. These tools may aid the programmer directly; for example, they include graphical debugging, aid query and browsing facilities to find quickly a particularly procedure or a variable.

**3. Maintenance Tools** – These tools deal with software after initial release. Then tools provide facilities in tracking bug fixes and enhancement requests, porting to new platform or performing new release.

**4. Support and Frameworks Tools** – These tools provide basic functionality required in tools of type 1, 2 and 3 support software includes basic operating system functionality as well as higher level support such as project management and scheduling software, and data base support to track different ver-

sion and configurations of software releases. Various projects have been directed toward standardizing frameworks to support and integrate CASE applications.




**Classification of CASE Tools**

CASE (computer-aided software engineering) is the use of a computer-assisted method to organize and control the development of software, especially on large, complex projects involving many software components and people. Using CASE allows designers, code writers, testers, planners, and managers to share a common view of where a project stands at each stage of development. CASE helps ensure a disciplined, check-pointed process. A CASE tool may portray progress (or lack of it) graphically. It may also serve as a repository for or be linked to document and program libraries containing the project's business plans, design requirements, design specifications, detailed code specifications, the code units, test cases and results, and marketing and service plans.

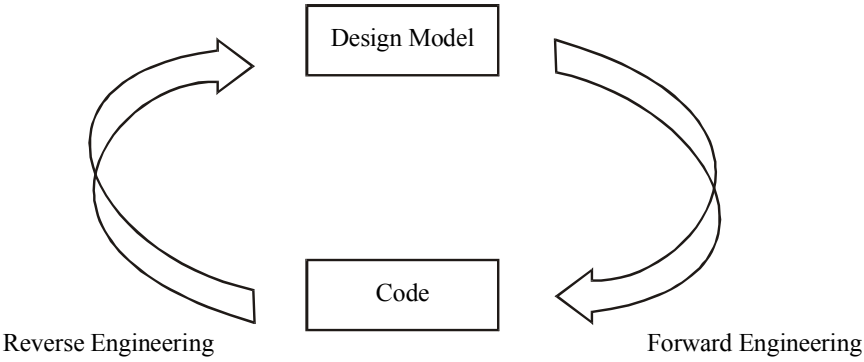
**Following are the three advantages of CASE tools :**

(i) Integrated development environment : CASE tools provide unique user interface for the developer and analyst, automate time consuming and tedious activities like code generation.

(ii) Guidance in development : It provides common platform for all the developers and helps methodical system development.

(iii) Consistency between the model and documentation: Documentation is generated out of the model automatically leading to consistency between the model and documentation.

**Reverse and Forward Engineering :** Below is the Figure depicting both Forward and Reverse Engineering.




Reverse Engineering is the process of recreation of model based on existing code. First, the existing code is scanned to generate the model. Then the model can be fine turned in accordance with requirements. Reverse engineering allows developers to create model for old systems, which were never modeled. It analyses existing software with purpose of understanding its design and specification. Reverse engineering tools read program source code and create graphical and textual representation of design.

Re-engineering efforts make the software up-to-date to current technology and hence easy to maintain. Forward engineering is the process of generation of skeleton code out of the models. First step is to create the model for a system, then generate the relevant code for the model and then allow modification of this code in tune with the requirements.

Re-engineering means "restructuring and rewriting the legacy system or part of it without changing its original functionality". The new system becomes restructured and re-documented. Re-engineering tools read program source code and interactively change and existing system to improve quality performance or maintainability.

# Chapter-11

# Implementation And Maintenance Of System

Implementation of system involves coding, testing, installation and user training. System design specifications are converted to computer programs, and database structure are created. The program are tested using a code walk through and by creating different test scenerios. System testing is the testing of software in its totality after individual modular had been tested.

Different methodology is adopted for conversion/installation like direct conversion, parallel conversion, phased conversion, single location conversion. User documentation is a written document of visual and textual information about the application and how to use it. Well designed user documentation can reduce training cost of the organization. Training of user is vital for success of any system. Training should be conducted after any significant changes are made to the system.

Maintenance of the system after implementation is a major activity considering the total life of the software product. Maintenance is important to make the system current and relevant in a changing organizational environment. Software maintenance is the activity of modifying the software once it is delivered to the customer depending the requirements of the customer or to add additional functionalities of the software. Software maintenance activity in general does not provide any quantifiable benefit to the organization. Different maintenance activities are adaptive maintenance, corrective maintenance, perfecting maintenance, and preventive maintenance. Out of all maintenance activity, corrective maintenance constitutes more than 60% of the total maintenance activity. Different issues related to software maintenance are technical, organizational and procedural.

It has been observed that even the best system cannot show good result if analysts managing the implemenation do not attend to every important detail. This is an area where the systems analysts need to work with utmost care.

**Q1. Explain at least five types of System testing. [JUNE05, Q1(d)]**

**Following are the five types of System testing :**

**(i) Security Testing :** System used for processing sensitive information are prone to high security risks. Individual often tries to access unauthorized data for various reasons. Threats could be external or internal. Hacking of pass-

words is a common problem. Individual can use software to generate random passwords to gain access of the system. Security testing takes care of these aspects of the system security.

**(ii) Recovery Testing :** Test the ability of the system to recover from errors. Errors or any other processing faults must not cause overall system to fail. The recovery time of the system after failure must be within a specific period and tolerance limit. System failures are forced during this phase of testing by introducing exceptions to see how the system responds to the case.

**(iii) Stress Testing :** Stress test is designed to test the system as to how the system behaves in abnormal situation. The aim of the stress test is to find the limit of quantity or frequency of input after which the system fails. Stress test cases are designed which require maximum memory and other resources; in excess of what normal situation demands.

**(iv) Response Testing:** Testing of response time is of special importance in OLTP (on-line transaction processing systems like railway reservation system points of sale, etc.) . Testing is done to measure the response time. The same is compared with desired maximum response time.

**(v) Performance Testing :** It checks the run time performance of the system. It is often coupled with stress testing. Performance testing is specifically important in embedded and real time systems.

**Q2. Briefly explain the four methods of conversion from an existing system to the new system. [DEC05, Q1(d)]**

**Following are the four methods of conversion from an existing system to the new system :**

**(i) Direct Conversion :** This kind of conversion plan is often the least preferred for critical business applications. This is abrupt approach. The old system is shutdown and the new system starts. This kind of conversion although economical, the users are at the mercy of the new system, hence direct installation can be very risky. Some times due to procedural reasons where two systems can't be run parallel, this kind of conversion is the only option. When the new system fails, there is no way to start the old system as a backup as it has been shutdown.

**(ii) Pilot Conversion :** Although this kind of pilot conversion plan is beneficial for the user, it places a substantial burden on the implementation team as it has to maintain two systems in parallel. This is the middle path approach. Instead

of converting all at once through the organization, this kind of pilot installation involves conversion/installation of system at a single pre-decided location. The location may be a branch office of the organization. Proper selection of the pilot site is important as it should be able to perform a true conversion process to test all functionalities of the new system. The advantage of the pilot conversion is that the potential risk in case of failure of the system is limited to a single location. Once the user is ascertained that the implementation of the system has been successful in a particular location, it is proposed to replicate the system in other locations.

**(iii) Parallel Conversion :** Parallel conversion is costly as two systems are run in parallel, but result of only one system are used for business operations but it is least risk prone. Under this kind of conversion, the old system is allowed to run alongside the new system until the management and the end user are satisfied with the result of the new system. It is compared with the new system to test whether the functionalities covered by the old system are thoroughly covered in the new system are not detrimental for normal functioning of the organization as the new system is replaced and normal functions are resumed by the old system.

**(iv) Phased Conversion :** is an incremental approach to switch over to the new system. Different sub-systems of the new system is used in conjunction until the whole new system is converted. This kind of approach for conversion limits the potential risk of failure of the new system. In a phased installation as a sub-system is made functional, actual results are visible before the whole new system is made functional.

Hardware and software selection is an important issue to be considered before actually carrying the conversion. Each conversion strategy not only involves data and software, but also other resources like personnel, hardware, etc.

**Q3. Briefly explain the four key stages of maintenance activity.**

**Four key stages of maintenance activity are as follows :**

**Help Desk –** The problem is received from the user through a formal change request. A preliminary analysis of the change request will be done, and if the problem is sensible, it is accepted.

**Analysis –** Managerial and technical analyses of the problems are undertaken to investigate the cost factors and other alternative solutions. Feasibility Analysis is done to assess the impact of the modification, to investigate alternative solutions, to assess short and long term costs, and to compute the benefit of making the change.

**Implementation –** The chosen change/solution is implemented and tested by the maintenance team. All infected components are to be identified and brought in to the scope of the change. Unit test, integration test, user-oriented functional acceptance tests and regression test strategies are provided.

**Release –** The changes are released to the customer, with a release note and appropriate documentation giving details of the changes.

**Q4. Define various types of maintenance.**

**Different Maintenance Activities –** Maintenance covers wide range of activities, including correcting, coding and design errors, updating documentation and test data, and upgrading user support. Maintenance means restoring something to its original condition. Many activities classified as maintenance are actually enhancements. Enhancement means adding, modifying or redeveloping the code to support changes in the specification.

Maintenance phase focuses on change associated with error correction, adaptations required as the software's environment evolves, and changes due to enchantments brought about by changing customer requirements. Four types of activities are encountered during the maintenance phase :

**1. Corrective maintenance**
**2. Adaptive maintenance**
**3. Perfective maintenance**
**4. Preventive maintenance**

**1. Corrective maintenance** – Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software to correct defects. Corrective maintenance means repairing processing or performance failure or making changes because of previously uncorrected problems or false assumptions. This type of maintenance is to rectify design, coding and implementation problem detected after the implementation of the system. This type of problem needs immediate attention as it hampers the day-to-day work of the end user. Proper planning and interaction with the end user during system development process can minimize corrective maintenance.

**2. Adaptive maintenance** – Adaptive maintenance means changing the program function. Over time, the original environment (e.g. CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software, to accommodate changes to its external envi-

ronment. Unlike corrective maintenance, this kind of activity adds value to the information system and affects a small part of the organization.

**3. Perfective maintenance** – As software is used, the customer/user will recognize additional functions that will provide benefit. Perfective maintenance extends the software beyond its original functional requirements. Perfective maintenance means enhancing the performance or modifying the program(s) to respond to the user's additional or changing needs. Sometimes, changes are made to improve performance of the software.

**4. Preventive maintenance** – Computer software deteriorates due to change and because of this, preventive maintenance, often called software reengineering, must be conducted to enable the software to serve the needs of its end users In essence, preventive maintenance makes changes to compute program so that they can be more easily corrected, adapted and enhanced. Preventive maintenance is done periodically to reduce the probability of system failure. Preventive maintenance could increase the volume of transaction that can be handled by the system. Preventive maintenance is done when the system is least used or not used at all.

**Issues involved in Maintenance** – The responsibility of software development team is not over after the product is released for implementation and installed. If software is not properly maintained, a well-documented and clearly designed system can decay into a poorly documented and ill maintained system. In a network environment, a bug has ramifications beyond just poor performance or functionalities. A bug can open up avenue for a hostile intruder.

It is very important that the software should be easily maintainable. Factors like availability of source code, availability of system manual, etc. are very important for maintainability. One of the most important issues is the cost factor for maintenance of software. Maintenance activity may sometimes introduce new bugs while rectifying it.

Following are various factors which affect the CASE maintenance:
**(i) Volume of Defects** – The inherent errors/bugs that are found in the system after installation. Cost of maintenance increases with the increase in volume of defects.
**(ii) Number of Customers** – More number of customers mean more request for changes in the system after installation.
**(iii) Availability of System Documentation** – The quality and availability of

system documentation is vital to carry out the maintenance. Poorly written system documentation increases the cost of maintenance.

Following are the various issues in software maintenance :

**1. Organizational Issues** – Most of the software maintenance activity is resource consuming as it has no clear qualifiable benefits for the organization.

**2. Process Issues** – Software maintenance requires a number of additional activities not performed at developed stage. Impact analysis and regression test on the software changes are crucial issues.

**3. Technical Issues** – How to construct software that it is easy to comprehend is a major issue and the technology to be this is still not available. Still, the following are some guidelines for the same:

- Make a decision on the best implementation route or to make no change.
- Translate the problem into software terms to decide if it is viable or not.
- Determine the origin of the change request and suggest solution.
- All solutions are investigated to determine that they are applied to all the software components affected.
- Ripple effect propagation is phenomenon by which changes made to a software component along the software life-cycle have a tendency to be felt in other components.

# Chapter-12

# Audits And Security Of Computer System

Every business and organization can experience a serious incident, which can prevent it from continuing normal operations. This can happen any day, at any time. The potential causes are many and varied: flood, explosion computer malfunction, accident, grievous act... the list is endless. Even best designed system cannot control the prevention of natural disaster. In today's ever-changing world of information, it is extremely difficult to keep up on the latest vulnerabilities, virus, patches, trends, technology, hacker's behaviours and activity. Security in any system should be commensurate with its risks. However, the process to determine which security policy and controls are appropriate and cost effective, is often a complex and sometimes a subjective matter. One of the prime functions of security risk analysis is to put this process on to a more objective basis. It's easy for the information system security professional to get caught up in attending the logical aspects of security such as reviewing the log files, making configuration changes, trouble shooting and many other technical duties.

**Q1. What are the objectives of Audit?**

Objectives of Audit are :

**1.** Improvement of Reliability, Security and Efficiency of Information Systems.

**2.** Information System Auditors

**3.** Computer Assisted Audit Techniques

**Q2. Define 'Information System Audit'. List three objectives of Audit. List at least five benefits of Audit. [DEC05, Q5(b)]**

**Information System Audit :** It is the process of collecting and evaluating evidence to determine whether information systems, related resources and the environment adequately safeguard assets, maintain data and system integrity, provide relevant and reliable information, achieve organizational/information system goals effectively, consume resources efficiently, and have in effect internal controls that provide reasonable assurance that operational & control objectives will be met, undesired events will be prevented or detected & rectified in a timely manner. The overall objective of an IS Audit is to ensure control maximization and risk mitigation.

**Following are the three objectives of Audit :**

**(i)** To improve the quality of information systems, prevent failure and minimize the effects of failure, and speed up the process of recovery in the event of a failure. This will help Information System to be more reliable.

**(ii)** To improve the cost performance of an information system by optimum utilization of its resources, which leads to increase in efficiency.

**(iii)** To make an information system more secure from natural as well as manmade disasters, unauthorized access, and other destructive actions.

**Following is the list of benefits of Audit :**

* Improve system and process controls.
* Detect and Prevent errors as well as fraud.
* Reduce risk and enhance system security.
* Plan for contingencies and disaster recovery.
* Manage information & developing systems.
* Prepare for the independent audit.
* Evaluating the effectiveness and efficiency related to the use of resources.
* Cost control.
* Competitive advantage.
* Standardization.
* Improved business efficiency.

**Q3. Explain at least four concurrent audit techniques.[JUNE05, Q5(b)]**

**Following are the four concurrent audit techniques :**

**(i) ITF (Integrated Test Facility Technique) :**

Here, the test data of Auditor is integrated and the same is processed with Client's real life input data. ITF ensures that files of the client are unchanged and any changes, if necessary, will be made only to the dummy files of the client'' files. At the end, these dummy files are studied to know the discrepancies.

**(ii) The Snapshot Technique :**

In this technique, Audit software is embedded in the software that is to be audited. It is embedded at those places where critical processing takes place. Then, it takes a snapshot of the process before and after the critical processing.

**(iii) SCARF (System Control Audit Review File) :**

It is one of the complex Audit techniques. This technique will embed Audit software in the host application. This will enable audit software to monitor the Systems transactions uninterruptedly. The information that is collected dur-

ing Audit process will be stored in a special audit file known as scarf master file.

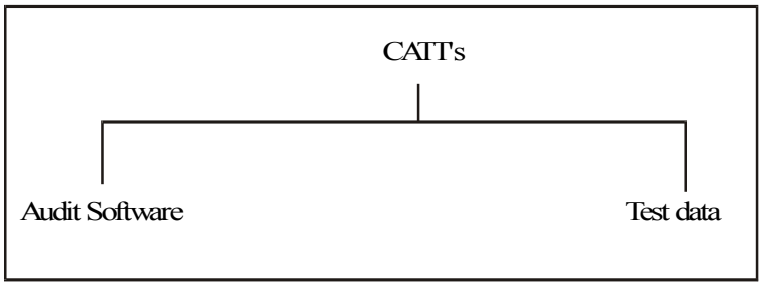
Usually, SCARF is used to collect the following information: Application System errors, Policy and procedural variances, System exceptions, Statistical samples, Snapshots and extended records, Data profiling, Data for performance measurement.

**(iv) CIS (Continuous and Intermittent Simulation technique) :**

This technique will use the Data Base management systems to trap exceptions. Whenever, there is a need for service, DBMS will inform the same to CIS. CIS will then carry out the suitable service.

**Q4. Write a detailed note on CAAT.**

**Computer Assisted Audit Techniques** – The system auditors use various types of automated audit software to carry out information system audit. The use of "computer assisted audit tools" should be controlled by IS Auditor to provide reasonable assurance that the audit objectives and the detailed specifications of the CAAT's have been met.




**Audit Software** – It is a computer program used to process data of significance for audit from E-R accounting system. The auditor should substantiate their validity for audit purposes before making use of these tools. These tools are:

**1. Package Programs** – These are generalized computer programs to perform data processing function like need of files, selecting info, performing calculations etc.

**2. Special Purpose Programs** – These are also some computer programs designed to perform audit tasks in a specific business activity.

**3. Utility Tools** – It is used by the auditors to perform common data processing functions like creating, sorting and printing files. These tools are not specifically designed for audit purpose.

Examples of various Audit Software available in the market are :
**(i)** Visual audit pro
**(ii)** E-Z Audit
**(iii)** IDEA

**Visual audit pro** – It audits automatically over the computer network. Its auditing activities are: User log on/off collection of information about software and its version. Collects information about hardware inventory like serial number, model, memory and associated peripheral devices, user information, registry information etc.

**E-Z Audits** – It collects the information about the capacity of RAM, name of network card with its speed of connection, MAC Address (Media Access Control Address) and TCP/IP information. One can find out how many local, removable and network drives are there on the system, detail of connected printers both local and networked etc.

It gives the detail of the installed software on a system e.g. operation system, with services packs, installed programs and their names, Exe files and DLL versions.

**IDEA(Interactive Data Extraction and Analysis)** – It can be used to impart information from database that is to be audited for further analysis to auditor. It helps to corroborate audit effectively e.g. it can be used to check for duplicate payments on a single invoice. It is used to analyze system log for fraud detection.

The potential of fraud occurrence in a payroll package is very high. The audit software helps to detect fraud for unusually high salary, extracting information without a department number, extracting information on Bank Account number. It can also extract information on fictitious employee, compare it with personnel database. It can also compare payment details for two different months.

**Test Data** – It is used to test the correctness and efficiency of the software. When test data is processed with entity's normal processing system, the auditor should ensure that the test transactions are subsequently eliminated from the system. When using the test data the IS auditors should be aware that the test data should only point out the erroneous processing and should not change the data that is produced by the system during actual operations.

**Audit Expert Systems** - Audit expert systems are also used by some of the IS auditor to assist them in auditing. When using these systems, the IS auditor should

have thorough knowledge of the operations of the system to confirm that the decision path followed are appropriate to the given audit environment.

**Audit Trail** - Audit trail is a system log of changes made in the data, setting and related change i.e. a record showing who has accessed a computer system and what operations he/she has performed during a period of time. Audit trail are useful both for maintaining security and for recovering lost transactions. Most accounting systems and Data management systems include Audit trail components. In addition there are separate Audit trail software products that enable network administrator to monitor use of network resources. Listing of terminals addresses and allocation can be used to look for incorrectly logged, missing or additional materials.

**Q5. What do you mean by Disaster Recovery Planning.**

Disaster recovery (DR) planning is the process of developing advance arrangements and procedures that enable an organisation to respond to a disaster by resuming critical business functions within a defined time frame, minimising loss, and restoring affected areas.

It is not a two-month project, neither is it a project that you can forget about, once it is completed. An effective recovery plan is a live recovery plan. The plan must be maintained and tested/ exercised regularly.

**An effective DR plan consists of the following stages:**

Programme description
Pre-planning activities (project initiation)
Vulnerability assessment and general definition of requirements
Business impact analysis
Detailed definition of requirements
Plan development
Testing programme
Maintenance programme
Initial plan testing and plan implementation.

The primary objective of a business resumption plan is to enable an organisation to survive a disaster and to re-establish normal business operations. In order to survive, an organisation must ensure that critical operations can resume within a reasonable time frame. Therefore, the goals of a business resumption plan should be to identify weaknesses and implement a disaster prevention programme, minimise the duration of a serious disruption to business operations, facilitate effective co-ordination of recovery tasks, and most importantly reduce

complexity of the recovery effort.
Historically, the data processing function alone has been assigned the responsibility for providing contingency planning. Frequently, this has led to the development of recovery plans to restore computer resources in a manner that is not fully responsive to the needs of the business. Contingency planning is a business issue rather than a data processing issue. In today's environment, the effects of long-term operations outage may have a catastrophic impact. The development of a viable recovery strategy must, therefore, be a product not just from the providers of the organisation's data processing, communications and operations centre services, but also the users of those services and management personnel who have the responsibility for protection of the organisation's assets.

**Q6. Explain all the classes of disaster recovery needs of any organization.**
**Following are the four classes of disaster recovery needs of any organization :**
**(i) Critical -** Critical needs are those which are absolutely essential to the running of the business and because of their high complexity cannot be replaced with manual methods.

**(ii) Vital -** These needs are those where the organisation could somehow continue for the better part of a working week, that is 4-5 days or maximum one week but the services must be restored in that time. These can not be replaced by manual operations at all but the organisation could bear with them for some time.

**(iii) Sensitive -** These belong to such operations that are more efficiently performed with the help of computers, but with some difficulty could be performed manually as well. For example typing of some kind of letters and making of indents and orders.

**(iv) Non-Critical -** Non-critical needs are those whose effects are seen over larger time scale of several weeks. If the services are restored over that period, externally die organisation is not much the worse for the breakdown. There would of course, be some loss in money terms, and possibly in terms of the quality of services although it is not likely to lead to total business failure.

**Q7. The concern about quality of software is sometimes referred to as a 'Software Crisis'. Elaborate on the meaning of the phrase and comment on the available approaches and practices to cope with the issue. #**
**Software Crisis**

The software crisis was a term used in the early days of software engineering, before it was a well-established subject. For the last several decades, tens of thousands of people, usually very intelligent and talented have been involved in the building of computer systems. The translation of a familiarity with computer hardware and software into the development of useful commercial or business information systems is not a straightforward or intuitive task. It is now well-known that the rate at which the hardware has been more and more accessible and at lower and lower prices, has created a matching demand for development of software in a similar scale. In essence, it refers to the difficulty of writing correct, understandable and verifiable computer programs. The roots of the software crisis are complexity, expectations, and change.
*Software crisis can be broadly classified in the following major areas :*

**From Programmer's Point of View :**

• Problem of compatibility
• Problem of portability
• Problem in documentation
• Problem in coordination of work of different people where a team is initiating to develop software.
• Problems that arise during actual run time in the organisation. Some time the errors are not detected during sample run.
• Problem of piracy of software.
• Customers normally expand their specifications after program design and implementation has taken place.
• Problem of maintenance in proper manner.

**From User's Point Of View :**

• Problem of different versions of software (user as well as operating system).
• Security problem for protected data in software.
• The customized software generally does not meet his total requirements
• Problem of virus
• Problem of software bugs, which comes to knowledge of customer after considerable data entry
• Certain software run only on specific operating system environment
• The problem of compatibility for user may be because of different size and density of floppy diskettes.
• Problem in learning all the facilities provided by the software because companies give only selective information in manual
• Certain software run and create files which expand their used memory spaces and create problem of disk management.

• Software crisis develops when system memory requirement of software is more than the existing requirements and/or availability.

**Available approaches and practices to cope with the issue :** The main purpose with software reuse is to improve software quality and productivity, and there by maximize a software development organizations profit. The software engineering community has had long-standing high hopes that **software reuse would be the answer to the software crisis**. A number of software reuse approaches have been presented over the years. One example of such an approach is the object oriented programming paradigm (OOP). OOP supports software reuse by techniques known as polymorphism, encapsulation and inheritance. These techniques help the developer in producing highly modular and to some extent reusable code. Much research has also been done on reuse libraries. The basic idea of such traditional software reuse approaches is that organizations create repositories where the outputs of practically all development efforts are stored. These repositories would typically contain components, modules and algorithms that developers are then urged to use. Unfortunately, it usually takes longer to find the desired functionality and adapting it to current needs than it would to build it anew. The typical programmer solution to this problem is to ignore the legacy and build most of the software from scratch. Traditional techniques, which support so called small-grained reuse, have therefore proved ineffective when trying to address the software crisis in practice. Another and more effective approach to software reuse is known as the "clone and own" approach. When a new product project is initiated using this approach, the development team tries to find another product within the organization that resembles the current product as much as possible. The organization then copies (clones) all for more information. One drawback with the clone-and-own approach is however inefficient maintenance. When "cloning" an existing product to create a new product, its maintenance trajectory is split into two separate paths. This could lead to considerable additional maintenance costs for the common parts of the products over their lifespan. Software product lines are about strategic reuse, this means that software product lines are as much about business practices as they are about technical.

# Chapter-13

# MIS

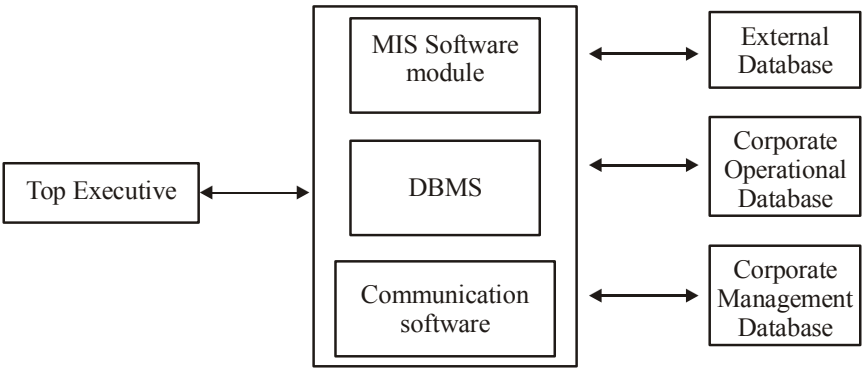
There are many kinds of information systems in the real world which use computer resources (hardware, software and people) to transform data to meaningful information for business needs and decision-making. Every business process depends on information for day-to-day activities and decision making. The concept of management information system (MIS) is not clearly understood by many developers as well as the end users. Of these MIS systems some consider any computer based system be an MIS, some other consider MIS to be a discipline of management education similar to other disciplines like accountably and marketing. Many view MIS as a philosophy of providing help to managers in decision making. Actually, MIS have been playing a key role in helping the managers at various levels of business functions for decision making. In early days of business information system data processing is used to generate various day-to-day reports. In today's world, as business is operating in a more complex and varied environment, managers have realized the need for specialized computer based information system for special activities and business needs. Keeping in view various types of business information systems have developed over a time such as transaction processing systems, management information systems, decisions support systems, expert systems and model management systems.

**Q1. What is MIS ? List at least four characteristics of MIS. Explain the architecture of MIS, with a diagram. [DEC05, Q5(a)]**

**MIS (management information systems)** is a general term for the computer systems in an enterprise that provide information about its business operations. It's also used to refer to the people who manage these systems. Typically, in a large corporation, "MIS" or the "MIS department" refers to a central or centrally-coordinated system of computer expertise and management, often including mainframe systems but also including by extension the corporation's entire network of computer resources. OR It can be said that it is a special kind of information system that helps managers to take decisions. MIS is tailored to provide specific information to individual managers for long term and strategic decision-making. MIS is used by the middle and top-management for their information requirements for decision-making.

**Following are the four characteristics MIS :**
**(i)** Provides reports to management usually in semi-structured format (in detailed, summary, and exception);
**(ii)** Usually uses shared database from many sources;
**(iii)** Often based on management or statistical models;
**(iv)** Information presented in both textual and graphical forms, but more often in graphical format; Provides information on trend analysis, exception reporting and what-if-analysis.

**Architecture of MIS :** The bulk of information requirements of Managers at middle and top levels comes from external non-computer sources like meeting documents, newspaper, telephonic talk, letters, memos, etc. Corporate database are important for day to day operations of the organization.




At the same time, data from external non computer sources provides managers with objective information that helps them to make strategic, long and short term decisions. Various components in the architecture of MIS are shown in Figure above and explained below.

**External Database :** Since organization operates in a social environment it is influenced by various external factors. Impact of theses external factors on the long-term goal and success of organization is very important. External databases are databases that are not owned by the organization and the organization pays royalty to access these database. Examples of these database are: databases of Market research groups, Statistical and Demographic organizations etc.Top management needs to analyse data from these sources for long term planning.

**Corporate database :** Corporate database stores data generated by various business processes through transaction processing Systems. These can be employee database, customer database, inventory database, etc.

**Management database :** These database store selected data from corporate databases. It generally stores summarized information for the requirement of managers.

**MIS Software :** This is used to extract and process information from various databases. It acts as a user interface to the managers.

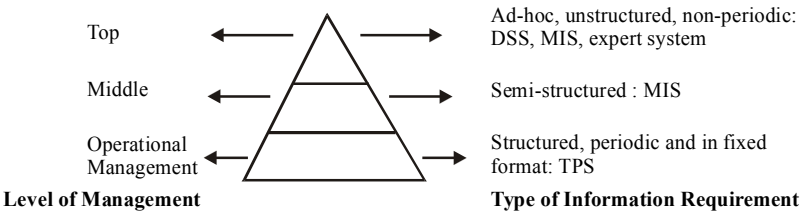
**DBMS:** Database Management System is a system to store, retrieve and manage data.

**Communication Software :** Used to communicate with customers, suppliers and other stakeholders of the organization. Examples are Organization's Bulletin board or Messaging Software.

**Q3. What are the different kinds of Information Systems? Write tabular representation of T.P.S., M.I.S., D.S.S., E.S.**

**Different Kinds of Information Systems**

Depending on the end use, the information systems may be classified broadly to operation information systems and management information systems. Operation information system generally helps to support business operation, whereas management information system helps in managerial decision-making. Transaction processing systems may be classified into Operation information system. Decision support systems, MIS and expert systems may be classified into different forms of MIS for specific purpose. The information requirements of managers are directly related to the position of the manager in hierarchy) ladder as shown below. Keeping in view, various types of information systems have evolved over time.




**Levels of Management and their Information requirements**

The types of information systems to be discussed subsequently have specific focus areas to support an organization's information requirements. Following table depicts various functions of Information Systems :

**Figure : Functions of Information Systems**

| Systems | Transaction Processing Systems | Management Infomation Systems | Decision Support System | Expert Systems |
|---|---|---|---|---|
| Information Source | Process data resulting from business operation | Process data form business operation as well as external data | Use analytical models and specialized database in addition to internal data. | Use knowledge of experts from a specific field |
| Types of Support | Provides support for day-to-day operation of business process | Provides data for managerial decision-making | Provides interactive decision support to manager for decision-making | Provide expert advice on a specific domain of activity |
| Format of Reporting | Periodic and routine type in fixed format | Reports are semi-structured and ad-hoc type | Provides report like sensitivity analysis and what-if analysis | Provides advice like human expert |
| Used by | Operational management | Strategic decision-making for managers | For decision support tailored made to individual managers | Managers for expert advice on a specific field |
| Examples | Sales transaction processing system, on-line railway reservation system | Marketing management information system | Geographic Information System (e.g. IBM's Geo-Manager, which integrates interactive computer graphics with geographic database | Expert system for medial diagnostic (e.g., MYCIN) |

**Batch and Online Transaction Processing**

| | Batch Processing | Online Transaction Processing |
|---|---|---|
| **Process** | Transaction data is accumulated in regular intervals for processing at a scheduled interval | Transaction data is processed as and when generated by the business process |
| **Updation of database** | When the batch is processed | When the transaction is processed |
| **Response time** | Several hours/day | Immediate |
| **Associated cost** | Economical with efficient utilization of resources | High |
| **Example** | Processing pay cheques received for clearance in a banking system | Point of sales terminal, Online Railway reservation system |

**Q4. Define an Expert system. Mention its characteristics. With the help of an example of expert system, explain various components of it.**
**[JUNE05, Q3(b)]**

**Expert System :** It is a computer program that simulates the judgment and behavior of a human expert or an organization that has expert knowledge and experience in a particular field. Typically, such a system contains a knowledge base containing accumulated experience and a set of rules for applying the knowledge base to each particular situation that is described to the program. Sophisticated expert systems can be enhanced with additions to the knowledge base or to the set of rules. The expert system is a knowledge-based information system to act as a consultant to the user. Ex pert systems are being used in many specialized field like medicine, engineering and business. An Expert System in the field of medicine can help diagnose illness. Unlike Decision Support System, an expert System interacts with the user to get input and provides expert advice on a problem in a specific domain.

**Following are the characteristics of Expert System :**

**(i)** The direction of interaction is from machine to the user.
**(ii)** Replaces a human advisor/expert for specific domain of knowledge;
**(iii)** Its domain of knowledge is narrow;
**(iv)** Has reasoning and explanation capability;
**(v)** Types of problem treated is repetitive; and
**(vi)** Captures knowledge and expertise of a problem solver or decision maker and simulates thinking for those with less knowledge;

**Various Components of Expert System :**
Coming Figure shows the components of an expert system. There are two major components knowledge base and software module called inference engine.

**THE KNOWLEDGE BASE**
The main purpose of the knowledge base is to provide the guts of the expert system—the connections between ideas, concepts, and statistical probabilities that allow the reasoning part of the system to perform an accurate evaluation of a potential problem. Knowledge bases are traditionally described as large systems of "if then" statements, but this description is misleading because knowledge bases may not contain definitive rules at all, but may contain only associative relationships among different concepts, statistical information about the probability of certain solutions, or simply large databases of facts that can be compared to one another based on simple conventions intrinsic to the expert system.

**THE USER INTERFACE**
For the last several years, interface designs for expert systems have hinged on graphical capabilities and unconventional methods of entering data into the system. **For example,** many expert systems used a mouse for data entry well before the Macintosh became popular. Graphical interfaces can supply information in any number of forms: simple text "dressed up" in windows, pop-up menus, or actual graphical objects. Recently, many of those formats have been integrated into conventional applications, but they are of particular use in expert systems. An expert system may express an idea, solution, or explanation using more complex conventions than rows of numbers, pie charts, or brief messages.

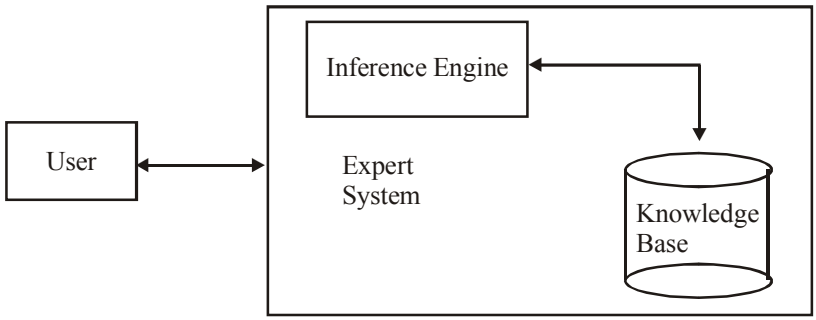



**Figure : Components of an Expert System**

**Q5. Compare Batch and Online Processing.**

| | **Management Information Systems (MIS)** | **Decision Support Systems (DSS)** |
|---|---|---|
| **Source of information** | Operational data/external database | Analytical models/external database and operational database |
| **Structure of Information** | Periodic and often in fixed format | Interactive inquiry and Response to support |
| **Target** | Support group decision-making by managers | Tailored to decision-making style of individual managers |

**Q6. Write a note on MIS as a Profession**

After going through the spectacular growth in the information Technology one might tend to feel that a good computer scientist can make a good MIS professional. Quite often such an assumption is not true. Developing information systems call for people interaction, organisational understanding and inter personal skills that are generally not the thrust areas of the computer science professional. Keeping this view in mind ACM Curriculum Recommendation clearly outlines a separate profession with a mastery of organisation, viz. the Information System Executive. There are various types of management information systems. Mason and Swanson (1981) describe four categories of management information systems:

(1) databank information system,

(2) predictive information system,

(3) decision-making information system, and

(4) decision-taking information system. The classification is based on the level of support that the information system provides in the process of decision making.

**Q7. Discuss the characteristics of an effective management Information System.**

**Following are the characteristics of an effective management Information System :**

**(1) Flexibility:** All organization are dynamic and changes occur for a wide range or reason. A good MIS must be able to adapt to meet their changes flexibility in the way the system is designed is crucial.

**(2) Reliability:** Reliability is crucial to performance and can be ensured only by thorough checking and testing. Good standards help to make the **MIS** reliable as do validation and security routines.

**(3) Simplicity:** Simplicity in design will find its way through the system any one can design a complicated system but it takes real skill and experience to design simple systems which are easy to operate and control. Systems are abstract and therefore not apparent in complicated **MIS**.

**(4) Economy:** The **MIS** should be cost effective there are many hidden costs in the design, development and operation of system the most important of which is the time of the people involved.. Costs should be carefully monitored once the system is working and compared with the original planned costs.

**(5) Help fullness:** Unless the **MIS** helps in the planning operation and control of the business, it is not upto the mark.

**(6) Consistency:** Information system should be consistent. There must be a link between all the data. Data should be collected by the same method and should be presented at equal intervals, e.g. It the data regarding sales of a particular concern is collected in lakh tons in one year then it should not be collected in term of units for the next year. Thus there must be consistency in all the information given.

**(7) Management Oriented:** It means that the development of the information system should start form an appraisal of management needs and overall business objectives such as system is not necessarily for top management only it may also meat the information requirements of middle level or operating levels of management as well.

**(8) Management Directed:** Because of management orientation of MIS, it is necessary that management should actively direct the system's development efforts. Therefore one time involvement is not enough for system's effectiveness, it is necessary for management to devote their sufficient time not only at the stage of designing the system but for its review as well, to ensure that the implemented system meets the specification of the designed system in brief, management should be responsible for setting system specification and it must play a key role in the subsequent trade off decisions that occur in system development.

**(9) Integrated:** Developed system of information should be an integrated one. The word integration here means telling a compressive view or a complete picture look at the inter locking sub system that operate within a company.

**(10) Common Data Flows:** It means the use of common input processing and output procedures are media whenever possible. Data is captured by system analysts only once and as close to its original sources as possible. They then try to utilise a minimum of data processing procedures and subsystem to process the data and string to minimise the number of output documents and reports produced by the system. This eliminates duplications in data collection documents and procedures. It also avoid duplication simplify operations and produce an efficient information system. However, some duplication is necessary in order to insure effective information in system.

**(11) Heavy Planning Element:** An **MIS** usually take 3 to 5 years and sometime even longer to get established firmly within a company therefore, a heavy planning element must be presented in **MIS** development. It means that **MIS** designer should keep in view future objectives and requirement of firms information in mind. The designer must avoid the possibility of system obsolescence before the system gets into observation.

**(12) Sub-System Concept:** Even though the information system is viewed as a single entity it must be broken down into digestible subsystems which can be implemented one at a time by developing a phasing plan.

**(13) Common Data Base-** Data base holds, the functional system together it is defined as a 'superfile' which consolidates and integrates data records formerly stored in many separate data files. Although it is possible to activate the basic objectives of **MIS** without a common data base, thus paying the price of duplicate storage and duplicate file updating more often than not the common database is a definite characteristic of **MIS**.

**(14) Computerised:** It is possible to have a **MIS** without using a computer but its use increases the effectiveness of the system in fact its use equips the system to handle a wide variety of application by providing quickly their information requirement, other necessary attributes of the computer to **MIS** are accuracy and consistency in processing data and reduction in clerical staff.

**(15) Relevance:** The information given to each manager should be relevant to his responsibilities. The flow of information should be from one level to another in the direction where the decision is to be taken. If information is being sent from top level to the middle level management then it should be sent to the proper responsibility centre so that the decision taken can be implemented.

**(16) Brevity:** Information should not only be clear but should also be brief. Brevity does not mean that certain matters be left out but it means that maximum information should be communicated in minimum words, graphs, charts, tables, figures and other such media helps in making the information brief.

**(17) Accuracy:** Information should be accurate as for as possible and if not, then the level of in accuracy should be within limits, for example in reports emphasis is laid on the accuracy of the reports. Delayed submission increases its costs.

**Q8. What are different types of MIS systems that will be in operation in any organization? Explain them.**

There are various types of management information systems. Mason and Swanson (1981) describe four categories of management information systems:

**(i)** databank information system,
**(ii)** predictive information system,
**(iii)** decision-making information system, and
**(iv)** decision-taking information system. The classification is based on the level of support that the information system provides in the process of decision making.

