



Republic of Rwanda
City of Kigali



GASABO DISTRICT

DISTRICT COMPREHENSIVE ASSESSMENT, RTQF LEVEL... 2023-2024

SECTOR: ICT AND MULTIMEDIA

TRADE: SOFTWARE DEVELOPMENT

MODULE CODE: SWDBS401

MODULE NAME: BACKEND SYSTEM DESIGN

DATE OF EXAM: 13/03/2024

DURATION: 3 HOURS

SCHOOL YEAR: 2023-2024

TERM: 2

Instructions:

- | | |
|---|------------|
| 1. Attempt all questions in section A | (55 Marks) |
| 2. Attempt three questions in section B | (30 Marks) |
| 3. Attempt one question in section C | (15 Marks) |

Q1. Define a) Entity-Relationship diagram(ERD) /2marks

b) Data flow diagram(DFD) /1marks

► **Answer:**

- a) Entity-Relationship diagram(ERD)** is a visual representation of the data model that shows the entities within a system, the attributes of those entities, and the relationships between them.
- b) Data flow diagram(DFD)** is a visual representation that depicts the flow of data within a system and how it is processed. DFDs are commonly used in system analysis and design to illustrate the relationships between different components of a system, emphasizing the flow of data between them.

Q2. Explain the element of Data flow Diagram(DFD) and Discuss about the rules of drawing Data flow Diagram(DFD) /5marks

Answer:

Element of data flow diagram:

Data Flow Diagrams (DFDs) consist of various elements that help in visualizing the flow of data within a system. These elements provide a structured representation of how data moves through processes, data stores, and external entities.

1. **External Entity:** External entities represent sources or destinations of data outside the system being modeled. They interact with the system by sending or receiving data. External entities are usually depicted as squares or rectangles on the edges of the diagram.
2. **Process:** A process, also known as a function or transformation, represents a specific activity or operation performed within the system. Processes receive input data, perform some processing or transformation, and produce output data. Processes are typically represented as circles or ovals within the diagram.
3. **Data Flow:** They show the direction of data movement and the connections between different components of the system. Data flows are depicted as arrows connecting the various elements of the diagram.
4. **Data Store:** Data stores represent repositories where data is stored within the system.

Rules of drawing Data flow Diagram(DFD):

- ▶ **Identify the System Boundaries:** This helps in understanding what processes and data are included within the DFD.
- ▶ **Use Standard Symbols:** Use standardized symbols for representing different elements in the DFD. The commonly used symbols include:
 - Rounded rectangles represent processes.
 - Arrows represent data flows.
 - Squares represent external entities.
 - Double lines represent data stores.

> **Naming Conventions:** This helps in understanding the purpose and function of each element.

- ▶ **Consistency in Notation:** Use the same symbols and labeling conventions consistently to avoid confusion.
- ▶ **Hierarchy:** Organize the DFD in a hierarchical manner, with the main process at the highest level and subprocesses at lower levels. This helps in managing complexity and understanding the system at different levels of detail.
- ▶ **No Decision Logic:** Avoid including decision logic or control flow in DFDs.
- ▶ **External Entities:** Clearly define external entities and their interactions with the system. External entities represent sources or destinations of data that are outside the system boundaries.

Q3. a) Define system Interaction /1marks

b) Explain the main component of system interaction /3marks

Answer:

- a) **System Interaction** refers to the ways in which different components, subsystems, or systems engage with each other to achieve a common goal or to perform a set of related tasks. In the context of technology and software, system interaction typically involves the exchange of information, communication, and collaboration between different elements within a larger system.
- b) The main component of system interaction
 - **Web server**
 - ▶ On the hardware side, a web server is a computer that stores web server software and a website's component files (for example, HTML documents,

images, CSS stylesheets, and JavaScript files). A web server connects to the Internet and supports physical data interchange with other devices connected to the web.

