DAY-1 (14/5/2025)

Emp ID : 110714877

NAME :Kinnera venkatamma

Task 1. WHAT IS SDLC-( Software Development Life Cycle )?

It's a structured, step-by-step process that software development teams use to create, deploy, and maintain high-quality software. SDLC helps in planning, analysis, designing, development, testing, deployment, and maintenance of software.

What it is:

SDLC provides a framework for managing the entire software development project, from initial planning to final maintenance.

It helps teams organize their work, track progress, and ensure that the software meets stakeholder needs.

It's not a single, rigid model, but rather a collection of methodologies and approaches that can be adapted to different project requirements.

Task 2 . WHY SDLC ?

It’s a structured approach to software development that helps organizations create high-quality software by providing a clear framework for planning, building, testing, and maintaining software.

It is important because it helps manage project complexities, improve quality, and ensure projects are completed on time and within budget.

Structured Approach: Ensuring that each phase is clearly defined and followed, leading to a more organized and efficient process.

Improved Quality: By incorporating rigorous testing and quality assurance throughout the lifecycle, SDLC helps ensure that the final product is of high quality and meets user expectations.

Reduced Costs: SDLC can help reduce costs by preventing costly mistakes early in the development process, ensuring that the project stays on track and within budget.

Increased Productivity: The structured approach of SDLC can lead to increased productivity by streamlining the development process and allowing teams to focus on the task at hand.

Better Communication: SDLC facilitates better communication between stakeholders by providing a common framework for understanding the project's progress and requirements.

Enhanced Risk Management: SDLC helps identify and mitigate potential risks early in the development process, ensuring that the project is less likely to face unexpected problems.

Increased Visibility: SDLC provides increased visibility into the development process, allowing all stakeholders to track progress and understand the project's status.

Customer Satisfaction: By focusing on user requirements and providing a clear roadmap for development, SDLC helps ensure that the final product meets or exceeds customer expectations, leading to higher customer satisfaction.

Task3. WHAT ARE DIFFERENT STEPS IN SDLC ?

The SDLC typically involves seven key phases. each phase contributes to a structured approach for software development, ensuring a systematic process from concept to continuous improvement.

1.Planning: This initial phase focuses on establishing the project's objectives, scope, resources, and overall strategy.

It involves defining the software's purpose, target audience, and key features.

2.Analysis: In this stage, the development team gathers and analyzes information to understand the user's needs and identify specific requirements for the software.

This includes gathering data from various sources, including users, stakeholders, and internal/external experts.

3.Design: The design phase involves creating the software's architectural blueprint, including the structure, components, and interfaces.

This stage focuses on how the software will work, including algorithms, data structures, and user interfaces.

4.Development: This phase is where the software is actually coded and built according to the design specifications.

It involves implementing the software's functionality, using various programming languages and tools.

5.Testing: This crucial phase involves evaluating the software's functionality, performance, and reliability.

It ensures that the software meets the defined requirements and performs as expected under various conditions.

6.Implementation: This phase focuses on deploying the software to its intended environment, integrating it with existing systems, and making it accessible to users.

It involves setting up the necessary infrastructure, configuring the software, and providing necessary documentation.

7.Maintenance: This ongoing phase involves maintaining the software's functionality, resolving bugs, and adapting to changing requirements.

It includes updating the software, providing user support, and addressing any issues that may arise after deployment.

Task 4 . SDLC MODELS ?

SDLC models are structured approaches used to guide software development, ensuring a systematic and efficient process from initial planning to deployment and maintenance.

They provide frameworks for project management, delegation, reporting, and quality assurance.

Some of the most common models include Waterfall, Agile, V-model, Spiral, and Iterative.

•Waterfall Model: A sequential approach where each phase (requirements, design, implementation, testing, deployment, maintenance) is completed before the next begins.

•V-Model (Verification and Validation Model): An extension of the Waterfall model that emphasizes testing at each stage of development, ensuring verification and validation at each phase.