**(i) Databank Information System :** The responsibility of this information system is to observe, classify, and store any item of data which might be potentially useful to the decision maker. Examples of the kind of data that might be recorded in such a database for a given village, region, or area are as follows:

· Number of farms
· Number of units of arable land (hectares, fedans, acres)

· Average farm size
· Amounts of selected farm inputs applied annually
· Production per year on a unit of land for selected crops

**(ii) Predictive Information System :** This system moves beyond pure data collection and the determination of trends over time. Predictive information systems provide for the drawing of inferences and predictions that are relevant to decision making. If data from the above examples were to be used in this way, it is possible to obtain information useful for making predictions or for drawing inferences. For example, tables containing the following information for a given village, region, or area might be produced:
· The ratio between the number of farms and the various categories of extension staff members
· The ratio between the amount of farmland and the various categories of extension staff members
· Amount of extension financial operating resources allocated per year to selected farmer problems or concerns
· Amount of extension financial resources, both salary and operating expenses, allocated per year to selected extension approaches to solving different farmer problems or concerns

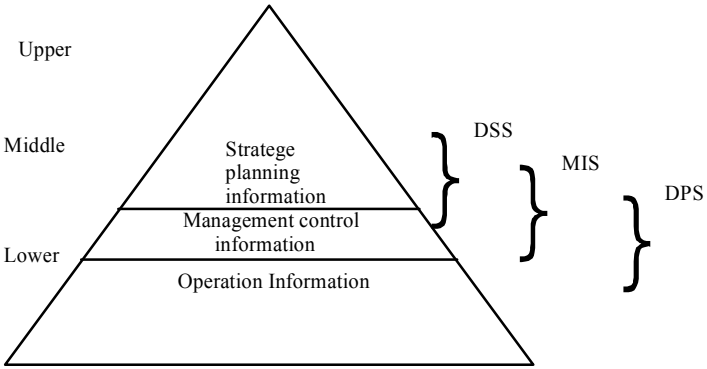
**(iii) Decision-Making Information System :** This system goes one step further in the process of decision making and incorporates the value system of the organization or its criteria for choosing among alternatives. An extension organization's values are many and varied. They include concerns for resolving farmer problems, increasing and providing for stability of farmer incomes, and improving the quality of farm life. But they also include an intent to provide well for staff members (training, adequate salaries, etc.) and to aid in the process of bringing about rural economic development.

**(iv) Decision-Taking Information System :** Examples of decision-taking information systems are not usually found in an extension organization. This is a decision system in which the information system and the decision maker are one and the same. Management is so confident in the assumptions incorporated in the system that it basically relegates its power to initiate action to the system itself. Airplanes carry automatic pilot systems, which are an example of a decision-taking system. Once activated, the system itself keeps the plane on course and at the proper speed and altitude (according to parameters determined by the pilot). Another example of decision-taking information systems is found in modem factory production. In automobile production, continuous inventories of parts are maintained by computer as cars move down an assembly line. Orders are placed

automatically by the computer when additional parts are needed. This is done without the intervention of a manager.

**Q9. How are MIS and DSS related? How do they differ from each other?**
It is viewed that DSS is an extension of MIS or can be defined as a second generation of MIS =>MIS is generated when we add predefined managerial reports that are span out of transaction processing, report generation and online inquiry capabilities - all integrated with a given functional area such as production or personnel MIS => DSS results from adding external data sources, accounting and statistical modules and interactive query capabilities, the outcome is system designed to serve all levels or management.

**Management level** **Information level System support**




| | CATEGORY | CHARACTERISTICS |
|---|---|---|
| 1. | Management Information System | Provides input to be used in the managerial decision process. Deals with supporting well Structured decision situations. Typical Information requirements can be anticipated. |
| 2. | Decision Support System | Provides information to managers who must make judgements about particular situations. Support decision makers in situation that are not well-structured. |

**Difference between MIS and DSS :**
The MIS has the characteristics of a decision support system. The decision support system (DSS) concept is an evolution of the term MIS. However, the DSS focuses more on support leading to decisions.

The term DSS became popular in late 70's due to the wide and diverse computer applications. Operations carried out under DSS may range from retrieving information, analyzing and aggregating data, evaluating the consequences of decisions, and proposing decisions.

The distinction between MIS and DSS is basically a difference in focus. With the DSS the focus is on top decision makers to provide them with sound judgements and visions. On the other hand, MIS is focused on the flow of information, regardless of the user.

The DSS is a part of a larger concept where computer-based knowledge is used to make decisions. The computer operations and their linkages to data and decisions.

**Q10. Describe the significance of decision support systems.**

It is an information system that offers the kind of information that may not be predictable, the kind that business professionals may need only once. These systems do not produce regularly scheduled management reports. Instead, they are designed to respond to a wide range of requests. It is true that all the decisions in an organisation are not of a recurring nature. Decision support systems assist managers who must make decisions that are not highly structured, often called unstructured or semi-structured decisions. A decision is considered unstructured if there are no clear procedures for making the decision and if nor all the factors to be considered in the decision can be readily identified in advance. Judgement of the manager plays a vital role in decision making where the problem is not structured. The decision support system supports, but does not replace, judgement of manager.

Decision support systems are intended to help individual managers in their decision making capability, involving generally the decisions belonging to the non-programmed category. Such system would need access to the large information generated by office automation and transaction processing systems. However, what they need is mainly the summary information, exception information or some patterns and trends. To generate such summarised information like trends, patterns, exceptions etc. they may need access to analytical models. Since the decisions are non-programmed the exact report, formats and the contents of decision support systems cannot be decided 'a priori'. The end users would, therefore; need tools that enable them to generate on the fly reports with case in a flexible manner. Since the end users are likely to be conversant with but not skilled in programming, the interaction will have to be at the higher level of abstraction. Fourth Generation Languages, Menu Driven Packages, use of point and clock or pull down menu based

access, form based database access (with a flexibility to paint screens easily with different data, format etc.) are likely to be more easily acceptable.

**Q11. Compare Decision Support Systems with Knowledge Based Systems. Describe the scope of each of them.**

| | DSS | KBS |
|---|---|---|
| Objective | Assist human | Copy human behaviour |
| Decision maker | Human | System |
| Major orientation | Decision making | Transfer of expertise |
| Query direction | Human asks machine | Machine asks human |
| Manipulation | Numerical | Symbolical |
| Problem area | Complex, wide | Deep but narrow |
| Database | Facts | Facts & procedural knowledge |

**Scope of DSS and KBS :** Systems that support decisions are simply the Decision Support Systems (DSS). DSS represents both a philosophy and a set of tools. At the philosophical level, it is the next stage of evolution of EDP & MIS. While EDP concentrated on accurate and timely processing of data. DSS goes one step further in emphasizing the organisation and presentation of information in a manner that has direct relevance to the supporting phase of a decision maker. Here the goal is to improve the quality of decision making through information. Naturally the issue of access and flexibility are also important in organising the information. While in knowledge base system, Knowledge Based Systems goes beyond the decision support philosophy to indicate the expert system technology into the decision making framework.

A more deeper concept that also gets emphasized in DSS philosophy is the recognition of the central role played by decision maker - a human manager. While Expert Systems (ES) have been the tools and the techniques perfected by artificial intelligence (AI) researchers to deduce decision influences based on codified knowledge. The codification of knowledge uses the principles of knowledge representation (part of the large theoretical ideas of knowledge engineering). Typically such codification uses rules like IF-THEN rules to

represent logical implications. Using first order predicate calculus, it is fairly easy to construct inference engines that use forward chaining or backward chaining to perform the induction.

There are several techniques of decision support. All of them involve an innovative integration of 'what-if' capabilities with data processing/information processing systems. One of the technique that evolved as part of the DSS growth is the influence diagram. This is a graphical tool to document the decision process in the form of logical diagram. It also allows an excellent top-down version of the documentation process of the logic. Using very simple graphical symbols to indicate goals, decisions and outcomes, influence diagrams capture the essence of the decision maker excellently.

**Q12. Define the following terms in detail ?**

**(i) Warnier-Orr Diagrams**

**Warnier/Orr diagrams** are very powerful design tools and offer some distinct advantages to system experts. They are quite simple in appearance and easy to understand. Also known as logical construction of programs/logical Construction of systems are powerful tools aimed at designing of program structures by identifying the output and processing results and then working backwards to determine the steps and combinations of input needed to produce them. The simple graphic methods used in Warnier/Orr diagrams make the levels in the system evident and movement of the data between them vivid. Warnier/Orr diagrams clearly show the various processes and sequences in which they are performed. Each process is defined in a hierarchical way. At each level, the process is shown in a bracket below that groups its components. Since a process consists of different subprocesses, a Warnier/Orr diagram employs a set of brackets to indicate each level of system clearly. Brackets denote set and subsets. Iteration structure (called repetition in the Warnier-Orr notation) is depicted by a parenthesis to the left of the series of elements to be repeated(see diagram next). A Number inside the parentheses indicates the amount of times the iteration should be performed. Thus we will repeat the four bracketed elements until there are no more bicycles to assemble.

**Warnier-Orr diagram for bicycle assembly.**

$$\text{Process} \quad \left\{ \begin{array}{l} \textit{Open Container} \\ \textit{Stock Parts} \\ \textit{Assemble Wheels} \\ \textit{Finish Assembly} \end{array} \right\}$$

$$(0, \text{End})$$

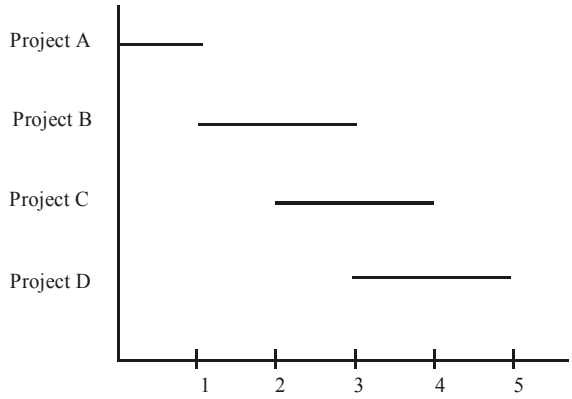
**BRIEF DESCRIPTION OF PERT AND CPM**

Prior to the development of PERT and CPM, the most popular technique for project scheduling. These Charts show a graphical representation of work on a time scale. The primary limitation of this technique is its inability to show the inter-relationships and interdependencies among the many activities which control the progress of the project. Although it is possible to redraw the chart to show the interrelationships, the confusion arises as the size of the project increases. To overcome such limitations, PERT and CPM were proposed.

Program Evaluation and Review Technique or PERT and Critical Path Method or CPM are two of the most widely used techniques in project management. The objectives of project management can be described in terms of a successful project which has been finished on time, within the budgeted cost and to technical specifications which satisfy the end users.

**A project is any human undertaking with a clear beginning and a clear ending.**

Planning, scheduling and controlling the work during any worthwhile project is the main task for any project manager. Project planning calls for detailing the project into activities, estimating resource requirements and time for each activity, and describing activity inter-relationships. Scheduling requires the details of starting and completion dates for each activity. Control requires not only current status information but insight into possible trade-offs when difficulties arise. Normally for any project, we may be interested in answering questions such as: (i) when do we expect the project to be completed; (ii) if any activity is delayed, what effect will this have on the overall completion time of the project; (iii) if there are additional funds available to reduce the time to perform certain activities, how should they be spent, and (iv) what is the probability of completing the project by the scheduled date.




**Q13. Differentiate between PERT & CPM. Also describe some of the key concepts necessary for proper understanding of network.**

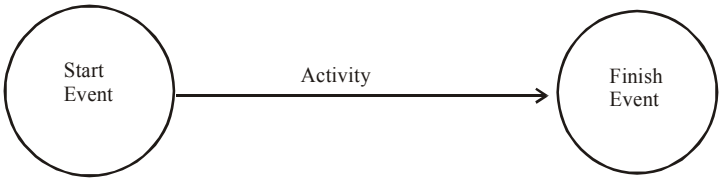
The **difference** between PERT and CPM arose primarily because of the original job for which each technique was developed. Initially the PERT technique was applied to research and development projects while the CPM was used towards construction projects. Both of them share the notion of a critical path and are based on the network analysis that determines the most critical activities to be controlled so as to meet completion dates. However, since the use of either technique is based on individual characteristics, the **main difference** is the PERT is particularly useful for non-repetitive and complex projects in which time estimates are uncertain. CPM is best utilised for repetitive and non-complex projects when time estimates can be made with some measure of certainty. The selection of the technique depends on the degree of uncertainty associated with time estimates and the cost of missing time estimates. Actually PERT restricted its attention to the time variable whereas CPM included time-cost trade-offs. For our purpose we will not differentiate between the two techniques but we can further define PERT and CPM as the process of employing network techniques to optimise the use of scarce project resources for the development of the software.

**For proper understanding of network, it is necessary to define some of the key concepts.**

*Activity:* All projects may be viewed as being composed of operations or tasks called activities, which require the expenditure of time and resources for their accomplishment. An activity is depicted by a single arrow ($\rightarrow$) on the project network. The activity arrows are called arcs.

*Event:* An event represents a specific accomplishment in the project and takes place at a particular instant of time, and does not, therefore, consume time of

resources. An event in a network is a time oriented reference point that signifies the end of one activity and the beginning of another. Events are usually represented in the project network by circles (±). The event circles are called nodes. Therefore, the major difference between activities and events is that activities represent the passage with event nodes as shown below:

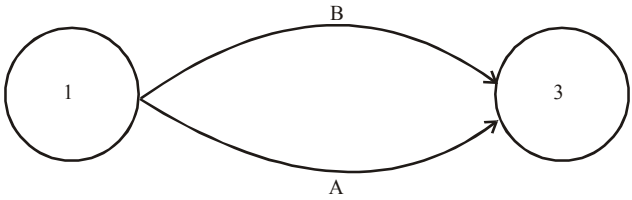



**Predecessor activity:** activities that must be completed immediately prior to the start of another activity are called predecessor activities.

**Successor activity:** Activities that cannot be started until one or more of the other activities are completed, but immediately succeed them are called successor activities.

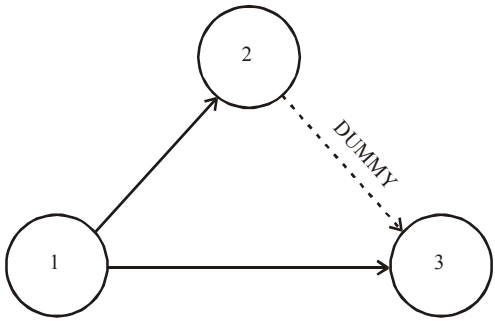
**Concurrent activity:** Activities which can be accomplished concurrently are known as concurrent activities. It may be pointed out that an activity can be a predecessor or a successor to an event or it may be concurrent with one or more of the other activities.

**Dummy activity:** In most projects many activities can be performed concurrently or simultaneously. It is possible that two activities could be drawn by the same beginning and end events. In situations where two or more activities can be performed concurrently, the concept of dummy activity is introduced to resolve this problem. Therefore there will be only one activity between two events. As a result of using the dummy activity, other activities can be identified by unique end events. Dummy activities consume time or resources. By convention, dummy activities are represented by a dashed arrow on the project network. In the following diagram, both activities A and B have the same beginning and end events:




The above network diagram is incorrect because it breaks the rule of assigning unique numbers to each activity for the purpose of identification. The following

network diagram demonstrates the principle of using a dummy activity for overcoming the problem of parallel activities with identical start and finish events.




Therefore, a dummy activity is created to make activities with common starting and finishing events distinguishable, and also to identify and maintain the proper precedence relationship between activities.

**1** Each activity is represented by one and only one arrow in the network. Therefore, no single activity can be represented twice in the network.

**2** No two activities can be identified by the same beginning and end events. In such cases, a dummy activity is introduced to resolve the problem.

**3** Two events are numbered in such a way that the event of higher number can happen only after the event of a lower number is completed.

**4** Dangling must be avoided in a network diagram. This happens when precedence and inter-relationships of the activities are not properly identified.

**5** To ensure the correct logical sequence and inter-relationships, one has to answer the following questions satisfactorily.

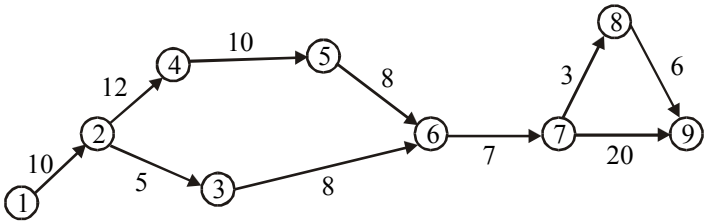
(i) Which activities precede this activity? That is, what other activities must be completed before this activity can be started?

(ii) Which activities follow this activity? Or, what activities cannot be started until this activity is completed?

(iii) Which activities can take place concurrently with this activity? Or, what activities can be worked on at the same time when this one is being performed?

The main determinant of the way PERT and CPM networks are analysed and interpreted is whether activity time estimates are deterministic or probabilistic. If time estimates can be made with a high degree of confidence so that actual time will not differ significantly from estimates, we say the time estimates are deterministic. On the other hand if estimated times are subject to variation, we say the time estimates are probabilistic. First, we shall describe the analysis of network with deterministic time estimates and at a later stage with probabilistic time estimates.

**Q14. Define critical path. Calculate the earliest start time and latest completion time for each activity. [7]**




**Critical Path ⇒** It is the longest path through the network. No task on the critical path can be held up without delaying the start of the next task and ultimately, the completion of the project.

**Earliest Start Time ⇒** To calculate earliest start time (Ej) for each event j, the following algorithm :

Step 1 : Set $E_1 = 0$ for the starting event.

Step 2 : SEt $E_j = Max_i (E_i - t_{ij})$, where maximization occurs over all events i that are immediate predecessors of event j.

| Event No. | Rule and Value |
|---|---|
| 1. | $E_1 = 0$ |
| 2. | $E_2 = E_1 + t_{12} = 0 + 10 = 10$ |
| 3. | $E_3 = E_2 + t_{23} = 10+15 = 25$ |
| 4. | $E_4 = E_2 + t_{24} = 10 + 12 = 22$ |
| 5. | $E_5 = E_4 + t_{45} = 22 + 10 = 32$ |
| 6. | $E_6 = \text{Max} \{(E_5 + 6_{56}), (E_3 + t_{36})\}$ <br> $= \text{Max} \{(32 + 8), (15 + 8)\}$ <br> $= \text{Max} \{40, 23\} = 40$ |
| 7. | $E_7 = E_6 + t_{67} = 40 + 7 = 47$ |
| 8. | $E_8 = E_7 + t_{78} = 47 + 3 = 50$ |
| 9. | $E_9 = \text{Max} \{(E_8+t_{89}), (E_7+t_{79})\}$ <br> $= \text{Max} \{ (50 + 6), (47 + 20)\}$ <br> $= \text{Max} \{56, 67\} = 67$ |

**Latest Complete Time ⇒** In network calculations, a backward pass is also made to determine the latest time each event can begin without delaying the completion of the project. The algorithms is :

$L_n = E_n$, where n represent the last event in the network.

$L_i = \text{Min}_j (L_j - t_{ij})$, where the minimization occurs over all events j that are immediate successors of event i.

Rule and Value

$L_9 = E_9 = 67$
$L_8 = L9 - t_{89} = 67 - 6 = 61$
$L_7 = \text{Min } \{ (L_8 - t_{78}), (L_9 - t_{79})\}$
$= \text{Min } \{ (61-3), (67-20)\}$
$= \text{Min } \{58, 47\} = 47$
$L_6 = L_7 - t_{67} = 47 - 7 = 40$
$L_5 = L_6 - t_{56} = 40 - 8 = 32$
$L_4 = L_5 - t_{45} = 32 - 10 = 22$
$L_3 = L_6 - t_{36} = 40 - 8 = 32$
$L_2 = \text{Min } \{( L_4-t_{24}), (L_3 - t_{23})\}$
$= \text{Min } \{ (22 - 12), (32 - 5)\}$
$= \text{Min } \{10, 27\} = 10$
$L_1 = L_2 - t_{12} = 10 - 10 = 0$

**Q15. What is Structured Design? What are the major System Design activities? Explain and support them with appropriate examples.**
**STRUCTURED DESIGN :** Structured design is a data flow based methodology. The approach begins with a system specification that identifies inputs and outputs and describes the functional aspects of the system. The specifications then are used as a basis for the graphic representation. The next step is the definition of the modules and their relationships to one another in a form called a structure chart, using a data dictionary and other structured tools.
Major development activities are carried out during structured design. They are database design, implementation planning, system test preparation, system interface specification, and user documentation.

**Major System Design Activities are defined below :**
**a) Data base design:** This activity deals with the design of the physical database. A key is to determine how the access paths are to be implemented. e.g. personnel file & payroll file maintenance

**b) Program design:** In conjunction with database design is a decision on the programming language to be used and the flowcharting, coding, and debugging procedure prior to conversion. The operating system limits the programming languages that will run on the system.
e.g. Languages FORTRAN, COBOL, C, BASIC & DBASE

**c) System and program test preparation:** Each aspect of the system has a separate test requirement. System testing is done after all programming and testing are completed. The test cases cover every aspect of the proposed system, actual operations, user interface and so on. System and program test

requirements become a part of design specifications- a pre requisite to implementation. e.g. discovering Logic & Syntax errors & various paths.

**Q16. What is a Gantt chart? What are the limitations of a Gantt chart?**

Gantt chart (developed by Henry L. Gantt) is a project control technique that can be used for several purposes, including scheduling, budgeting and resource planning. It is used almost exclusively for scheduling purpose and therefore controls only the time dimension of projects. It is also known as "Bar-chart" in which each bar representing an activity. The bars are drawn against a time line. The length of each bar is proportional to the length of time planned for activity.

Gantt charts can be used for resource allocation and staff planning. They can take different forms depending on their intended use. They are best for resource scheduling, while, they show the tasks and their duration clearly, however, they do not show intertask dependencies plainly. When a Gantt chart is used as a project control method, milestones or checkpoints usually are placed at the completion of each task (that may also be placed within tasks). They indicate the completion of a particular task and are the basis for determining whether the task and the project are on schedule; when a check point is reached, the task just completed and the entire project are reviewed and evaluated.

**Limitations of Gantt chart (Bar chart) :-** A Gantt chart helps in scheduling the activities of a project, but it does not help in identifying them. Once can begin with the activities identified in the work breakdown structure, during the scheduling activity, and also during implementation of the project, new activities may be identified that were not envisioned during the initial planning. The manager must than go back and revise the break down structure and the schedules to deal with these new activities. However, because the bar chart incorporates only the scheduling dimension of a project it gives little indication of which task must be completed before others are begin, and project costs must be accumulated and evaluated using other methods.

**Q17. What are various methods that are commonly used to verify data entering the system as input? Explain two of them in brief.**

If incorrect data enters the system, it is usually very costly to make the necessary corrections. Also, how expensive would it be to have your operator record a quantity of 100 rather than 10 for a shipment of color TVs? The shipping charges for sending the TVs to the customer and then of having them returned would be only one of the costs. While the 90 extra TVs were in transit, they would not be available to other customers (which could result in a loss of sales) or could be damaged. There are many methods which are

commonly used to verify data entering the system as input. Some of the them are :

**(i) Key Verification**

A second operator rekeys the data already recorded. This method is used for verifying data recorded in Punched cards or on diskettes and magnetic tape. Then two floppies are compared to correct record by record which mismatched during comparison after verifying, from the original documents. This is most effective method used by Computer Service bureaus for data validation.

**(ii) Use of Self-checking Numbers**

The computer can be programmed to reject numbers that have been transposed or have one or more wrong digits. Check digits and self-checking number routines can be effectively used for numbers in a series, such as student roll numbers, account numbers, part numbers. Or invoice numbers are popular for such jobs.

**(iii) Visually Displaying an Identifying Characteristics**

When using a terminal, a part number is entered. Displayed in the VDT is the description of the part, which is then visually confirmed by the operator.

**(iv) Hash Totals**

Sometimes numbers are added to produce a meaningless total called a hash total. For example, totaling is made of the quantity of all items purchased. When the records are entered and processed, the hash total is compared to the original total. If the two totals agree, it is an indication that all quantities were entered correctly and all records were processed.

**(v) Checking Between a Range of Numbers**

The numbers on the orders being processed on a given day should fall between, say, 4999 (the last number from the previous day) and 6001 (the next order number that will be on all of the orders processed by the next day). If the order number recorded on the input record does not fall within that range, an error message will be generated.

**(vi) Reasonableness Test**

Based upon past history, some input can be checked to see if it is reasonable. For example, because of long-standing company policy, it is unlikely that any employee will have more than 20 hours of overtime. If more than 20 hours of overtime are recorded in an employee's current transaction record, an error message will be generated as the data is being edited.

Similarly in 'Date of Birth' field, it is checked that no date is more than 31,

month number is not more than 12 and the year is not more than the current year or current year minus minimum age prescribed.

**(vii) Verification of Codes**

The pay and fringe benefits are calculated for employees based upon their payroll status. Assuming that the valid status code must be either an H (hourly), S (salaried), T (trainee), or a P (part-time), an error message would be generated if the code used was not an H, S, T, or P.

**(viii) Verification of Data Type**

Some input fields should contain only numeric data while others should contain only alphabetic data. The fields can be edited to make certain that only the right type of data is recorded in each field.

**(ix) Verification That Certain Combinations of Data Exist**

For example, all students may be coded with either a W or a V. The V denotes a non-work-study student while the W indicates that the student is on work-study. The only valid account numbers for a work-study student are 2155 and 2156. Any other account number for a W-coded student is invalid.

**(x) Sequence Check**

If the numbers in the source documents are serial and the documents are in order, the input records will also be in numerical sequence. A check can be made by the program to determine whether the records are in either ascending or descending order.

**Q18. What are the benefits of using flowchart ?**

**The benefits of using flowcharts are :**

o **Promote process understanding by explaining the steps pictorially.** People may have differing ideas about how a process works. A flowchart can help you gain agreement about the sequence of steps. Flowcharts promote understanding in a way that written procedures cannot do. One good flowchart can replace pages of words.

o **Provide a tool for training employees.** Because of the way they visually lay out the sequence of steps in a process, flowcharts can be very helpful in training employees to perform the process according to standardized procedures.

o **Identify problem areas and opportunities for process improvement.** Once you break down the process steps and diagram them, problem areas become more

visible. It is easy to spot opportunities for simplifying and refining your process by analyzing decision points, redundant steps, and rework loops.

o **Depict customer-supplier relationship,** helping the process workers understand who their customers are, and how they may sometimes act as suppliers, and sometimes as customers in relation to other people.

o They are **self-documenting**.

o **Allow easy review and maintenance** of a software system. This results in less time spent on debugging

o **Helps to develop reusable algorithmic structures** hence they are useful for identifying the logic of generic systems. Hence they help the user to develop concise generalize algorithmic solutions to similar problems.

o Force **careful and concise solution** to the logic of a program problem.

o A simplified **consistent means** for developing program logic .

**Q19. What are the various steps in the process of selection of Hardware? Explain them. What are the criteria for selection of Software? Explain them.**

The selection process should be viewed as a project and a project team should be formed with the help of management. The selection process consists of several steps which are discussed below :

**1. Requirement analysis :** The first step in selection is understanding the user's requirements within the framework of the organisations' objectives and the environment in which the system is being installed.

**2. System specification :** System specifications must be clearly defined. These specifications must reflect the actual applications to be handled by the system and include system objectives, flowcharts, input-output requirements, file structure and cost.

**3. Request for proposal :** After the requirement analysis and system specification have been defined, a request for proposal is prepared and sent to selected vendors for bidding.

**4. Evaluation and validation :** The evaluation phase ranks various vendor proposals and determines the one best suited to the user's requirements. It looks into items such as price, availability and technical support. System validation ensures that the vendor can, in fact, match his/her claims, specially system performance.

**5. Vendor selection :** This step determines the vendor with the best combination of reputation, reliability, service record, training, delivery time, lease/finance terms. The selected vendors are invited to give a presentation of their system. The system chosen goes through contract negotiations before implementation.

**The criteria for software selection are :**

**(a) Reliability :** gives consistent results without any failure for a specified time period.

**(b) Functionality :** functions to standards.

**(c) Capacity :** satisfies volume requirements of the user.

**(d) Flexibility :** adapts to the changing needs

**(e) Usability :** is user-friendly.

**(f) Security :** maintains integrity and prevents unauthorised user.

**(g) Performance :** delivers the results as expected.

**(h) Serviceability :** has good documentation and vendor support.

**(i) Ownership :** has right to modify and share use of package.

**(j) Minimal costs :** is justified and affordable for intended application.

**Q20. Explain briefly the importance of a system reliability. Can a system be hundred percent reliable?**

A system is said to be reliable if it does not produce dangerous or costly failures during its normal use. The definition recognises that system may not always be used according to designer's expectations. There are changes in the ways users use a system and also in business operations. However, there are steps analysts can follow to ensure that the system is reliable at the installation stage and its reliability will continue even after implementation.

There are two levels of reliability. The first level shows that the system is meeting the right requirements. This is possible only if thorough & effective determination of systems requirements was performed by the analyst. A careful and thorough systems study is required for this aspect of reliability. The second level of systems reliability involves the actual working of the system delivered to the user. At this level, system reliability is interwoven with software engineering & development.

**No,** it is not possible that a system will be 100% reliable. Reliability has three approaches : (i) Error avoidance (ii) Error detection & correction, and (iii) Error tolerance. It is not necessary that a system consists of all of these approaches.

**Q21. What is the difference between verification and validation?**

Before System acceptance few tests are to be performed by end-user using real data over an extended period. Extensive test that addressees three levels of test of acceptance testing :

1. Verification testing.
2. Validation Testing

3. Audit testing.

Verification testing runs the system in simulated environment using simulated data. Sometime this testing called "ALPHA TESTING". The simulated test primarily looking for errors and omissions regarding end-user and design specification which were specified in the earlier phases but not fulfilled during construction.

Validation Testing runs in the live environment using real data. Sometime this testing called "BETA TESTING". During validation number of tests are considered as follow.

**(i)** System Performance. During the normal processing workload the time and response should be adequate.

**(ii)** Peak Workload processing performance. During the peak workload the performance is tested, if not performed efficiently than Hardware/Software efficiencies should be increased.

**(iii)** Human Engineering test.

**(iv)** Methods and procedures test.

**(v)** Recovery test.

**Q22. Differentiate between Unit testing and System testing. Describe also the various steps involved in System testing.**

The **main difference** between system testing and unit testing is the scope of the interdependencies. When we think about system testing for a web application, for example, we take all of the pieces into consideration, such as the Views and Controls presented to the end user. When we think about unit testing, however, we think about testing one family(or a single module) of components at once. This implies, that Unit testing gives stress on the modules independently of one another, to find errors. While, in system testing all the subsystems are gathered into one pool and the whole system is tested to determine whether it meets the user requirements. System testing is know for last chance to correct the errors while after unit testing there are integration, and system testing to correct the errors. But it is considered bad to un-detect the module level errors in unit testing. We can say unit testing is subset of system testing while vice-verse is not true.

**System testing consists of the following five steps :**

- Program testing
- String testing
- System testing
- System documentation
- User acceptance testing

**Q23. Advantages of using Data Centres**

The various advantages of using data centres :-

**(a)** The major benefit of using data centre is that they can make use of computer without spending large initial amount.

**(b)** It eliminates staff and management problems caused by the employment of a team of highly paid technical professionals in rapidly changing field of computer.

**(c)** The small organisation can utilise the expertise knowledge of experienced and qualified staff of data centre in his data processing job which is not available within the organisation.

**(d)** There is no fear of equipments becoming obsolete.

**(e)** The small organisation can get valuable experience of working on computer before deciding whether or not to install an in-house computer.

**Q24. Do you think that it is always essential to train the employees of an organization who will be using software that is developed by a software company? Explain your answer.**

A crucial phase in the system life cycle is the successful implementation of the new system design. Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be completely new, replacing an existing manual or automated system or it may be major modification to an existing system. In either case, proper implementation becomes necessary so that a reliable system based on the requirements of the organisation can be provided. Successful implementation may not guarantee improvement in the organisation using the new system, but improper installation will prevent it. It has been observed that even the best system cannot show good result if the analysts managing the implementation do not attend to every important details. This is an area where the systems analysts need to work with utmost care. Where first activity regarded as most important is "Training Employees" to make them familiar with the system.

Even well designed system can succeed or fail because of the way they are operated and used. Therefore, the quality of training received by the personnel involved with the system in various capacities helps or hinders and may even prevent the successful implementation of management information system. Those who are directly or indirectly related with the system development work must know in detail what their roles will be, how they can make efficient use of the system and what the system will or will not do for them. Both systems operators and users need training.

**Systems Operators Training :**

Running of the system successfully depend on the personnel working in the

computer centre. They are responsible for providing the necessary support. Their training must ensure that they are able to handle all possible operations, both routine and extra-ordinary in nature.

If the system calls for the installation of new equipment, such as a new computer system, special terminals or different data entry machines, the operators' training should include such fundamentals as how to turn the equipment on and use it, how to power off and a knowledge of what constitutes normal operation. The operators should also be trained on different type of malfunctioning, how to recognise them and what steps should be taken whenever they arise.

As part of their training, operators should be given both a troubleshooting list that identifies possible problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problems arise. Training also involves familiarisation with run procedures, which involves working through the sequence of activities needed to use a new system on an ongoing basis.

**User Training :**

User may be trained on use of equipment, particularly in the case where, for example, a micro-computer is in use and the individual involved is both operator and user. In such cases, user must be given training on how to operate the system also. Questions that may be trivial to the analyst, such as how to turn on a terminal, how to insert a diskette into a micro-computer, or when it is safe to turn off equipment without danger of data loss, are significant problems to new users who are not familiar with computers.

In most of the cases, user training deals with the operation of the system itself, with proper attention given to data handling techniques. It is imperative that users be properly trained in methods of entering transactions, editing data, formulating inquiries, deleting and inserting of records. No training is complete without familiarising users with simple systems maintenance activities. Weakness in any aspect of training may lead to awkward situations that create user frustration and errors. So, it is must to give training to these user if he does not knows about software & system concepts.

**Q25. Distinguish between transaction processing systems, office automation systems, decision support systems and management information systems.**

| TPS | OAS | DSS | MIS | ESS | KWS |
|---|---|---|---|---|---|
| This is a system used for storing information which needs to be updated frequently and often online 24 hours a day. | This system are configurations of networked computer hardware and software used in business and communication functions, such as preparing written communications and strategic planning. | This system allow analysts to extract information quickly and easily which are analyzed very oftenly , daily, weekly, and yearly . | This system is used to collect and process data(information) and provides it to managers at all levels who use it for decision making, planning, program implementation, and control. | This system provide high level views of an organisation by aggregating data from various sources from within the organisation and also external sources. | This is the Information system that aid knowledge workers in the creation and integration of new knowledge in the organization. |
| Need High data transfer rate | No need of High data transfer rate | High data transfer rate but less then TPS. | Moderate data transfer rate | Moderate data transfer rate | Slow data transfer rate |
| Operations perform very frequently | Operations performed frequently | Operations performed on regular basis | Operations performed rarely | Operations performed frequently | Operations performed very rarely |
| Distributed in nature | Restricted to small geographic area | Restricted to small geographic area | Discrete in nature | Restricted to a building or a campus in nature | Discrete in nature |
| Requires large number of operator to operate the system | Every user operate the system by his own | Every user operate system by his own | Requires less number of operator to operate the system | Requires proportional number of operator to user operate the system | Requires very less number of operators to operate the system |
| Requires basic knowledge about system understability | Basic knowledge is sufficient | Basic knowledge is required | Require moderate knowledge | Requires some skills | Basic knowledge is sufficient |
| No redundancy | redundancy | redundancy | redundancy | No redundancy | No redundancy |

**Q26. Explain the meanings of the terms 'hot sites' and 'cold sites' in the context of planning for an organization's information systems. What are the advantages and disadvantages of each of these choices?**

**# Hot sites**

The term hot site is used for a site which is almost a replica of the afflicted site and is commercially available 'on demand'. The advantage of such an arrangement is that it is very fast and a continuity of operations can be maintained fairly easily. The pre-dominant disadvantage is that it is likely to be quite expensive.

**Private Hot Sites**

In the absence of the availability of a hot site, or the decision not to use such a site for confidentiality or other strategic reasons, some organisations establish their very own hot site for such emergencies. The biggest advantage is

that there is no dependency on any other organisation and, therefore, no sharing of resources. Naturally, the biggest advantage also flows from this and that is it would cost exactly twice cost of the original site. Since this site would not be used very often, it effectively doubles the cost of the capital invested in the business.

| Alternative | Features | Advantages | Limitations |
|---|---|---|---|
| "Cold" backup service | Lease terms on site with no computer but with adequate electric communication to support a computer facility | Low cost, high security | Difficulty replacing hardware, especially custom-made equipment limited use, usually up to sixweeks shared membership of site complicates security and privacy, easy to outgrow site due to undercapacity |
| Private cold site | Own an empty computer facility in a remote area with features similar to "cold" backup serivice | Guaranteed access, no compatibility problems, relatively low cost | Problem obtaining and installing necessary equipment after a disaster |
| "Warm" backup service | Contract with a computer service bureau to use its facility in case of disaster | Time saving, good environment, security | High cost, restrictiveness of use, compatibility problem |
| Private warm backup site | Ranges from having a complete facility left idle to duplicate computer installation that can be switched to service immediately | Reliability, quick availability | High cost, possible loss of management and control |

**Cold Site :** A disaster recovery facility that provides only the physical space for recovery operations while the organization using the space provides its own hardware and software systems.

**Q27. What do you mean by Disaster Recovery Planning.**

Disaster recovery (DR) planning is the process of developing advance arrangements and procedures that enable an organisation to respond to a disaster by resuming critical business functions within a defined time frame, minimising loss, and restoring affected areas.

It is not a two-month project, neither is it a project that you can forget about, once it is completed. An effective recovery plan is a live recovery plan. The plan must be maintained and tested/ exercised regularly.

**An effective DR plan consists of the following stages:**

Programme description

Pre-planning activities (project initiation)
Vulnerability assessment and general definition of requirements
Business impact analysis
Detailed definition of requirements
Plan development
Testing programme
Maintenance programme
Initial plan testing and plan implementation.

The primary objective of a business resumption plan is to enable an organisation to survive a disaster and to re-establish normal business operations. In order to survive, an organisation must ensure that critical operations can resume within a reasonable time frame. Therefore, the goals of a business resumption plan should be to identify weaknesses and implement a disaster prevention programme, minimise the duration of a serious disruption to business operations, facilitate effective co-ordination of recovery tasks, and most importantly reduce complexity of the recovery effort.

Historically, the data processing function alone has been assigned the responsibility for providing contingency planning. Frequently, this has led to the development of recovery plans to restore computer resources in a manner that is not fully responsive to the needs of the business. Contingency planning is a business issue rather than a data processing issue. In today's environment, the effects of long-term operations outage may have a catastrophic impact. The development of a viable recovery strategy must, therefore, be a product not just from the providers of the organisation's data processing, communications and operations centre services, but also the users of those services and management personnel who have the responsibility for protection of the organisation's assets.

**Q28. Define a DFD. Write the conventions that govern the construction of DFDs. Design a DFD for a study centre management system.**

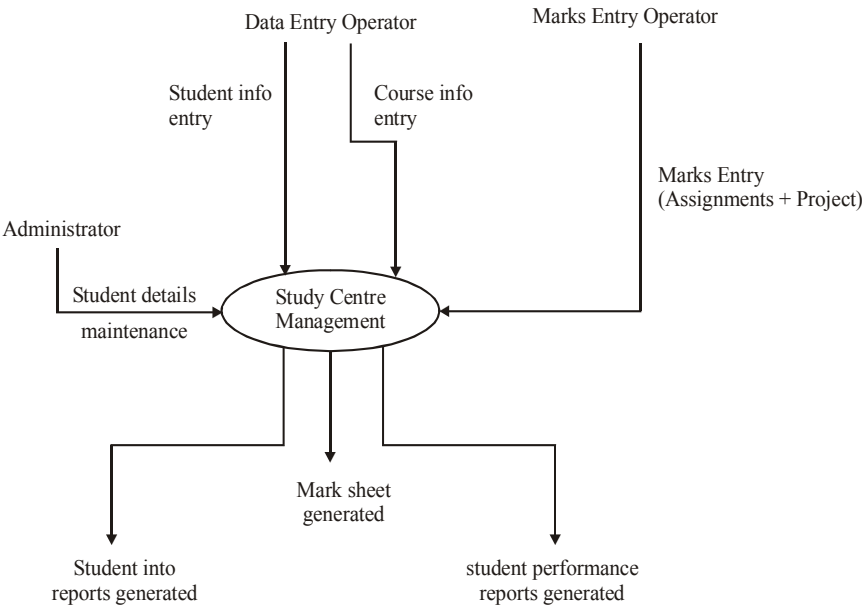
**Data Flow Diagram (DFD)** is a powerful diagram that can be used to document the information flow. It also presents itself to be broken down in top-down fashion. At the top level, data flows are represented at very abstract aggregate level. Each component of the data flow is further broken down to different levels, so that at each level we have just a few entities to concentrate on. DFD have developed a representation scheme to represent data store (storage & retrieval of data), processes (where some changes are made to the system) and entities (the player in the game) and the actual information flows.

**The following seven conventions governing the construction of DFDs :**

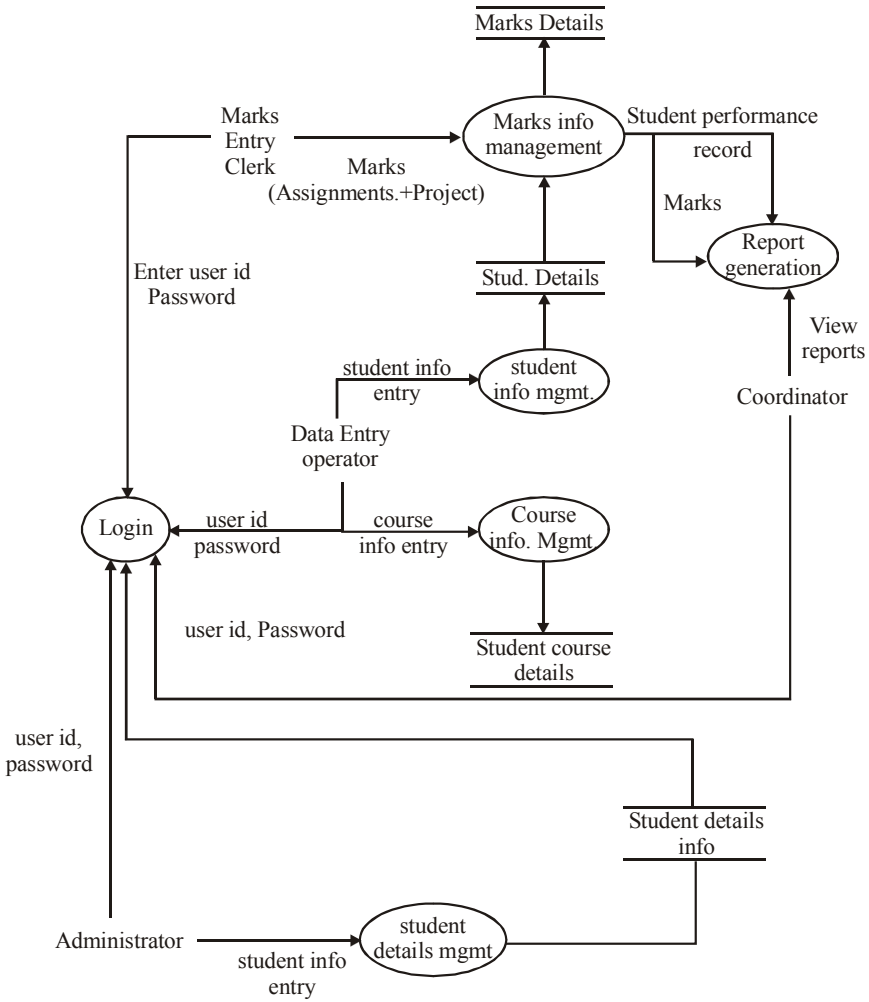
**1.** Arrows should not cross each other.