- **Application Server**

- ▶ An application server is a modern form of platform middleware. It is system software that resides between the operating system (OS) on one side, the external resources (such as a database management system [DBMS], communications and Internet services) on another side and the users' applications on the third side.

.Database Server

- ▶ A database server is a type of hardware that runs database software. Database software helps users or companies store, manage, retrieve, update or change files, information logs and other forms of digital data.

.External Services and API

- ▶ What is external service API?
External APIs expose a business's internal resources to outside users or applications. For instance, third-party developers who need to access data or services that belong to a business, or who want to build apps that integrate with the business's platform, can do so using external APIs.

Q4. Distinguish back-end and front-end development, and give at least 2 examples for Each. **/4marks**

Answer:

Back-End Development	Front-End Development..
.Handles the technical aspects of a website or application, such as server management, data storage.	.Focus on the visual aspects of a website or application. Including the user interface, layout ,and fonts
.Develop the invisible structure that helps websites function properly.	.Developers work directly with the client to create and interactive elements.
.Back-End developers work on the parts of a website or application that users cannot see, ensuring that servers and databases function quickly.	.Uses technologies like HTML,CSS, and JavaScript to translate website design into functional code.
.Back-End development focuses on the technical aspects that make the front-end possible.	.Front-End Developer Ensure that visitors can interact with and navigate sites.

Example of backend development

- **Amazon** is a good example of Back End development in action. You go on Amazon and search for a new shirt. A Back End dev uses server-side language to pull all the information about shirts from a database. That information is then processed in an application and returned to the user through Front End language.

Example of frontend development

Frontend development involves creating the user interface and experience of a website or web application. This includes designing layouts, implementing interactions, and ensuring that the website or application is visually appealing, responsive, and easy to use. Here's an example of a simple frontend development project:

-HTML, CSS, JavaScript

Q5. Answer **True(T)** Or **False(F)**

i.Front-End Development Develop the Invisible structure that helps websites function properly? / **1marks**

ii. Back-End Development is included by the user interface, layout and fonts? /1marks

iii. Do you think System Development Life Cycle (SDLC) models is tools that you can use to better deliver your software project? /1marks

Answer: i. FALSE

ii. FALSE

iii. TRUE

Q6. Define a) User Documentation /1marks

b) Normalization /1marks

Answer:

- a) **User Documentation** refers to the set of documents or materials created to assist users in understanding and effectively using a product or service.
- b) **Normalization** process applied to ensure data integrity and reduce redundancy I database and to organize data which is un-organized.

Q7. Define the following term and give example for each

- a. Database /1marks
- b. Server /1marks
- c. A system /1marks
- d. Application Programming Interface (API) /1marks

Answer:

- ▶ Database often store information about people, such as customers or users. For example, social media platforms use databases to store user information, such as names, email addresses and user behavior. The data is used to recommend content to users and improve the user experience
- ▶ Server is a computer program or device that provides a service to another computer program and its user, also known as the client.

-stores, sends, and receives data. In essence, it "serves" something else and exists to provide services. A computer, software program, or even a storage device may act as a server, and it may provide one service or several.
- ▶ A system We can define a system as a group of components OR interacting elements that act according to set of rules to form a unified whole, or subsystems that integrate and function together in order to achieve a specific goal.

- ▶ Application Programming Interface (API) is a server that you can use to retrieve and send data to using code. APIs are most commonly used to retrieve data, and that will be the focus of this beginner tutorial. When we want to receive data from an API, we need to make a request. Requests are used all over the web.

Q8. Discuss about key component of system documentation. /3marks

Answer:

- ▶ **User Manuals:** Guides or manuals created to help end-users understand how to interact with the system.
- ▶ **System Architecture Document:** Describes the overall structure of the system, including its components, modules, and their interactions
- ▶ **Technical Specifications:** Details the technical aspects of the system, including hardware requirements, software dependencies, and specific technologies used in the development.
- ▶ **Source Code Documentation:** Comments and documentation within the source code that explain the purpose and functionality of different code segments.
- ▶ **Testing Documentation:** Includes test plans, test cases, and results of testing activities. It ensures that the system meets the specified requirements and functions correctly

Q9. Define JSON (JavaScript Object Notation) and Operating System. /3marks

Answer:

.JSON (JavaScript Object Notation)

is a standardized format commonly used to transfer data as text that can be sent over a network. It's used by lots of APIs and Databases, and it's easy for both humans and machines to read. JSON represents objects as name/value pairs, just like a Python dictionary.