•Agile Model: An iterative and incremental approach that emphasizes flexibility, customer collaboration, and rapid adaptation to changing requirements.

•Spiral Model: A risk-driven model that combines elements of Waterfall and Iterative approaches, allowing for iterative development with risk management at each stage.

•Iterative Model: A model where the SDLC process is divided into iterations, with each iteration producing a working prototype of the software.

•Big Bang Model: A model where development is initiated without a specific, defined process, often with minimal planning and a focus on immediate implementation.

•Incremental Model: Delivers working software in small increments, with each iteration adding new features until all requirements are met.

•Prototyping Model: Involves building and testing a prototype, allowing for feedback and revisions before full development.

•RAD (Rapid Application Development) Model: An iterative and incremental model that rioritizes quick development and iteration cycles, often used for projects with user interface requirements.

Task 5. NETWORK TYPES ?

Network types are categorized by size, scope, and purpose.

These networks connect devices to share resources, communicate, and access the internet.

Elaboration:

•LAN (Local Area Network): Connects devices within a small geographical area, such as a home or office.

•PAN (Personal Area Network): Connects devices around an individual, like a smartphone and Bluetooth headphones.

•WLAN (Wireless Local Area Network): A LAN that uses wireless technology like Wi-Fi.

•WAN (Wide Area Network): Connects devices across a wide geographic area, like the internet or a national network.

•MAN (Metropolitan Area Network): Connects networks within a city or metropolitan area.

•Other Types: Other network types include Campus Area Network (CAN), Virtual Private Network (VPN), Storage Area Network (SAN), and Enterprise Private Network (EPN).

•Network Topology: In addition to network types based on size, network topology refers to the physical arrangement of network devices and connections. Common topologies include point-to-point, bus, star, ring, mesh, tree, and hybrid.

Task 6. SERVERS ? TYPES ?

A server is a computer or system that provides resources, data, services, or programs to other computers (clients) over a network.

There are many different types of servers, each with its own specific purpose.

Some common types include web servers, mail servers, file servers, database servers, and application servers.

Here's a more detailed look at different types of servers:

1. Web Servers: These servers host websites and web applications, delivering web pages and content to users' browsers. Examples include Apache and IIS.

2. Mail Servers: These servers handle email communication, receiving, storing, and delivering emails.

3. File Servers: These servers provide a central location for storing and managing files, making them accessible to users and applications on the network.

4. Database Servers: These servers manage and store databases, allowing users and applications to access and manipulate data stored in structured formats.

5. Application Servers: These servers provide a runtime environment for executing applications, managing resources, and handling requests.

Other Important Types of Servers:

•Proxy Servers: These servers act as intermediaries between clients and other servers, potentially improving security, performance, and content filtering.

•DNS Servers: These servers translate domain names (like google.com) into IP addresses, allowing users to access websites using human-readable names.

•Print Servers: These servers manage and provide access to network printers, allowing multiple users to share printing resources.

•Game Servers: These servers host online multiplayer games, allowing players to connect and interact with each other.

•Virtual Servers: These servers are created within a physical server using virtualization technology, allowing multiple virtual machines to run on the same hardware.

•Backup Servers: These servers are responsible for creating and maintaining backups of data, ensuring that data can be restored in case of system failures.

Task 7 .WHAT IS DNS ?

The Domain Name System (DNS) acts as the internet's "phonebook", translating human-readable domain names (like example.com) into IP addresses (like 192.0.2.1) that computers use to communicate.

Here's a more detailed explanation:

•IP Addresses: Every device connected to the internet has a unique IP address, which is a numerical address that other devices use to find it.

•Domain Names: Humans find it easier to remember and use domain names (like google.com) than IP addresses.

•DNS Function: DNS servers are responsible for mapping domain names to their corresponding IP addresses, allowing browsers to load websites.

•How it works: When you type a domain name into your browser, DNS resolves it to the IP address, and your browser can then connect to the server hosting the website.