**2.** Squares, circles, and files must bear names.
**3.** Decomposed data flows must be balanced (all data flows on the decomposed diagram must reflect flows in the original diagram).
**4.** No two data flows, squares, or circles can have the same name.
**5.** Draw all data flows around the outside of the diagram.
**6.** Choose meaningful names for data flows, processes, and data stores. Use strong verbs followed by nouns.
**7.** Control information such as record counts, passwords, and validation requirements are not pertinent to a data-flow diagram.
If too many events seem to be occurring at a given point, an analyst can decompose a data conversion (circle). The new data conversions form a parent-child relationship with the original data conversion: the child circle

**Design a DFD for a study centre management system**




O-Level DED




**Users** are Coordinotor, Student, Administrator, Data Entry Operator, Marks Entry Operator.

**Q29. Develop an analysis and design document for a video library management information system. Make necessary assumptions.**

***Analysis :***

**Technical Feasibility :** This is designed by considering the existing and aspecting technology. This system is compatible for the existing system and can work efficiently even if small changes occurred.

**Economic Feasibility :**The developing running and managing cost of this

system is very low. This system requires very less maintenance cost . This system is very economical from the developer, customer, and management point of view.

**Operational Feasibility :**This system has designed to overcome all the problems, faults and lacks of a manual or any existing system . This system is very user friendly. Any one can understand its operations by the help of its user-friendly interface.

***Design :***

Once it is found that the project development is feasible, Design has to be developed for the requirements listed in the analysis phase.

**Data Dictionary :** A Data Dictionary is a catalogue of all elements in a system. It consists of data about data which is called metadata. It is a document that collects co-ordinates and confirms what specific data terms mean to different people in the team. It is important for the reasons like To manage the details, Communicate meaning, Document system features, Facilitate analysis and Locate errors and omission. The data dictionary stores to following description.

(i) Item ( CD/DVD/VCD/ACD ) details

This stores the information about the stock of video library.

| **Attributes** | **Stores** |
|---|---|
| ITEM –ID | Unique ID of item |
| NAME | Name of Album / Movie |
| CATEGORY | Pop / Classical / English/ Hindi / Punjabi etc. |
| SINGER | Singer of that album. |
| PRICE | Price of that item. |
| DATE-RELEASE | Date of release |
| DATE-ADDED | Date on which added in library |
| COPIES | Total no. Of copies of that item in Library. |

(ii) ITEM_STATE

Stores information regarding current status of the item.

| **Attributes** | **Stores** |
|---|---|
| Item – id | Unique ID number. |
| Status | Gives info. About current status of the item.<br>(R)→ Item is on rent.<br>(L) → Item is lost.<br>(P) → Item is present. |

(O) →Returned data has gone.
(D) →Item is deleted.
(S) → Item is sold

(iii) Item_Rent
This stores the information about the items, which are on rent.

| **Attributes** | **Stores** |
|---|---|
| TID | Identification no. of rent transaction. |
| DEPOSIT | Security deposit. |
| Item – ID | Unique id no. item. |
| Customer – Name | Name of the customer who take item on rent. |
| Customer – Address | Address of the customer. |
| Customer –Contact no. | contact no. of customer. |
| Time | Time at which item is issued |
| Date–return | Date on which item take to be returned. |

(iv) ITEM_DELETE
It stores the information about the items that are deleted and the date on which it was deleted.

| **Attributes** | **Stores** |
|---|---|
| ITEM_ID | Unique id of each item. |
| DATE | Stores the date on which the item was deleted. |
| REASON | Reason why it is deleted. |

(v) COMING_ITEM
It stores the information about the item ( album / movies ) that are about to come

| **Attributes** | **Stores** |
|---|---|
| ID | a unique ID for this table only. |
| Proposed – date | Date on which item is supposed to come. |
| Title | Title of that item |
| Category | Pop/ Classical / Hindi / Punjabi etc. |
| Singer | Singer of that item |
| Copy | No. of copies aspect. |
| Price | Approximate price of that item. |

**KEY FEATURES**

**(1)** Item ( CD/DVD /VCD /ACD/ Cassettes ) can be taken on rent.
**(2)** Add new item in to the database of video Library.
**(3)** Delete existing item from database.
**(4)** Keep record of item which are accepted. To added / released.
**(5)** Account maintenance.
**(6)** Query processing:
(i) Query about availability of a item.
(ii) Query about customer who have taken a item on rent.
(iii) Query about number of copies.
(iv) Query about upcoming issues.
**(7)** Return an item.

**DESCRIPTION**

**(1) Item on rent**

(i) A customer can issue a item on rent and at that time complete details about that customer will be stored in the database.
(ii) A limitation about the time period for which a customer can have that item is imposed on customer.
(iii) Full payment or advance will be taken from that customer.
(iv) Respective database will be updated.

**(2) Adding new item in video library.**

In this feature new item ( CD /DVD etc.) will be added in to the database then video library's respective database will be updated according to requirement and constraints.

**(3) Deleting existing item from the Video Library**

Existing item sometimes need to be deleted from the library. This is may be due to less space, corrupted item, least item or trend has changed. After deleting any item, the respective database will be updated.

**(4) Upcoming – Item / Issues**

Keep records of all the upcoming albums / movies which are about to release or already released but havn't arrived yet. So, a advance records will be kept for processing customer queries.

**(5) Account Maintenance**

Keep record of status of item that are given to customer on rent or sale Also give alert to operator if any item's return date has passed.

**(6) Query Processing**

Solve the query of customers and library owner about the items status, condition etc. Customer can know about the price, category, number of copies, date of released, date of arriving, and date of return etc. Library owner can also process query about the item which are in grace or in overdue.

**(7) Returning of Item**

When a customer is returning an item, checks about the date, time, condition of that item is to be done. Also check about the remaining balance (if any) or security deposit. According to that, update the respective database

**Q30. Develop Analysis and Design documents for a Library Information System. Make necessary assumptions.**

***Analysis :***

**Technical Feasibility :** This is designed by considering the existing and aspecting technology. This system is compatible for the existing system and can work efficiently even if small changes occurred.

**Economic Feasibility :**The developing running and managing cost of this system is very low. This system requires very less maintenance cost . This system is very economical from the developer, customer, and management point of view.

**Operational Feasibility :**This system has designed to overcome all the problems, faults and lacks of a manual or any existing system . This system is very user friendly. Any one can understand its operations by the help of its user-friendly interface.

***Design :***

Once it is found that the project development is feasible, Design has to be developed for the requirements listed in the analysis phase.

**Data Dictionary :** A Data Dictionary is a catalogue of all elements in a system. It consists of data about data which is called metadata. It is a document that collects co-ordinates and confirms what specific data terms mean to different people in the team. It is important for the reasons like To manage the details, Communicate meaning, Document system features, Facilitate analysis and Locate errors and omission.

**Key Features of Library Information System :**

(1) Issue of Books

(2) Return of Books

(3) Query processing

Query about availability.

Query about number of books available.
Query about return /issue dates.

**Description :**

**(1) Issue of books**

(i) A student of any course should be able to get books issued.
(ii) Books from general section are issued to all but the book bank books are issued only for their respective courses.
(iii) A limitation is imposed on the number of books a student can issue.
(iv) A maximum (say 4) number of books from book bank and some pre-defined (say 3) books from general section is issued for 15 days only.
(v) The system takes the current system date as the date of issue and calculates date of return.
(vi) A due date for return of the book is stamped on the book.

**(2) Return of books**

(i) Any person can return the issued books.
(ii) The system displays the student details on whose name of books were issued as well as the date of issue and return of the book.
(iii) The system operate verifies the duration for the issue.
(iv) The information is saved and the corresponding updating take place in the database.

**(3) Query Processing**

(i) Any user ( librarian, student ) can find the availability of a particular books.
(ii) A user can query about the availability of books of any particular author. In this query the system will show all the written by that author.
(iii) Number of copies available of the desired books. This will also show to whom the remaining books issued.

**In our system Data Dictionary record store the following descriptions:**

Books Details

Stores information about books in the library. This table contains the following attributes:

| Attributes | Stores |
|---|---|
| Book Id (primary key) | ISBN Number |
| Name | Name of the book |
| Category | Category of the book |
| Subject | Subject of the book |
| Author | Author of the book |
| Price | Price of the book |
| | |
| | |
| | |

| | |
|---|---|
| Date | Date on which the book is added in the library |
| Edition | Edition of the book |
| Copies | Total no. of copies of the book present in the library. |

Book _copy
Stores information about various copies of a book available in the library.

| Attributes | Stores |
|---|---|
| Book Id | ISBN Number |
| Book Idg | Ident no. of the book whose multiple copies are present |

Book _ State
Stores information regarding current status of the book.

| Attributes | Stores |
|---|---|
| Book Id (primary key) | ISBN Number |
| Status | It gives information about current status of the book<br>Status of a book can be- |
| | * I => Book is used |
| | * L => Book is lost |
| | * P => Book is present |
| | * M => Book is lost by member. |
| | * D => Book is deleted |

Book_IR
It gives information about the state of the book which is issued to a member.

| Attributes | Stores |
|---|---|
| TID (primary key) | Identification no of issue/return transaction. |
| Book ID<br>MemID | ISBN Number<br>Stores indent of member who issued the book |
| Mode | Mode can be - |
| | * I => book is issued for home |
| | * R => book is issued for reading in library |
| State | State can be- |
| | * 0 => book is with the member<br>* 1 => member returned the book.<br>* 2 => member lost the book |

Book_Irdetail
It gives information about issue /return and date/time of the issue of the book.

| Attributes | Stores |
|---|---|
| TID | Identification number of the transaction of issue/return. |
| Issue_Date | Stores the date on which the book is issued. |
| Issue_Time | Stores the time of issue of the book to the number. |
| Return_Date | Stores the date on which a member returns the book. |
| Return_Time | Stores the time at which a member returns the book. |

Books_Delete
It stores the information about the books that are deleted and the date on which it was deleted.

| Attributes | Stores |
|---|---|
| Book Id | ISBN Number |
| Date | Stores the date on which the book was deleted. |

Book _Renewed
It store the information about the book ,when it is brought back to the library and the date on which it was renewed.

| Attributes | Stores |
|---|---|
| Book Id | Stores the bookId of the book being deleted. |
| Date | Stores the date on which the book was resumed. |

Member_Details – It stores the information about members of the library.

| Attributes | Stores |
|---|---|
| Memid | Stores the ID of the member. |
| First | Stores the first name of the member. |
| Middle | Stores the middle name of the member. |
| Last | Stores the last name of the member. |
| Sex | Stores the gender of the member. |
| Address | Stores the address of the member. |
| City | Stores the city of the member. |
| State | Stores the state of the member. |
| Country | Stores the country of the member. |
| PIN | Stores the pin code of the member. |
| Age | Stores the age of the member. |
| Phone | Stores the phone number of the member. |
| Issue_limit | Stores the maximum number of books that can be issued to the member |
| Date_of_Joining | Stores the date on which member enrols in the library |
| Books_issued | Stores the number of books issued to the member. |

Member_State
Storing information about state of the member in the library.

| Attributes | Stores |
|---|---|
| MemID | Stores the ID of the member |
| State | It can have two values. |
| | * P => Member can access the library<br>* D => Member is deleted for library access |

Member_Deleted
It stores information about the member deleted and the date on which the member was deleted.

| Attributes | Stores |
|---|---|
| MemID | Stores identification no of the member being deleted. |
| Date | Stores date on which member was deleted. |

**Q31. Suppose that you are the analyst charged with development of a "Student Information System" for a university. Develop an Analysis Plan for the same.**

**Objective**

To develop an analysis plan for Student Information System that support the delivery of academic and support services to students & to improve communication about students and their progress, and lets any institute to collect, manage and analyze whatever student information they choose to track. Student Information System is a system for students, parents, prospective students, and employees; an administrative transaction processing environment for university staff to conduct day-to -day business; an information environment for all levels of faculty and staff to do reporting, data extraction, and information analysis.

Student Information System have to accept, process and generate reports accurately and at any point of time any user can get the student information. Student Information System applications are proven to reduce time spent on administrative tasks so you can concentrate on raising student achievement. SIS facilitates networking of colleges/schools to University/Board, colleges/ schools to Society, colleges/schools to each other, provided they are equipped with this system.

The Student Information System is designed at integrating the four elements of an educational system - Management, Faculties, Students and Parents. It starts from a simple student portfolio to a complex open communication channel which facilitates synchronisation between the four elements with information and helps institutions be efficient and effective.

**Different Modules of a student information system are :-**

- Student Database
- Scheduling
- Attendance
- Communication
- Report Writer
- Grade Card
- Skills / Standards
- Discipline
- Fees / Billing

**Database Module**

The database of student information system saves hours of the staff in time-consuming data entry and record-keeping. When your staff enters student

information into Data Base, that information appears through all Administrator's modules and any Office Suite programs like scheduling, attendance, discipline, grading, cafeteria, library, health records, tuition billing, and more.

**Database Features:**

• Collect and report on student and staff information.
• Customize the program to meet data collection and reporting needs.
• Search and report based on any information in the database.
• Break down the student body by race, gender, age, etc.
• Batch enter information into the records of all or a group of specified students.
• Flexible security lets you choose the level of security appropriate .
• The Data Base module report writer supports multiple languages for many reports.

**Class Scheduling module**

Scheduling is our module for scheduling and master schedule building. This system can make schedules available for parents and students using our center, while through others module allows staff members to access student schedules .

**Class Scheduling Features**

· Easily view, edit and print student schedules.
· Support multi-tasking environment: View student schedules while accessing their information in other modules at the same time.
· Enter course requests manually.
· Reports and tools to easily resolve scheduling conflicts.
· Communicate with students about their class schedules.
· View students' schedules and confirm which class a student should be in at any time.
· Create numerous reports to facilitate scheduling.

**Attendance Module**

Attendance is our attendance management module, designed to handle the attendance requirements . The propose of this module is recording attendance, to produce daily attendance bulletins and letters and documents.

**Attendance Features :**

• Easily track and view attendance information for students and staff.
• Quickly and easily produce the daily attendance bulletin.
• Communicate with teachers the daily attendance bulletin.
• Produce reports analyzing your attendance data.
• Give format of Period Attendance that allow teachers to mark students as absent, to entry later.
• Customize the calendar to count each day the student attends.

**Grade Cards Management module**

Grade Card is module for managing and editing grades, generating grade-related reports, and producing customized report cards, honor rolls, and transcripts. Grade Cards module, can be used to generate Grade cards .

**Grade cards module features:**

· Manage Grades and Produce Grade Cards.
· View and edit students' grades for any year.
· Calculate honor rolls and class rankings using the criteria and ranges .
· Generate reports on honor rolls, class ranks, grade searches and grade distribution.
· Customize the program according to periods or semesters.
· Save copies of students' assignments and Grade cards for final assessment.

**School communication module**

Communication module enables reports generation and organizing, managing and sending administrative notices. This module save staff time of repetitious tasks by automating frequent communications between the institute and staff and/or families.

**Features:**

· Enable reports, letters, notices, schedules, report cards, and other documents.
· Send staff members the daily attendance bulletin each day .
· Stores which previous letter, notice or report.
· Creates addresses by groups of students, parents and staff.
· Request/solve query information about our communication module.

**Report Generation Module**

This module is used to produce the exact desired report is required format like letter, notice, report card, statistical report, or other document.

**Features:**

· Produce any type of report using data - from letters to statistical reports.
· Can analyze previous data and reports.

**Skills Module: Skills-Based Report Cards**

The module's flexibility makes it easy to produce standards-based and skills-based report cards in a variety of formats.

**Features:**

· Include attendance, school announcements and grading scales/keys on report cards
· Allows up to a 3-letter grade code.
· Include narrative comments by course or as a group .

**Discipline Module**

Discipline management module, simplifies the difficult and time consuming task of tracking student discipline and helps improve discipline by ensuring that students are held accountable for their actions. Discipline lets educators document and track disciplinary incidents, maintain related records, notify parents of infractions, and produce reports, notices, forms and other documents.

**Features:**

· Instantly access any student's complete disciplinary history when speaking with parents.

· Document multiple discipline incidents, follow-up actions, teacher comments, incident locations, and involved students and staff.

· Create incident reports containing multiple discipline incidents.

· Easily and quickly produce reports, including statistical analysis of discipline data, penalty reports.

· Use our communication module to easily inform parents discipline notices abut their child's involvement in a disciplinary incident.

**Fees/Billing Management module**

Fees/Billing module for billing management, enables to save administrative staff time in student invoicing, data entry and financial record keeping. Billing management module allows school staff to easily enter, view and edit invoices and payments, efficiently track details of past, present, and current fees tasks, and produce billing-related reports and correspondence.

**Features:**

· Keep records for tuition, class dues, graduation charges and other student fees.

· Easily enter, view and edit billing information.

· Enter charges of all types for students: class dues, athletic fees, graduation charges, tuition, overdue books, and more.

· Built-in report generator to create invoices, billing letters, forms, notices, reports and other documents using data.

· Print statistical and financial reports, invoices, statements, aging reports, receipt logs and family summary statements.

· Use our communication module to send parents invoices and statements for tuition, graduation charges, book fees etc.

# Solved Papers
# MCS-14

## MCS-014: SYSTEMS ANALYSIS AND DESIGN
## JUNE, 2007

**Note :** Question number 1 is **compulsory.** Attempt any **three** questions from the rest.

**1. (a) Prepare a SRS for a "Railway Reservation System". Design the DFD for the context level, first level and second level for the system. Follow the rules/conventions given in your study material to draw the DFDs.**

**Ans.** *The Software Requirements Specification* (SRS) is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional description, a representation of system behavior, and indication of performance requirements and design constraints, appropriate validation criteria, and other information pertinent to requirements.

Below are the S/W requirements specification to manage Railway Reservation System**:**

**Abstract :** The system to be developed is for working of the train information like arrival, departure etc., train ticket reservation, train ticket cancellation based on input about the traveller, Name, address, Phone no., Journey date, departure, destination, source.

**1. Introduction**

**1.1 Purpose :** The purpose of this document is to describe the external requirement for a Railway Reservation System. It also describes the interfaces for the system.

**1.2 Scope :** The document is meant for use by the developers and will be the basis for validating the final delivered system.

**1.3 Definitions, Acronyms, Abbreviations :** Related to various types of classes for tickets, departure, arrival, fare and classification of food served in train(in first class only).

**1.4 References:** can be taken from catalogues, pamphlets or from site visits.

**1.5 Responsibility of the developer :** The developer is responsible for

(a) Developing the system.

(b) Installing the software on client's hardware

(c) Training for using the system to the users at the client site.

(d) Providing maintenance for a specified period.

**2. General Description**

**2.1 Product Functions Overview :** The railway has 2500 trains which are

running for the different destinations. Each ticket carries its own unique PNR number, traveller name, plane number and class. The railway reservation system aims at finding information about any ticket quicker and easier, booking any ticket with a click of mouse, also cancellation of ticket. It also maintains a record of regular passengers, their names, addresses, tickets issued, date of return, etc.

**2.2 User characteristics :** The main users of the system will be operator who will book the tickets, will be familiar with the computer based system, as the training is given to them. Admin staff will also be familiar with computers for basic day to day accounts maintenance.

**2.3 General constraints :** The system should be run on a Windows NT version 4.

**2.4 General Assumption and Dependencies**

1. Each plane is classified as per the standard train classification system.
2. The user is supposed to be familiar with various categories of the seats.
3. Each ticket is know as per the standard PNR number.

**3. Specific Requirements**

**3.1 Inputs and Outputs :** The system is using various files to accept input and store it into the database, also for producing output various files are taken into account.

Following are the files used as I/O in this system : -

1. User_Master
2. Agent_Master
3. Discount_Master
4. Station_Master
5. Train_Master
6. Seat_Master
7. Fare_Master
8. Cancellation_Master
9. Passenger_Master
10. Train_Status
11. Passenger_Details

**Output_1:** Contains information about seats available or not.

**Output_2 :** Generate ticket cancelled report.

**Output_3 :** Error messages. At the minimum following error

**e1** Data file has error

**e2** No direct train for this destination

**Functional requirements**

**3.2 Updating input data:** Whenever any new train or station or agent is introduced in the system, the data for them is to be stored in database.

**3.3 External Interface Requirements User Interface:** Only one user command is required. System should prompt input data from the user.

**3.4 Performance constraints:** Information desired by the user should be

made available by the system in one second.

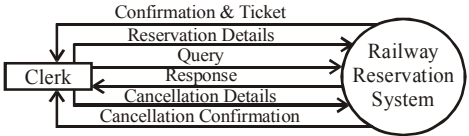
**3.5 Design constraints**

**3.5.1 Software constraints:** The system is to run under Windows operating system

**3.5.2 Hardware constraints:** The system will run on a Windows Operating System with 256 MB RAM. It will be connected to a 10-pages per minute printer.

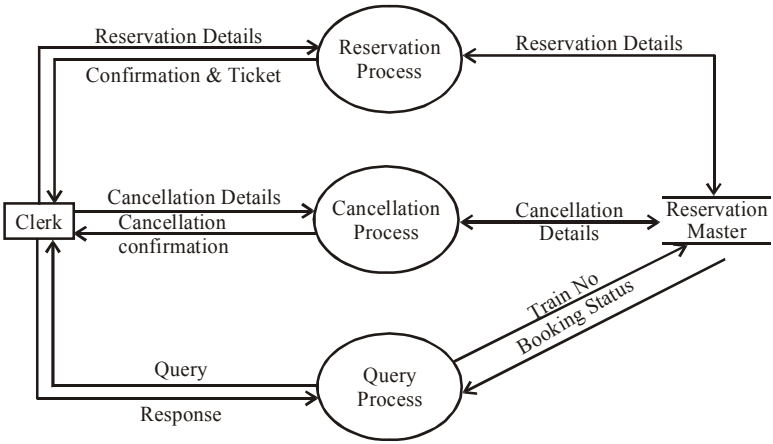
**3.6 Acceptance criteria :** Before accepting the system the developer must demonstrate that the system works satisfactorily. The test cases should test all things like data structure, interface, Basis path or other types of error detecting tests. White box & Black box tests can be applied to test it thoroughly.

**Context Level DFD**

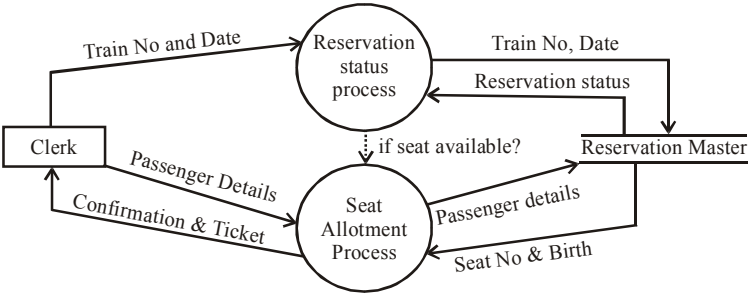



Level O D F D

**Level 1 DFD**




Level 1 DFD




Level 2 DFD for Reservation Process

**(b) What is the significance of the prototype approach? Mention the steps for the prototype design with the help of the flow diagram. Also, list the advantages and disadvantages of this approach.**

**Ans.** A prototyping approach emphasizes the construction model of a system. Designing and building a scaled-down but functional version of a desired system is the process known as Prototyping. A prototype is a working system that is developed to test ideas and assumptions about the new system. It consists of working software that accepts input, performs calculations, produces printed or displayed information or performs other meaningful activities. It is the first version or iteration of an information system i.e. an original model. Customer evaluates this model. This can be effectively done only if the data are real and the situations are live. Changes are expected as the system is used. This approach is useful when the requirements are not well defined. A prototype is usually a test model. It is an interactive process. It may begin with only new functions and can be expanded to include others that are identified later. The steps of Prototyping process are depicted in Figure. They are:

· Identify the user's known information requirements and features needed in the system.
· Develop a working prototype.
· Revise the prototype based on feedback received from customer.
· Repeat these steps as needed to achieve a satisfactory system.

Actual development of a working prototype is the responsibility of a systems analyst. The difference between a prototype model and an actual information system is that, a prototype will not include the error checking, input data validation, security and processing completeness of a finished application. It will not offer user help as in the final system.

But, sometimes, the prototype can evolve into the product to be built. The prototype can be easily developed with tools of fourth generation language (4GL's) and with the help of Computer Aided Software Engineering (CASE) tools. Prototyping approach is a form of rapid application development (RAD).

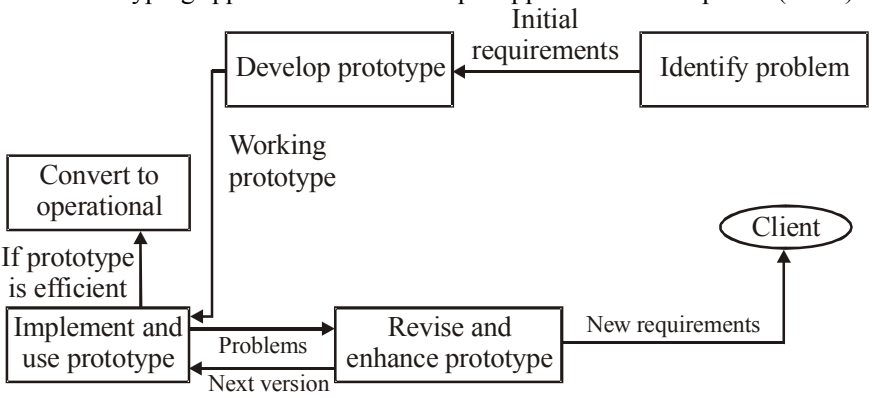



Figure : Prototype approach

Prototype approach is **advantageous** for determining requirements for the software to be built in the following situations:

**1.** Requirements are not clear.

**2.** For any complex systems, prototypes are more useful.

**3.** In the cases where communication problems exist between customer and analyst, this model is useful.

**4.** Tools and data are readily available for building the working system.

There are some **disadvantages** of the prototype approach :

**1.** The customer sees what appears to be working version of the software, unaware that the prototyping is held together "with chewing gum and baling wire." Unaware that in the rush to get it working no one has considered over all software quality or long-term maintainability.

**2.** Usually, prototypes are stand alone systems. Building prototypes is difficult in cases where data has to be shared.

**3.** Important issues, such as security and validation, are not given importance

**4.** The developer often makes implementation compromises in order to get a prototype working quickly. An inappropriate operating system or programming language may be used simply because it is available and known; an inefficient algorithm may be implemented simply to demonstrate capability.

**(c) "The design principles are meant to effectively handle the complexity of the process of design." List and explain all the design principles. Also, explain the top-down design approach with the help of "an inventory system" design.**

**Ans.** There are certain principles that can be used for the development of the system. These principles are meant to effectively handle the complexity of process of design. These principles are:

**Problem Partitioning:** It is concerned with partitioning the large problems. Divide and Conquer is the policy adopted here. The system is divided into modules that are self dependent. It improves the efficiency of the system. It is necessary that all modules have interaction between them.

**Abstraction:** It is an indispensable part of design process and is essential for problem partitioning. Abstraction is a tool that permits the designer to consider a component at an abstract level (outer view) without worrying about details of implementation of the component. Abstraction is necessary when the problem is divided into smaller parts so that one can proceed with one design process effectively and efficiently. Abstraction can be functional or data abstraction. In functional abstraction, we specify the module by the function it performs. In data abstraction, data is hidden behind functions/operations. Data abstraction forms the basis for object-oriented design.

Design principles are necessary for efficient software design. Top down and

bottom up strategies help implement these principles and achieve the objectives. A system consists of components called modules, which have subordinate modules. A system is a hierarchy of components and the highest-level module called super ordinate module corresponds to the total system to design such a hierarchy, there are two approaches namely top down and bottom up approaches.

The top down approach starts from the highest-level module of the hierarchy and proceeds through to lower level.

**1. Top Down Design:** This approach starts by identifying major components of the system decomposing them into own subordinate level components and interacting until the desired level of detail is achieved. Top down design methods often result in some form of stepwise refinement, starting from an abstract design, in each step, the design is refined to a more concrete level until we reach a level where no more refinement is required and the design can be implemented directly. This approach is explained by taking an example of "An Inventory System" depicted through figures here :


Figure : The top (root) of software system

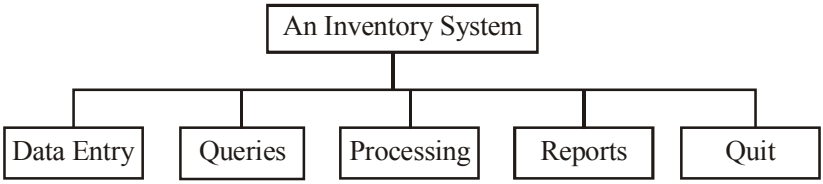



Figure : Further decomposition of the "top"of software system

Now, we can move further down and divide the system even further as shown in Figure.

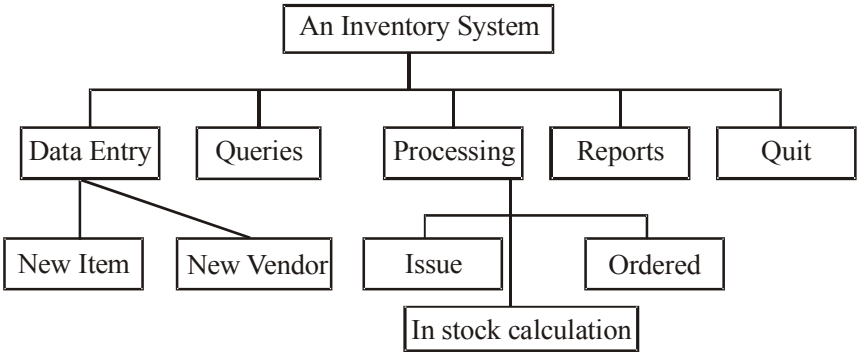



Figure : Hierarchy Chart of An Inventory System

This iterative process can go on till we have reached a complete software system. A complete software system is a system that has been coded completely using any front-end tool (ex Java, Visual Basic, VC++, Power Builder etc.) Top down design strategies work very well for system that is made from the scratch. We can always start from the main menu and proceed down the hierarchy designing data entry, modules, queries modules, etc.

**(d) Describe the aims of the following testing techniques:**
**(i) Stress testing**
**(ii) Performance testing**
**(iii) Response testing**
**(iv) Recovery testing**
**(v) Security testing**
**Ans.** Refer to Chapter-11, Q.No.-1, Page No.-106

**2. (a) What are the various components associated in a Entity-Relationship diagram? List the various symbols used and their purpose of use. Also, draw an ERD for a "Pay slip generation" application. Assumptions can be made wherever necessary.**
**Ans.** "E-R diagram". It is a detailed logical representation of the data for an organization and uses three main constructs i.e., data entities, relationships, and its own associated attributes.
**Entities :** An entity is a fundamental thing of an organization about which data may be maintained. An entity has its own identity, which distinguishes it from each other entity. An entity type is the description of all entities to which a common definition and common relationships and attributes apply. Consider an insurance company that offers both home and automobile insurance policies. These policies are offered to individuals and businesses. POLICY and CUSTOMER are both entity types in this example, while home and automobile are entities of POLICY, and individual and business are entities of CUSTOMER.
**Relationships :** A relationship is a reason for associating two entity types. These relationships are sometimes called binary relationships because they involve two entity types. Some forms of data model allow more than two entity types to be associated. A CUSTOMER is insured by a POLICY. A POLICY CLAIM is made against a POLICY

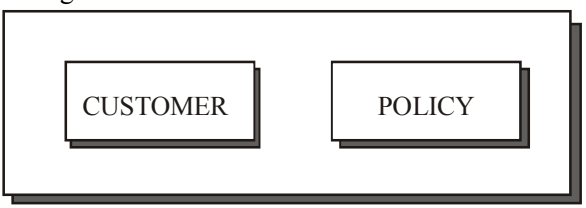



FIG: Two entity types in an E-R diagram

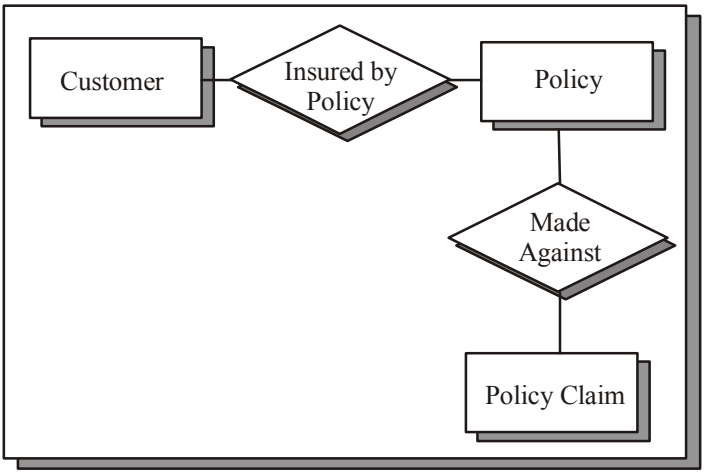



Fig: Relationship added to ERD

**Attribute:** An attribute is a descriptive property or characteristic of an entity. Synonyms include element, property and field.

A compound attribute is one that actually consists of other attributes. It is also known as a composite attribute. An attribute "Address" is the example of compound attribute.

**Cardinality** defines the minimum and maximum number of occurrences of one entity that may be related to a single occurrence of the other entity. Because all relationships are bi-directional, cardinality must be defined in both directions for every relationship. Figure depicts various types of cardinality.

**Degree:** The degree of a relationship is the number of entities that participate in the relationship.

**Recursive relationship 4:** A relationship that exists between different instances of the same entity is called recursive relationship. Figure depicts recursive relationship between the instances of the Course entity.

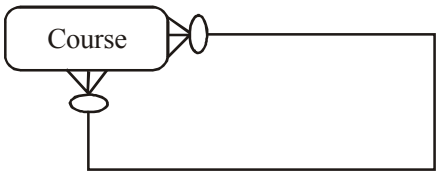
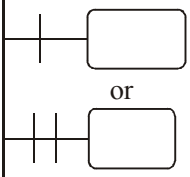
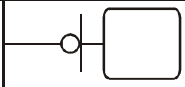
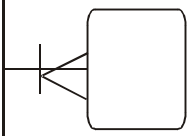
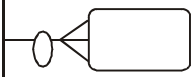
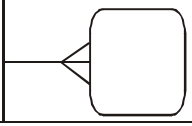



Figure : Example of Recursive relationship

| Cardinality Interpretation | Minimum Instances | Maximum Instances | Graphic Notation |
|---|---|---|---|
| Exactly one (One and only one) | 1 | 1 |   |
| Zero or one | 0 | 1 |  |
| One or more | 1 | Many(>1) |  |
| Zero, one or more | 0 | Many(>1) |  |
| More than one | >1 | >1 |  |

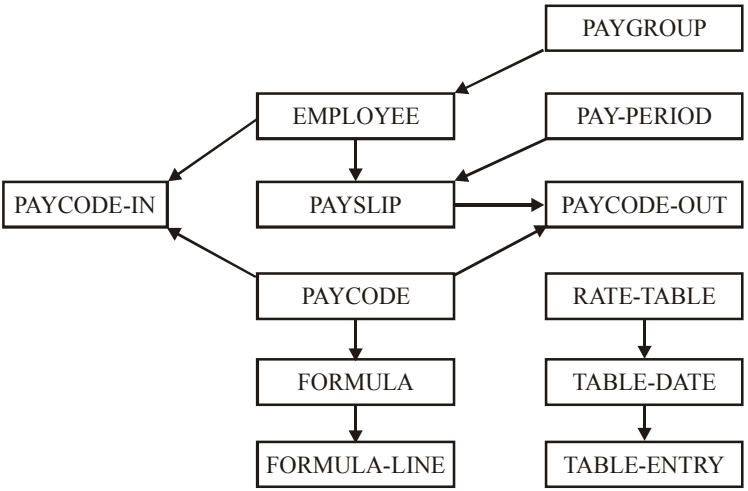



Figure : ERD for a "Pay slip generation

**In this E-R diagram the arrowhead identifies the MANY in each ONE-to-MANY relationship.**

| table name | contents |
|---|---|
| PAYGROUP | PAYGROUP_ID, paygroup_desc |
| EMPLOYEE | EMPLOYEE_ID, paygroup_id, employee_name |
| PAYCODE-IN | EMPLOYEE_ID, PAYCODE_ID, DATE-SEQ, input-hours, input-value, input-percent, start-date, end-date, node_id |
| PAY-PERIOD | PAYGROUP-ID, PAY-YEAR, PAY-PERIOD, start-date, end-date, pay-date |
| PAYSLIP | EMPLOYEE-ID, PAYSLIP-ID, paygroup-id, pay-year, pay-period |
| PAYCODE-OUT | EMPLOYEE-ID, PAYSLIP-ID, PAYCODE-ID, output-value |
| PAYCODE | PAYCODE-ID, sort-seq, paycode-desc, paycode-type, allow-hours, allow-value, allow-percent, allow-node |
| FORMULA | PAYCODE-ID, DATE-SEQ, start-date, end-date |
| FORMULA-LINE | PAYCODE-ID, DATE-SEQ, LINE-NO, formula-line |
| RATE-TABLE | TABLE-ID, table-desc |
| TABLE-DATE | TABLE-ID, DATE-SEQ, start-date, end-date |
| TABLE-ENTRY | TABLE-ID, DATE-SEQ, ENTRY-SEQ, rate-band, rate-amount |

I have simplified the contents here in order to remove unnecessary complications. The primary key fields are shown in upper case.

**(b) Define an Expert system. Mention the basic characteristics of the expert systems. Also describe various components of them.**
**Ans.** Refer to Chapter-13, Q.No.-4 Page No.-124

**3. (a) What are CASE tools? List the broad classification of various types of CASE tools and their usage.**
**Ans.** CASE (computer-aided software engineering) is the use of a computer-assisted method to organize and control the development of software, especially on large, complex projects involving many software components and people. Using CASE allows designers, code writers, testers, planners, and managers to share a common view of where a project stands at each stage of development. CASE helps ensure a disciplined, check-pointed process. A CASE tool may portray progress (or lack of it) graphically. It may also serve as a repository for or be linked to document and program libraries containing the project's business plans, design requirements, design specifications, detailed code specifications, the code units, test cases and results, and marketing and service plans.
Classification of CASE tools – CASE tools can be broadly classified into four

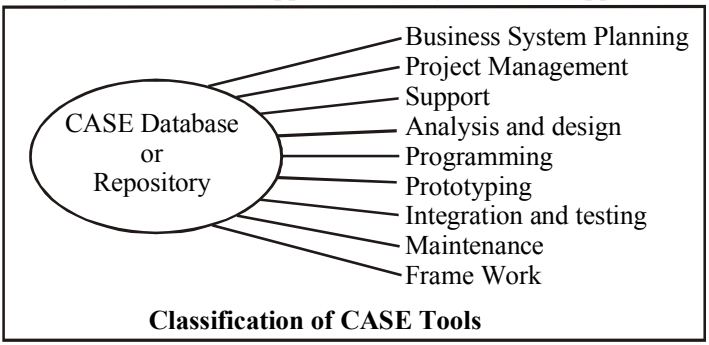
categories:

**1. Front-end CASE Tools** – These tools deal with the high level design, specification and analysis of software and requirements. These tools may include computer-aided diagramming tools, oriented towards a particular programming design methodology.

**2. Back-end CASE Tools** – These tools deal with the detailed design, coding, assembly and testing of the software. These tools may aid the programmer directly; for example, they include graphical debugging, aid query and browsing facilities to find quickly a particularly procedure or a variable.

**3. Maintenance Tools** – These tools deal with software after initial release. Then tools provide facilities in tracking bug fixes and enhancement requests, porting to new platform or performing new release.

**4. Support and Frameworks Tools** – These tools provide basic functionality required in tools of type 1, 2 and 3 support software includes basic operating system functionality as well as higher level support such as project management and scheduling software, and data base support to track different version and configurations of software releases. Various projects have been directed toward standardizing frameworks to support and integrate CASE applications.




**Classification of CASE Tools**

**(b) Define Systems Audit. List the objectives of it. Discuss the issues involved in Transaction, Security and Application audits.**

**Ans. First Refer :** Chapter-12, Q.No.-2, Page No.-112

**Issues involved in Transaction, Security and Application audits are given below :**

**1. Transaction Audit :** Transaction audit is a process to find

· Who did changes?

· What changes are made?

· Whether the changes are authorized or not as per the security policy of the Organization?

The details of the above transactions are written to either a media or printed. This allows Database Administrators to track changes and helps the organization

to satisfy regulatory requirements such as tracking specific users actions, general security screening, validating user permission etc.