.An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is the most important type of system software in a computer system.

Q10. Compare Hardware and Software and Give example for Each. /2marks

Answer:

Hardware and software are two fundamental components of a computer system, each playing distinct roles in its operation. Here's a comparison between hardware and software:

1. Definition:

- **Hardware:** Refers to the physical components of a computer system that you can touch and see, such as the processor, memory modules, hard drives, graphics cards, and input/output devices like keyboards and monitors.
- **Software:** Refers to the intangible programs, applications, and data that instruct the hardware on what tasks to perform. Software includes operating systems, applications, utilities, and user-generated data.

2. Nature:

- **Hardware:** Tangible and physical components that make up the computer system. It includes electronic circuits, mechanical parts, and other physical devices.
- **Software:** Intangible and non-physical instructions or programs that tell the hardware how to operate. Software is stored in digital form and executed by the hardware.

3. Examples:

- **Hardware:** Examples include CPU (Central Processing Unit), memory (RAM), storage devices (hard drives, SSDs), input devices (keyboard, mouse), output devices (monitor, printer), and networking hardware (routers, network cards).
- **Software:** Examples include operating systems (Windows, macOS, Linux), application software (Microsoft Office, Adobe Photoshop), programming languages, utilities (antivirus software, disk cleanup tools), and user-generated data (documents, photos, videos).

4. Functionality:

- **Hardware:** Performs physical tasks such as processing data, storing information, displaying output, and communicating with peripherals.
- **Software:** Provides instructions to the hardware, enabling it to perform specific tasks. Software controls the hardware and facilitates user interaction with the computer system.

5. Modifiability:

- **Hardware:** Generally less flexible and harder to modify compared to software. Hardware changes typically involve physical alterations or replacements.
- **Software:** Highly flexible and modifiable. Software can be easily updated, modified, or replaced without physical changes to the hardware.

6. Cost:

- **Hardware:** Generally more expensive than software due to the physical components involved.
- **Software:** Can vary widely in cost depending on factors like licensing, complexity, and functionality. Some software may be expensive, while others are freely available or open source.

7. **Interdependency:**

- **Hardware and Software:** Both are interdependent and require each other to function effectively. Hardware provides the platform for software to run, while software controls and utilizes the hardware resources

Q11. List and explain the types of documentation of system design. /5marks

Answer:

- ▶ **Functional Specification Document (FSD):** is a comprehensive document that outlines the detailed functionalities and features of a software system, about the factors that influence the design and implementation of the system.

Explains system functionalities, user interactions, expected inputs and outputs, error handling, and performance requirements.

- ▶ **Technical Specification Document (TSD):** Outlines system architecture, hardware and software requirements, communication protocols, algorithms, and technical specifications of individual components, Illustration of how data move through the system.

> **System requirement documentation:** out line the functional and non-functional requirements of the system from user perspective.

- ▶ **System Architecture Documentation:** describe the high level structure and organization of the system, include information about the component modules and their relationship.
- ▶ **Detailed design specification:** provide an in depth description of the internal design of the system component and modules.
- ▶ **Data base Design Documentation:** Describe the structure and organization of the data base that the system relies on.
- ▶ **Interface Design Document:** specifies how different system components or modules interact with each other.
- ▶ **User Interface Documentation:** Out line the design and layout of the user interface(include prototype to illustrate how the user will interact with the system.
- ▶ **Data Flow Diagram:** Visual representation of how data flows through the system process, datastores, and external entities.
- ▶ **System Testing Documentation:** Describe the testing strategy, test case, and deploy the system in production environment.