**2. Audit of Computer Security:** Issues of security of computer involve both physical and logical security. Physical security involves restricting physical access to the computing resources from unauthorized person. Logical security involves restricting the use of computing resources by unauthorized person by providing logical control mechanism (e.g. password protection). The audit of computer security involves review of physical and logical security measures. Review of parameters, plans, practices, and policies that are developed and implemented by the organization over the computer resources, and how security measures are followed for Computers, Networks and Data communication. They are also included in the Audit.

**3. Audit of Applications:** Here, both manual and programmed internal controls related to information systems are assessed. Primarily, there are four areas of audit coverage for an application being reviewed.

The four areas are given below:

**Control environment:** This includes reviewing the system's security, its operating platform, system documentation and the interaction it has with other systems.

**Data Input Controls:** This involves reviewing the controls which ensure that data that enters into the system is accurate, complete and valid as per the standard. Examples include verifying system tables, limit checks, range checks and redundant data checks.

**Processing Controls:** These controls ensure that the data is properly processed and that automatic calculations performed by the system are accurate. This is tested by assessing controls built into the programs and by processing test data through the system and comparing the results of processing with expected results. Also, there will be checks on currency of stored data, default values and reporting exceptions.

**Output Controls:** In this, review of the system generated reports to ensure that they are accurate and the reports produced are reliable, timely and relevant is done. Also, it is checked whether cost savings can be achieved by reducing the number of reports produced. Data control personnel perform visual review of computer output and reconciliation of totals.

**4. (a) Elaborate the role of the Systems Analyst in the overall system development. Also, discuss the inter-personal skills that s/he should possess.**

**Ans. First See :** June-2005, Q.No.-1(a), Page No.-167

**The Interpersonal skills in context with the system are as follows:**

• **Communication :** The system analyst should have a very good communication

skill. He is supposed to interact with the other person of the organisation. Communication means listening others, feeling what other says and coming to one another.

• **Understanding:** Identifying the problems and assessing their solutions with the consideration and constraints proposed by the organisation.

• **Teaching:** Educating the people to understand computer, its user how it works, what more and more can be obtained through the computer.

• **Selling:** Selling ideas and promoting innovations in problem solving using computers.

• **Creativity:** Helping users model ideas into concrete plans and developing candidate systems to match user requirements.

• **Project management:** Scheduling, performing well under time constraints, coordinating team efforts and managing costs and expenditure.

• **Dynamic interface:**

• Questioning attitude and inquiring mind.

• Knowledge of basic of the computer and business functions.

**(b) Why are software documentation standards followed in any organization? Explain any three documentation standards, mentioning their high-points.**

**Ans.** This software documentation standard is used in the organization for uniform practices for documentation preparation, interpretation, change, and revision, to ensure the inclusion of essential requirements of different standards. Sometimes, documentation as per various standards is stated in the contractual agreement between the software vendor and the customer.

This standard will also aid in the use and analysis of the system/sub-system and its software documentation during the system/software life cycle of a software project. Documentation comes in many forms, e.g., specifications, reports, files, descriptions, plans, source code listings, change requests, etc. and can be in electronic or paper form.

**The following are the three documentation standards, mentioning their high-points :**

**1. ISO/IEC 18019** (Guidelines for the design and preparation of user documentation for application software) : This standard describes how to establish what information users need, how to determine the way in which that information should be presented to the users, and then how to prepare the information and make it available. It covers both on-line and printed documentation. It describes standard format and style to be adopted for documentation. It gives principles and recommended practices for documentation.

**2. ISOIEC 15910** (Software user documentation process) : This standard

specifies the minimum process for creating user documentation for software that has a user interface, including printed documentation (e.g., user manuals), on-line documentation, help text and on-line documentation systems.

**3. IEEE 1063** (Software user Documentation) : It provides minimum requirement for structure, information content and format for user documentation. It does not describe the process to be adopted for documentation. It is applicable for both printed and on-line documentation.

**5. Explain the following:**

**(i) Cost-Benefit Analysis**

**Ans.** Refer to Chapter-5, Q.No.-8, Page No.-48

**(ii) Data Coupling**

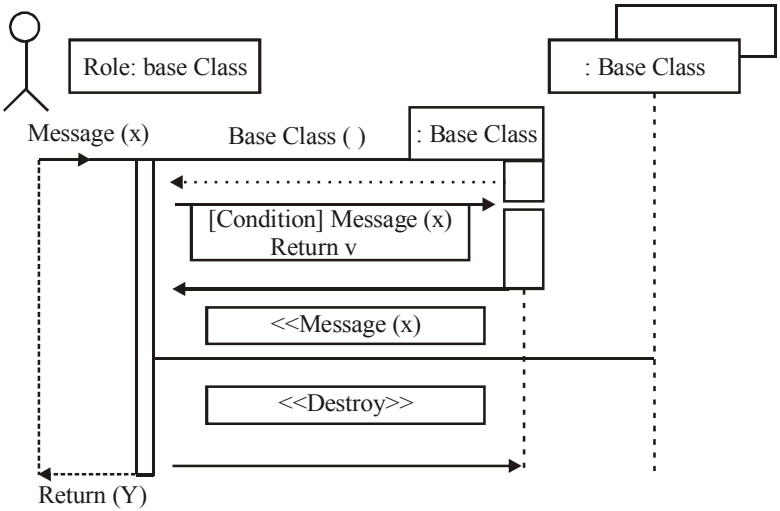
**Ans.** Refer to Chapter-6, Page No.-55

**(iii) Criteria for the form design**

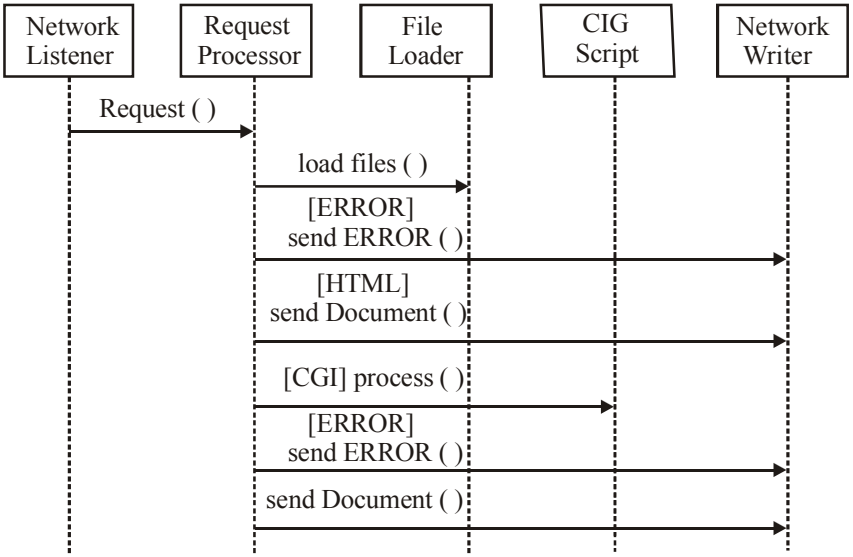
**Ans.** Refer to Chapter-5, Q.No.-5 Page No.-82

**(iv) Sequence diagrams**

**Ans.** These are interaction diagrams emphasizes the time ordering of message. It is shown as a table that shows objects along x-axis and messages in increasing time, along the y-axis. It has global life line and the focus of control.




The sequence diagram for sending the document along the matter is shown as:


Network Listener
Request Processor
File Loader
CIG Script
Network Writer
Request ( )
load files ( )
[ERROR]
send ERROR ( )
[HTML]
send Document ( )
[CGI] process ( )
[ERROR]
send ERROR ( )
send Document ( )

## MCS-014: SYSTEMS ANALYSIS AND DESIGN
## DECEMBER, 2007

**Note :** Question number 1 is **compulsory.** Attempt any **three** questions from the rest.

**1. (a) Define SRS. What are basic issues that the SRS writer should address?**

**Ans. System requirement specification(SRS)** is a set of complete and precisely stated properties along with the constraints of the system that the software must satisfy. A well designed software requirements specification establishes boundaries and solutions of system to develop useful software. All tasks, however minute, should not be underestimated and must form part of the documentation.

The basic issues that the SRS writer(s) shall address are the following:

**(a) Functionality:** What is the software supposed to do?

**(b) External interfaces:** How does the software interact with people, the system's hardware, other hardware, and other software?

**(c) Performance:** What is the speed, availability, response time, recovery time of various software functions, etc.?

**(d) Attributes:** What are the portability, correctness, maintainability, security etc. considerations?

**(e) Design constraints imposed on an implementation:** Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

**(b) Write the importance of feasibility study. Describe different type of feasibility reports.**

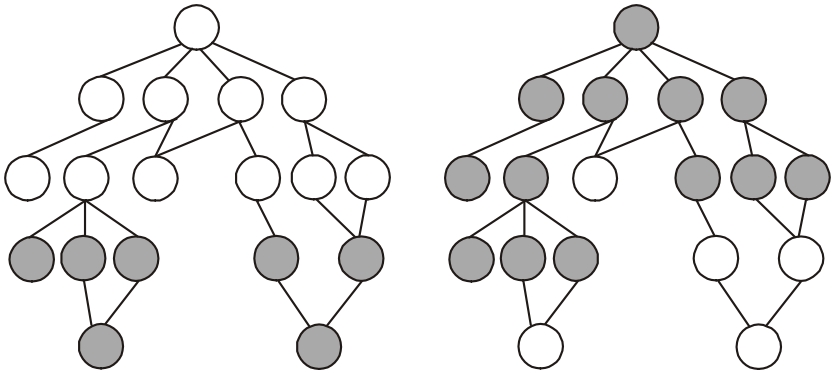
**Ans. Importance of a feasibility study:**

- Give focus to the project and outline alternatives
- Narrow business alternatives
- Surfaces new opportunities through the investigative process
- Identifies reasons not to proceed
- Enhances the probability of success by addressing and mitigating factors early on that could affect the project
- Provides quality information for decision making
- Helps to increase investment in the company
- Provides documentation that the business venture was thoroughly investigated
- Helps in securing funding from lending institutions and other sources

Types of feasibility study reports:
**Now Refer :** Chapter-5, Q.No.-5, Page No.-43

**(viii) Cultural Feasibility study:** In this stage, the project's alternatives are evaluated for their impact on the local and general culture. For example, environmental factors need to be considered.
**(ix) Marketing Feasibility study:** This will include analysis of single and multi-dimensional market forces that could affect the commercial, along with the company that is carrying out the feasibility achieving more and more reputation as they have carried out safety checks which allow the system to run appropriately.
**(x) Schedule feasibility :** Looks at how long the system will take to develop, or whether it can be completed in a given time period using some methods like payback period.

**(c) List principles of system design. Also, differentiate between Top-Down and Bottom-Up design with the help of a suitable example.**
**Ans. First Refer :** June-2007, Q.No.-1(c), Page No.-228
Top down–integration or design proceeds down the invocation hierarchy, adding one module at a time until an entire tree level is integrated; and thus it eliminates the need for drivers. The bottom-up strategy works similarly from the bottom and has no need of stubs. Both the approaches are shown in the following figure :



Bottom-up integration Top-down integration

Integration should follow the lines of first putting together those subsystems that are of great concern. This prioritisation might dicatate top-down integration if control and the user interface were the most worrisome or complex part of software. This can occur, for instance, if the customer is anxious to see a running program early in the development phase. In another situation, the machine interface and performance might be of special interest, and then

bottom-up integration would be dictated. With bottom-up integration we stand better chance of experiencing a high degree of concurrency during our integration. It has also been said that *top-down design is an exercise in faith that everything will work out well, whereas bottom-up design shows that things really do work.*

**(d) Why is Joint Application Development (JAD) used? Explain the different participants in JAD. Also, write the various benefits of JAD.**
**Ans.** Refer to Chapter-5, Q.No.-6, Page No.-45

**(e) Differentiate between Decision Support System (DSS) and Expert System. Also, write the main characteristics of both.**
**Ans.**

**Differentiate between Decision Support System and Expert System**

| | **Decision Support System (DSS)** | **Expert System (ES)** |
|---|---|---|
| **Objective** | To assist human decision | To mimic human decision |
| **Reasoning capability** | No or limited | Yes |
| **Database** | Adhoc, factual information | Procedural and factual knowledge |
| **Domain** | Broad | Narrow, very specific domain |
| **Types of Data** | Numerical, character based | Mostly symbolic |
| **Direction of Query** | Human to machine | Machine to human |
| **Decision Maker** | Human takes the decision with support from DSS | Computer makes the decision |

**DSS possess the following desirable characteristics:**
**1.** DSS should aid the decision maker in decision making.
**2.** DSS should be able to address semi/unstructured decision making situations.
**3.** DSS should support decision makers particularly at tactical/strategic levels.
**4.** DSS should be able to create general purpose models, simulation capabilities and other analytical tools available to decision maker.
**5.** DSS should enable users to use DSS without assistance from MIS/Technical professionals.
**6.** DSS should be readily adapted to meet information requirement for any decision environment.

**The following characteristics of the Expert System:**
**(a)** Ability to explain their reasoning or suggested decisions.
**(b)** Ability to display "Intelligent" behaviour.
**(c)** Ability to draw conclusions from complex relationship.

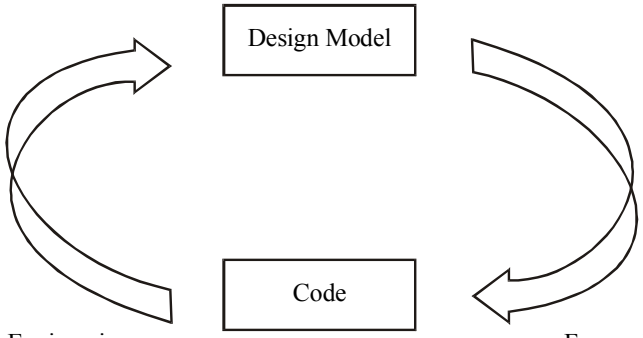
**(d)** Ability to provide 'Portable Knowledge'.
**(e)** Ability to deal with certainty.
**(f)** Not widely used or tested, due to difficulty of use.
**(g)** Limited to relatively narrow problems.
**(h)** Inability to deal with 'Mixed Knowledge'.
**(i)** Inability to refine own knowledge base.
**(j)** Difficult to maintain.

**2. (a) What are the important disadvantages of CASE tools? Explain, how CASE tools are used in reverse and forward engineering.**

**Ans. Disadvantages of CASE Tools:** The following are some of the disadvantages of CASE tools:

**1.** Complex functionality of an organization.
**2.** Many project management problems are not amenable to automation. Hence, CASE tools can't be used in such cases.
**3.** CASE tools cannot automatically provide a functional, relevant system.
**4.** They cannot easily interface with database and 4GL.
**5.** They cannot radically transform the systems analysis and design process.
**6.** They cannot automatically force analysts to use a prescribed methodology when one does not exist.
**7.** There can be many project problems which are not amenable to automation. Thus CASE tool implementation fails.

**Reverse and Forward Engineering are the two important concepts related to CASE :** Below is the Figure depicting both Forward and Reverse Engineering.




Reverse Engineering is the process of recreation of model based on existing code. First, the existing code is scanned to generate the model. Then the model can be fine turned in accordance with requirements. Reverse engineering allows developers to create model for old systems, which were never modeled. It analyses existing software with purpose of understanding its design and specification. Reverse engineering tools read program source code and create

graphical and textual representation of design.
Forward engineering is the process of generation of skeleton code out of the models. First step is to create the model for a system, then generate the relevant code for the model and then allow modification of this code in tune with the requirements.

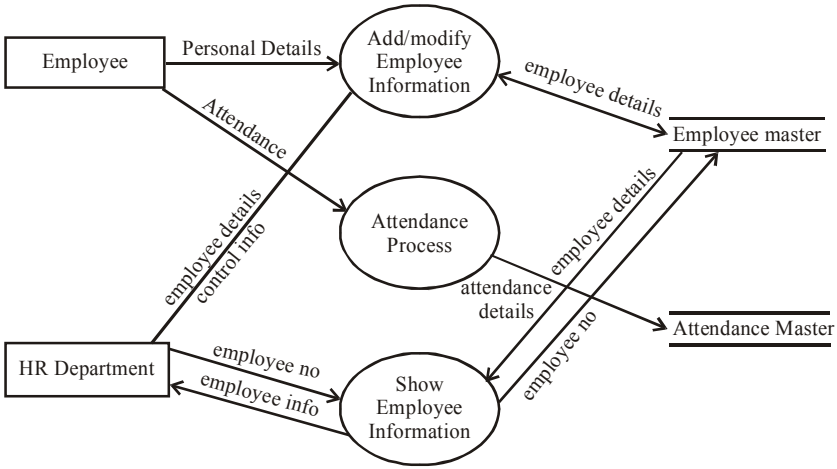
**(b) Explain required steps taken by software development team at the time of delivery of software.**

**Ans.** As IT organizations push to keep up with business, they realize they must find a way to add agility and simplification to their existing software delivery process. SOA has been touted as the great white hope but architecture alone will not make it happen. What is needed is a set of best practices that supports a new collaborative and interactive software delivery processes. The software development team is required to take the following steps at the time of software delivery:

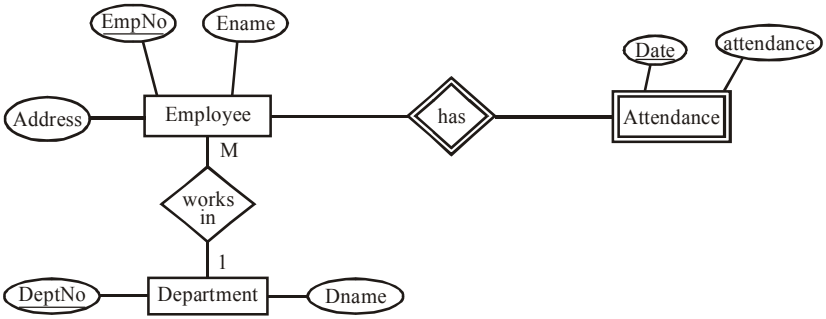
- How to deliver applications that meet the needs of your customer – on time and within budget
- How to blend existing steps within the software delivery lifecycle to speed up application development and deployment
- The reason business services are the building blocks of SOA-based applications
- How building out services allows you to leverage other investments
- How to incorporate collaborative process design concepts and best practices into application development and deployment

**3. (a) Design the detailed process modelling and data modelling diagrams for employee information system. Make necessary assumptions.**

**Ans.** Process Modelling : Level 1 DFD



Data Modelling : ER Diagram




**(b) "Inaccurate, incomplete, out of date, of missing documentation is a major contributor to poor software quality." Justify the statement. Also, explain how software maturity model (CMM) contributes in the quality of a software.**

**Ans.** Inaccurate, incomplete, out of date, or missing documentation is a major contributor to poor software quality. That is why documentation and document control has been given due importance in ISO 9000 standards, SEI CMM software Maturity model. In SEI CMM Process Model and assessment procedure, the goal is to improve the documentation process that has been designed. A maturity level and documentation process profile is generated from the responses to an assessment instrument.

One basic goal of software engineering is to produce the best possible working software along with the best possible supporting documentation. Empirical data show that software documentation products and processes are key components of software quality. Studies show that poor quality, out of date, or missing documentation are a major cause of errors in software development and maintenance. Although everyone agrees that documentation is important, not everyone fully realizes that documentation is a critical contributor to software quality. Documentation developed during higher maturity levels produces higher quality software.

**4. (a) Define process, procedure and process specification tools. Make the decision table equivalent to the following structured English notation:**
**If balance $\leq$ 0**
**then {issue exception note}**
**else if balance in account < minimum balance**
**then {issue warning to department}**
**else {issue items to department}**

**endif**
**endif**

**Ans.** Refer to Chapter-7, Q.No.-5, Page No.-70

**Decision table for the given situation :**

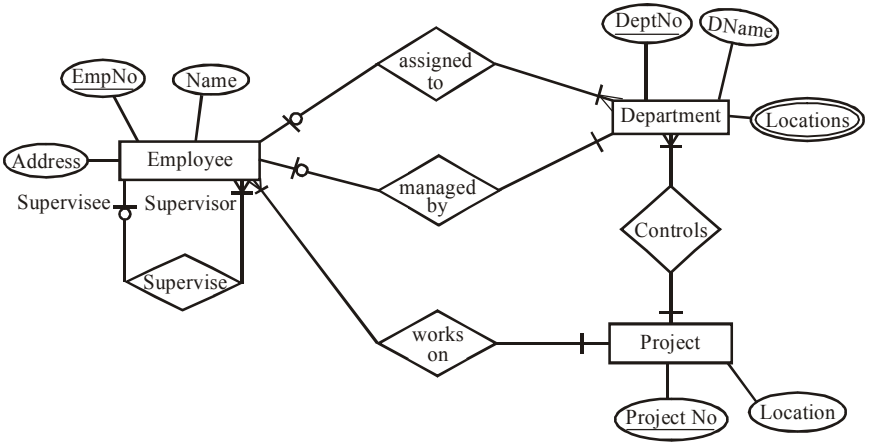
C1=balance in account <=0;
C2=balance in account < min. balance
A1=issue exception note and mark "no future issues"
A2=issue item to dept
A3=issue warning to dept.

| C1 | N | N | N |
|---|---|---|---|
| C2 | – | Y | N |
| A1 | X | – | – |
| A2 | – | X | X |
| A3 | – | X | – |

**(b) Draw E – R diagram for statement written below:**

**"The LG company is organized into departments. Each department has a particular employee who manages the department. A department may have several locations. A department controls a number of projects. Each project has a single location. An employee is assigned to one department but may work on several projects. Every employee has a direct supervisor.**

**Ans.**




**5. (a) Explain the important tasks performed by a Systems Analyst.**

**Ans.** Refer to Chapter-1, Q.No.-3, Page No.-8

**(b) Write the importance of testing in software development. Explain**

**the important contents of Test Design Document.**

**Ans.** The importance of testing in software development is to ensure that stable software is operating your business, serving your customers, and supporting your partners, you must make testing an integral part of your software development process.

During system development, this document provides the information needed for adequate testing. It also lists approaches, procedures and standards to ensure that a quality product that meets the requirement of the user is produced. This document is generally supplemented by documents like schedules, assignments and results. A record of the final result of the testing should be kept externally.

This document provides valuable input for the maintenance phase.

The following IEEE standards describe the standard practices on software test and documentation:

**1.** 829-1998 IEEE Standard for Software Test Documentation

**2.** 1008-1987 (R1993) IEEE Standard for Software Unit Testing

**3.** 1012-1998 IEEE Standard for Software Verification and Validation

**The following are the important contents of Test Design Document:**

**1. Introduction**

**Purpose :** The purpose of this document and its intended audience are clearly stated.

**Scope :** Give an overview of testing process and major phases of the testing process. Specify what is not covered in the scope of the testing such as, supporting or not third party software.

**Glossary :** It gives definition of the technical terms used in this document.

**References :** Any reference to other external documents stated in this document including references to related project documents. They usually refer the System Requirement Specification and the System Design Specification documents.

**Overview of Document :** Describe the contents and organization of the document.

**2. Test Plan :** A test plan is a document that describes the scope, approach, resources and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, and the person who will do each task, and any risks that require contingency planning.

**2.1 Schedules and Resources :** An overview of the testing schedule in phases along with resources required for testing is specified.

**2.2 Recording of Tests :** Specify the format to be used to record test results. It should very specifically name the item to be tested, the person who did the testing, reference of the test process/data and the results expected by the test, the date tested. If a test fails, the person with the responsibility to correct and retest is also documented. The filled out format would be kept with the specific

testing schedule. A database could be used to keep track of testing.

**2.3 Reporting test results :** The summary of what has been tested successfully and the errors that still exist which are to be rectified is specified.

**3. Verification Testing**

**3.1 Unit Testing :** For each unit/component, there must be a test which will enable tester to know about the accurate functioning of that unit.

**3.2 Integration testing :** Integration test is done on modules or sub-systems.

**4. Validation Testing**

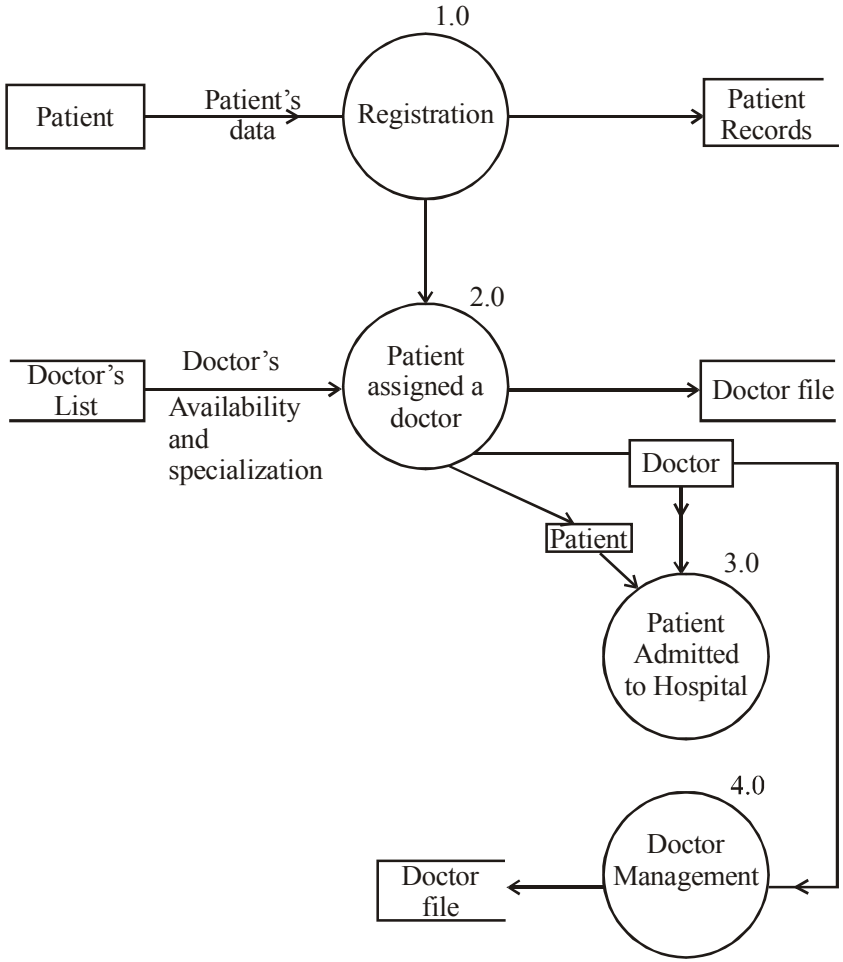
**4.1 System Testing :** This is the top level of integration testing. At this level, requirements are validated as described in the SRS.

**4.2 Acceptance and Beta Testing :** List test plans for acceptance testing or beta testing. During this test, real data is used for testing by the development team (acceptance testing/alpha testing) or the customer (beta testing). It describes how the results of such testing will be reported back and handled by the developers.

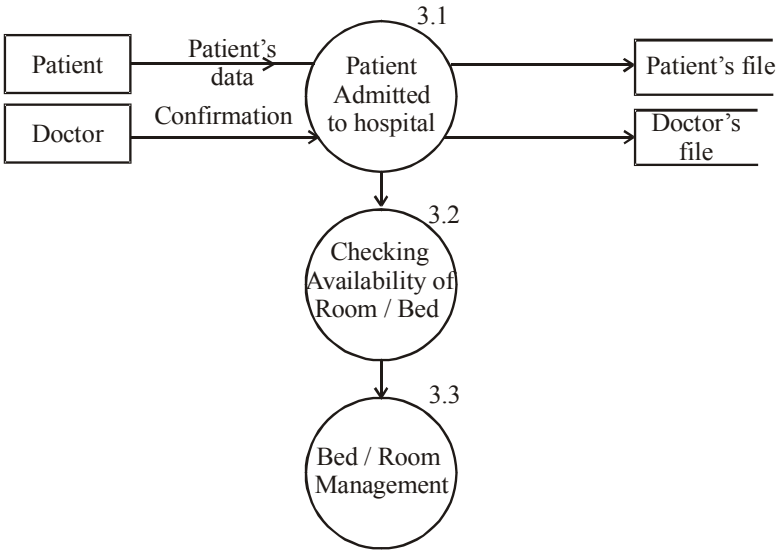
**MCS-14 : Systems Analysis and Design**
**June, 2008**

**Note :** Question number 1 is **compulsory.** Attempt any **three** questions form the rest.

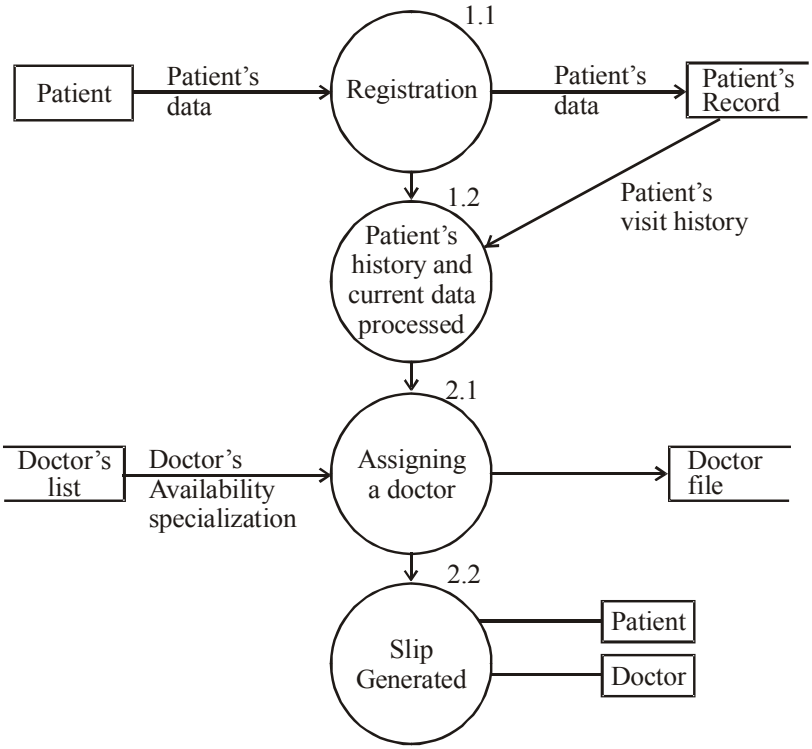
**Q1. (a) Design a SRS Document for a hospital management system. Also draw a DFD (context-level), (1-level) and (2-level) for the above said system.**
**Ans.**



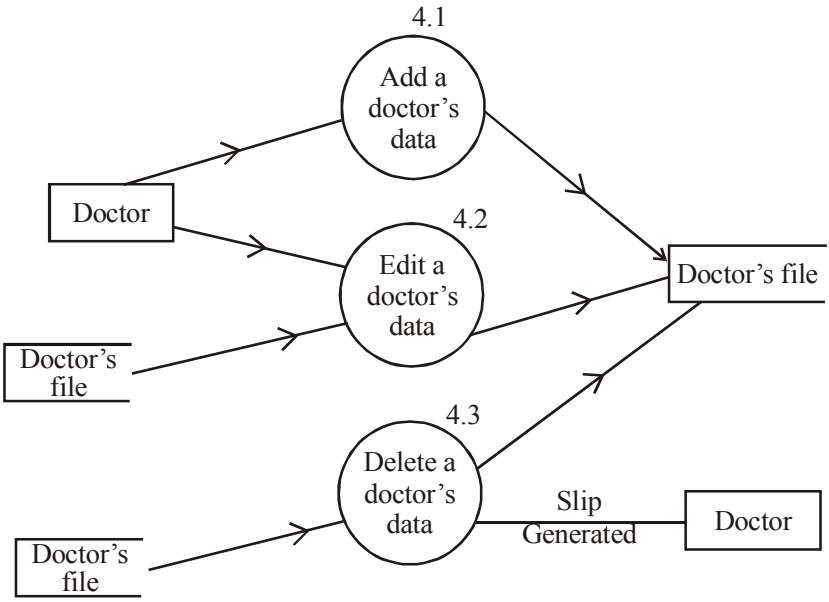

Zero level DFD for
Hospital Management System




Ist level DFD for Process 3.0




I[st]level DFD for Process 1.0 and 2.0




Ist level DFD for Process 4.0

**Software Requirements Specification Document (SRS)**

**1** Abstract

This document follows the IEEE standard for a requirements specification document, with some variations.

**2** Introductions

**2.1** Purpose

The purpose of this document is to describe the external requirements for hospital management system. It also describes the interfaces for the system.

**2.2** Scope

This document is the only one that describes the requirements of the system. It is meant for use by the developers and will be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

**2.3** Definitions, Acronyms, Abbreviations

Not applicable.

**2.4** References

Not applicable.

**2.5** Developer's Responsibilities

The developer is responsible for (a) developing the system, (b) installing the

software on the client's hardware, (c) conducting any user training that might be needed for using the system, and (d) maintaining the system for a period of one year after installation.

**3. General Descriptions**

**3.1 Problem Statement**

Create a Management System for a hospital with the following features

• **Registration** - This feature allows to add/edit/delete patients in the system.

• **Patient Management** - Before we let a user to visit a doctor, we should classify whether he is an in-patient or an out-patient. Then basic symptoms should be captured and entered in the system. Existing patients can be searched by name, admission date, patient ID etc.

• **Patient Visit History -** Previous visit histories of the patient can be accessed

• **Assign Doctor -** Assign a patient to a doctor based on the basic information of the patient's problems. Based on the specialization and availability of the doctors, patients should be assigned.

• **Doctor Management** - Add/Edit/Delete the doctors working in the hospital.

• **Bed/Room Management** - The beds or rooms in a hospital can be managed. Add/Edit/Update the rooms with details. Check availability and assign a room to in-patients

• Doctors can update the patient's record with their observations.

**3.2 User Characteristics**

The main users of this system will be department secretaries, who are somewhat literate with computers and can use programs such as editors and text processors.

**3.3 General Constraints**

The system should run on Sun 3/50 workstations running UNIX 4.2 BSD.(can vary)

**3.4 General Assumptions and Dependencies**

Not applicable.

**4. Specific Requirements**

**4.1 Inputs and Outputs**

The system has inputs and produces outputs.

Input file 1: Contains the list of room numbers and their capacity; a list of all the patients in the patients file; and the list of various doctors. The format of the file is :

rooms

room1 : cap1

room2 : cap2

:
;
;
times
time1 , time2 , time3 ;
Input file 2: Contains information about the doctors visiting the hospital. For each doctor it specifies the code number, and various details.

**Reports :**
The following reports can be generated. You can implement more reports which you think can be useful.
• Utilization of doctors
• List of Patients diagnosed by a disease
• List of patients who visited during a given time period

**Output** : Error messages. At the minimum, the following error messages are to be given:
e1. Input file does not exist.
e2. Input-file-1 has error
e2.1. The code number has wrong format
e2.2. time has wrong format.
e2.3. room number has wrong format.
e2.4. capacity out of range.
e3. Input-file-2 has error
e3.1.code does not exist
**4.2** Functional Requirements
**4.3** External Interface Requirements

The important roles associated with the system would be
**Administrator** – Administrator will have complete control of the system. She/he can Add/Edit/Delete patients, Add/Edit/Delete Doctors, Add/Edit/Delete Beds, Search for patients, Assign patients to doctors.
**Doctor** – Doctor can access a patient's record and update his observations about the patient in that particular visit
**User Interface :** Only one user command is required. The file names can be specified in the command line itself or the system should prompt for the input file names.
**4.4** Performance Constraints
For input file 2 containing 10 patients details and their respective doctor's detail, the reports should be printed in less than 1 minute.

**4.5** Design Constraints
Software Constraints
The system is to run under the UNIX operating system.
Hardware Constraints
The system will run on a Sun workstation with 256 MB RAM, running UNIX. It will be connected to an 8-page-per-minute printer.
**5. Acceptance Criteria**
Before accepting the system, the developer must demonstrate to the user that the system works on the data for the last 4 years. The developer will have to show through test cases that all conditions are satisfied.

**(b) What do you mean by Risk Management ? Categorize Risk and how we can assess these risks.**
**Ans.** Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem – it might happen, it might not.
**(a) Reactive Risk Strategy :** At best, a reactive strategy monitors the project for likely risks. Resources are set aside to deal with them, should they become actual problems. More commonly, the software team does nothing about risks until something goes wrong. Then, the team flies into action in an attempt to correct the problem rapidly.
**(b) Proactive Risk Strategy :** A more intelligent strategy for risk management is to be proactive. A proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability and impact are assessed, and they are ranked by importance. Then, the software team establishes a plan for managing risk. The primary objective is to avoid risk, but because not all risks can be avoided.
Risk categorization involves developing a list of all project risks, in part using a list of the type. For each risk determine first whether the probability that the risk will occur is high, medium, or low, then whether the impact of the risk, should it materialize is high, medium or low. Finally, use a cross-reference such as the following figure to establish the degree of each risk.
Risk categorization also includes one more step. The project manager will need to decide which risks to focus on. Clearly the extreme or high risks merit attention: by definition, they are both probable and dangerous. Whether the project manager deals with the medium risk becomes a question of the time available and the ease with which a risk can be mitigated. Ideally, every risk should be managed, but in reality, a project manager will not have the time to handle all of them, so that risk management becomes a balancing act between the degree of a risk and the effort needed to manage it. Nevertheless, there are some medium and even low risks that are easily mitigated. For example, the

impact of a fire can be reduced by the use of off-site backups.

| Probability | Impact | | |
|---|---|---|---|
| | High | Medium | Medium |
| High | Extreme | High | Medium |
| Medium | High | Medium | Low |
| Low | Medium | Low | Minimal |

Fig : Determination of Degree of Risk

**(c) What are the attributes of a good system analyst? Also explain the need of documentation in a system design.**
**Ans.** First Refer to Q3, Page No.-8, Now Refer to Q4, Page No.21

**(d) Explain the testing phase of SDLC in detail.**
**Ans. Testing Phase:** The code is tested at various levels in software testing. Unit, system and user acceptance testing are often performed. This is a grey area as many different opinions exist as to what the stages of testing are and how much if any iteration occurs. Iteration is not generally part of the waterfall model, but usually some occurs at this stage.

**Types of testing :**
**Unit testing :** Unit testing is a method of testing that verifies the individual units of source code are working properly. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class.
**System testing :** 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.
**Integration testing :** 'Integration testing' (sometimes called Integration and Testing, abbreviated I&T) is the phase of software testing in which individual

software modules are combined and tested as a group.

**Black box testing :** Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid input and determines the correct output. There is no knowledge of the test object's internal structure.

**White box testing :** White box testing (a.k.a. clear box testing, glass box testing or structural testing) uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software.

**Module testing :** At this step of testing a smallest working unit called module is tested alone.

**Regression testing :** Regression testing is any type of software testing which seeks to uncover software regressions. Such regressions occur whenever software functionality that was previously working correctly stops working as intended. Typically regressions occur as an unintended consequence of program changes.

**Automation testing :** Test automation is the use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions.

**User acceptance testing :** acceptance testing is black-box testing performed on a system (e.g. software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery.

**Q2 (a) Explain Joint Application Development (JAD) and prototyping as an approach for Information System development.**

Refer to Q6, Page No.-45

**(b) Give five tasks performed by a project team leader in software development.**

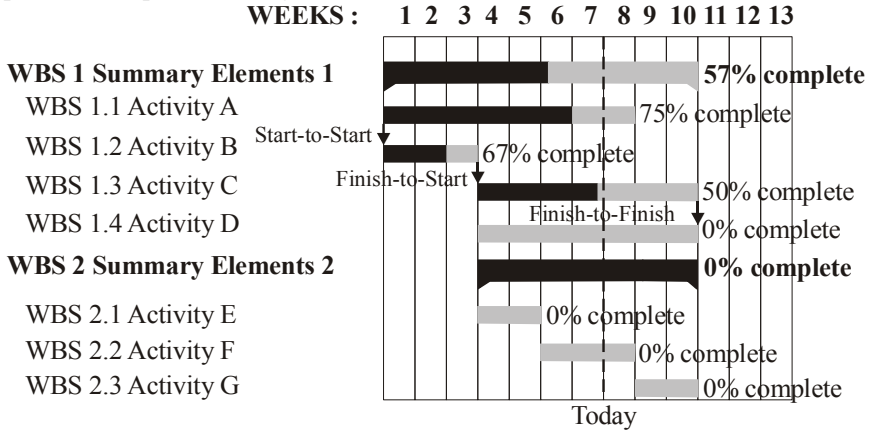
**Ans.** Ensure the organization's strategic plan, mission, vision and values are communicated to the team and integrated into the team's strategies, goals, objectives, work plans and work products and services. Articulate and communicate to the team the assignment, project, problem to be solved, actionable events, milestones, and/or program issues under review, and deadlines and time frames for completion. Coach the team in the selection and application of appropriate problem solving methods and techniques, provide advice on work methods, practices and procedures, and assist the team and/or individual members in identifying the parameters of a viable solution. Lead the team in: identifying, distributing and balancing workload and tasks among employees

in accordance with established work flow, skill level and/or occupational specialization; making adjustments to accomplish the workload in accordance with established priorities to ensure timely accomplishment of assigned team tasks; and ensuring that each employee has an integral role in developing the final team product. Train or arrange for the training of team members in methods and techniques of team building and working in teams to accomplish tasks or projects, and provide or arrange for specific administrative or technical training necessary for accomplishment of individual and team tasks. Monitor and report on the status and progress of work, checking on work in progress and reviewing completed work to see that the supervisor's instructions on work priorities, methods, deadlines and quality have been met. Serve as coach, facilitator and/or negotiator in coordinating team initiatives and in consensus building activities among team members.

**(c) Explain categories of feasibility analysis.**
Refer Q.No.7, Page No.-46,47,48

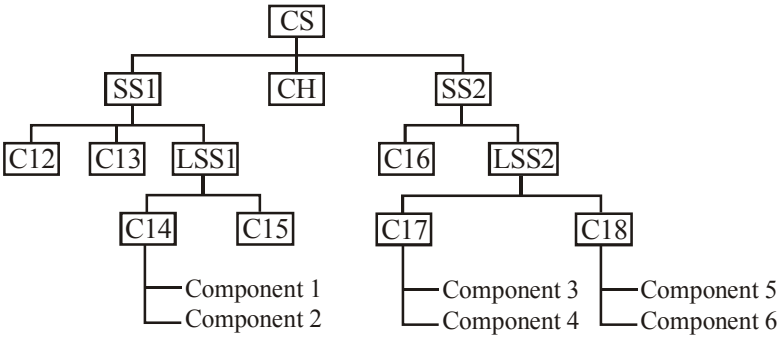
**Q3. (a) Give a diagram each of a Gantt chart and Structured chart. Explain their functionality also.**

**Ans.** A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency (i.e, precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.




A Gantt chart showing three kinds of schedule dependencies and percent complete indications.

A **Structure Chart** (CS) in software engineering and organizational theory is a chart, that shows the breakdown of the configuration system to the lowest manageable levels.
It is used to show the hierarchical arrangement of the modules in a structured program. Each rectangular box represents on module. The names of the modules are written inside the box. An arrow joins two modules that have an invocation relationship.




A Configuration System Structure Chart.

**(b) Explain the techniques for gathering of requirements for system analysis. Give their examples also.**
Refer (b), Page No.-202

**(c) Differentiate between a bottom-up and a top-down design (with diagrams)**
Refer (c), Page No.-240

**Q4. (a) Categorize cohesion and coupling. Also discuss the relative performance and advantages of each type of cohesion and coupling.**
Refer Page No.-57, 58, Now Refer Q5, Page No.-55

**(b) Explain logical and physical design with appropriate diagrams.**
Refer Page No.-63,64,65

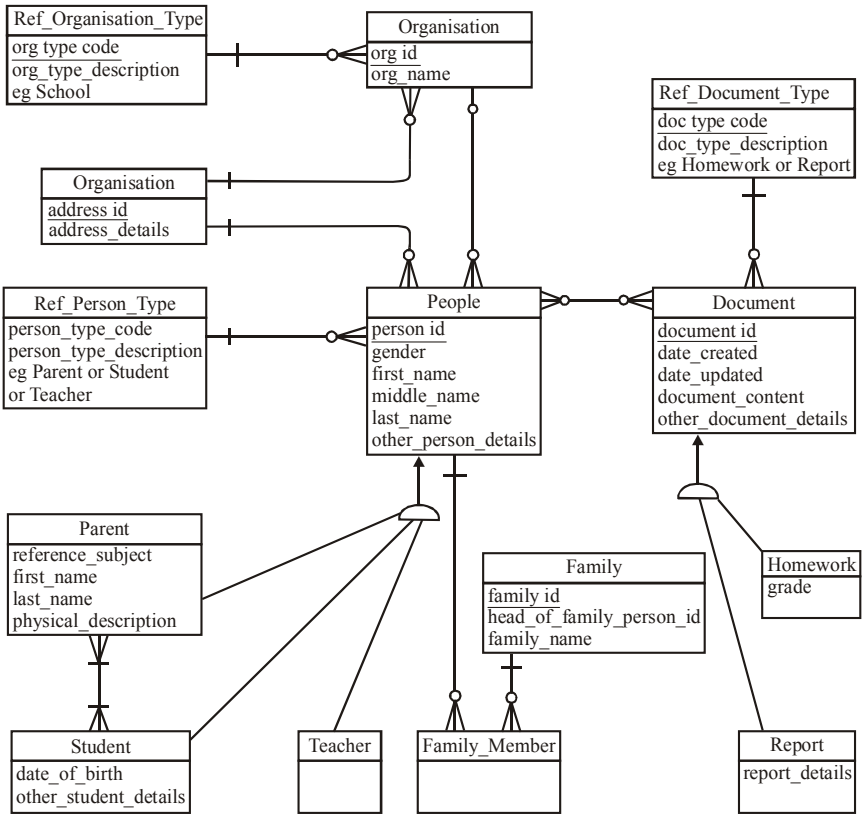
**(c) Categorize system maintenance. Explain the types in brief.**
**System maintenance** can be categorized into three groups, namely corrective, preventive, and perfective maintenance.
Now Refer to Q.No.-4, Page No.-109

**Q5 (a) What are the factors which concern a form and report design? What are the types of information which have to be delivered by a form or a report?**

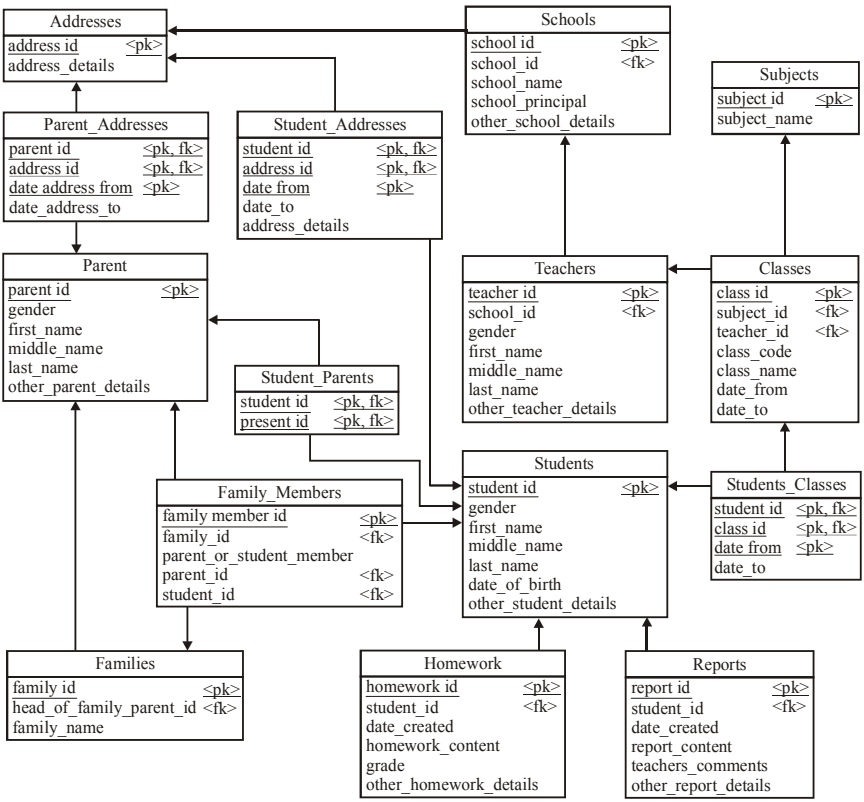
Refer to Q5, Page No.-82. Now Refer to Q4(a), December-2008.

**(b) Construct ERD for a School Management System. Explain it.**

ERD for a SMS




Physical Data Model is given to understand the above E-R Diagram properly.




**The Area being Modelled is :** A School Management System. <pk> implies "Primary Key", <fk> implies foreign key.

**The User Requirements have been defined as follows :-**

**A. Generic School Management System :** A Database is required to support an Online school system which should be so generic that any number of schools in an area, (e.g. Town/country/world) can register to use the System. The Database should include Teachers,Student,Parents and so on.

**(1)** A Teacher could submit reports, datasheets of student with their remarks.

**(2)** Parents, (after entering their Password) could view the progress, dues, notices etc. of their children plus any query or discussion regarding their child etc..

**(3)** Student can view their progress, notices, dues.....with their passwords. This would be like a general platform to provide School-Specific facilities.

**B. The Things of Interest, ('THINGS'), include -**

B.1 Attendance

B.2 Classes

B.3 Parents

B.4 Register
B.5 Schools
B.6 Students
B.7 Teachers

**C. These THINGS are Related as follows :**

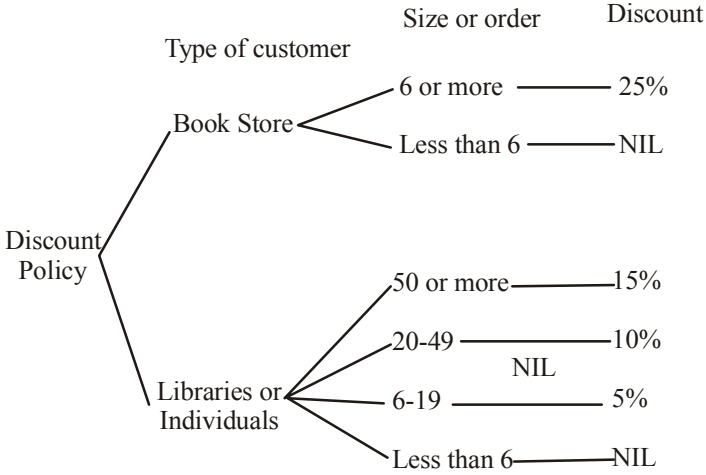
C.1 A STUDENT can register for zero, one or many CLASSES.

**(c) Explain the following :**

**(i) Decision trees**

**Ans.** Once the data elements are defined in the data dictionary, we begin to focus on the process. For example, we need to know what goes into particular transaction. For example, particular transaction. For example, particular book store gets various discount on the purchase of the book from the publishers, let us assume the following discount policy :

Book stores get a trade discount of 40% for orders from libraries and individuals, 50% allowed on orders of 6-19 copies for a book title: 10% on order for 20-49 copies per book title: 15% on orders for 50 copies or more per book title. A policy statement like this can be time-consuming to describe and confusing to implement. The analyst needs to use tools to implement the logic of the policy. The first such tool is DECISION TREE. Decision Tree has many branches as there are logical alternatives. It is excellent tool, below is an example of Decision Tree :




**(ii) CASE tools**

Refer to Page No.-98

**(iii) MIS**

Refer to Q1, Page No.-120

**(iv) Expert systems**

Refer to Q4, Page No.-124

## MCS-14 : Systems Analysis and Design
## December, 2008

**Note :** Question number 1 is **compulsory.** Attempt any **three** questions form the rest.

**Q1 (a) Explain Analysis and Design phases of Software development life cycle in detail. Give a problem statement of a problem of your choice and implement the states of analysis and design and it.**
Refer to Page No.-15, Now Refer to Q29, Page No.-153

**(b) What are the steps in a database design? Categorize file organisation and give examples to explain the categories.**
Refer to Page No.-86, Now Refer to Page No.-90

**(c) Develop and design Entity Relationship Diagram (ERD) and 0-level and 1-level DFDs for a student admission system. Design a decision tree and a data dictionary to give the design level details of the system.**
First Refer to Q5(b), June-2008. Now Refer to (e), Page No.-210
Now Refer to www.gullybaba.com/ignou/mcs14

**(d) How can security be implemented in a MIS based project?**
**Ans.** In MIS based project we can have different modules for different level of management as well as for different departments.
• To secure the data among different departments, we have to give access rights of particular department module to the users which belongs to that department only.
• To secure the data among hierarchy of management levels, we have to provide the accessibility of the information and/or reports which are relevant for that level of user only.

**(e) Explain any five types of feasibility study.**
Refer to Q5, Page No.-43

**Q2 (a) List any ten components of software user documentation.**
Refer Q15, Page No.-38

**(b) Explain in detail, the analytical and management skills of a system analyst.**
**Ans. Analytical Skills :**Ability to see things as systems, identify, analyze, and solve problems in an optimal way for a specific organization.

**Management Skills :** Include organization's recourse management, project management (people and money), risk management, and change management.
Now Refer to Q6, P10

**(c) What is an object interaction diagram. Give an example.**
**Ans.** The object-interaction diagram of UML depicts the massages and message arguments that objects send to one another. This diagram performs a job similar to the structure chart in prehistoric structured techniques: It shows the run-time communication structure of the system.
The object-interaction diagram (often referred to as the interaction diagram) is ideal for modeling the procedure of a single use case, which is defined in (Jacobson 1992) as a "behaviorally related sequence of transactions in a dialogue with a system." In other words, an object-interaction diagram often depicts the dialogue between a user and a system as they perform a useful unit of work, such as entering a purchase order or scheduling a machine run.
The object-interaction diagram comes in two styles : the collaboration diagram and the sequence diagram. The two styles of diagram capture the same content-indeed, one style of diagram can be automatically converted into the other.
The collaboration diagram tends to be more free-form and flexible in how it's drawn. It also gives similar weight to static object relationships and dynamic object messaging. The sequence diagram, on the other hand, is more tightly organized and emphasizes temporal sequence.

**Q3. (a) Explain the structures of a typical text design document.**
Refer to Q5(b), Page No.-245,246

**(b) Give a short note on Expert Systems. Where does an Expert System find its utility in a MIS organisation?**
Refer to Q4, Page No.-124

**Q4 (a) Explain the three types of information along with examples.**
Refer to Q4, Page No.-82

**(b) Explain the four guidelines for displaying contents of Form or Report.**
**Ans.** The way the form or a report appears to the human eye has a lot of impact on the user and by following specific guidelines for highlighting information such as using colour to display text, and presenting numeric tables and lists, we can make the form or report more presentable.
**Highlighting Information :** Highlighting the information will enhance the appearance of the output. However, highlighting should be used sparingly to draw the user to or away from certain information and to group together

related information.
Highlighting will be very useful in situations such as the following :
• Notifying users of errors in data entry or processing;
• Drawing attention to high priority messages; and
• Providing warnings to users in situations like unusual data values or an unavailable device.
**Using Colour :** Colour influences the usability of the system. Use of appropriate colours while designing has several advantages.
**Displaying Text :** Now a days, as the text based applications such as e-mail, bulletin boards, chatting etc., are being widely used, textual output is becoming increasingly important.
**Designing Tables and Lists :** The content and meaning of tables and lists are significantly derived from the format of the information. There are few simple guidelines that are to be followed while designing tables and lists. Use meaningful labels to all columns and rows and separate labels from other information by using highlighting.
Numeric, textual and alphanumeric data should be properly formatted. For example, numeric data should be right justified and columns should be aligned using decimal points or other delimiter.
Textual data should be left justified and line length should be somewhere between 30-40 characters per line. If there are long sequences of alphanumeric data, then they should be broken into small groups of three to four characters each.

**Q5 (a) List any 10 benefits of audit.**
Refer to Page No.-113

**(b) Explain process modelling and data modelling.**
**Ans.** The term process model is used in various contexts. For example, in business process modeling the enterprise process model is often referred to as the business process model. Process models are core concepts in the discipline of Process Engineering.
Data modeling in software engineering is the process of creating a data model by applying formal data model descriptions using data modeling techniques.
Data modeling is a technique for defining business requirements for a database. It is sometimes called database modeling because a data model is eventually implemented in a database.
**Now Refer to Q3(a), P243**

## MCS-14 : Systems Analysis and Design
## June, 2009

**Note :** Question number 1 is **compulsory.** Attempt any **three** questions form the rest.

**Q1. (a) Draw an ERD (Entity Relationship Diagram) for university admission system. Explain its cardinality. Also, draw DFDS for the same system upto level-I.**
Refer to Chapter-13, Page No.-152

**(b) Construct a data dictionary for a Railway Reservation System. Make necessary assumptions.**
Refer to Chapter-4, Q.No.-11, Page No.-35

**(c) Explain the following:**
**(i) Prototyping**
Refer to Chapter-5, Q.No.-6(ii), Page No.-46

**(ii) User acceptance Testing (with examples)**
**Ans.** An acceptance test has the objective of selling the user on the validity and reliability of the system. It verifies that the system's procedures operate to system specifications and that the integrity of important data is maintained. Performance of an acceptance test is actually the user's show. User motivation is very important for the successful performance of the system. After that a comprehensive test report is prepared. This report shows the system's tolerance, performance range, error rate and accuracy.

**(d) What do you understand by CASE tools? Categorise the Case tools. What are the advantages and disadvantages of CASE tools?**
Refer to Chapter-10, Page No.-100-101, and now
Refer to June-2006, Q.No.-3(a), Page No.-200

**Q2. (a) Explain different types of User Documentation. What will be the typical content of a User Manual?**
Refer to Chapter-4, Q.No.-14, 15, Page No.-37

**(b) What is meant by Feasibility Study? Explain various types of Feasibility Study.**
Refer to Chapter-5, Q.No.-5, Page No.-43

**Q3. (a) Describe the rules and steps in designing a Form and a Report. Give an example of each.**

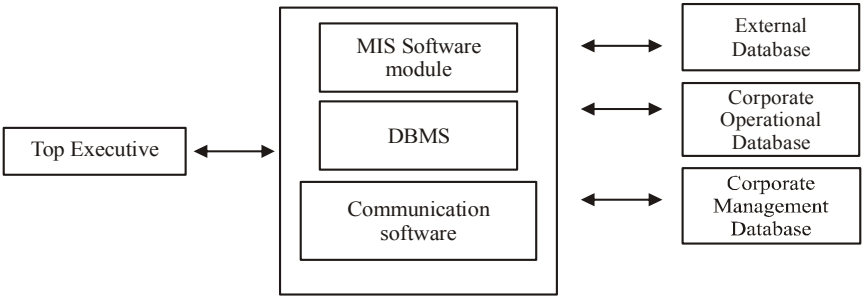
Refer to Dec-2005, Q.No.-4(b), Page No.-191 and now
Refer to Chapter-8, Q.No.-3, Page-80

**(b) Explain the role of system audit in a software development life cycle. Discuss the responsibility and authority of the System Auditor.**

Refer to Chapter-12, Q.No.-2, Page No.-112

**(c) Discuss the various components of an MIS along with diagram.**

**Ans. Components of MIS:**




**External Database:** External databases are databases that are not owned by the organization and the organization pays royalty to access these databases. Examples of these databases are: databases of Market research groups, Statistical and Demographic organizations etc. Since organization operates in a social environment it is influenced by various external factors. Impact of these external factors on the long-term goal and success of organization is very important. Top management needs to analyse data from these sources for long term planning.

**Corporate Database:** Corporate database stores data generated by various business process through transaction processing Systems. These can be employee database, customer database, inventory database etc.

**Management Database:** These databases store select data from corporate databases. It generally stores summarized information for the requirements of managers.

**MIS Software:** This is used to extract and process information from various databases. It acts as a user interface to the managers.

**DBMS:** Database Management System stores, retrieves and manages data on various databases.

**Communication Software:** This is used to communicate with customers, suppliers and other stakeholders of the organization. Examples are Messaging Software or Organization's Bulletin board.

**Q4(a). What is SRS? Why is it required? What is the step after development of SRS?**

**Ans.** SRS stands for software requirement specification. It is a document in which we write the detailed description about client requirement and their respective solutions in it. It is a set of complete and precisely stated properties with the constraints of the system that the software must satisfy.

It is required because:-

**a)** It reduces communication gap between developer team and client team.

**b)** It gives the detailed description about the product to be developed to developer's team.

**c)** It help to co-operate audits and other requirement of the organization.

**d)** It is used it communicate among different teams of software project.

Feasibility study is implemented after SRS developments which includes hardware, software specification for the system and the cost benefit analysis.

**(b) Propose a modified version of Water Fall Model which will remove disadvantages of Water Fall Model.**

**Ans.** The modified waterfall uses the same phases as the pure waterfall, but is not done on a discontinuous basis. This enables the phases to overlap when needed. The pure waterfall can also split into subprojects at an appropriate phase (such as after the architectural design or detailed design).

**Strengths**

More flexibility than the pure waterfall model.

If there is personnel continuity between the phases, documentation can be substantially reduced. Implementation of easy areas do not need to wait for the hard ones.

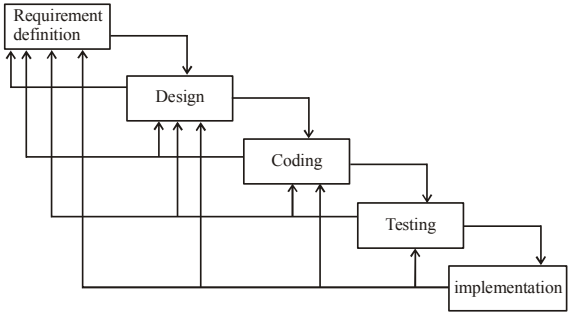
Milestones are more ambiguous than for the pure waterfall.

**Weaknesses**

Activities performed in parallel are subject to miscommunication and mistaken assumptions. Unforeseen interdependencies can create problems.

**Modified Waterfall Summary**

Risk reduction spirals can be added to the top of the waterfall to reduce risks prior to the waterfall phases. The waterfall can be further modified using options such as prototyping. JADs or CRC sessions or other methods of requirements gathering done in overlapping phases.




**(c) Explain following phrases w.r.t. SDLC.**
**(i) Logical Design**
Refer to Chapter-7, 2nd Para, Page No.-63

**(ii) Implementation**
Refer to Chapter-11, Ist Para, Page No.-106, and now
Refer to Chapter-3, Q.No.-1(vi) Page No.-16

**(iii) Maintenance**
Refer to Chapter-11, 2nd Para, Page No.-106, and now
Refer to Chapter-3, Q.No.-1(vii), Page No.-16

**Q5. Write a short note on the following with example of each:**
**(a) Coupling**
Refer to Chapter-6, Q.No.-5, Page No.-55

**(b) Inheritance Diagram**
Refer to Dec-2006, Q.No.-2(b)(iii), Page No.-213

**(c) Indexed Sequential File Organization**
Refer to Chapter-9, Last Para, Page No.-91

**(d) Decision Support System**
**Ans. DSS :** Short for Decision Support system, the term refers to an interactive computerized system that gathers and presents data from a wide range of sources, typically for business purposes. DSS applications are systems and subsystems that help people make decisions based on data that is called from a wide range of sources.

For example: a national on-line book seller wants to begin selling its products internationally but first needs to determine if that will be a wise business decision. The vendor can use a DSS to gather information from its own resources to determine if the company has the ability or potential ability to expand its business and also from external resources, such as industry data, to determine if there is indeed a demand to meet. The DSS will collect and analyze the data and then present it in a way that can be interpreted by humans. Some decision support systems come very close to acting as artificial intelligence agents. DSS applications are not single information resources, such as a database or a program that graphically represents sales figures, but the combination of integrated resources working together.

## MCS-14 : Systems Analysis and Design
## December, 2009

**Note :** Question number 1 is **compulsory.** Attempt any **three** questions form the rest.

**Q1. (a) Give characteristics of a SRS. Develop a SRS for a store management system.**
Refer to Chapter-3, Q.No.-2, Page No.-16

**(b) Give levels and components of MIS. Who are the key persons at all the levels of MIS? Explain their responsibilities.**
**Ans.** Refer to Chapter-13, Q.No.-1, Page No.-120, also
Refer Chapter-1, Page No.-2, and 3

**(c) List atleast 6 attributes of a good system analyst. Explain the phase of software maintenance in a SDLC.**
Refer to Chapter-2, Q.No.-3, Page No.-8
Now Refer to Chapter-12, Q.No.-4, Page No.-109

**(d) Explain database design. Also differentiate between hierarchical and network model.**
Refer to Chapter-9, Page No.-85
Now Refer to Chapter-9, Page No.-93

**Q2. (a) Differentiate between coupling and cohesion.**
Refer to June-2006, Q.No.-1(c), Page No.-197

**(b) Give steps in a form design. Explain atleast 5 common GUI controls for inputs.**
**Ans. Form Design:** System inputs are designed through forms. The general principles for input design are:
**1.** Capture only variable data.
**2.** Do not capture data that can be calculated or stored in computer programs.
**3.** Use codes for appropriate attributes.
**4.** Include instructions for completing the form.
**5.** Data to enter should be sequenced.
**Common GUI controls for Inputs:**
**Text Box:** It can allow for single or multiple lines of characters to be entered.
**Check Box:** It consists of a square box followed by a textual description of the input field for which the user is to provide the Yes/No value.

**List Box:** A list box is a control that requires the user to select a data item's value from the list of possible choices. It is rectangular and contains one or more rows of possible data values.
**Dropdown List:** It is like a list box, but is intended to suggest the existence of hidden list of possible values for a data item.
**Combination Box:** It is also known as combo box. It combines the capabilities of a text box and list box. A combo box gives the user, the flexibility of entering a data item's value or selecting its value from a list.

**(c) Categorise and differentiate between various types of object oriented CASE tools.**
Refer to June-2007, Q.No.-3(a), Page No.-233

**Q3. (a) What are the security issues in a computer system? How does an organization prevent its database from security concerns? Illustrate with an example.**
**Ans.** Refer to June-2005, Q.No.-3(a), Page No.-172

**(b) Draw ERD for a student information system for a college. Explain the concept of cardinality through it.**
**Ans.** Refer similar question.
Minimum-Cardinality is a facet on a slot on a class. Minimum-Cardinality represents the minimum number of values which should be on that slot for that class. When you define a relationship, you can specify a minimum, maximum, and default cardinality. Cardinality refers to the quantity of the products the user can select from a relationship. For example, you define a relationship called Hard Drives. It contains a 20 GB drive and a 30 GB drive. If you set the minimum cardinality to 2, the user must pick 2 items from this relationship. The user can do this in any of the following ways:
§ Pick one 20 GB drive and one 30 GB drive
§ Pick two 20 GB drives
§ Pick two 30 GB drives
The three types of cardinality you can define for a relationship are as follows:
§ **Minimum Cardinality**. Governs whether or not selecting items from this relationship is optional or is required. If you set the minimum cardinality to 0, selecting items is optional. If you set the minimum cardinality to greater than 0, the user must select that number of items from the relationship.

**(c) What do you mean by internal information, external information and turnaround document?**
Refer to Chapter-8, Q.No.-4, Page No.-82

**Q4. (a) What role does an end user play in the system development? Give examples of SDLC models to support the answer.**
**Ans.** www.gullybaba.com/ignou/mca/mcs14.html

**(b) Categorise system documentation. Give few examples of documentation.**
Refer to Chapter-4, Q.No.-1, Page No.-19 and
Refer to Chapter-4, Q.No.-3, Page No.-21

**(c) What activities are performed during Design phase? Explain them.**
Refer to Chapter-13, Q.No.-29, Page No.-153
Now Refer to Chapter-13, Q.No.-30, Page No.-157

**Q5. Write short notes on the following with example of each:**
**(a) CASE tools**
Refer to June-2007, Q.No.-3(a), Page No.-233

**(b) Objectives of SDLC**
Refer to Chapter-1, Q.No.-5, Page No.-5

**(c) Prototype**
Refer to Dec-2005, Q.No.-2(b)(ii), Page No.-186

**(d) Modularity**
Refer to June-2005, Q.No.-2(a), Page No.-170

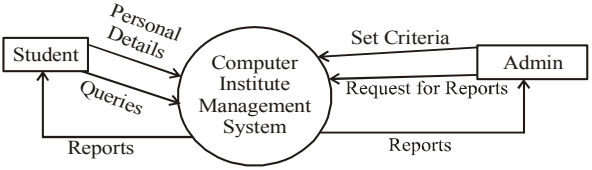
## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## June, 2010

**Note:** *Question No.1 is compulsory. Attempt any three questions from the rest.*

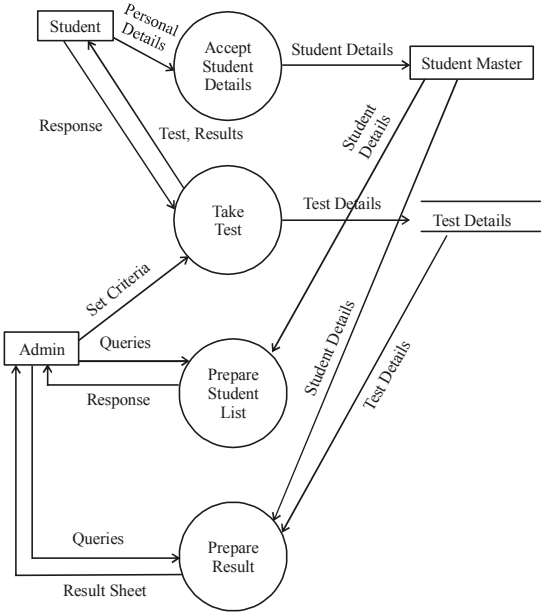
**Q1. (a) Draw a data flow diagrams (DFDs) fill 2-levels depicting the various processes, data flow and data repositories for a "Computer-Institute Management System". Follow the appropriate conventions/symbols as given in your study material.**

**Ans.**

**Level 0 DFD**




**Level 1 DFD**




**(b) What kind of fact finding techniques would you use for investigating the information requirements for a Multispeciality hospital, which is presently doing manual registrations? Which kind of techniques do you**

**think will be more effective? Also, mention the problems you anticipate in conducting the investigations.**

**Ans.** I will use the following fact finding techniques for investigating the information requirements for a multispecialty hospital:

1. Interviews - To collect requirements of individuals
2. Group Discussion – To collect the requirements of the groups.
3. Questionnaires – To gather hidden or intrinsic requirements.

I think all the above 3 techniques are equally important but amongst these interviewing is the best.

The following problems may be anticipated in conducting the investigation:

1. It is difficult to conduct personal interview with each individuals.
2. It is difficult to collect a group of person at the same time for group discussion.

**(c) Explain the role of MIS in an organisation. List and explain different kinds of Information Systems.**

**Ans.** MIS performs various roles in an organization:

- Supports day-to-day business operation;
- Supports managerial decision-making;
- Supports strategic decision-making and competitive advantages;
- Optimizing operational cost;
- Provide timely and accurate information; and
- Provide expert advice to the managers on selected domains.

and now refer to Page-122, Q.No.-3, Chapter-3

**(d) With the help of a diagram explain different components of a typical CASE tool. Also cite certain examples of CASE tools, being used at different SDLC levels.**

**Ans.** Refer to Page-101-105, Chapter-10

**Q2. (a) Describe the role of a Systems Analyst. Also, list various skills that the Systems Analyst should possess.**

**Ans.** Refer to Page No.-7, Q.No.-2, Chapter-2; and now

A systems analyst must name following skills:

- Working knowledge of information technology
- Computer programming experience and expertise
- General business knowledge
- Problem solving skills
- Communication skills
- Interpersonal skills
- Flexibility and adaptability
- Thorough knowledge of analysis and design methodologies.

**(b) Identify various reports/outputs/products associated with each and every phase of a SDLC.**

**Ans.** · *Project Identification and Selection:* Priorities for systems and project, architecture for data, networks, hardware and Information System Management are the result of the associated system.

· *Project Initiation and Planning:* Detailed work plan for project, specification of system scope and high level system requirements, assignment of team members and other resources.

· *Analysis:* Description of current system, need to enhance or replace current system, explanation of alternative systems and justification of alternatives. (data, process, input and output).

· *Physical design:* Technical, detailed specification of all system elements, i.e., programs, files, network, system software, etc. and acquisition plan for new technology.

· *Implementation:* Code, documentation, training programs and support capabilities.

· *Maintenance:* New version of software with associated updates of documents, training and support.

**Q3. (a) Mention various criteria for the Report design.**

**Ans.** Refer to Page No.-191, Q.No.-4(b)

**(b) With the help of a neat diagram discuss various components of Decision Support Systems (DSS).**

**Ans. Components of a DSS :**

**Data Management System**

This is a system where various activities associated with retrieval, storage, and organization of the relevant data for the particular decision context are managed. It also provides security functions, data integrity procedures, backup and recovery, concurrency control, and general data administration. It can be a relational, object oriented or any other suitable database.

**Model Management System**

Similar to Data Management Systems, Model Management Systems perform retrieval, storage, and organization activities associated with various quantitative models that provide the analytical capabilities for Decision Support Systems. This software module is responsible for analytical and limited reasoning capability of a DSS. This may contain various statistical and operation research models.

**Knowledge Engine**

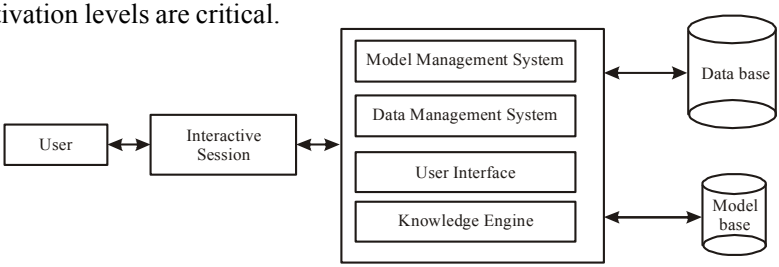
This module is responsible for activities related to problem recognition and generation of interim or final solutions. The knowledge engine is the "brain" of the Decision Support System. Decisions require reasoning, and less structured decisions require more reasoning.

**User Interface**

This software module provides functionalities for input/output, error capturing and reporting. A common user interface for various Decision Support Systems is not possible as their designs vary in accordance with the environment of the organization when they are deployed.

Types of User interface: Keyboard, Joystick, Mouse, Scanner, Voice, Pen mouse, Touch screen, etc.

Like all information systems, issues related to the user such as training, skill, motivation levels are critical.




**Q4. (a) With reference to Risk Assessment and management, write short notes on the following:**

**(i) Quantitative Risk analysis :**This approach gives an idea about the amount of risk involved with the event. This basically employs two fundamental elements i.e. The probability of occurrence of the loss making event and probability of occurrence of the event.

Estimated Loss = Potential loss due to the event * probability

It is therefore possible to rank the events in order of estimated loss. But the problem associated with the quantitative approach is estimating the probability of occurrence of the event, also in some cases the events are interrelated making the probability calculation even more difficult.

**(ii) Qualitative Risk analysis :** The extent of the information security program should commensurate with the degree of risk associated with the institution's systems, networks, and information assets. For example, compared to an information-only Web site, institutions offering transactional Internet banking activities are exposed to greater risks. Further, real-time funds transfers generally pose greater risks than delayed or batch-processed transactions because the items are processed immediately.

**(b) Explain various issues involved in software maintenance, with appropriate specific examples.**

**Ans.** Refer to Page-109, Q.No.-4, Chapter-11

**Q5. Write short notes on the following:**

**a) Knowledge Acquisition by Expert Systems.**

**Ans.** Expert systems must liase with people(experts) in order to gain knowledge and the people must be specialized in the appropriate area such as Medicine, Geology and Chemistry to name a few. Knowledge Engineer acts as an intermediary between the specialist (human expert) and the expert system. This process of picking the brain of an expert is a specialized form of data capture and makes use of interview techniques. The Knowledge Engineer is also responsible for the self-consistency of the data loaded to the expert system. Thus, a number of specific tests have to be performed to ensure that the conclusions reached are sensible and accurate. Figure depicts communication between expert system, knowledge base and human expert.


Communication between Expert System, Knowledge Engineer and Human Expert.

There are various applications for expert systems in business, engineering and medicine. Expert systems ask the user, a series of queries and based on the feedback from the user, deliver expert advice on the specific subject. Expert systems are used in the field of Medical diagnosis, Sales forecasting etc. Expert Systems are being used by managers for credit management, employees performance evaluation, portfolio analysis and production monitoring. Although expert systems are used in many fields, it can never replace a human expert. Expert system can provide expert advice based on the available information and knowledge. Expert systems lack learning capability like human being and have very limited focus area. It fails in the areas where advice requires a broad knowledge base.

**b) Disaster recovery.**

**Ans.** Refer to Page No.-116, Q.No.-5 & 6, Chapter-12

**c) Criteria for form design.**

**Ans.** Refer to Page No.-191, Q.No.-4(b), Dec-2005

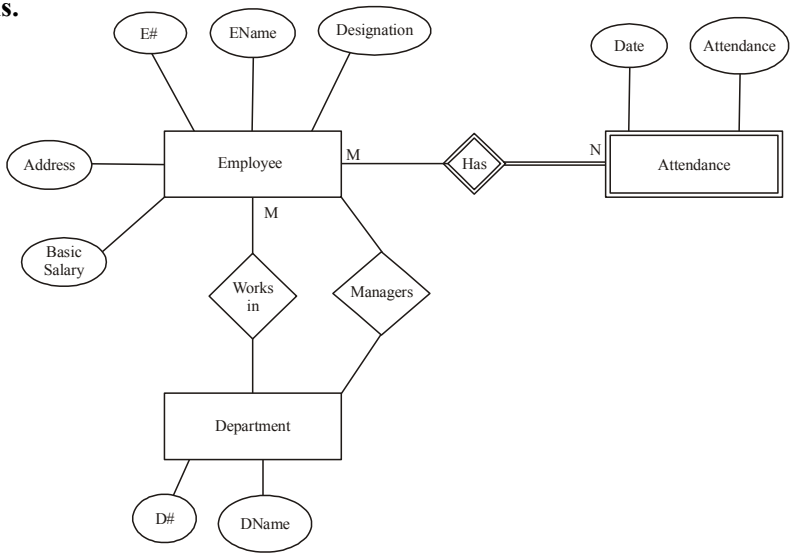
**d) System Audit.**

**Ans.** Refer to Page-112, Q.No.-2, Chapter-12

**MCS-014 : SYSTEM ANALYSIS AND DESIGN**
**December, 2010**

**Note:** *Question No.1 is compulsory. Attempt any three questions from the rest.*

**Q1. (a) Construct an Entity Relationship Diagram for employee management system. Also construct a relational database schema for it.**
**Ans.**




Relational Database schema
Employee (E#, Ename, Designation, Address, Basic Salary, D#)
Department (D#, DName, E#)
Attendance (E#, Date, Attendance)

**(b) With the help of diagrams, explain the steps in physical and logical design.**
**Ans.** Refer to Page No.-63-65, Chapter-7

**(c) What are the CASE tools? Explain object oriented CASE tools in detail. Also explain differentiate between Reverse and forward engineering.**
**Ans.** Refer to Page-190, Q.No.-4(a) and now
**Object Oriented CASE tools**
Object oriented CASE tools are similar to other CASE tools. These differ from others only in terms of their capability to create class diagrams and text specifications for reports. Object oriented CASE tools support an O-O methodology such as Rumbaugh's Object Management Technique (OMT).

A large number of object oriented CASE tools are available in the market. These include: Paradigm Plus from Protosoft, Rational Rose from Rational and WithClass from MicroGold Software.

The following are some of the features found across most of the Object Oriented CASE tools:

1. Create graphics, such as class diagrams, message diagrams, state diagrams etc.
2. Create text specifications such as system specification, class specification and relationship specification.
3. Generate source code.
4. Repository of models.

All of these object oriented CASE tools are similar in terms of their capabilities to create class diagrams, code generation. However, they may differ in terms of their extendibility, number of supported operating systems and additional features and capabilities, such as support for different methodologies and computer language code generation (C++, Ada, Java etc). Most provide capability to automatically generate code of C++ and other languages from a class diagram and class specifications. Some provide the capability to generate class diagrams from code (reverse engineering).

**(d) Describe the various levels and components of MIS. Explain the role of a system analyst in system development.**

**Ans.** Refer to Page-122, Q.No.-3, Page-193, Q.No.-5(a) and Page-7, Q.No.-2

**Q2. (a) Develop a SRS for inventory management system. Give any three characteristics of SRS.**

**Ans.** Refer to Page No.-16, Q.No.-2, Chapter-3 and now

**SRS for Inventory Management**

**[I] Introduction**

Inventory management system automate the inventory control process of an organisation. Storekeeper keep track of all inventory and controls them. Sales department sends the request for issue of inventory to storekeeper and storekeeper generates the requisition for purchase department for purchase of inventory.

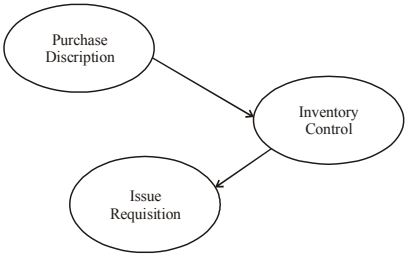
**[II] Information description about the system**

Inventory purchased by purchase department, managed by store keeper and issue to sales department.

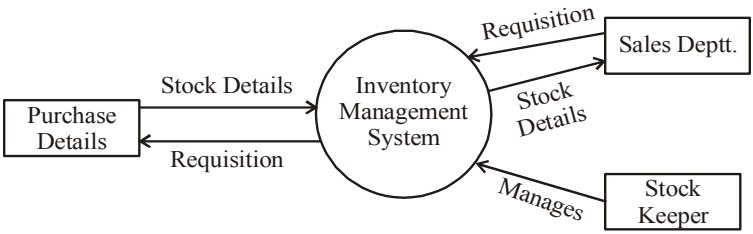
There will be three subsystems/interfaces of the system.

1. Purchase Requisition
2. Inventory Management
3. Issue Requisition

**[III] Functional description of the system**




**Context DFD**




**(b) Why is feasibility study considered an important task in system development? Classify feasibility and elaborate the key features of each.**
**Ans.** Refer to Page No.-46, Q.No.-7

**(c) Classify Information systems and describe the characterizing features, in brief.**
**Ans.** Refer to Page No.-122, Q.No.-3

**Q3. (a) Explain in brief, the following SDLC models (with suitable diagrams):**
**(i) Waterfall Model. (with feedback)**
**Ans.** Refer to Page No.-266, Q.No.-4(b)
**(ii) Spiral Model.**
**Explain the key features of the above models.**
**Ans.** • Balances all of the risk elements, i.e. the high-risk elements must be lowered first. • Offers prototyping as a risk-reduction option at any stage of development. • It allows reworks of earlier stages as more attractive alternatives are identified. • Detail isn't necessary until detail is needed.

**(b) What are the various types of documentation? Give an example of each type.**
**Ans.** Refer to Page No.-24, Q.No.-7, Chapter-4

**(c) Differentiate between a decision tree and a decision table. Give an example and explain the decision table.**

**Ans.** Refer to Page No.-260, Q.No.-5(c)(i) & Page No.-67, Q.No.-3

**Q4. (a) Give steps in Report design. Classify Reports.**

**Ans.** a) The first is to collect all relevant information regarding the form or Report by asking questions like who, what, when, where, and how.

b) Refine the above information into an initial prototype.

c) Review and evaluate the prototype.

d) Design, validate and test the outputs using some combination of prototyping and code generation tools.

**Classification of Report:**

· **Detailed reports:** These reports present information with little or no filtering or restrictions.

Ex.: Detailed listing of all customer accounts, orders or products in inventory.