- **Maintenance and Support Documentation:** Provide information on a system maintenance procedures, troubleshooting guidelines and support mechanisms.

Q12. Explain the five criteria followed in software development. /3marks

Answer:

Specific: Goals should be clear and well defined, like "Create a login page for the website"

Measurable: making goals specific enough to track progress, like "Increase downloading speed by 20 Mbps (MEGABIT PER SECOND)"

Achievable: goals should be possible to reach with the resources and limitations you have

Relevant: goals should match the main aims and purpose of the project or organization

Time bound: Set a deadline for when the goal should be achieved, like "Launch the mobile app by the end of next month".

Q13. Outline the requirements needed in software project. /4marks

Answer:

The FURPS model provides a structured approach to capturing and prioritizing diverse requirements in a software project. It helps ensure that developers and stakeholders have a comprehensive understanding of what the software needs to achieve, and it guides the planning and execution of the development process.

- **Functionality(F):** Define the system's ability to perform its functions accurately and reliably over time.

Include considerations for error handling, fault tolerance, and system availability.

- **Usability(U):** Who will use the product? How will they use it? What look-and-feel do you want?

Describe how users will interact with the system.

- **Reliability (R):** Describe the procedures and mechanisms in place to restore the system after a failure.

- **Performance (P):** Specify the maximum acceptable time for the system to respond to user inputs.

- **Supportability (S):** Describe the software's ability to accommodate changes in the external environment, such as technology upgrades or regulatory requirements.

Q14. Define operating system and give the function of operating system. /4marks

Answer:

Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. The operating system is the most important type of system software in a computer system.

the function of operating system:

The operating system (OS) serves as the primary software interface between the user and the hardware of a computer system. Its functions include:

1. **Process Management:** The OS manages processes, allocating system resources like CPU time and memory to different programs running simultaneously.
2. **Memory Management:** It handles memory allocation and deallocation, ensuring efficient and secure use of system memory by processes.
3. **File System Management:** The OS provides a hierarchical structure for organizing and storing files on storage devices, as well as managing file access and permissions.
4. **Device Management:** It controls device communication by providing drivers for hardware components, enabling interaction between software applications and peripherals like printers, keyboards, and storage devices.
5. **User Interface:** The OS provides a user interface (UI) that allows users to interact with the computer system, either through a graphical user interface (GUI) or a command-line interface (CLI).
6. **Security and Access Control:** It enforces security policies, controlling access to system resources and protecting against unauthorized access, malware, and other security threats.
7. **Networking:** Many modern operating systems include networking capabilities, allowing computers to communicate with each other over local area networks (LANs) or the internet.
8. **Error Handling:** The OS detects and handles errors that occur during system operation, providing error messages and logging information for troubleshooting purposes.
9. **System Performance Monitoring:** It monitors system performance metrics such as CPU usage, memory usage, and disk I/O, providing tools for system administrators to optimize performance and diagnose issues.
10. **File System Backup and Recovery:** Some operating systems include built-in tools for backing up and restoring files and system configurations to prevent data loss in the event of hardware failure or other disasters.

Q15. Discuss about framework and library. /2marks

Answer:

The terms "framework" and "library" are often used in the context of software development, and while they both provide reusable pieces of code to help developers build software, they have distinct roles and characteristics.

- A library and framework are both reusable code written by someone else to help solve common programming problems.
- A framework is a comprehensive, pre-designed set of rules, guidelines, and tools that provides a structured way to build and organize software.
- It dictates the overall architecture of your application and provides a skeleton where you insert your code.
- Examples: Django (web framework for Python), Ruby on Rails (web framework for Ruby).