· **Summary reports:** These reports categorize information for managers who do not want to wade through details. The data in summary reports are categorized and summarized to indicate trends and potential problems. We can also include charts and graphs in the summary report so that it clearly summarizes trends at a glance.

Ex.: Report that summarizes the months and years total sales by product types and category.

· **Exception reports:** These reports filter data before they are presented to the manager as information, i.e., these reports include exceptions to some conditions or standards.

Ex.: A report that identifies items, which are low in stock.

**(b) Write short notes on the following:**

**(i) JAD**

**Ans.** Refer to Page No.-45, Q.No.-6(i), Chapter-5

**(ii) Data dictionary**

**Ans.** Refer to Page No.-35, Q.No.-11, Chapter-4

**(c) Why is audit important for system development? Explain the types of audit.**

**Ans.** Audit is important for system development because it is used:

• To improve the quality of information systems, prevent failure and minimize the effects of failure, and speed up the process of recovery in the event of a failure. This will help Information System to be more reliable.

• To make an information system more secure from natural as well as manmade disasters, unauthorized access, and other destructive actions.

• To improve the cost performance of an information system by optimum utilization of its resources, which leads to increase in efficiency.

Audit can be broadly of two types namely auditing manual processes and audit through computer. Audit through computer is important to find out the accuracy and integrity of information system output. This types of audit are done by information system expert and use test data to check the adequacy and accuracy of control mechanism built-in to the system.

A typical audit looks at the following factors:

Audit of response time: In this audit the actual response time of the system verses the desired response time is compared to the performance of the system

Audit of broken links: This is applicable to web site and other intranet applications. The most irritating things on a web site is not finding a link document. There are automated software to find broken/unavailable links on web site.

**Database Audit:** Database audits involve checking the database integrity and availability. The information that is sent to the database should be checked with the information actually stored on the database.

Network audit: Network audit involves checking the venerability of network. It checks whether the network configuration is giving optimal performance or not.

**Transaction Audit**

Transaction audit is a process to find –

• Who did changes?

• What changes are made?

• Whether the changes are authorized or not as per the security policy of the Organization?

The details of the above transactions are written to either a media or printed. This allows Database Administrators to track changes and helps the organization to satisfy regulatory requirements such as tracking specific users actions, general security screening, validating user permissions etc.

**Audit of Computer Security**

Issues of security of computer involve both physical and logical security. Physical security involves restricting physical access to the computing resources from unauthorized person. Logical security involves restricting the use of computing resources by unauthorized person by providing logical control mechanism (e.g. password protection). The audit of computer security involves review of physical and logical security measures. Review of parameters, plans, practices, and policies that are developed and implemented by the organization over the computer resources, and how security measures are followed for Computers, Networks and Data communication. They are also included in the Audit.

**Audit of Application**

Here, both manual and programmed internal controls related to information systems are assessed. Primarily, there are four areas of audit coverage for an application being reviewed.

*The four areas are given below:*

**Control environment:** This includes reviewing the system's security, its operating platform, system documentation and the interaction it has with other systems.

**Data Input Controls:** This involves reviewing the controls which ensure that data that enters into the system is accurate, complete and valid as per the standard. Examples include verifying system tables, limit checks, range checks and redundant data checks.

**Processing Controls:** These controls ensure that the data is properly processed and that automatic calculations performed by the system are accurate. This is tested by assessing controls built into the programs and by processing test data through the system and comparing the results of processing with expected results. Also, there will be checks on currency of stored data, default values and reporting exceptions.

**Output Controls:** In this, review of the system generated reports to ensure that they are accurate and the reports produced are reliable, timely and relevant is done. Also, it is checked whether cost savings can be achieved by reducing the number of reports produced. Data control personnel perform visual review of computer output and reconciliation of totals.

**Q5. (a) What is cost Benefit analysis? Explain any one model with a suitable example.**

**Ans.** Refer to Page No.-48, Q.No.-8; and now

Different costs are:

**Cost of human resources**

It includes the salaries of system analysts, software engineers, programmers, data entry operators, operational, and clerical staff. In other words, the amount that is going to be spent on all the people involved.

**Cost of infrastructure**

The cost of infrastructure including those of computers, cables, software, etc., comes under this head.

**Cost of training**

Both the developing staff and operating staff need to be trained for new technologies and new system. So, the training cost has to be considered for calculating the cost of the system.

We may consider the case of an insurance company's branch office. There are 15 employees in the office which include one manager, two business development officers, one accounts officer, one administrative officer, seven clerical staffs, two security guards, one peon. If the branch is converted to a fully computerized branch, the total hardware and software cost will be Rs.10 lakhs. Training of the existing manpower will be Rs.50,000.Total investment is 10.5 lakhs. Total

maintenance cost of software and hardware is Rs.1.5 lakhs per year. Interest of the investment is Rs.1,26,000 per year. So, the total expenditure is increased by Rs.2, 76,000 per year. But the branch can reduce the clerical staff. Now it needs two clerical staffs and two data entry operators. Total cost saving by reducing 3 staffs will be Rs.4.5 lakhs per year. Here, it is clear that the tangible benefit is more than the expenditure. Besides, the tangible benefit also improves the customer's satisfaction. So, it is clear that the project is feasible.

**(b) What are the various types of file organisation? Describe in brief.**
**Ans.** Refer to Page No.-89, Q.No.-3

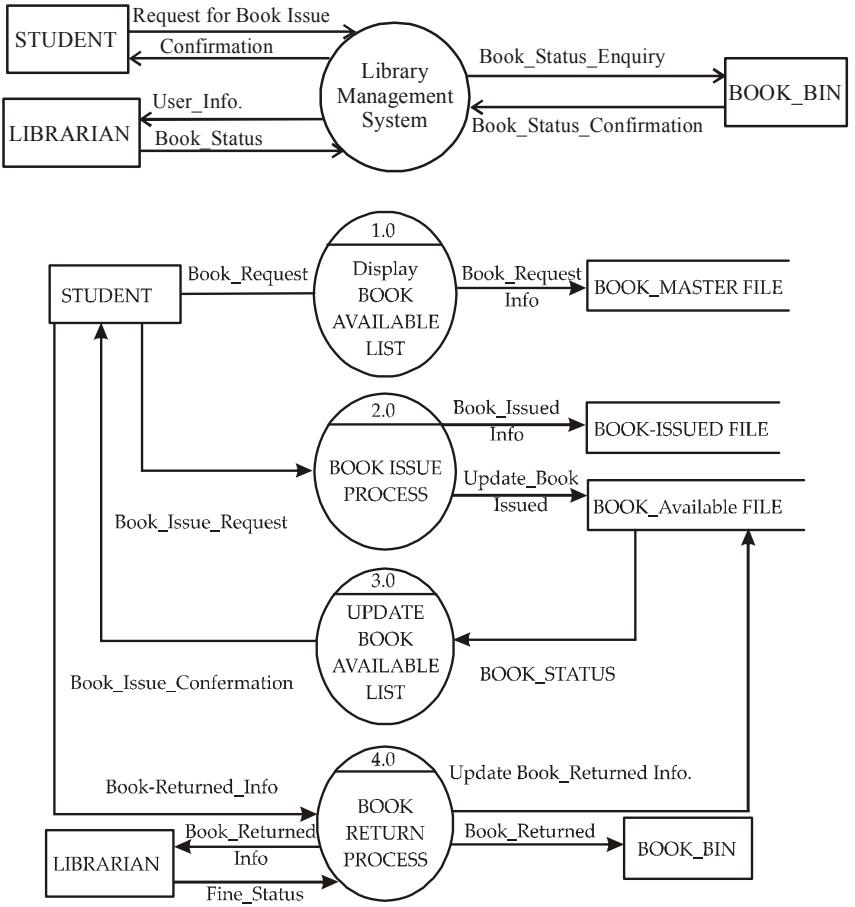
**(c) Differentiate between various types of coupling and cohesion. Which of these types (in each) is the best and worst and why?**
**Ans.** Refer to Page No.-55-62, Q.No.-5 & 6

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## June, 2011

**Note:** *Question No.1 is compulsory. Attempt any three questions from the rest.*

**Q1. (a) Construct a data flow diagram (level 0 and 1) for a library management system. Also draw the respective ERDs for it.**

**Ans.** Refer to Page No.-207, Q.No.-1(b)



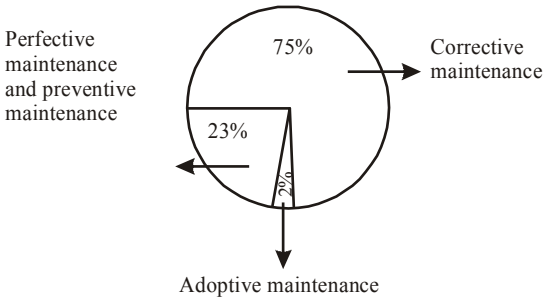

**Level 1 DFD for Library Management System**

**(b) Explain the steps and activities in the design phase of SDLC.**

**Ans.** Refer to Page No.-13, Q.No.-1, Chapter-3

**(c) Explain the key features of the followings:**
**(i) Prototyping Model**
**Ans.** Refer to Page No.-46 [Prototyping]

**(ii) Spiral Model:** Refer to Page No.-279, Q.No.-3(a)(ii)
**Give a brief note on system maintenance and explain any 1 model of system maintenance (with a diagram).**
**Ans.** Refer to Page No.-109, Q.No.-4, Chapter-11
Model of System Maintenance




Generally, we adopt spiral model for entry of a new project at software maintenance phase. In spiral model a new project can enter in any phase of development, so a fully complete and implemented project can also be maintained using spiral model.

**(d) What is the importance of security and protection of databases for any organization? What are the various policies and methods to ensure security of databases?**
**Ans.** Refer to Page No.-172, Q.No.-3(a)
The various methods and policies used to ensure security of database are as follows:
(1) In-House Backup: This level is the minimum acceptable and is mandatory for all installations and application's system. Define in detail all in-house back up procedures, the techniques used, files copied, frequency, etc.
(2) Alternate Storage Area: This level of protection is necessary for mission critical components. It consists of offsite storage of at least one copy of all AIS files and databases, programs, and procedures necessary to operate the high priority application systems, either at the installation or at an alternate site of operation (including copies of contingency plans and related materials).
The alternate storage area should be located in an area reasonably accessible to the installation, but not subject to the same degree of major threat as the site. It is recommended that, as a rule of thumb, the alternate storage area be

no closer than one mile from the site. However, the distance may vary from location to location.

(3) The Disaster Recovery Toolkit: The Disaster Recovery Toolkit is a highly valuable collection of items and documents to assist in ensuring business continuity in the face of serious incident or disaster. Many organizations use these documents as a checklist and add element specific to their need. Although they vary from organization to organization, they generally comprise the following:

(a) A contingency audit questionnaire.
(b) A dependency analysis document – questions and guidance.
(c) A business Impact Analysis questionnaire.
(d) An audit questionnaire for disaster recovery or business continuity plan.
(e) A checklist, action list and framework for disaster recovery.

**Q2. (a) Explain the key players and their roles and responsibilities at various levels of a MIS. Explain the tasks performed at DSS in detail.**

**Ans.** Refer to Page No.-273, Q.No.-1(c) & Page No.-122, Q.No.-3

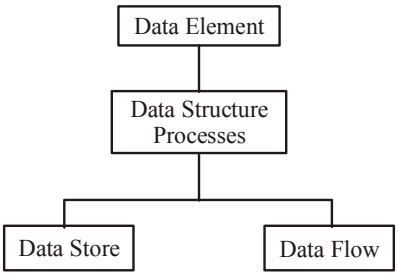
Role & responsibilities of MIS are as follows:

→ Support Day-to-Day business operation.
→ Support managerial decision-making.
→ Support strategic decision-making & competitive advantages.
→ Optimizing operational cost.
→ Provide timely and accurate information.
→ Provide expert advice to the manager on selected domains.
→ It is responsible for providing reports to management usually in semi-structured format.
→ It also responsible for information presented in both textual and graphical forms but more often in graphical format.
→ It is responsible for providing information on trend analysis exception reporting & what-if-analysis.
→ It is responsible to allow the user to ask question, such as what if we increase price by 10% the effect on sales of the product? If inflation increases by 5 per cent what will be the effect on the sales forecast?

**(b) Differentiate between a decision table and a decision tree. Prepare a sample data dictionary for an employee management system.**

**Ans.** Refer to Page No.-67, Q.No.-3 & Page No.-72 [Decision tree]

A dictionary has four parts:




I Data Elements

(1) Employee Number (2) Employee Name (3) Date of Birth
(4) Sex (5) Date of joining (6) Salary Grade
(7) Department Name (8) Basic Salary (9) Dearness Allowance
(10) Provident Fund (11) LIC (12) Income Tax
(13) Gross Salary (14) Net Pay (15) Deductions
(16) Address (17) City (18) Pin Code
(19) Telephone Number (20) House Rent Allowance
(21) City Compensatory Allowance

II Data Store [Files]

(1) Employee Master File (2) Attendance file
(3) Current Payroll file (4) Leave Record
(5) Deduction Summary file (6) Printed Pay Register
(7) Salary Summary file (8) Gross earning file
(9) Deduction file

III Data flows

(1) Leave Application Record (2) Pay slip
(3) Bank Statement (4) Due details
(5) Earning details (6) Deduction details
(7) Current Payment details

IV Processes

(1.0) Prepare Attendance and leave Record
(1.1) Verify Master
(1.2) Sanction leave
(1.3) Prepare leave Record
(1.4) Prepare leave Summary
(2.0) Prepare Payroll Register
(2.1) Calculate Current Pay
(2.2) Calculate deductions and Earnings
(2.3) Calculate Net Pay

**(c) Prepare a brief note on user documentation, as well as utility.**

**Ans.** Refer to Page No. -25 [User Documentation]

Utilities:

→ Review the progress or development of an application software.

→ Communicates facts about system to users.

→ Communicates between personnel working on a development project.

→ Provides necessary guide lines to allow correction or revision of a system or its computer programs.

→ Assists in the reconstruction of the system in case it is destroyed.

→ Documentation serves as a focal point from which the analysts design can be assessed and as a standard to be utilized as a reference once the system is implemented.

→ Provides operating instruction to users and operating staff.

→ It helps the management to determine if the new design achiever the objectives of the company within the established constraints and if it is justifiable from a cost standpoint.

**Q3. (a) What is cost estimation? Give one model used by an organization for cost benefit analysis.**

**Ans.** Refer to Page No.-48, Q.No.-8

Visit us at: www.GullyBaba.com/BCA/MCS-014

**(b) Explain the essential features of a SRS and prepare it for a project on 'Student Enrollment System'.**

**Ans.** Refer to Page No.-16, Q.No.-2, Chapter-3

Software Requirements Specification (SRS) is a complete description of the behaviour of the system to be developed. It includes a set of use cases hat describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

SRS for SES (STUDENT ENROLLMENT SYSTEM):

**(1) GENERAL DESCRIPTION**–SES is STUDENT ENROLLMENT SYSTEM for managing the records of the alumni's of the university as well as staff, faculty and higher authorities.

1.1 *Purpose*– The purpose for developing this type of software or introducing this SES is to facilitate everyone who is concerned with the university.

1.2 *Scope*– The scope of SES is global i.e. it should be able to be accessed from anywhere through internet i.e. registered users must be able to login to their accounts by directly accessing the university's website and then signing in with their username and password anytime and anywhere.

1.3 *Abbreviation*– SES STUDENT ENROLLMENT SYSTEM

1.4 *Overview*– As the SES is able to have a user interface. It should have a drop down boxes and if we drag mouse on any control at our welcome screen information regarding that the control should be displayed. Help menu should be there. As a teacher it should provide them to upload the various assignments and the attendance of the students. As a developer it should make a user interface which is user friendly. He should make the SES as simple as he can. Backup at the main server should be made.

**(2) OVERALL DESCRIPTION**

2.1 *Product Perspective*– Product i.e. SES should be able to provide a basic and easy interchange of information i.e. it should be able to remove the communication gaps between a teacher and the student. It should have chat facilities for all the users that are online. It should be compatible with the operating systems.

2.2 *Product Functions*– The following are the product functions of the SES:

• The SES login box should on the official website of the university.

• The password field should be secured.

• After signing in all updates and new announcements for users should be displayed.

• By clicking on the dropdown box of the options the user should be able to view progress reports, assignments, notes, attendance, placement services and results.

• User should be able to change the passwords.

• Web pages should support pdf, ppt, doc and similar supported formats so that they can be easily downloadable and unloadable.

2.3 *User Characteristics*– A user can only have his/her registration number as username so if he joins the university then only he can then only he can login. This prevents misuse, unauthorized access and hacking of the product.

2.4 *General Constraints*– Server capacity is how many users can access or can be online at once. More is the number of users more will be the network traffic and hence the server comes in a down state. Personal firewall and updating is a tough task, it should be such that it should not block the network traffic, making the system slower. Firewall of the SES should not collide with the firewall of the user system.

2.5 *Assumption and Dependencies*– SES should work even at when the network traffic is high. Server should have a power backup as well as a database backup. The SES should be compatible with most of the operating systems i.e. previous and latest ones.

**(3) SPECIFIC REQUIREMENTS**

3.1 *External Interface Required*

3.1.1 User Interfaces– The external users are the students and the teachers of the university. The students can have an access to their accounts for their attendance, assignments etc. The teachers have also an account to access

their account for uploading of the students' attendance and the assignments to be submitted by them.

3.1.2 Hardware Interfaces– The external hardware interface used for accessing the SES is the personal computers of the teachers and the students. The PCs may be laptops with wireless LAN as the internet connections provided will be wireless.

3.1.3 Software Interfaces– The Operating Systems can be any version of Windows, Linux, Unix or Mac which supports TCP/IP protocols.

3.1.4 Communication Interfaces– The communication interface is a local area network through wireless network routers.

3.2 *Performance Requirements*– The PCs used must be at least Pentium 4 machines so that they can give optimum performance of the product.

3.3 *Design Constraints*– The constraints at the designing time are that the needs of the university students and the teachers may keep on changing so the designers must keep this in view and design the product in this way that it is easily updatable.

3.4 *Attributes*– The following are the attributes of the product SES:

(a) It should be equipped with current and archive database.

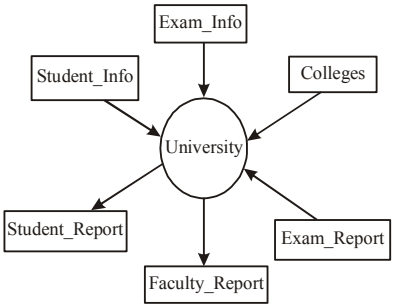
(b) All records can easily be updated.

(c) It should have its personal firewall.

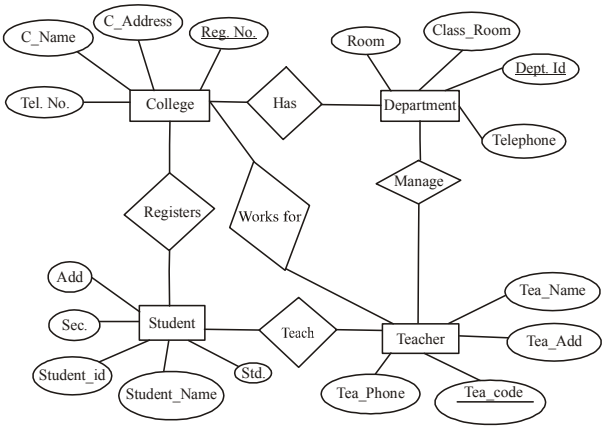
(d) It should facilitate student with updating his/her account, downloading or uploading of assignments from anywhere.

(e) It should also do the same for teachers they can also have their pay checks online i.e. SES should be capable of online transaction.

3.5 *Other Requirements*– The software is such that as the time goes by the need of the university management, students and teachers may keep on changing thus it is made to change from time to time.



Context Analysis Diagram For Student Enrollment System




**(c) Write a brief note on an 'expert system and its key characteristics'.**
**Ans.** Refer to Page No.-173, Q.No.-3(b)

**Q4. (a) Give differences between internal information and turnaround information. Also describe the process of designing a form.**
**Ans.** Refer to Page No.-82, Q.No.-4 & 5

**(b) What is denormalization? Give a suitable example to explain it. Also explain the process of denormalization of tables.**
**Ans.** Refer to Page No.-88, Q.No.-2

| STUDENT | COURSE | INSTRUCTOR |
|---|---|---|
| Narayan | Database | Mark |
| Smith | Database | Navathe |
| Smith | Operating System | Ammar |
| Smith | Theory | Schulman |
| Wallace | Database | Mark |
| Wallace | Operating System | Ahamad |
| Wong | Database | Omiecinski |
| Zelaya | Database | Navathe |

A relation TEACHES that is in 3NF but not in BCNF

TEACH (STUDENT, COURSE,INSTRUCTOR) relation with the functional dependencies $\{\{STUDENT, COURSE\} \rightarrow INSTRUCTOR, INSTRUCTOR \rightarrow COURSE\}$. A lossless decomposition of TEACH into $T_1$ (STUDENT, INSTRUCTOR) and $T_2$ (INSTRUCTOR, COURSE) does not allow queries of the form what course did student smith take from Instructor Navathe to be

answered without joining and. Therefore storing $T_1, T_2$ & TEACH may be a possible solution, which reduces the design from BCNF to 3NF. Here TEACH is a materialized join of the other two tables, representing an extreme redundancy. Any updates to $T_1$ & $T_2$ would have to be applied to TEACH. **An** alternate strategy is to consider $T_1$ & $T_2$ updatable base tables whereas TEACH can be created as view.

$T_1$ (STUDENT, INSTRUCTOR)

| STUDENT | INSTRUCTOR |
|---|---|
| Narayan | Mark |
| Smith | Navathe |
| Smith | Ammar |
| Smith | Schulman |
| Wallace | Mark |
| Wallace | Ahamad |
| Wong | Omiecinski |
| Zelaya | Navathe |

$T_2$ (INSTRUCTOR, COURSE)

| COURSE | INSTRUCTOR |
|---|---|
| Database | Mark |
| Database | Navathe |
| Operating System | Ammar |
| Theory | Schulman |
| Database | Mark |
| Operating System | Ahamad |
| Database | Omiecinski |
| Database | Navathe |

**(c) Differentiate between various types of databases and differentiate between a DBMS and a file-system.**

**Ans.** Refer to Page No.-121, Q.No.-2

Differences b/w a DBMS & file-system:

$\rightarrow$ Traditionally the term file has been used to refer to the folder that hold all the related to refer to the folder that hold all the related material in ordered storage for quick reference. Consider an example of flat file, which stores the pin code of vendor's address. So any change in the field size will enforce changes in the entire program which uses the data related to the vendor's address & if it is the file being used by more than one application the problem expands it area.

The traditional/flat file oriented approach to information processing has a special rate master file & its own set of personal files for each application.
Traditional file approach significantly lacking sharing of data across various applications.
(1) Program/data dependency: if a data field is to be added to a master file, all such programs that access the master file would here to be changed to all for this new field.
(2) Data redundancy: The same piece of information may be stored in two or more files.
(3) Lack of flexibility: As there is a strong coupling b/w data & program, most of the data/information can be retrieved in a way it has been programmed.

**Q5. Write short notes on the following with example of each:**
**(a) Information System Design**
**Ans.** Refer to Page No.-122, Q.No.-3

**(b) Modularity**
**Ans.** Refer to Page No.-170, Q.No.-2(a)

**(c) System testing**
**Ans.** Refer to Page No.-211, Q.No.-2(b)(i)

**(d) CASE tools**
**Ans.** Refer to Page No.-98-99, Chapter-10

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2011

***Note:*** *Question* ***No.1*** *is compulsory. Attempt any three questions from the rest.*

**Q1. (a) Draw a data flow diagrams (DFD) till second level depicting the various processes, data flow and data repositories for a "Railway Reservation System".**

**Ans.** Refer to Page No-224, Q.No.-1

**(b) What is MIS? Explain the role of MIS in a business organization. Also explain how MIS can be used for management control and strategic planning of an organization.**

**Ans.** MIS: Stands for Management Information System.

→ MIS is a general term for the computer system in an enterprise that provides information about its business operation.

→ It is also used to refer to the people who manage these systems.

→ MIS also refers to a centrally or centrally coordinate system of computer expertise & management, often including main frame system but also including by extension the corporation entire network of computer resources.

→ MIS Performs following roles in a business organization:

• Supports day-to-day business-operation.
• Supports managerial decision-making.
• Optimizing operational cost.
• Provide timely & accurate information.
• Provide expert advice to the managers on selected domains.
• Supports strategic decision-making & competitive advantage.

MIS is used for management control & strategic planning of an organisations:

→ Management information system helps the organization to produce information that organization need to improve decision-making problem solving, control operation and creating new products or services.

→ Management Information System are used for:

• Operational Control: Information for control of day-to-day business operations. Information required by operational managers to control their daily-work. This includes information on current stock of items, employee attendance, employee performance sheet etc. Such information is very much structured & computational in nature & is produced in fixed format.

• Management Control: Information for short term planning. Information is rather un-structured or semi-structured such as cash flow statement sales, trend analysis, monthly & annual financial statements. This type of information is used by mid-level manager for planning & control or organizational sub-units.

→ Strategic Planning: Information for long-term planning, developing policies & long-term goals for the organization. Such information is adhoc& unstructured such as human resource forecast, market trend analysis etc.

**(c) What is information? Explain how information can be classified in an organization.**

**Ans.** Refer to Page No.-82, Q.No.-4 & Page No.-122, Q.No.-3

**(d) What is system design? Explain top-down design with an example.**

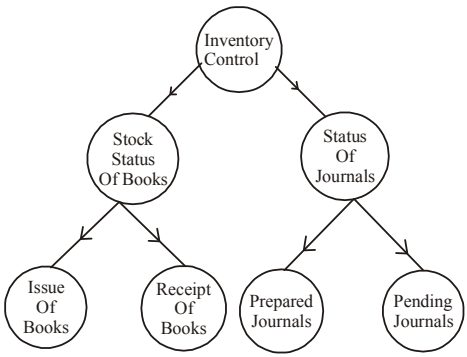
**Ans.** System Design is the specification or construction of a technical, computer based solution for the business requirements identified in a system analysis. System design is a process through which requirements are translated into representation of software. Preliminary design is concerned with the transformation of requirements into data and software architecture. Detail design focuses on refinements to the architectural representation that lead to detailed data structure and algorithmic representation for the software.

System design consists of logical and physical design of the system. We must design all aspects of the system from input and outputs to reports, databases and computer processes. Design occurs in two phases viz logical design and physical design.

**Top down design Approach:** According to top down design approach is based on the fact that large problems become more manageable if they are divided into a number of smaller and simpler tasks which can be tackled separately.

→ Top down design approach is performed in a special way.

The main module is designed first and then sub modules are designed. After main module is written and checked, each sub module is written and tested in turn. Therefore, in Top-down design we try to minimize the complexity of the problem and make it manageable by sub-dividing it into smaller segments which is called modularization or de-composition. Example of Top-down design approach is given below.



**(e) What is System Analysis? Explain role and responsibilities of a System Analyst in system development.**

**Ans.** System Analysis is the study of a business problem domain to recommend improvements and specify the business requirements for the solution independent of any technology that can or will be used to implement a solution to that problem. During the System Analysis the requirements are determined. In this phase, analysis works with the users to determine what the users went from a proposed system.

Now Refer to Page No.-7 & 8, Q.No-2 & 3, Chapter-2

**Q2. (a) What are CASE tools? Briefly describe various types of CASE tools and their usage.**

**Ans.** Refer to Page No.-98 & 101, Chapter-10

**(b) What is E-R Diagram? Construct an E-R Diagram for "Retail Management System". Make assumptions wherever necessary.**

**Ans.** Refer to Page No.-26, Q.No.-9, Chapter-4 & Page No.-77, Q.No.-8

**Q3. (a) What is Form? Why proper Form design is essential for a system? Briefly, discuss the criteria used for Form design.**

**Ans.** Refer to Page No.-79, 84, & 82, Q.No.-1, Q.No.-6 & Q.No.-5

**(b) Define an Expert System. List various components of Expert System. Also explain the characteristics of an Expert System.**

**Ans.** Refer to Page No-124, Q.No.-4, Chapter-13

**Q4. (a) Identify various outputs/products/reports associated with each and every phase of a SDLC.**

**Ans.** Refer to Page No.-13, Q.No.-1 & Page No.-17, Q.No.-3

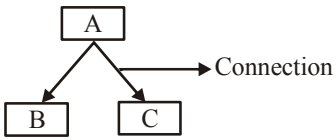
**(b) What is a Structure Chart? Draw a structure chart for "Banking System". Make assumptions wherever necessary.**

**Ans.** Structure Chart: It is graphical tool for representing the hierarchy of the flow of the software. Structure chart has 3 elements:

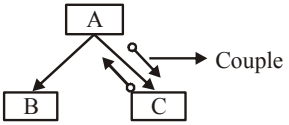
**(a) Module:** A module is represented by a rectangle with proper Name i.e. Module Name.

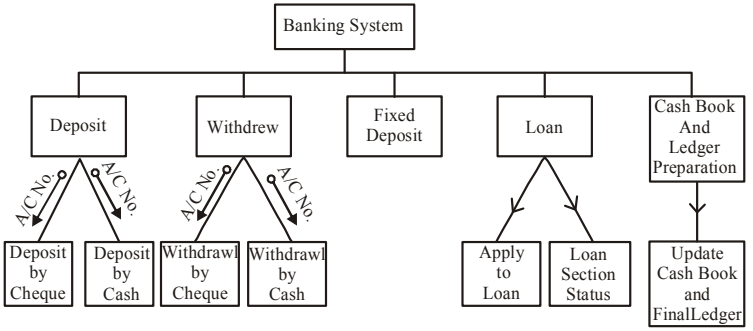
**(b) Connection:** A connection is represented by vector linking two modules.

$\rightarrow$ It shows that one module has called another module.



**(c) Couple:** A couple is represented by an arrow with a circular tail.
⟶ It represents the direction of data item moving from one module to another




**Q5. (a) What is data dictionary? Explain why data dictionary is used by system analyst.**
**Ans.** Refer to Page No.-35, Q.No.-11 & 12, Chapter-13

**(b) Write short note on following:**
**(i) Decision Support System**
**Ans.** Refer to Page No.-132, Q.No.-10

**(ii) System Audit**
**Ans.** Refer to Page No.-112, Q.No.- 2, Chapter-12

**(iii) Software Testing**
**Ans. Software Testing:** Software Testing is the process of executing a program or system with the intent of finding errors.
Thus, the fundamental purpose of software testing is to find errors in the software. Finding problems & having them fixed is the care of what a test engineer does. Following are the types of testing techniques:
**(1) White-Box Testing:** It is also called as Glass Box Testing. It is a test care design method that uses the control structure of the procedural design to device test cases. Using White Box Testing methods the software engineer can derive test cases that: (a) Guarantee that all independent paths within a module have been exercised atleast once. (b) Exercise all loops at their boundaries & within their operational bounds. (c) Exercise all logical decisions can their true or false sides. (d) Exercise interval data structure to assure their validity.

**(2) Black-Box Testing:** If focuses on the functional requirements of the software i.e black-box testing. Enables the software engineer to derive sets of input condition that will fully exercise all functional requirements for a program. Black Box Testing is not an alternative to white box testing. Rather, it is a complementary class of errors that white-box testing. Black Box Testing attempts to find errors in the following categories: (a) In correct errors (b) Interface errors (c) Errors in the data structures or external database.

**(3) System Testing:** System testing is designed to uncover weakness that were not found in earlier tests. This includes forced system failure & validation of total system as it will be implemented by its user in the operational environment.

→ Under this testing, generally we take low volumes of transaction based on live data.

→ This volume is increased until the maximum level for each transaction type is reached.

→ The total system is also tested for recovery after various major failures to ensure that no data are lost during the emergency.

→ All this is done with the old system still in operation, when we see that the proposed system is successful in the test, the old system is discontinued.

**(4) User Acceptance Testing:** An acceptance test has the objective of selling the user on the validity & reliability of the system.

→ It verifies that the system's procedures operate to system specifications & that the integrity of important data is maintained.

→ Performance of an acceptance test is actually the user's show.

→ User motivation is very important for the successful performance of the system.

After that a comprehensive test report is prepared. This report shows the system's tolerance, performance range, error rate & accuracy.

**(5) System Documentation:** All design & test documentation should be well prepared & kept in the library for future reference. The library is the central location for maintenance of the new system.

**Experience each of your qualities and virtues. Don't keep your treasures locked up in the treasure store of your intellect, but use them. In this way you will become victorious over all adverse situations.**

**MCS-014 : SYSTEM ANALYSIS AND DESIGN**
**June, 2012**

**Note:** *Question **No. 1** is compulsory. Attempt any three questions from the rest.*

**Q1. (a) What are different phases in SDLC? What is need of System Analysis? Explain the role of System Analyst.**

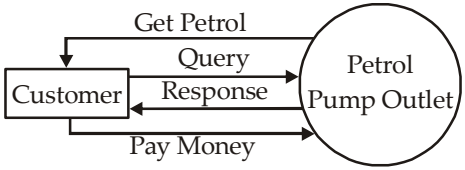
**Ans.** Refer to Page No.-13, Q.No.-1, Chapter-3

A computerised system enables an organisation to provide accurate information and respond faster to the queries, events etc. If a business needs computerised information system, a systems analyst is required for analysis and design of that system. Information systems evolved from the need to improve the use of computer resources for the information processing needs of business application. Customer defines the business problems to be solved by the computer. Project managers, Analysts, Programmers and Customers apply information technology to build information systems that solve those problems. Information technology offers the opportunity to collect and store enormous volume of data, process business transactions with great speed and accuracy and provide timely and relevant information for taking correct decision by management. This potential could not be realised without the help of a systems analyst since business users may not fully understand the capabilities and limitations of modern information technology. Similarly, computer programmers and information technologists do not fully understand the business applications they are trying to computerise or support. A communication gap has always existed between those who need computer based business solutions and those who understand information technology. Systems analyst bridges this gap.

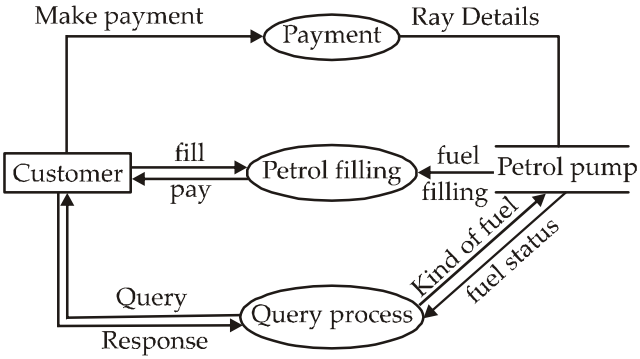
***Now**, Refer to Page No.-7 & 8, Q.No.-2 & Q.No.-3, Chapter-2*

**(b) Draw a Data Flow Diagram (DFD) till second level Depicting the various processes, data flow and data repositories for a "Petrol-Pump retail outlet".**

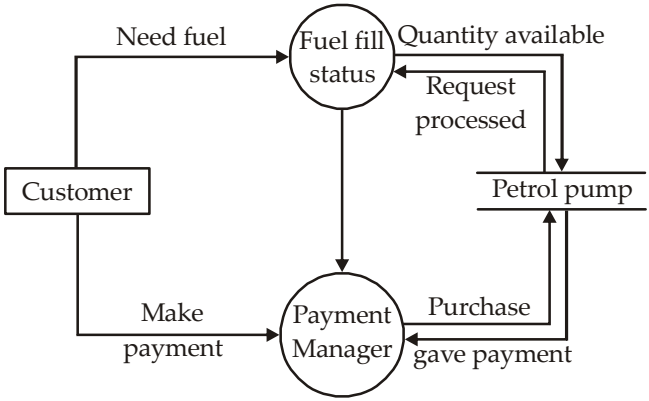
**Ans. Context Level DFD**



**Level 1 DFD**




**Level 2 DFD**




**(c) What is meant by "System Design Specification"? Explain the purpose of the following system design tools:**

**(i) Object Interaction Diagram**

**(ii) Inheritance Diagram**

**(iii) Aggregation Diagram**

**Ans.** The system design specification or software design specification as referred to has a primary audience, the system implementer or coder. It is also an important source of information for the system verification and testing. The system design specification gives a complete understanding of the details of each component of the system, and its associated algorithms, etc.

The system design specification documents all as to how the requirements of the system are to be implemented. It consists of the final steps of describing the system in detail before the coding starts.

The system design specification is developed in a two stage process: In the first step, design specification generally describes the overall architecture of the system at a higher level. The second step provides the technical details of

low-level design, which will guide the implementer. It describes exactly what the software must perform to meet the requirements of the system.

**(i) Object Interaction Diagram**

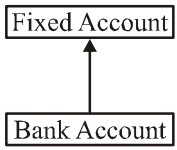
The purposes of interaction diagrams are to visualise the interactive behaviour of the system. Now visualising interaction is a difficult task. So the solution is to use different types of models to capture the different aspects of the interaction.

That is why sequence and collaboration diagrams are used to capture dynamic nature but from a different angle.

So the purposes of interaction diagram can be described as:

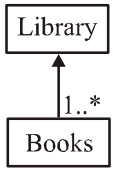
- To capture dynamic behaviour of a system.
- To describe the message flow in the system.
- To describe structural organisation of the objects.
- To describe interaction among objects.

**(ii) Inheritance Diagram**




Inheritance refers to a type of relationship wherein one associated class is a child of another by virtue of assuming the same functionalities of the parent class. In other words, the child class is a specific type of the parent class. To depict inheritance in a UML diagram, a solid line from the child class to the parent class is drawn using an unfilled arrowhead.

**(iii) Aggregation Diagram**




Aggregation refers to the formation of a particular class as a result of one class being aggregated or built as a collection. For example, the class "library" is made up of one or more books, among other materials. In aggregation, the contained classes are not strongly dependent on the life cycle of the container. In the same example, books will remain so even when the library is dissolved. To render aggregation in a diagram, draw a line from the parent class to the child class with a diamond shape near the parent class.

**(d) What is DSS? Explain various components briefly with the help of a suitable diagram.**

**Ans.** Refer to Page No.-267, Q.No.-5(d), June-2009 & Page No.-274, Q.No.-3(b), June-2010

**(e) What is reverse engineering? Explain the need of reverse engineering with an example.**

**Ans.** Refer to Page No.-104[Reverse and Forward Engineering]

**Need For Reverse Engineering:**

• The original manufacturer of a product no longer produces a product
• There is inadequate documentation of the original design
• The original manufacturer no longer exists, but a customer needs the product
• The original design documentation has been lost or never existed
• Some bad features of a product need to be designed out. For example, excessive wear might indicate where a product should be improved
• To strengthen the good features of a product based on long-term usage of the product
• To analyse the good and bad features of competitors' product
• To explore new avenues to improve product performance and features
• To gain competitive benchmarking methods to understand competitor's products and develop better products
• The original CAD model is not sufficient to support modifications or current manufacturing methods
• The original supplier is unable or unwilling to provide additional parts
• The original equipment manufacturers are either unwilling or unable to supply replacement parts, or demand inflated costs for sole-source parts
• To update obsolete materials or antiquated manufacturing processes with more current, less-expensive technologies.

**For example**, medical scientists try to reverse engineer the human body and football coaches try to reverse engineer the opposing team.

**Q2. (a) What are object oriented Case Tools? List basic features OO Case Tools. Also write differences between various types of OO Case Tools.**

**Ans. Object Oriented CASE Tools**

Object oriented CASE tools are similar to other CASE tools. These differ from others only in terms of their capability to create class diagrams and text specifications for reports. Object oriented CASE tools support an O-O methodology such as Rumbaugh's Object Management Technique (OMT).

**Examples of object oriented CASE tools**

A large number of object oriented CASE tools are available in the market. These include: Paradigm Plus from Protosoft, Rational Rose from Rational and With Class from MicroGold Software. Our discussion here will describe object oriented CASE tools in general without making references to a specific product.

The following are some of the features found across most of the Object Oriented CASE tools:

(1) Create graphics, such as class diagrams, message diagrams, state diagrams etc.

(2) Create text specifications such as system specification, class specification and relationship specification.
(3) Generate source code.
(4) Repository of models.

**Differences between various types of object oriented CASE tools**

All of these object oriented CASE tools are similar in terms of their capabilities to create class diagrams, code generation. However, they may differ in terms of their extendibility, number of supported operating systems and additional features and capabilities, such as support for different methodologies and computer language code generation (C++, Ada, Java etc). Most provide capability to automatically generate code of C++ and other languages from a class diagram and class specifications. Some provide the capability to generate class diagrams from code (reverse engineering). Figure below depicts drawing and text tools.

An object oriented CASE tool has the capabilities of drawing class diagrams and state diagrams. Specification tools create text reports.

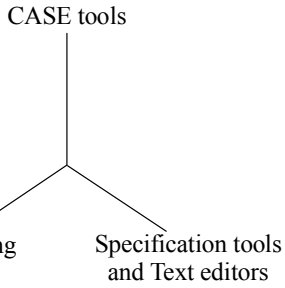



**Figure:** Depicts major outputs of object oriented CASE tools.

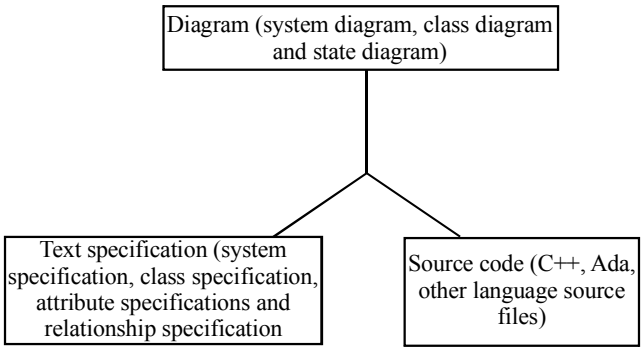



**Figure:** Major Output of Object Oriented CASE Tools

The class diagram is core to object-oriented design. It describes the types of objects in the system and the static relationships between them. The core element of the class diagram is the class. In an object-oriented system, classes are used to represent entities within the system. Entities are often related to real world objects.

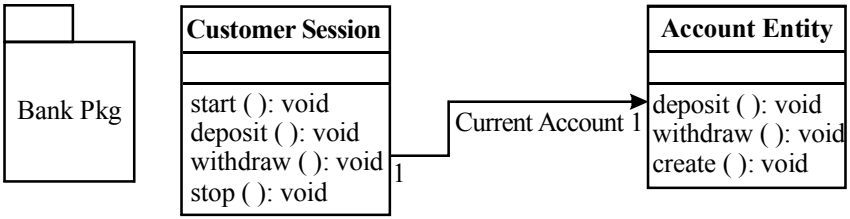



**Figure:** A class Diagram for a typical Banking Application

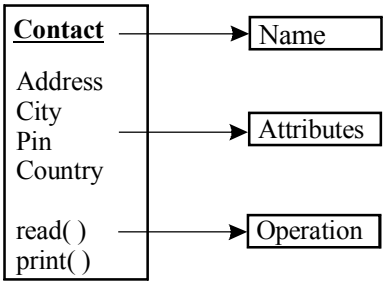



**Figure:** An Example of a Simple Class Contact

**(b) What is on-line transaction processing? Give an example of on-line transaction processing. Also differentiate between on-line transaction, processing and batch processing.**

**Ans. On-Line Transaction processing (OLTP):** In this technique, the data is processed by the system immediately after the transaction is over. This type of processing is well suited for small transactions and where turn around time is important. The database is always up-to-date since these are updated as when the transaction data is generated. Responses to the user inquiry are immediate. Since the database is accessed and updated after every transaction, care must be taken to protect the integrity of the database. Controls are some times built-in to the software for this kind of applications. Information systems related to Banking are examples of OLTP. The drawback of OLTP is high costs associated with the necessary security and fault tolerance features.

**Example:** In a banking System, you withdraw amount through an ATM. Then account Number, ATM PIN Number, Amount you are withdrawing, Balance amount in account etc are operational data elements.

- Operational Data
- Operational data are usually of local relevance
- Frequent Updates
- Normalised Tables
- Point Query

***Now**, Refer to Page No.-124 [Batch and Online transaction processing]*

**Q3. (a) What is transaction audit? Explain four advantages of transaction audit.**

**Ans.** Refer to Page No.-281[Transaction Audit]

**(b) What is Joint Application Development (JAD)? Explain six typical participants in a JAD.**

**Ans.** Refer to Page No.-45, Q.No.-6, Chapter-5

**Q4. (a) What is need of software testing? Describe the objectives of the following testing techniques.**

**(i) Stress Testing**

**(ii) Performance Testing**

**(iii) Recovery Testing**

**(iv) Security Testing**

**Ans.** Software development involves developing software against a set of requirements. Software testing is needed to verify and validate that the software that has been built has been built to meet these specifications. If not we may probably lose our client. So in order to make it sure, that we provide our client a proper software solution, we go for testing. Testing ensures that what you get in the end is what you wanted to build. We check out if there is any problem, any error in the system, which can make software unusable by the client. This helps in the prevention of errors in a system.

***Now,*** *Refer to Page No.-106, Q.No.-1, Chapter-11*

**(b) What is need of standard software documentation? Explain any three documentation standards briefly.**

**Ans.** Refer to Page No.-21, Q.No.-4, Chapter-4

**Q5. (a) Briefly, discuss various criteria for Form and Report design, using an example.**

**Ans.** Refer to Page No.-82, Q.No.-5, Chapter-5

**(b) Write a short note on following:**

**(i) Coupling and cohesion**

**Ans.** Refer to Page No.-57, Q.No.-6, Chapter-6

**(ii) Distributed system.**

**Ans.** Refer to Page No.-186, Q.No.-2(b)(i), Dec-2005

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2012

**Note:** *Question* ***No. 1*** *is compulsory. Attempt any three questions from the rest.*

**Q1. (a) List the underlying principles for prototype design. Under which conditions, this design will be useful? With the help of an example system, explain its approach.**

**Ans.** Refer to Page No.-227, Q.No.-1(b), June-2007

**(b) Draw a data flow diagram (DFD) till second level, depicting various processes, data flow and data repositories for a "Study centre Management System". Follow appropriate conventions/symbols as were given in your study material.**

**Ans.** Same as Page No.-272, Q.No.-1(a), June-2010

**(c) Explain what kind of fact finding techniques would you use for deciding on the design of a new application/system for some "Assignments Management System" at a study centre.**

**Ans. Fact Finding Techniques for Assignment Management System**

The set of tools that a systems analyst uses to learn more and more about the area to be computerised.

• **Record scanning:** A systems analyst studying the existing system leads to go through several records that the users build up, move around, refer to over and over again and finally preserve for some time period. Such records could take several forms. For example, in a banking environment, it could be vouchers, scrolls, registers, ledgers, etc. in a laboratory it could be journals, charts, experiment templates, etc. in the government scenario, it may be citisen's applications, supporting documents, notes or memos prepared etc. in a warehouse it could be the bin card, delivery challans, goods inspection note, stock ledger, etc.

• **Interviews:** While record scanning helps in getting snap shots of the process at various time intervals as reflected in documents, an interview is the best tool to appreciate the reasoning that tells us exactly how the processing takes place, how the decisions are arrived at, whether some small yet significant data carried in the heads of concerned people that is not getting documented, etc. an interview is a dynamic tool that could squeeze a lot of information that is typically in a latent form in the minds of people.

• **Questionnaires:** Where the systems analyst needs to interact with several people of comparable designation, role or status and who may also be posted

at far-off places, a questionnaire is a natural choice that supplements an initial interview of a few people.
Questionnaires presuppose several factors. Firstly, it should be a well thought, neatly structured document that is to be filled up by the respondent without the systems analy6st being personally available. It should be developed after the initial interviews so that is contents appeal to and interest the respondents. It has therefore to be explicit, self-contained and still concise. Ease of its forward and return transmission also impacts on its success.
• **Observations:** this tool believes that 'Seeing is believing'. While records may depict every single detail, interviews with users may also leave some points untouched. This is not to doubt user intentions, but out of sheer habit some workflow points may be too obvious for them and hence may not be expressed. Observations come in handy to cover all such issues.

**(d) Explain briefly the criteria for the following:**
**(i) Form Design**
**(ii) Report Design**
**Ans.** Refer to Page No.-191, Q.No.-4(b), Dec-2005

**Q2.(a) Define a Decision Support System (DSS). What are its characteristics? Also list and explain various components of DSS.**
**Ans.** Refer to Page No.-124, Q.No.-4, Chapter-13

**(b) Explain Reverse and Forward Engineering with the help of an example for each.**
**Ans.** Refer to Page No.-190, Q.No.-4(a), Dec-2005

**Q3. (a) Define Internal Information. Where is it useful? Also discuss various sub classes of internal information with the help of an example for each.**
**Ans.** Refer to Page No.-82[Internal Information]
**Subclasses of Internal information**
A *subclass*, *heir class*, or *child class* is a modular, derivative class that inherits one or more language entities from one or more other classes (called *superclasses*, *base classes*, or *parent classes*). The semantics of class inheritance vary from language to language, but commonly the subclass automatically inherits the instance variables and member functions of its superclasses
**Uses:**
(i) In marketing and sales information, e.g. Revenues market share, etc.
(ii) Production and operational information, e.g. Assets, Quality, Lead time, etc.

(iii) Financial information, e.g. Profits, Costs, Margins, Investments, etc.
(iv) Research & Development information, e.g. Knowledge base, Copyright, etc.

**(b) What is Cohesion? Explain any four types of Cohesion.**
**Ans.** Refer to Page No.-58[Cohesion]

**Q4. (a) List and explain the Goals of a Good Design. Also discuss as set of guidelines to be followed basing upon the good design.**
**Ans.** Refer to Page No.-54, Q.No.-4, Chapter-6

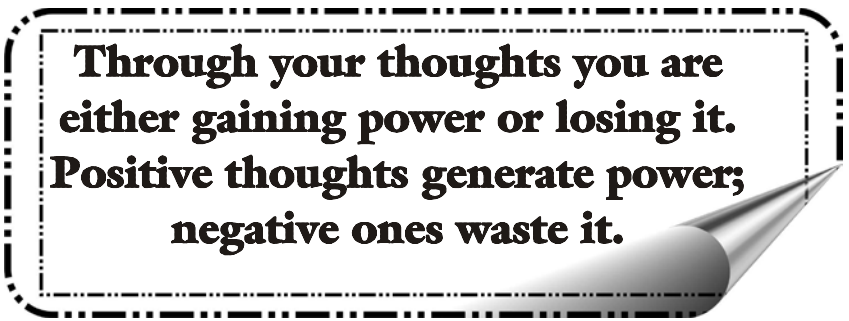
**(b) Explain various steps involved in the process of documentation.**
**Ans.** Refer to Page No.-21, Q.No.-3, Chapter-4

**Q5. Write short notes on the following:**
**(a) Data Dictionary**
**Ans.** Refer to Page No.-35, Q.No.-11, Chapter-4

**(b) Joint Application Development**
**Ans.** Refer to Page No.-45[Joint Application Development]

**(c) Cost Benefit Analysis**
**Ans.** Refer to Page No.-48, Q.No.-8, Chapter-5

**(d) Any two types of system testing.**
**Ans.** Refer to Page No.-106, Q.No.-1, Chapter-11

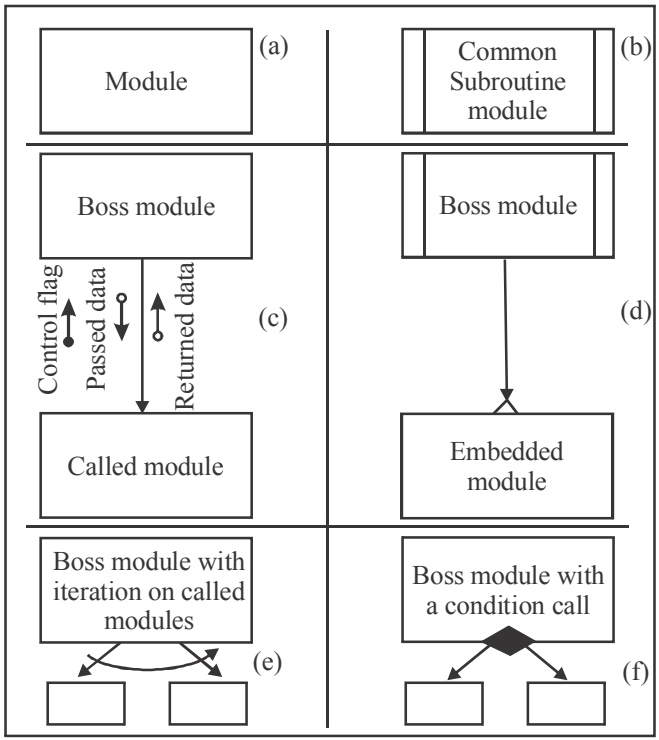


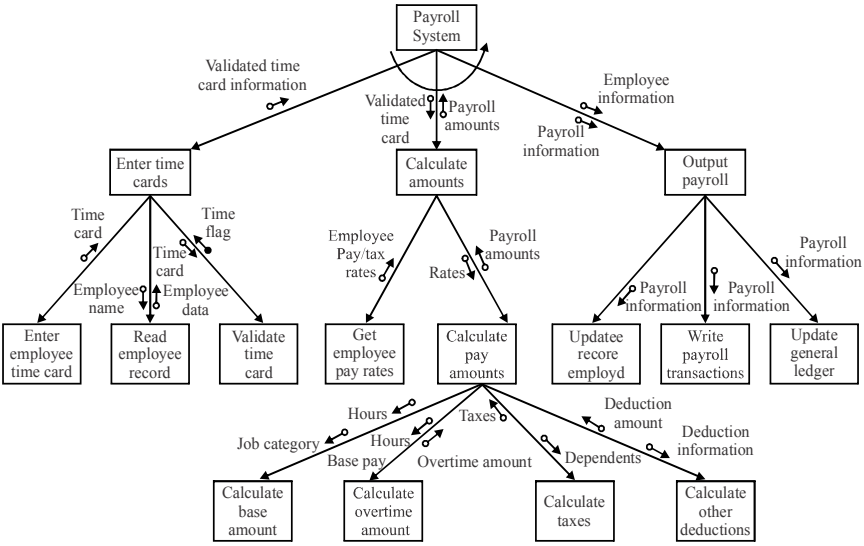
**MCS-014 : SYSTEM ANALYSIS AND DESIGN**
**June, 2013**

**Note:** *Question* ***no. 1*** *is* ***compulsory*** *and carries* ***40*** *marks. Attempt any* ***three*** *questions from the rest.*

**Q1. (a) Draw a Data Flow Diagram (DFD) till second level depicting various processes, data flow and data repositories for a "Library Management System". Follow the conventions.**
**Ans.** Refer to Page No.-284, Q.No.-1(a), June-2011

**(b) Define a Structure Chart. Draw a Structure Chart for a Payroll Processing System. Also, explain the symbols used in the chart.**
**Ans.** A Structure Chart (SC) in software engineering and organisational theory, is a chart which shows the breakdown of a system to its lowest manageable levels and Describes functions and subfunctions of each part of system. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure shows relationships between modules of a computer program.






**(c) Define modularity. Describe the ways and means to achieve modularity. Explain with the help of an example.**

**Ans.** Refer to Page No.-54, Q.No.-4, Chapter-6

**(d) Write the importance of quality in software development. List specifications of various quality factors.**

**Ans.** In software development the quality is software is very important because quality factors are properties such as speed or ease of use, whose presence or absence in a software product may be detected by its users. Other qualities applicable to a software product, such as being modular, or readable, are internal factors, perceptible only to computer professionals who have access to the actual software text. There are various factors that affect software quality:

**Correctness:** The ability of software products to perform their tasks, as defined by their specification.

**Robustness:** The ability of software systems to react appropriately to abnormal conditions.

**Extendibility:** The ease of adapting software products to changes of specification.

**Reusability:** The ability of software elements to serve for the construction of many different applications.

**Compatibility:** The ease of combining software elements with others.

**Efficiency:** The ability of a software system to place as few demands as possible on hardware resources, such as processor time, space occupied in internal and external memories, bandwidth used in communication devices.
**Portability:** The ease of transferring software products to various hardware and software environments.
**Ease of use:** The ease with which people of various backgrounds and qualifications can learn to use software products and apply them to solve problems. It also covers the ease of installation, operation and monitoring.
**Timeliness:** The ability of a software system to be released when or before its users want it.
**Other qualities:** Other qualities beside the ones discussed so far affect users of software systems and the people who purchase these systems or commission their development. In particular: Verifiability, Integrity, Repairability, and Economy.

**Q2. (a) Explain Real Time Systems and Distributed Systems.**
**Ans.** Refer to Page No.-186, Q.No.-2(b), June-2005

**(b) Explain Structured Analysis and Structured Design. What are its goals?**
**Ans.** Structured Analysis is a development method for the analysis of existing manual systems or automated systems, leading to development of specifications (expected functionality or behaviour) for proposed system. Structured analysis introduced a process-modeling tool called the Data flow diagram, used to illustrate business process requirements The objective of structured analysis approach is to organize the tasks associated with requirement determination to provide an accurate and complete understanding of a current situation. The major tasks of structured system analysis approach are:
• Preliminary Investigation
• Problem Analysis
• Requirement Analysis
• Decision Analysis.
Structured Design utilizes graphic description (Output of system analysis) and focuses on development of software specifications. The goal of structured design is to lead to development of programs consisting of functionally independent modules that perform relatively independently of one another. Structured Design introduced a modeling tool called Structure Charts It is a specific program design technique, not a comprehensive design method. Thus it does not specify file or database design, input or output layout or the hardware on which the application will run. It provides specification of program modules that are functionally independent.
It is a process-centered technique that transforms the structured analysis models into good software design models. They are used to illustrate software (program) structure to fulfill business requirements. Structure charts describe the interaction between independent module and the data passing between the modules.

The goal of structured system analysis and design is to reduce maintenance time and effort. Modeling is the act of drawing one or more graphical representations of a System. Model driven development techniques emphasize the drawing of models to help visualize and analyze problems, define business requirements and design Information systems. The first model driven approach is Structured Analysis and Design approach.

**Q3. (a) What are the various activities involved in the feasibility study? Discuss any three feasibilities need to be studied during feasibility analysis.**

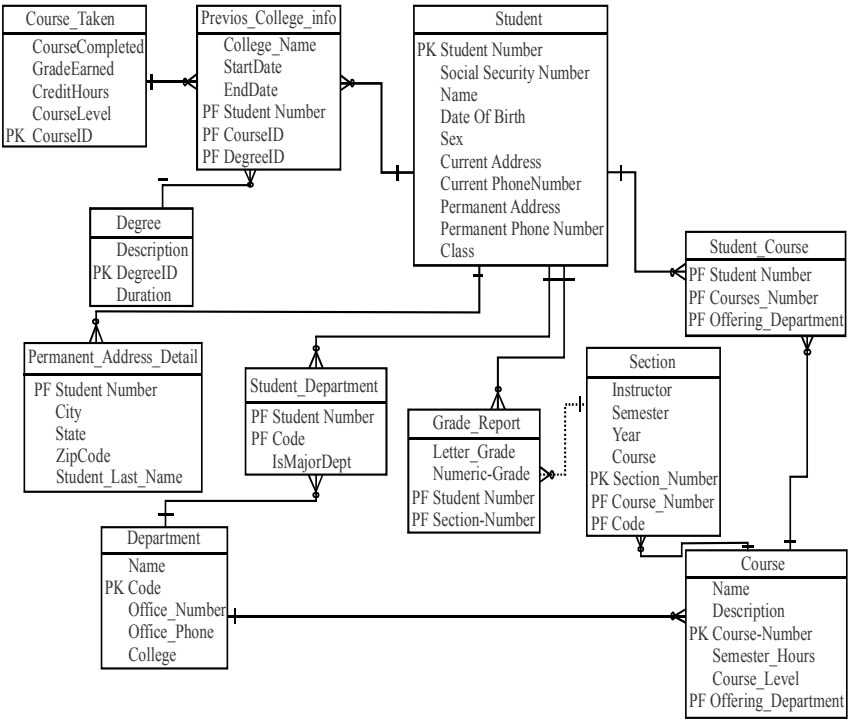
**Ans.** Refer to Page No.-46, 47 & 48 Q.No.-7, Chapter-5

**(b) What are top-down design and bottom-up approaches? Design top-down approach for any system of your choice.**

**Ans.** Refer to Page No.-240, Q.No.-1(c) Dec-2007, and Page No.-229, June-2007 [top down design on inventory system].

**Q4. (a) Describe all the steps involved in database design for the system discussed in 1 (a) question.**

**Ans.** Refer to Page No.-86, Q.No.-1, Chapter-9



**(b) Define CASE tools. Explain their role.**

**Ans.** Refer to Page No.-98 & 100, Chapter-10

**(c) With the help of an example, explain a sequence diagram.**

**Ans.** Refer to Page No.-237, Q.No.-5(iv), June-2007

**Q5. Write short notes on the following :**

**(a) Participatory Design**

**Ans.** Participatory Design (PD) represents a useful alternative approach to the SDLCPD emphasizes the role of the user much more than other techniques do. In some cases, PD may involve the entire user community in the development process. Each user has an equal share in determining system requirements and in approving system design. In other cases, an elected group of users control the process. These users represent the larger community. Under PD, systems analysts work for the users. The organization's management and outside consultants provide advice rather than control. PD is partly a result of the role of labour and management in the workplace where labour is more organized and is more intimately involved with technological changes.

**(b) Test Design Document**

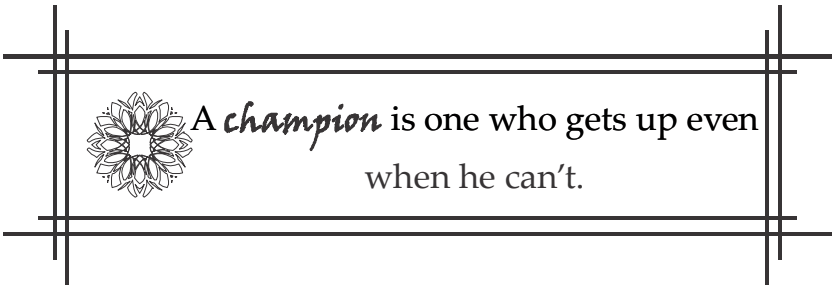
**Ans.** Refer to Page No.-246, Q.No.-5(b), Dec-2007

**(c) Coupling**

**Ans.** Refer to Page No.-55, Q.No.-5, Chapter-6

**(d) Decision Tables**

**Ans.** Refer to Page No.-67, Q.No.-3, Chapter-7



## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2013

**Note:** *Question* ***no. 1*** *is* ***compulsory*** *and carries* ***40*** *marks. Attempt any* ***three*** *questions from the rest.*

**Q1. (a) Explain how both waterfall model and prototyping model can be accommodated in the spiral process model.**

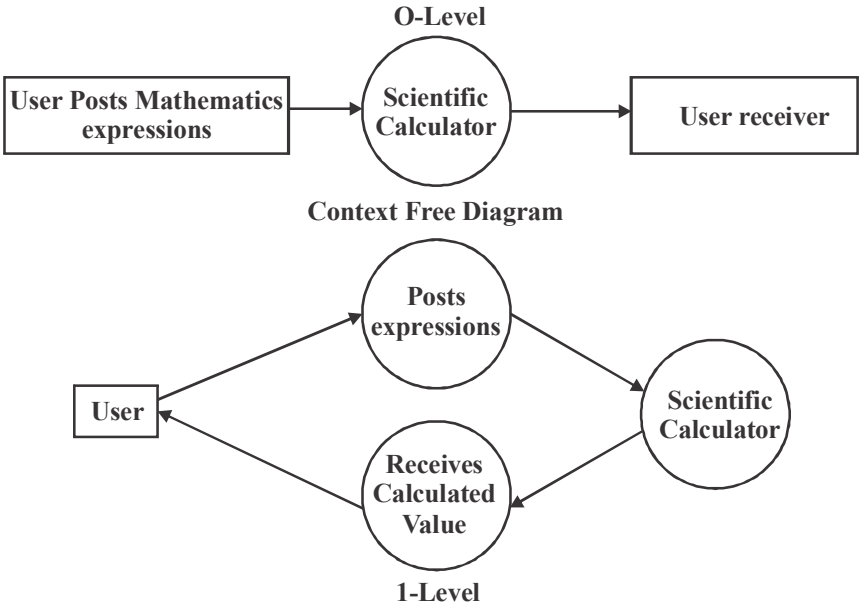
**Ans.** The Spiral Life Cycle Model is a type of iterative software development model which is generally implemented in high risk projects. It was first proposed by Boehm. The spiral model was introduced, due to the shortcomings in the waterfall and prototype models of software engineering. It is a combination of the said two models of software development. From the name of the model, it can be derived that the activities of software development are carried out like a spiral. Spiral model is an evolutionary software process model which is a combination of an iterative nature of prototyping and controlled and systematic aspects of traditional waterfall model. To explain the model further,the entire software development process is broken down into small projects. the phases of the spiral model are as follows :

• Planning phase
• Risk Analysis phase
• Engineering phase
• Coding and implementation phase
• Evaluation phase

Each of the regions is populated by a set of work tasks, called a task set, that are adapted to the characteristics of the project to be undertaken. For small projects, the number of work tasks and their formality is low. For larger, more critical projects, each task region contains more work tasks that are defined to achieve a higher level of formality.

**(b) A program is to be developed to simulate the operations of a scientific calculator. List the facilities to be provided by this calculator. Analyze this using a DFD 0-level and 1-level diagram.**

**Ans.**




**The facilities to be provided by this calculator:**

- scientific notation
- floating point arithmetic
- logarithmic functions, using both base 10 and base e
- trigonometric functions (some including hyperbolic trigonometry)
- exponential functions and roots beyond the square root

**(c) Give full form of CASE/ Draw CASE diagram showing all components. Describe each component briefly.**

**Ans.** Refer to Page No.-98, 101 & 103, Chapter-10

**(d) Distinguish between Analysis and design. Explain with example.**

**Ans.** Refer to Page No.-52, Q.No.-1, Chapter-6

**(e) Explain the following terms:**

**(i) Debugging**

**Ans.** Debugging is a methodical process of locating and reducing the number of bugs, or defects, in a computer program or a piece of electronic hardware, thus making it behave as expected.

To debug a program or hardware device is to start with a problem, isolate the source of the problem, and then fix it. A user of a program that does not know how to fix the problem may learn enough about the problem to be able to avoid it until it is permanently fixed. When someone says they've debugged a program or "worked the bugs out" of a program, they imply that they fixed it so that the bugs no longer exist. Debugging tools (called debuggers) help identify coding errors at various development stages. Some programming language packages include a facility for checking the code for errors as it is being written.

Debugging is a necessary process in almost any new software or hardware development process, whether a commercial product or an enterprise or personal application program. For complex products, debugging is done as the result of the unit test for the smallest unit of a system, again at component test when parts are brought together, again at system test when the product is used with other existing products, and again during customer beta test, when users try the product out in a real world situation. Because most computer programs and many programmed hardware devices contain thousands of lines of code, almost any new product is likely to contain a few bugs. Invariably, the bugs in the functions that get most use are found and fixed first. An early version of a program that has lots of bugs is referred to as "buggy."

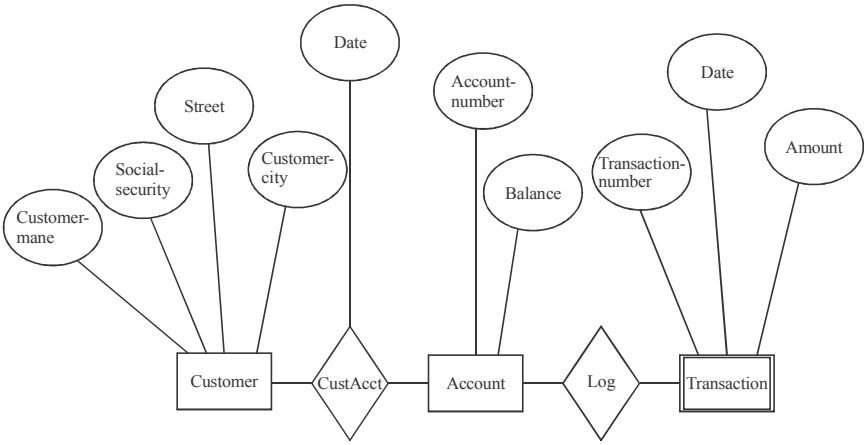
**(ii) Cost-Benefit Analysis.**

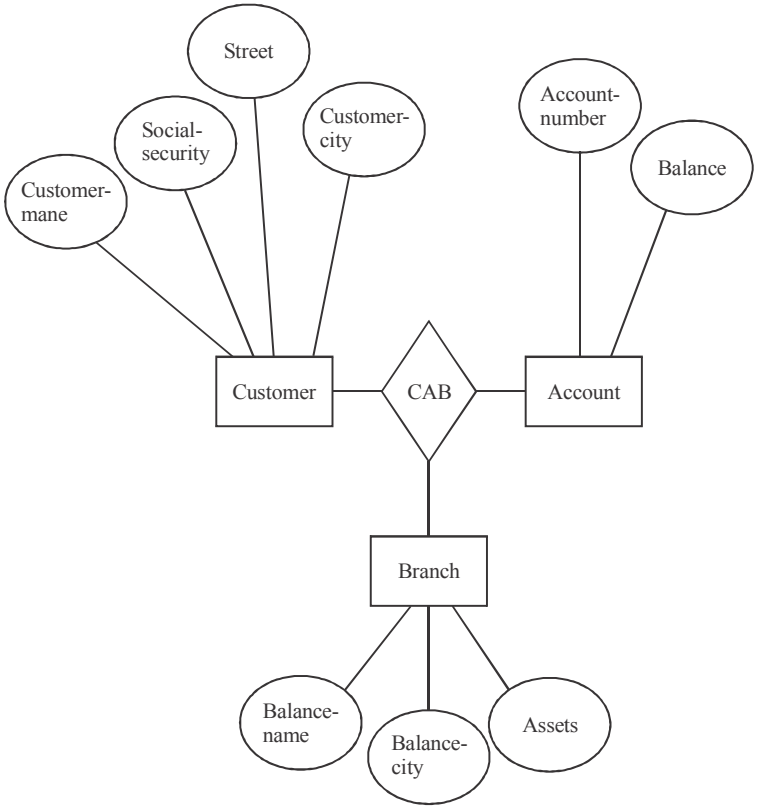
**Ans.** Refer to Page No.-48, Q.No.-8 & 9, Chapter-6

**Q2. (a) Draw ER diagram for the following situation:**

**An account is a relationship between a customer and a bank. A customer has a name. A bank has a branch. A customer may have several accounts of different types and balance.**

**Ans.**


Date
Street
Account-number
Date
Social-security
Customer-city
Transaction-number
Amount
Customer-mane
Balance
Customer
CustAcct
Account
Log
Transaction


Street
Account-number
Social-security
Customer-city
Balance
Customer-mane
Customer
CAB
Account
Branch
Balance-name
Balance-city
Assets

**(b) Describe characteristics of a good interface. List some guidelines to design output screens icons and graphics.**

**Ans.** User interface design is the design of websites, computers, applications and likewise. While designing the first and foremost thing that needs to be kept in mind is the user experience and interaction. As the name says its an interface design in which users are of prime importance The main target behind designing an interface design is to make the user interaction as simple and efficient as possible There are some characteristics of a good interface:

**Clarity:** Clarity is the most important element of user interface design. Indeed, the whole purpose of user interface design is to enable people to interact with your system by communicating meaning and function. If people can't figure out how your application works or where to go on your website they'll get confused and frustrated.

**Conciseness:** concise which means if something can be explained without sparing extra words do that! The idea is to save the valuable time of the users by keeping things as concise as possible.

**Consistency:** Consistent interfaces allow users to develop usage patterns – they'll learn what the different buttons, tabs, icons and other interface elements look like and will recognize them and realize what they do in different contexts.

**Familiar:** Familiar is just that: something which appears like something else you've encountered before. When you're familiar with something, you know how it behaves – you know what to expect. Identify things that are familiar to your users and integrate them into your user interface.

**Attractive:** Attractive in a sense that it makes the use of that interface enjoyable. When your software is pleasant to use, your customers or staff will not simply be using it – they'll look forward to using it.

**Efficient:** A user interface is the vehicle that takes you places. Those places are the different functions of the software application or website. A good interface should allow you to perform those functions faster and with less effort.

**Forgiving:** A forgiving interface is one that can save your users from costly mistakes. For example, if someone deletes an impo0rtant piece of information, can they easily retrieve it or undo this action.

**(c) Why is documentation important to information system life cycle? Explain. Also describe the term program specification in brief. What information does it convey?**

**Ans.** Refer to Page No.-21, Q.No.-4, Chapter-4

**Q3. (a) Explain the term 'Expert System'. Briefly explain the characteristics of an expert system. Give an example.**

**Ans.** Refer to Page No.-124, Q.No.-4, Chapter-13

**(b) Why is maintenance of a software important? Discuss some of the problems that are faced during maintenance of software.**

**Ans.** The modification of a software product, after delivery, to correct faults, to improve performance or other attributes, or to adapt the product to a changed environment. Maintenance is an important part of the software life-cycle. It is expensive in manpower and resources, and one of the aims of software engineering is to reduce its cost.

The most important problem during maintenance is the before correcting or modifying a program, the programmer must first understand it.

The problems are:

• Often another person or group of persons working over the years in isolation from each other writes the program.

• Often the program is changed by person who did not understand it clearly, resulting in a deterioration of the program's original organisation.

• There is a high staff turnover within information technology industry. Due to this persons who are not the original authors maintain many systems. These persons may not have adequate knowledge about the system.

• Some problems only become clearer when a system is in use. Many users know what they want but lack the ability to express it in a form understandable to programmers. This is primarily due to information gap.

**(c) Differentiate between software Re-engineering and software Reverse Engineering. Explain their need with suitable examples.**

**Ans. Re-engineering:**

Re-structuring or re-writing part or all of a legacy system without changing its functionality is called Reengineering. It involves putting in the effort to make it easier to maintain and may also be restructured and should be redocumented. The system may be re-structured and re-documented Re-structuring or re-writing part or all of a legacy system without changing its functionality.

The Reengineering is needed When system changes are confined to one subsystem.

The subsystem needs to be reengineered When hardware or software support becomes obsolete, When tools to support restructuring are readily available.

**Reverse Engineering:**

Reverse Engineering is Analysing software with a view to understanding its design and specification. It May be part of the reengineering process but may also be used to re-specify a system for re-implementation and May be used to specify a system for prior to reimplementation. Reverse Engineering is builds a program data base and generates information from this Program understanding tools (browsers, cross-reference generators, etc.) may be used in this process. Reverse engineering has its origins in the analysis of hardware for commercial or military advantage. The purpose is to deduce design decisions from end products with little or no additional knowledge about the procedures involved in the original production. The same techniques are subsequently being researched for application to legacy software systems, not for industrial or defence ends, but rather to replace incorrect, incomplete, or otherwise unavailable documentation.

**Q4. (a) Explain the difference between a structured interview and an unstructured interview? When is each type of interview appropriately used?**

**Ans.** Structured interviews work through a specific set of questions in a method proscribed by the HR department for that company. It is one way to insure that that same questions get asked to all of the candidates. It is often used to be able to show that the best person for the job was selected and that it was a fair interview for all of the candidates.

Unstructured interviews are less structured and the individual interviewer can determine what they are going to ask the candidate. Unstructured interviews are more like an everyday conversation. They tend to be more informal, open ended, flexible and free flowing. Questions are not pre-set, although there are usually certain topics that the researchers wish to cover.

Unstructured interview technique was originated in anthropology and sociology as a method to bring out people's social realities.

Structured text has a pre-planned order, while unstructured text does not. For examples: Companies spend thousands of dollars for web developers to organize data on web pages and websites, so there is order to content, images, videos, etc. In contrast, millions of teenagers make their first websites without much regard for the order or presentation; they use an unstructured plan to place unstructured text. Two students read the same book to use in their term papers. One student uses index cards to organize his text, but the other student just jots down ideas on a loose leaf sheet of paper. A book consists of structured text, while a stack of journal pages is not structured text.

**(b) Explain the following (with an example).**

**(i) Inheritance Diagram**

**Ans.** Refer to Page No.-213, Q.NO.-2(a)(iii), Dec-2006

**(ii) Aggregation Diagram**

**Ans.** The E-R model cannot express relationships among relationships. An aggregation diagram shows relationships among objects. When a class is formed as a collection of other classes, it is called an aggregation relationship Introduction to Systems Development between these classes. Each module will be represented by its name. The relationship will be indicated by a directed line from container to container. The directed line is labelled with the cardinality of the relationship. It describes "has a" relationship. Figure shows an aggregation between two classes (circle has a shape).

**(c) Prepare SRS for an "Inventory Management System" for a retail medical store. Also explain the Risk assessment and management is brief. Make necessary assumptions.**

**Ans.** Refer to Page No.-278, Q.NO.-2(a), Dec-2010

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Since there could be various risks associated with the software development projects, the key to identify and manage those risks is to know about the concepts of software risk management. Risk management is the most important issue involved in the software project development. Risk management is the most important issue involved in the software project development. This issue is generally managed by Software Project Management (SPM). During the life cycle of software projects, various risks are associated with them. These risks in the software project is identified and managed by software risk management which is a part of SPM. Many concepts about software risk management could be identified but the most important are risk index, risk analysis, and risk assessment.

**Risk Index:** Generally risks are categorized into two factors namely impact of risk events and probability of occurrence. Risk index is the multiplication of impact and probability of occurrence. Risk index can be characterized as high, medium, or low depending upon the product of impact and occurrence. Risk index is very important and necessary for prioritisation of risk.

**Risk Analysis:** There are quite different types of risk analysis that can be used. Basically, risk analysis is used to identify the high risk elements of a project in software engineering. Also, it provides ways of detailing the impact of risk mitigation strategies. Risk analysis has also been found to be most important in the software design phase to evaluate criticality of the system, where risks are analyzed and necessary counter measures are introduced. The main purpose of risk analysis is to understand risks in better ways and to verify and correct attributes. A successful risk analysis includes important elements like problem definition, problem formulation, data collection.

**Risk Assessment:** Risk assessment is another important case that integrates risk management and risk analysis. There are many risk assessment methodologies that focus on different types of risks. Risk assessment requires correct explanations of the target system and all security features. It is important that a risk referent levels like performance, cost, support and schedule must be defined properly for risk assessment to be useful.