Q16. Answer True(T) Or False(F)

- a) Functional Specification Document (FSD) is a comprehensive document that outlines the detailed functionalities and features of a software system? **1marks**
- b) Interface Design Document, specifies how different system components or modules interact with each other? **/1marks**
- c) Maintenance and Support Documentation is visual representation of how data flows through the system process, datastores, and external entities? **/1marks**

Answer:

a) TRUE

b) TRUE

c) FALSE

Q17. Choose the correct answer:

- a) Process applied to ensure data integrity and reduce redundancy in database and to organize data which is un-organized. **/1marks**

I. Documentation

ii. Normalization

iii. Data Flow Diagram

Answer:

- a) **ii) Normalization**

Section B: Attempt any THREE questions/30MARKS

Q18. a. Discuss about the key aspects of the backend. / 5marks

b. Give at least five advantages of back-end. **/5marks**

a. Discuss about the key aspects of the backend.

Answer:

- **Data storage:** The back-End is responsible for storing and accessing data in a computer system.

- ▶ **Business Logic:** The Back-End handles the processing of data and the execution of Business rules.
 - ▶ **Code:** The Back-End is typically Composed of one or more programming languages, and it may interact directly with the front-end or be called from an external source.
- b. Give at least five advantages of back-end.

Answer:

The backend of a software application refers to the server-side components responsible for processing data, managing databases, and handling business logic. There are several advantages to having a robust backend in a software system:

1. **Data Management:** Backend systems handle data storage, retrieval, and manipulation. This centralized control ensures data consistency, integrity, and security. It also allows for efficient querying and indexing of data, enabling faster retrieval and analysis.
2. **Scalability:** Backend systems can be designed to scale horizontally or vertically to accommodate growing user bases and increasing data loads. This scalability ensures that the application can handle a large number of concurrent users and maintain performance under heavy traffic conditions.
3. **Security:** Backend systems provide a layer of security by implementing authentication, authorization, and encryption mechanisms to protect sensitive data and prevent unauthorized access. They also enable secure communication between clients and servers, reducing the risk of data breaches and cyberattacks.
4. **Business Logic Implementation:** Backend systems encapsulate the business logic of an application, including workflows, rules, and algorithms. This separation of concerns makes the application easier to maintain, modify, and extend over time. It also allows for consistent enforcement of business rules across different client platforms (e.g., web, mobile).
5. **Integration:** Backend systems facilitate integration with third-party services, APIs, and external data sources. This integration capability enables developers to leverage existing software components and infrastructure, reducing development time and effort. It also enables interoperability between different systems and platforms.
6. **Performance Optimization:** Backend systems can optimize performance by caching frequently accessed data, precomputing results, and fine-tuning database queries. These optimizations help reduce response times, improve user experience, and lower server load.

7. **Cross-Platform Compatibility:** Backend systems enable the development of cross-platform applications that can run on various client devices (e.g., web browsers, mobile devices). By centralizing core functionality on the server side, developers can create consistent user experiences across different platforms.
8. **Fault Tolerance and Reliability:** Backend systems can be designed with redundancy, failover mechanisms, and backup strategies to ensure high availability and fault tolerance. This resilience helps minimize downtime and data loss, providing a reliable user experience even in the face of hardware failures or network disruptions.

Q19.a) Define system documentation. /5marks

- b) Differentiate System Architecture Documentation and Functional Specification Document. /5marks**

a) Define system documentation.

Answer:

- is a collection of documents that provide detailed information about a software system, Describe the purpose, scope, and the key feature of the system.

Describe or provide the instruction on how to use the system.

b) Differentiate System Architecture Documentation and Functional Specification Document.

Answer:

- **Functional Specification Document (FSD):** is a comprehensive document that outlines the detailed functionalities and features of a software system, about the factors that influence the design and implementation of the system.

Explains system functionalities, user interactions, expected inputs and outputs, error handling, and performance requirements.

- **System Architecture Documentation:** describe the high level structure and organization of the system, include information about the component modules and their relationship.

Q20. Explain the phase in object-oriented software development and give one Advantage and one disadvantage of object-oriented analysis and design.