**Q5. (a) Explain various types of file organisation. Differentiate between a master file and a transaction file.**

**Ans.** Refer to Page No.-89, 90 & 91, Chapter-9

**(b) Differentiate between the following :**

**(i) Batch and Online Transaction Processing.**

**Ans.** Refer to Page No.-124, Q.No.-3, Chapter-13

**(ii) TPS and DSS**

**Ans.** Refer to Page No.-149, Q.No.-25, Chapter-13

**(c) Describe the aims and methodology of the following testing techniques:**

**(i) Black Box Testing**

**Ans.** Refer to Page No.-297, Q.No.-5(b)(iii), Dec-2011

**(ii) White Box Testing**

**Ans.** Refer to Page No.-297, Q.No.-5(b)(iii), Dec-2011

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## June, 2014

**Note:** *Question **no. 1** is **compulsory.** Attempt any **three** questions from the rest.*

**Q1. (a) What type of projects are amenable for Spiral model? Justify your answer with an example.**
**(b) Explain the characteristics of an Information System in detail.**
**(c) Consider a software which registers students for different programmes. The students fill a form and submit it. This is sent to the departments for confirmation. Once it is confirmed, the form and the fees is sent to the account section. Draw a DFD 1 - level and 2 - level for development of software mentioned.**
**(d) Differentiate between function-oriented design and object-oriented design. Also, explain the problems which arise if two modules have high coupling.**
**(e) Explain various criteria and specifications to be considered while designing forms and Reports.**

**Q2. (a) List any five fact finding techniques for system study and explain any two of them in detail. Differentiate between these two with respect to their merits and demerits.**
**(b) Explain the characteristics of MIS. Give the components of MIS. Also differentiate between DSS and expert system.**
**(c) Give an example of a structure chart and explain it.**

**Q3. (a) Distinguish between technical, operational and economic feasibility with suitable examples.**
**(b) Prepare SRS for 'Airline Reservation System.' Make assumptions wherever necessary.**
**(c) What are the roles of a system analyst in system development? What is a data dictionary? Mention the uses of data dictionary by a system analyst, with the help of an example. Explain the contents of data dictionaries.**

**Q4. (a) What is a Test Design Document? Explain the contents of this document.**
**(b) Differentiate between the following:**
**(i) Flat files and Database (ii) System Testing and Unit Testing**
**(c) Categorise CASE tools. Give a diagram to explain various components of a CASE tool.**

**Q5. (a) Give short notes on:**
**(i) Transaction Audit (ii) Types of Coupling (iii) Decision Trees.**
**(b) What is the need of functional decomposition in software development? Explain with an example.**
**(c) Explain the concept of Information Security and its architecture. Write a short note on threats and risks in a system.**

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2014

**Note:** *Question* ***no. 1*** *is* ***compulsory.*** *Attempt any* ***three*** *questions from the rest.*

**Q1. (a) What is an Information System? Explain classification of systems in brief. Also explain the need of SDLC for proper development of a system.**
**(b) What is SRS? Briefly explain any four characteristics of SRS. Develop an SRS for Library Management System. Make appropriate assumptions.**
**(c) What is coupling? Explain advantages and disadvantages of a highly coupled system.**
**(d) Explain basic guidelines for user interface design.**
**(e) What is the need of documentation in System Development? Explain, in brief, any two categories of documentation.**
**(f) What is CASE tool? Categorise various types of CASE tools.**

**Q2. (a) What are the various security issues in a computer system? Explain how an organisation may prevent security concerns from data stored in its centralised and distributed databases.**
**(b) What are Forms? Also explain the process of Form design.**

**Q3. (a) Why is a proper database design essential for a successful information system? Explain rules for designing tables, fields and constraints.**
**(b) What are the advantages of modular system design? Draw a structure chart for Banking Management System.**

**Q4. (a) What is a decision tree? Draw a decision tree for a system of your choice.**
**(b) What is a flow chart? Draw a flow chart to find the average marks of MCS-014 of the 10 students in a batch at a study centre.**
**(c) What is the need of system maintenance? Explain different types of maintenance.**

**Q5. Write short notes on the following:**
**(a) Expert System**
**(b) Prototyping Model**
**(c) Decision Table**
**(d) System Testing**

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## June, 2015

**Note:** *Question* ***no. 1*** *is* ***compulsory.*** *Attempt any* ***three*** *questions from the rest.*

**Q1. (a) What is the role of a system analyst in system development? Explain any six attributes of a good system analyst.**
**(b) What is modularity? How can modularity in design be achieved? Explain the advantages of modular design.**
**(c) What is MIS? What are the various functions of MIS in a business organisation?**
**(d) What is the need of feasibility study of an information system? Explain the different operational feasibility issues of a system.**
**(e) What is database design? Explain the different issues involved in the designing of a database.**

**Q2. (a) What is cost and effort estimation? Give a -brief explanation of any cost and effort estimation model.**
**(b) What is decision table? Design a decision table for Library Management System. Make appropriate assumptions.**
**(c) Explain the responsibility and authority of the system auditor.**

**Q3. (a) What is DFD? Draw a DFD (0 and 1 level) for Online Admission System to offer online admission to graduate and post graduate students of a university.**
**(b) Explain the steps and activities in the testing and maintenance phases of SDLC.**

**Q4. (a) What is Decision Support System (DSS)? What are its basic characteristics? Briefly explain the various components of DSS, with the help of a diagram.**
**(b) Draw the object interaction diagram for a system of your choice.**
**(c) What is a report? Briefly explain the criteria for report design.**

**Q5. Write short notes on the following:**
**(a) Software Quality Assurance**
**(b) CASE Tools**
**(c) Real Time System**
**(d) Software Documentation**

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2015

**Note:** *Question* ***no. 1*** *is* ***compulsory.*** *Attempt any* ***three*** *questions from the rest.*

**Q1. (a) Draw a 0-level and 1-level DFD of a library information system. Make necessary assumptions.**
**(b) Explain the term 'Feasibility study'. Explain the categories of feasibility with suitable examples.**
**(c) Explain waterfall model with the help of a diagram. How is it different from spiral model? Explain the merits and demerits of both the models.**
**(d) Differentiate between various types of file organisations.**
**(e) Explain the term MIS. Explain the components of a transaction processing system with suitable diagram.**

**Q2. (a) Construct an Entity Relationship diagram for an airlines reservation system.**
**(b) What is meant by the term CASE? Describe the various components and tools of CASE.**
**(c) Explain the terms 'User manual' and 'System documentation'.**

**Q3. (a) What is the relevance of an Information Security Architecture in MIS? Draw its diagram.**
**(b) Differentiate between System Analysis and System Design. What are the issues and constraints to be taken care of while designing a software product?**
**(c) Differentiate between Reverse engineering and Forward engineering. What is meant by a legacy system?**

**Q4. (a) Prepare SRS for a "University admission system". Make necessary assumptions.**
**(b) Differentiate between coupling and cohesion. Explain the various types of coupling.**
**(c) What is meant by "User Interface"? What are the basic guidelines to design a user interface?**

**Q5. (a) What are the essential criteria for form design and report design? Explain.**
**(b) Explain software maintenance and differentiate between various types of software maintenance.**
**(c) Give any one method of cost benefit analysis and explain it.**

**MCS-014 : SYSTEM ANALYSIS AND DESIGN**
**June, 2016**

**Note:** *Question* ***no. 1*** *is* ***compulsory.*** *Attempt any* ***three*** *questions from the rest.*

**Q1. (a) Prepare SRS for a library management system. Make necessary assumptions.**
**(b) Draw ERD for a library management system. Explain various components and relationships in detail.**
**(c) Differentiate between DFD and Flow chart. Discuss the importance of levels of DFD.**
**(d) Explain the following diagrams with examples:**
**(i) Structured chart**
**(ii) State-machine diagram**
**(e) What are the guidelines for database design? Explain the term 'Relational Database Schema'. Give an example of this schema.**

**Q2. (a) Differentiate between Software Re-engineering and Reverse engineering.**
**(b) Draw a decision table and a decision tree for a problem statement of your choice. Make necessary assumptions.**
**(c) What is meant by 'Expert System'? Why is it required? Give its features.**

**Q3. (a) What are the components of a MIS? Differentiate between DSS and TPS. Also explain various components of TPS in detail.**
**(b) Explain various types of cohesion. Give an example of a cohesion module. Explain.**
**(c) Explain any three fact finding techniques. Give their merits and demerits.**

**Q4. (a) What is a 'User Interface'? Explain the guidelines for designing a user interface.**
**(b) Differentiate between various concurrent audit techniques explaining their merits and features.**
**(c) What are the various standards of documentation? Also, explain in brief, the various components of software user documentation.**

**Q5. (a) Explain the differences between functional testing and structural testing.**
**(b) What are the various steps in logical design? Explain the use of normalization in database design.**
**(c) Explain the responsibilities of a system analyst. Give any two essential qualifications of a system analyst.**

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2016

**Note:** *Question* ***no. 1*** *is* ***compulsory.*** *Attempt any* ***three*** *questions from the rest.*

**Q1. (a) Draw a Data Flow Diagram (DFD) from context level till second level clearly showing various processes, data flow and data repositories for a "Library Management System".**
**Ans.** Refer to Page No.-284, Q.No.-1(a), June-2011

**(b) Define 'Modularity'. Write any two goals of a good design. Also write at least four guidelines for achieving the goals mentioned.**
**Ans.** Refer to Page No.-54, Q.No.-4, Chapter-6

**(c) Explain how both waterfall model and prototyping model can be accommodated in the spiral model.**
**Ans.** Refer to Page No.-314, Q.No.-1(a), December-2013

**(d) Explain relational, hierarchical and network models of database design, with examples.**
**Ans.** Refer to Page No.-93 & 94, Q.No.-3, Chapter-9

**Q2. (a) Describe the characteristics of a good interface. List some guidelines to design output screens, icons and graphics.**
**Ans.** Refer to Page No.-318, Q.No.-2(b), December-2013

**(b) What is an E-R diagram? Construct an E-R diagram for a "Railway Reservation System". Make necessary assumptions.**
**Ans.** Refer to Page No.-26, Q.No.-9, Chapter-4

**Q3. (a) Prepare an outline of SRS for a "Railway Reservation System". Make suitable assumptions wherever necessary.**
**Ans.** Refer to Page No.-224, Q.No.-1(a), June-2007

**(b) Define the term 'Cohesion'. Explain any two types of cohesion.**
**Ans.** Refer to Page No.-57, Q.No.-6, Chapter-6

**Q4. (a) Explain any two ways of Requirements Gatherings. What are their shortcomings? Use examples to explain.**
**Ans. (1) Success criteria is not defined clearly:** It is normal for stakeholders to know that they have a problem or an opportunity to explore, but not know

exactly what they want. The key to addressing this issue is to break the project into smaller pieces, and start from a section that the client is clearer about. Collaborative modeling tools allow for giving the clients a high level vision of the end product, and getting their feedback early on in the process. Also helpful is asking for examples of systems that they like and what they like about them, to better understand their current business process and identify pain points that need to be rectified or improved. In any case, it is important to make requirements quantifiable and testable, in order to have a solid basis for measuring results later.

**(2) Stakeholders have Conflicting priorities:** When a new system is created, it must adhere to the needs of several groups of stakeholders, such as end users and senior management. It is possible for these various groups to have conflicting views and priorities. The best approach to this problem is to have a designated authority in the organization who is in charge of negotiating the conflicting matters and makes the final decision. The requirement gathering process requires having tough, open ended questions for the stakeholders to answer. Stakeholders need time to fully articulate their ideas and perspective. Rushing the process may result in proposed terms that are considered out of scope, or promoting individual agendas rather than the organization's vision. A good practice is to have multiple meetings with enough time between meetings for the stakeholders to digest the outputs, thus ensuring that the requirement gathering process is on the right track.

**(b) Define the term "Decision Support System". Explain its various components.**

**Ans.** Refer to Page No.-267, Q.No.-5(d), June-2009 and Page No.-274, Q.No.-3(b), June-2010

**Q5. (a) What is a Form? What are the advantages of having forms as part of user interface?**
**What are the criteria used for Form Design?**

**Ans.** Refer to Page No.-79, 82 & 84, Q.No.-1, Q.No.-5 & Q.No.-6, Chapter-8

**(b) How does Reverse Engineering differ from Forward Engineering? Explain with example.**

**Ans.** Refer to Page No.-104, Chapter-10

**MCS-014: SYSTEM ANALYSIS AND DESIGN**
**June, 2017**

**Note:** *Question **No. 1** is **compulsory**. Attempt any **three** questions from the rest.*

**Q1. (a) Draw a Data Flow Diagram (DFD) from context level till second level clearly showing various processes, data flow and data repositories for a "Student Information System".**
**Ans.** Refer to Q.No.-1 (a), June-2013

**(b) What is meant by "Joint Application Development"? Who are the participants in it? What are their roles? Explain briefly.**
**Ans.** Refer to Page No.45, Q.No.-6, Chapter-5

**(c) Explain any five fact finding techniques along with their advantages and disadvantages.**
**Ans.** Refer to Q.No.-2, Chapter-5
**(1) Group Discussions:** In this method, a group of staff members are invited who are expected to be well versed in their own wings of the organisation. The analysts will have a discussion with the members for their views and responses to various queries posed by them.

In this process, individuals from different sections gather together and will discuss the problem at hand. Ultimately, they come to an optimum solution. In this process, the problems of all sections are taken care of most of the cases, solutions are found which are acceptable to everyone. The main disadvantage of this process is that it is very difficult to get all the concerned people together at a time. But, the major advantage is that a mutually acceptable solution can be found.
**(2) Presentations:** It is another way of finding the facts and collecting data. Presentation is the way by which the systems analyst gathers first hand knowledge of the project. The customer makes a presentation of the existing system or about the organisation. Participants in the meeting are representatives from the IT company and key personnel of the client organisation. When a company needs to develop a software project, it may present its requirements for IOE (interest of expression) from the interested IT Company. In that case, the client presents his/her requirements. Based on the requirements, the IT companies make prototype and show the demo of the prototype. It is very difficult to obtain information in detail from a presentation. But, information available through presentation is sufficient to develop a prototype. Presentation is made by the concerned department in consultation from other departments and senior officials.

**(d) What is a CASE tool? Explain the various components of a CASE tool with the help of a diagram.**

**Ans.** Refer to Page No.216, Q.No.-4 (a), Dec-2006

**Q2. (a) What is the need for Documentation? Draw a diagram depicting various stages of the process of Documentation.**

**Ans.** Refer to Q.No.-2, Chapter-4

The following are various steps involved in the process of documentation:

**(1) Collection of source material:** The very first step of any documentation process is to acquire the required source material for preparation of document. The material is collected including specifications, formats, screen layouts and report layouts. A copy of the operational software is helpful for preparing the documentation for user.

**(2) Documentation Plan:** The documenter is responsible for preparation of a documentation plan, which specifies the details of the work to be carried out to prepare the document. It also defines and the target audience.

**(3) Review of Plan:** The plan as set out in the process above is reviewed to see that material acquired is correct and complete.

**(4) Creation of Document:** The document is prepared with the help of document generator.

**(5) Testing of Document:** The document created is tested for usability as required by the target audience.

**(6) Maintain Document:** Once the document is created and distributed, it must be kept up to date with new version of the software product. It must be ensured that the latest document is available to the user of the software.

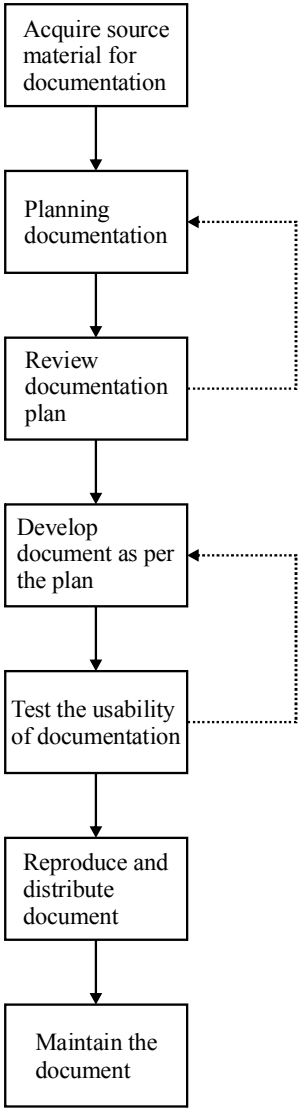



**Fig. Depicts various stages of the process of documentation.**

**(b) What is the role of a Systems Analyst? Explain any two skills that are required to be a Systems Analyst.**

**Ans.** Refer to Q.No.-1, 9, Chapter-2

**Q3. (a) How does a structure chart differ from a flow chart? Draw a structure chart for a "Student Information System". Make necessary assumptions.**

**Ans. Difference Between Flowcharts and Structure Charts:** A structure chart differs form a flowchart in the following ways:

- It is usually difficult to identify different modules of the software form its flowchart representation.
- Data interchange among different modules is not represented in a flowchart.
- Sequential ordering of tasks inherent in a flowchart is suppressed in a structure chart.
- A structure chart has no decision boxes.

Unlike flowcharts, structure charts show how different modules within a program interact and the data that is passed between them.

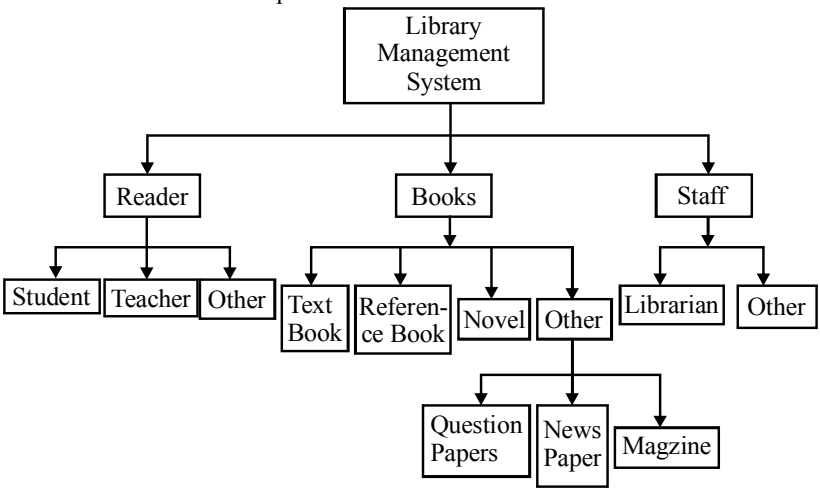



**Fig. Structure Chart**

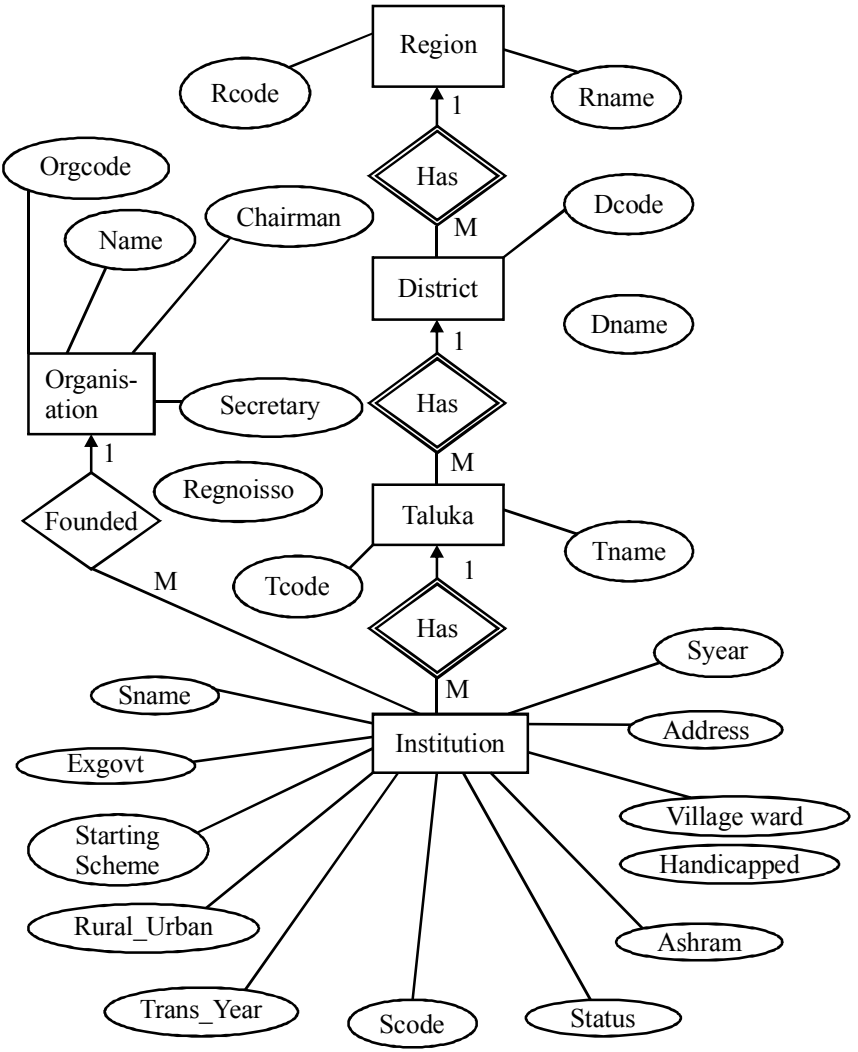
**(b) How does an E-R diagram differ from a DFD? Draw an E-R diagram for a "Student Information System". Make necessary assumptions.**

**Ans.** ERD and DFD are data presentation models that help in identifying the flow of data as well as inputs and outputs. They are important as they enable effective communication between members of different departments in an organisation. There are similarities in the two types of data presentation models although there are differences that will be talked about in this article.

DFD's are systematic representation of how data flows in an organisation, how and from where it enters the system, how it moves from one process to another and how it is stored in the organisation. On the other hand, a semantic data model of a system in a top down manner is called Entity Relationship Diagram or ERD. ERD demonstrates how a system will look like without telling how to implement it. Since it is entity based, ERD shows the relationship between entities in a system or process. On the other hand, DRD being data flow diagrams focus upon flow of data in a system and how this data is utilized in different stages of a process.

Both DFD and ERD are important for an organisation. While entities, whether they are people, places, events or objects are represented in an ERD, DFD talks about how data flows between entities. One gets to know about the entities for which data is stored in the organisation through ERD while DFD gives information about the flow of data between entities and how and where it is stored.

Different tools are made use of while preparing DFD and ERD. While it is common to use circles, ovals, rectangles and arrows to make DFD, ERD uses only rectangular boxes. Diamonds are used to represent relationships between entities in ERD and you find description of relationship whereas naming in DFD is through a single word.



**Q4. (a) What is an Expert System? Give an example of an Expert System. What are its features? Explain the risks associated with its use.**

**Ans.** Refer to Q.No.-3 (b), June-2005

Expert systems are distinct from traditional Information Systems because of two main reasons:

**Representation of Knowledge:** Information is expressed in declarative form in contrast to procedural expressions used in other types of Information Systems. Here, knowledge is stored in a structured non-procedural way.

**Perform Inexact Reasoning:** Reasoning – A process by which new information is derived from a combination or combinations of existing, or previously derived, information. In this aspect, an expert system comes closer to human mind, which is hardly seen by traditional software. The ability to perform in exact reasoning leads to easier decision-making because irrelevant alternatives are reasoned out before the execution of the software.

**(b) Prepare an outline of SRS for a "Student Information System". Make necessary assumptions.**

**Ans.** This SRS Document contains the complete software requirements for the Online Student Information Management System (OSIMS) and describes the design decisions, architectural design and the detailed design needed to implement the system. It provides the visibility in the design and provides information needed for sofware support. New reliable and fast school management software with the great customers support. It'll help you with your daily school management routines and deliver you from your paper work.

Online Student Information Management System is developing for general purpose and used to replace old paper work system and PUMS. OSIMS is to build upon the existing information system PUMS in order to efficiently provide student information to teachers and school administration. This increase in efficiency of result making, provide result to parents, give feedback to student, finally, publication and email student result. It provides a mechanism to edit the student information form which makes the system flexible.

**Overall Description**

The student management system allows authorized members to access the recors of academically registered students. It can be used in various educational institutes across the globe and simplifies working of institutes.

**(1) Product Perspective**

The proposed system shall be developed using client/server architecture and be compatible with Microsoft Windows Operating System. The front end of the system will be developed using Visual Basic 6.0 and backend will be developed using MS SQL Server 2000.

**(2) User Interfaces**

The ONSMS will have following user-friendly and menu driven interfaces:

***(a) Login:*** To allow the entry of only authorised users through valid login Id and password.

***(b) School Details:*** To maintain school details.

***(c) Programme Details:*** To maintain programme details.

***(d) Scheme Details:*** To maintain scheme details of a programme.

***(e) Paper Details:*** To maintain paper details of a scheme for a particular programme.

***(f) Faculty Details:*** To maintain the faculty details.

**(3) Hardware Interfaces**

(a) Screen resolution of at least 640 X 480 or above.

(b) Support for printer (dot matrix, deskjet, laserjet)

(c) Computer systems will be in the netwoked environment.

As It is a multi-user system.

**(4) Software Interfaces**

(a) MS-Windows Operating System

(b) Microsoft Visual Basic 6.0 for designing front-end

(c) MS SQL Server 2000 for backend

(d) Plateform: Java Language

(e) Integrated Development Environment (IDE): ECLIPSE

**(5) Communication Interfaces**

None

**(6) Memory Constraints**

At least 512 MB RAM and 500 MB space of hard disk will be required to run the software.

**Assumptions:** The login Id and password must be created by system administrator and communicated to the concerned user confidentially to avoid unauthorized access to the system.

It is assumed that a student registering for the subsequent semester has been promoted to that semester by the university as per rules and has paid desired university fee.

Registration process will be open only for specific duration.

**Q5. (a) What do you mean by Information Security? Write a short note on Threats and Risks in a system.**

**Ans. Information security**, sometimes shortened to **InfoSec**, is the practice of preventing unauthorised access, use, disclosure, disruption, modification, inspection, recording or destruction of information. It is a general term that can be used regardless of the form the data may take (e.g. electronic, physical).

he protection of information and information systems against unauthorised access or modification of information, whether in storage, processing, or transit, and against denial of service to authorised users. Information security includes those measures necessary to detect, document, and counter such threats. Information security is

composed of computer security and communications security. Also called INFOSEC. See also communications security; computer security; information security; information system.

**Threats and risks in a system:**

The security of any system should be commensurate with the risk involved. Threat and risk assessment involves identification of applicable threats to IS infrastructure, recognition of vulnerability and probable loss calculation. In this context, it is necessary to identify the source of threat.

Historically, an organization's computer systems were centrally located and the management of issues related to it were responsibility of the computer center staff and as such security related issues were also the responsibility of computer center staff whose focus were to make available the application on the centrally located computer as required. In comparison, today's computing infrastructure are far more diverse and complex to manage. Business information is dispersed.

The source of threats can be either external or internal. Historically virus has been the major potential external security threat but as organizations are diversifying their activity over multiple locations and with evolution of new technology it is difficult to perceive when an unauthorized intruder may try to hack upon organization's vital information and cause damage. Internal security threats are more common although the integrity of employee is checked before being inducted into the organization. Employee of an organization can pose serious threats to information security as they are closely associated with the system and know the vulnerabilities that can be targeted.

**(b) What is MIS? How is it useful for management control and strategic planning in an organisation?**

**Ans.** Refer to Q.No.-1, Chapter-13 and Q.No.-1 (b), Dec-2011

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2017

**Note:** *Question **no. 1** is **compulsory.** Attempt any **three** questions from the rest.*

**Q1. (a) List the underlying principles for a prototype design. Under which conditions will this design be useful? With the help of an example system, explain its approach.**
**(b) Draw a 0-level and 1-level DFD of a "Library Information System". Make necessary assumptions.**
**(c) Draw ERD for a "Library Information System". Make necessary assumptions.**
**(d) Assume that an organisation does not implement MIS. However, after a couple of years, it decides to implement it. What problems do you anticipate in implementing MIS in an organisation which did not implement it earlier? How would you handle them?**

**Q2. (a) Define a "Decision Support System (DSS)". What are its characteristics? Also, list and explain various components of a DSS.**
**(b) Define CASE. Explain its various components and tools. Give any two examples of CASE tools.**

**Q3. (a) Develop SRS outline for a "Library Information System". Make necessary assumptions.**
**(b) Explain the five types of system testing.**

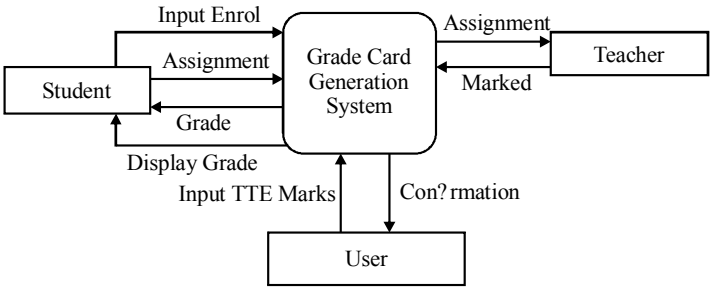
**Q4. (a) Explain the process of designing Forms and Reports.**
**(b) Explain any five different fact finding techniques.**

**Q5. Write short notes on the following:**
**(a) Different Standards for Documentation**
**(b) Audit of Transactions on Computer**

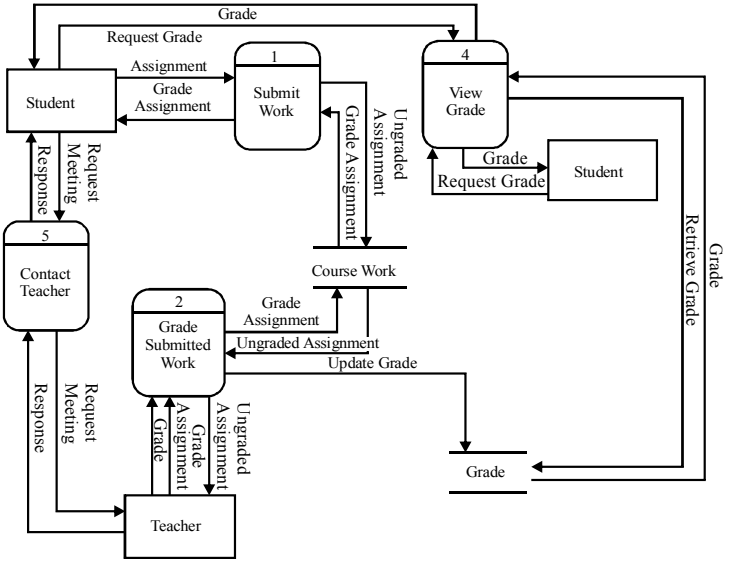
**MCS-014 : SYSTEM ANALYSIS AND DESIGN**
**June, 2018**

**Note:** *Question number **1** is **compulsory.** Attempt any **three** questions from the rest.*

**Q1. (a) Draw a 0-level and 1-level DFD of a "Grade Card Generation System for a University". Make necessary assumptions.**
**Ans.**



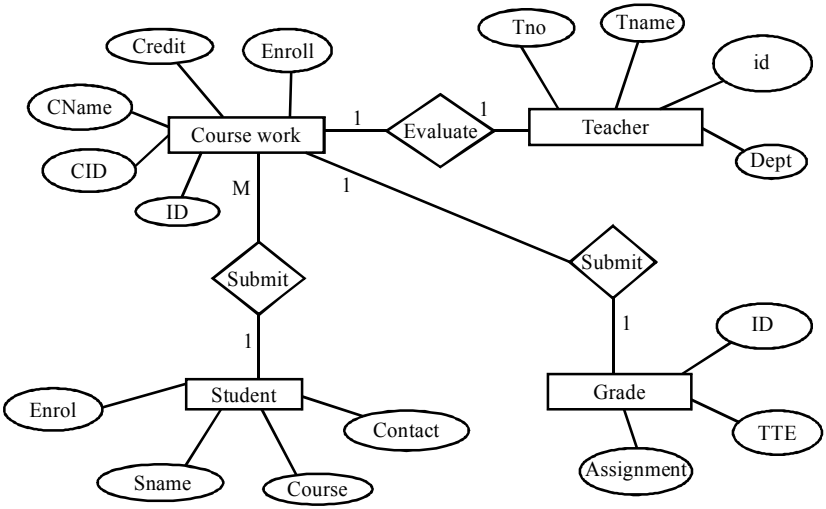

**O level**




**1 level**

**(b) Draw an ERD for a "Grade Card Generation System for a University". Make necessary assumptions.**

**Ans.**




**(c) Write SRS outline for "Grade Card Generation System for a University". Make necessary assumptions.**

**Ans. Introduction:** Grade card generation system is very useful for university. It helps to prepare students grade card using an automated system.

(a) Purpose: The purpose of this SRS document is to specify software requirements of the proposed system. This main purpose of the system is to automate the task carried out by different people in the organisation to perform the exam form submission.

(b) Project scope: This study aim is to automate the system by validating student's details. The data used by the system is stored in a database that will be the centre of all information held about students and exams:

- Manage large number of student details.
- Manage all details of students who registered for the course and send appropriate details about the course to the student account.
- View grades details of the students

(c) Definitions, Acronyms and abbreviations:

Personal details:

Details of candidate such as name, qualification, phone number, address, email address etc.

**(d) Explain various approaches for development of information systems.**

**Ans.** Refer to Chapter-1, Q.No.-4 (Pg. No.-5)

**Q2. (a) What do you mean by "Feasibility Study"? Explain different types of feasibility studies.**

**Ans.** Refer to Chapter-5, Q.No.-7 (Pg. No.-46)

**(b) With the help of an example, explain different types of fields in a table. Also, explain rules for naming tables and fields.**

**Ans.** Any conventional file system or database stores two kind of fields namely descriptive fields and primary key. Descriptive fields comprise the customer names, inventory numbers, item descriptions, and so on, which are used by the application. Keys refer to the primary and foreign key that are used to find database records and relate them to one another.

A table must have a primary key i.e. an attribute or combination of attributes that are guaranteed to be unique and not null. It is sometimes helpful to introduce a surrogate field to act as a key. This could be a table attribute, which has no business meaning, but simply added to serve as a unique identifier for each record in the table. This is sometimes referred to as plumbing.

The requirements for a primary key are very hard. It must conform to the following rules:

- They should exist.
- Be unique in the table.
- The values must not change or become null during the life of each entity instance.
- It must have a not-null value for each instance of the entity.

Surrogate keys are often required because sometimes, real business data does not fulfil the requirement of a primary key. Furthermore, the surrogate key is typically a single field (not a composite key), which simplifies the database schema, particularly when the key is used in other tables as a foreign key.

Most of modern RDBMS are tuned in for queries on integers, so it is advisable to use this datatype as a primary key. Many RDBMS provide a special serial number or sequence number of integer type, which generate a sequence of unique integers as a row is inserted into the table. Declaring a column to be of this type guarantees that a unique key is generated for each inserted row.

**Secondary key:** Also known as Alternate key. This is a field or collection of fields in the table which can be used as primary key in addition to the already existing primary key.

Foreign keys are table attributes, the values of which are the same as those of primary keys of another table. It is often desirable to label foreign key columns explicitly. For instance, by adopting a naming convention. Existence of foreign

key enforces the referential integrity constrains (discussed later in this Unit). A referential integrity constraint (references) should be declared as part of the CREATE statement in a DBMS while creating the table.

**Rules for Naming Tables and Fields**

Names for all database elements should be:

• Unique
• Meaningful
• Short

Restrictions for naming tables:

• Use no acronyms or abbreviations. Should be descriptive to convey meaning.
• Should not imply more than one subject

Restriction for naming fields:

• No acronyms
• Use abbreviations only if clear and meaningful
• Should not imply more than one subject
• Should be singular.

While designing database fields, it is required to set the properties of the fields.

**Name:** A name is used to refer the attribute in the DBMS that uniquely labels the field. The name of the attribute in the logical data model and the name of the field in the physical data model must be same. For example, student name in a student table.

**Data type:** Type of data the field is expected to store. This could be numeric, alphanumeric etc. The data type, supported by various RDBMS varies to a great extent. For example, student_name CHAR(25), indicates that the name of the student is of character data type, 25 indicates the maximum size of the data that can be stored in the field. The data type selected should ensure the following:

• it involves minimum usage of memory and represents all possible values
• supports all types of data manipulation that is expected from the business transaction.

**Size:** It indicates the size of the database fields. Many RDBMS support sizes that are variable. For example, VARCHAR data type in Oracle.

**Null or not Null:** specifies whether the field will accept null value. Not null constrains applied in DBMS ensure that null values are not entered to the respective fields. A null value is a special value distinct from 0 or blank. A null value indicates that the value is either missing or unassigned yet. We may specify that customer_name in a customer table to be not null. When a field is declared a primary key DBMS automatically ensures that the field in not null.

**Domain:** It indicates the range of values that are accepted by the fields. For example: Basic_Pay in a employee table can assume any value between the lowest basic_pay and highest basic_pay existing in the company. In such cases, the value

of the field can be restricted to the one between the highest and lowest value to avoid entry of non-existing basic_pay.

**Default value:** It refers to the value that is stored by default in the field. For example, ship_date in a invoice is most of the time same as invoice_date (current date). When a default value is assigned to a field, it reduces a lot of data entry time and reduces the chances of error.

**Referential integrity:** It refers to a set of rules that avoid data inconsistency and quality problems. Referential integrity ensures that a foreign key value cannot be entered unless it matches a primary key value in another table. RDBMS automatically enforces the referential integrity once the database designer identifies and implements primary and foreign key relationship.

• It prevents orphaned records. i.e. when a row containing a foreign key is created, the referential integrity constrains enforced vide the RDBMS ensure that the same value also exists as a primary key in the related table.

• When a row is deleted, it should be ensured that no foreign key in related tables is the same value as primary key of the deleted row.

| **Customer_id** | **Customer_ name** | **Customer_city** | **Customer_phone** |
|---|---|---|---|
| 5466 | John | New Delhi | 2345678 |
| 5678 | David | Mumbai | 2567890 |

**Table Customer**

| **Order_no** | **Order_date** | **Customer_id** | **Amount** |
|---|---|---|---|
| 123456 | 12/3/2004 | 5466 | Rs. 345 |
| 345678 | 11/3/2003 | 5678 | Rs. 567 |

**Table Order**

**Figure: Primary key and Foreign key relationship**

Customer_id is the primary key in customer table and is a foreign key in order table. This referential integrity constraints ensure that the value customer_id in order table must exist in the customer table. The primary key is shown in bold and the foreign key in intalics.

**Q3. (a) Explain different types of maintenance activities of a software system.**

**Ans.** Refer to Chapter-11, Q.No.-4 (Pg. No.-109)

**(b) Explain the differences between various types of Object-oriented CASE Tools.**

**Ans.** Refer to Chapter-10 (Pg. No.-101)

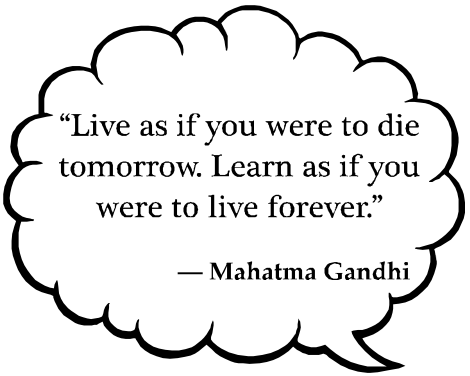
**Q4. (a) Define the term "Data Dictionary". Give an example.**

**Ans.** Refer to Chapter-4, Q.No.-11 (Pg. No.-35)

**(b) Why do analysts use data dictionaries? List the reasons.**
Ans. Refer to Chapter-4, Q.No.-11 (Pg. No.-36)

**Q5. Write a short notes on the following:**
**(a) Real Time Systems and Distributed Systems**
**Ans.** Refer to Dec.-2005, Q.No.-2(b) (Pg. No.-186)

**(b) Cost-Benefit Analysis**
**Ans.** Refer to Chapter-5, Q.No.-8 (Pg. No.-48)



## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## December, 2018

**Note:** *Question **no. 1** is **compulsory.** Attempt any **three** questions from the rest.*

**Q1. (a) Explain the advantages and disadvantages of the following software development models:**
**(i) Waterfall**
**(ii) Prototype**
**(iii) Spiral**
**(b) What are the advantages of DFDs? Draw DFDs of level-0 and level-1 for *online examination system* in which examinations are conducted using multiple choice questions and true/false questions. Make necessary assumptions.**
**(c) Explain why a system analyst should be familiar with technical skills. Also list any five technical skills which may be used by a system analyst.**
**(d) Explain data coupling with the help of an example.**
**(e) What is an expert system? Explain its two advantages and two disadvantages.**
**(f) What is MIS? Explain any two functions of MIS in a business organisation.**

**Q2. (a) Explain various roles of a system analyst in modern business.**
**(b) Explain the need of feasibility study for the development of an enterprise application system. Explain various issues involved in operational feasibility.**
**(c) What is risk analysis? Explain quantitative risk analysis.**

**Q3. (a) What is software re-engineering? How is it different from software reverse-engineering? Explain the advantages of both.**
**(b) Draw ER Diagram for the following situation:**
**An account is a relationship between Customer and Bank. A customer may have multiple accounts of different types (but only one account of one type). Each customer belongs to a branch of bank. Different types of accounts require different minimum balances. A customer may transfer money from one account to another account owned by him/her. Make necessary assumptions.**

**Q4. (a) Explain the following with an example of each:**
**(i) Use case diagram**

**(ii) Class diagram**

**(b) What is a CASE tool? How are CASE tools useful in different aspects of software engineering projects? Also briefly explain any two disadvantages of CASE tools.**

**Q5. Write short notes on the following:**

**(a) Software Maintenance**

**(b) User Interface Design**

**(c) Real Time Systems**

**(d) Software Documentation**

## MCS-014 : SYSTEM ANALYSIS AND DESIGN
## June, 2019

**Note:** *Question number **1** is **compulsory.** Attempt any **three** questions from the rest.*

**Q1. (a) Describe the significance of a Data Flow Diagram (DFD). Draw DFD's upto 2nd level for a *Study Centre management System* depicting various processes, data flow and data repositories. Follow all the conventions properly.**
**Ans.** Refer to Chapter-13, Q.No.-28 (Pg. No.-151)

**(b) Describe Open systems and Closed systems. Give two examples for each.**
**Ans.** Refer to Chapter-1, Q.No.-2(b) (Pg. No.-3)

**(c) What is the role of fact finding techniques in systems development? Mentioning their advantages and disadvantages, explain the following fact finding techniques.**
**(i) Interviews**
**Ans.** Refer to Chapter-5, Q.No.-1 and Q.No.-2 (Pg. No.-39)

**(ii) Group Discussions**
**Ans.** Refer to June-2017, Q.No.-1(c) (Pg. No.-330)

**(d) List and explain any two object oriented CASE tools along with a suitable example.**
**Ans.** Refer to June-2012, Q.No.-2(a) (Pg. No.-302)

**Q2. (a) Describe the following types of maintenance activities:**
**(i) Corrective maintenance**
**(ii) Adaptive maintenance**
**(iii) Perfective maintenance**
**(iv) Preventive maintenance**
**Ans.** Refer to Chapter-11, Q.No.-4 (Pg. No.-109)

**(b) Define the term Audit. List its objectives Also, discuss the responsibility of system auditor.**
**Ans.** This is an assessment of an information system performed by an information systems professional or IS auditor to provide recommendations

and advice to improve system performance and security. Audit should be done regularly and the result should be used to refine the system.
Is auditors are those people who make it sure that the system does what it is supposed to do. Although the audit can be carried out by the internal team of IT professionals, it is advisable that the audit is carried out by external auditors as they are neither stakeholders nor friendly with the stakeholders. Above all there is nothing like an unbiased opinion.
**Responsibility of the System Auditor:**The system auditor shall make the basis for each of his or her assessment clear. The system auditor may demand data and materials from the division being audited. The system auditor may also demand the head of an organisation to issue a report on the implementation of improvement to an audited division as suggested by him.
The system auditor shall firmly maintain professional ethics as an impartial evaluator. The system auditor shall be aware of the ethical demands on himself or herself and meet the internal and external trust by performing an accurate and sincere system audit.
**Now,** Refer to Chapter-12, Q.No-1 (Pg. No.-112)

**Q3. (a) Define an expert system. How are they different from traditional information systems? Explain various components of an expert system. Mention two examples of expert systems.**
**Ans.** Refer to Chapter-13, Q.No.-4 (Pg. No.-124)
Expert systems are distinct from traditional Information Systems because of two main reasons:
**Representation of Knowledge:** Information is expressed in declarative form in contrast to procedural expressions used in other types of Information Systems. Here, knowledge is stored in a structured non-procedural way.
**Perform Inexact Reasoning:** Reasoning – A process by which new information is derived from a combination or combinations of existing, or previously derived, information. In this aspect, an expert system comes closer to human mind, which is hardly seen by traditional software. The ability to perform in exact reasoning leads to easier decision-making because irrelevant alternatives are reasoned out before the execution of the software.
**Examples of Expert Systems**
Following are examples of Expert Systems
• **MYCIN:** It was based on backward chaining and could identify various bacteria that could cause acute infections. It could also recommend drugs based on the patient's weight.
• **DENDRAL:** Expert system used for chemical analysis to predict molecular structure.
• **PXDES:** Expert system used to predict the degree and type of lung cancer.
• **CaDet:** Expert system that could identify cancer at early stages.

**(b) Describe the criteria for form design and report designs.**
**Ans.** Refer to Dec-2005, Q.No.-4(b) (Pg. No.-191)

**Q4. (a) With reference to RDBMS, explain the use of the following files in a system:**
**(i) Master file**
**(ii) Transaction file**
**(iii) Archive file**
**(iv) Audit file**
**(v) Work file**
**Ans.** Refer to Chapter-9, Q.No.-3 (Pg. No.-89)

**(b) How does a system analyst contribute to the success of a system?**
**Ans.** Refer to Chapter-2, Q.No.-2 and Q.No.-3 (Pg. No.-7, 8)

**Q5. Write short notes on the following:**
**(a) Prototype Approach**
**Ans.** Refer to Chapter-5, Q.No.-6(ii) (Pg. No.-46)

**(b) Joint Application Development (JAD)**
**Ans.** Refer to Chapter-5, Q.No.-6(i) (Pg. No.-45)

**(c) SRS**
**Ans.** Refer to Chapter-3, Q.No.-2 (Pg. No.-16)

**(d) Decision Support Systems**
**Ans.** Refer to June-2009, Q.No.-5(d) (Pg. No.-267)

# Why Students Choose GPH Books?

- Syllabus covered as prescribed by Universities/ Boards/Institutions.
- Easily understandable language and format that help students prepare for exam in short period of time.
- Published with exam-oriented approach, hence prepared in question-answer format which provides students the instant understanding of a correct answer.
- Maximum solved previous year question papers included which help students to understand unique examination structure and equip them better for exam.
- Both semesters' question papers (June-December) are included with solutions.
- Instant updation of data as and when any change occurs.
- Use of recycled paper.
- Handy books and reasonable prices.
- For every book sold, we contribute for society/institution/NGOs/underprivileged