Answer:

- ▶ **Object-Oriented Analysis and Design (OOAD)** is a methodology for analyzing, designing, and implementing software systems based on the principles of object-oriented programming.

Advantage and one disadvantage of object-oriented analysis and design:

- ▶ Reducing the amount of code that needs to be written and improving the quality and consistency of the software.
- ▶ OOAD is a technical approach for analyzing and designing a system, application, or business by applying object-oriented programming and using visual modeling throughout the software development process to guide stakeholder communication.

Disadvantages

-OOAD Can be more expensive than other software engineering methodologies due to the upfront planning.

-OOAD Is that developers may not receive sufficient training, leading to missed deadlines, schedule slippage, and project failures.

Q21. a) Explain Tools/Techniques/Methods used for data Collection. /5marks

Answer:

1. The interview

Interview methods can be more personal and involve face-to-face discussions about a topic between the researcher and participant.

2. The observation

This might include observing individual animals or people in their natural spaces and places. Avoiding direct interactions between researchers and the subjects they are observing can ensure that results are more accurate. (Process of perceiving something or someone, often involving the use of senses or scientific instruments to acquire information.)

The observation allows the analyst to realize with his own eyes how activities are performed. This tool can complement what has been collected using other methods.

3. The questionnaire

Is a set of written or printed question used to gather information from individuals for statistical or personal purposes.

4. Documentation

The documentation is to read the various writings (books, reports, newspapers etc) speaking on the system to analyze, the activities it does and so the sequence of operations, etc. (It can be provided on paper, online, or in digital or analog media.)

b)Discuss about System Developing life cycles (SDLC) models. /5marks

Answer:

Waterfall Model

This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. The waterfall approach is the earliest approach and most widely known that was used for software development.

V-Shaped Model

The V-shaped model shows the relationships between each phase of development and the associated phase of testing. It's also referred to as the 'verification and validation model'. This is because each verification phase is associated with a validation phase.

Prototyping Model

It is an activity that can occur in software development and It used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team.

The usage

This process can be used with any software developing life cycle model. While this shall be chosen when you are developing a system has user interactions. So, if the system does not have user interactions, such as a system does some calculations shall not have prototypes.

Iterative and Incremental Model

It is developed to overcome the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between, . The basic idea behind this method is to develop a system through repeated cycles (iterative(doen something again and again, usually to improve it)) and in smaller portions at a time (incremental(used to describe something that increases in value, in regular amount)).

Agile Model

It is based on iterative and incremental development, where requirements and solutions evolve through collaboration between cross-functional teams.

Q22. Discuss about five criteria followed in software development. /10marks

Answer:

► **Specific:** Goals should be clear and well defined, like "Create a login page for the website"

Measurable: making goals specific enough to track progress, like "Increase downloading speed by 20 Mbps (MEGABIT PER SECOND)"

Achievable: goals should be possible to reach with the resources and limitations you have

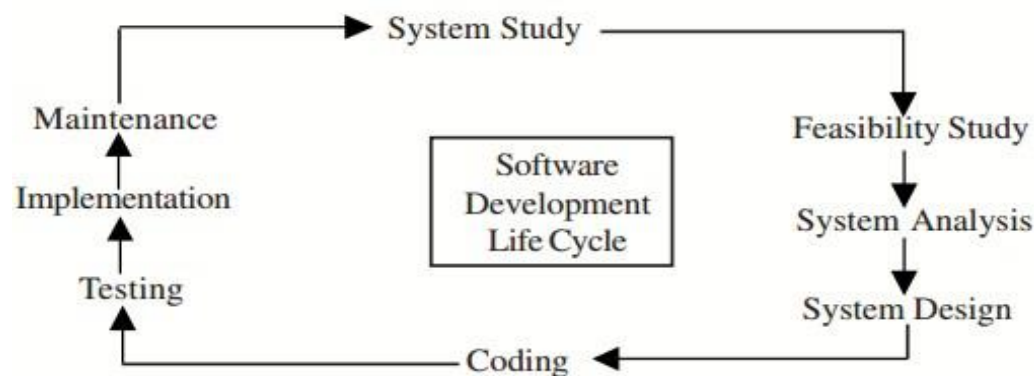
Relevant: goals should match the main aims and purpose of the project or organization

Time bound: Set a deadline for when the goal should be achieved, like "Launch the mobile app by the end of next month".

Section C: Attempt only ONE question/15MARKS

Q23. Using Diagram Illustrate and explain the system development life cycle(SDLC) Phases. /15marks

Answer:



- ▶ **Preliminary system study** is the first stage of system development life cycle. This is a brief investigation of the system under consideration and gives a clear picture of what actually the physical system is.

- ▶ **) Feasibility Study**

In case the system proposal is acceptable to the management, the next phase is to examine the feasibility of the system. The feasibility study is basically the test of the proposed system in the light of its workability, meeting user's requirements, effective use of resources and of course, the cost effectiveness(*cost and benefits*).

- ▶ **Detailed System Study**

This involves detailed study of various operations performed by a system and their relationships within and outside the system. During this process, data are collected on the available files, decision points and transactions handled by the present system. *Interviews, on-site observation* and *questionnaire* are the tools used for detailed system study.

- ▶ **System Analysis**

This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals

- ▶ **System Design**

Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing.

- ▶ **Coding**

The system design needs to be implemented to make it a workable system. This demands the coding of design into computer understandable language, i.e., programming language.

- ▶ **Testing**

Before actually implementing the new system into operation, a test run of the system is done for removing the bugs, if any.

- ▶ **Implementation**

After having the user acceptance of the new system developed, the implementation phase begins.

Implementation is the stage of a project during which theory is turned into practice.

► **Maintenance**

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environments. It has been seen that there are always some errors found in the systems that must be noted and corrected.

Q24. Out line the Implementation stage for drawing Structure system analysis and design methods(SSADM). / **15marks**

Answer:

- In the Structured Systems Analysis and Design Method (SSADM), drawing stages refer to the different phases of creating system models and diagrams using the SSADM methodology. SSADM typically involves several stages, each with its own set of activities and deliverables.

1. Stage 0: Feasibility Study:

1. Identify the need for the system and assess its feasibility.
2. Produce a Feasibility Study Report outlining the problem, objectives, scope, constraints, and potential solutions.

2. Stage 1: Investigation of the Current Environment:

1. Gather information about the current system or business processes.
2. Document existing processes, data flows, and problems.

3. Stage 2: Business System Options:

1. Analyze potential options for the new system, including alternative solutions.
2. Evaluate the feasibility and cost-effectiveness of each option.
3. Recommend the preferred solution.

4. Stage 3: Requirements Specification:

1. Elicit and document user requirements for the new system.
2. Develop Data Flow Diagrams (DFDs) to model the data flows and processes within the system.
3. Produce a Requirements Specification Document detailing functional and non-functional requirements.

5. Stage 4: Technical System Options:

1. Investigate technical options for implementing the system, including hardware and software platforms.
2. Evaluate technical feasibility, performance, and compatibility.
3. Recommend the preferred technical solution.

6. Stage 5: Logical Design:

1. Develop a logical design of the system based on the requirements specification.
2. Create Entity Relationship Diagrams (ERDs) to model the data entities and relationships.
3. Design system processes and interfaces using Data Flow Diagrams (DFDs).

7. Stage 6: Physical Design:

1. Specify the physical implementation details of the system, including hardware, software, and network components.
2. Design database schema and file structures.
3. Develop system architecture and deployment diagrams.

8. Stage 7: Implementation:

1. Develop, code, and build the system based on the physical design specifications.
2. Test and debug the system components to ensure they meet the requirements.
3. Install and deploy the system in the production environment.

► **Each stage in the SSADM methodology involves various activities, such as analysis, modeling, design, and implementation.**

SUCCESS!!!!!!!!!!!!!!