•DNS is essential for: Making it easier to navigate the internet by using familiar domain names instead of complex IP addresses.

Task8. WHAT IS TCP & UDP ? DIFFERENCE ?

TCP and UDP are both transport layer protocols, meaning they handle the transmission of data between applications.

TCP is a connection-oriented, reliable protocol, while UDP is a connectionless, best-effort protocol. TCP prioritizes reliability, while UDP prioritizes speed.

TCP (Transmission Control Protocol):

•Connection-oriented: Establishes a connection before data transfer, ensuring a reliable communication channel.

•Reliable: Guarantees delivery of data, retransmitting lost packets, and maintaining the order of packets.

•Slower: The connection establishment and error recovery mechanisms introduce overhead, making it slower than UDP.

•Common Uses: File transfer (FTP), email (SMTP), web browsing (HTTP), and other applications requiring reliable data transfer.

UDP (User Datagram Protocol):

•Connectionless: Does not establish a connection before data transfer, allowing for faster transmission.

•Best-effort: Does not guarantee delivery, error recovery, or order of packets.

•Faster: Lacks the overhead of connection establishment and error recovery, resulting in faster transmission.

•Common Uses: Streaming media (video, audio), online gaming, VoIP, and other applications where some data loss is tolerable.

Task9. WHAT IS MAC ADDRESS ? DIFFERENCE BETWEEN MAC & IP ADDRESS ?

A MAC address (Media Access Control address) is a unique identifier assigned to a network interface controller (NIC) for devices like computers, laptops, and routers. It's like a physical address for the device on a local network.

An IP address (Internet Protocol address), on the other hand, is a logical address used for communication across networks, including the internet.

MAC Address:

•Purpose: Identifies a device on a local network (LAN).

•Function: Ensures that data packets are delivered to the correct device within the LAN.

•Location: Hardcoded on the NIC of a device.

•Format: A 12-character hexadecimal address, typically in the format XX:XX:XX:XX:XX:XX.

•Uniqueness: Unique within a local network.

•Consideration: Remains constant regardless of the network.

IP Address:

•Purpose: Identifies a device on a network (including the internet) for communication across networks.

•Function: Facilitates routing of data across networks, ensuring data reaches the correct destination.

•Location: Assigned by a router or ISP, not hardcoded.

•Format: A numerical address in the format of four numbers separated by periods (e.g., 192.168.1.1).

•Uniqueness: Can be dynamically assigned, and may not be unique on a local network.

•Consideration: Can be changed or dynamically assigned using DHCP.

Key Differences:

•Scope:

MAC addresses are local, while IP addresses are global.

Task10. WHAT IS OSI MODEL ?

The OSI (Open Systems Interconnection) model is a conceptual framework that organizes network communication into seven distinct layers, each with specific functions and responsibilities.

It provides a standardized way to understand and analyze how different computer systems communicate across a network. While the modern internet primarily uses the TCP/IP model, the OSI model is still valuable for visualizing network operations and troubleshooting issues.

Task11. WHAT IS IPv4 & ITS CLASSES ?

An IPv4 address is a 32-bit numerical address used to identify devices on a network. These addresses are typically represented in a dotted decimal format, like 192.168.1.1, where each number represents a group of 8 bits (an octet). IPv4 addresses are divided into five classes (A, B, C, D, and E) based on their first octet, with each class having a different default subnet mask and intended usage.

Classes of IPv4 Addresses:

•Class A: 0-127.0.0.0 to 127.255.255.255 (Large networks, 8-bit network ID, 24-bit host ID)

•Class B: 128-191.0.0.0 to 128.255.255.255 (Medium-sized networks, 16-bit network ID, 16-bit host ID)

•Class C: 192-223.0.0.0 to 223.255.255.255 (Small networks, 24-bit network ID, 8-bit host ID)

•Class D:224-239.0.0.0 to 239.255.255.255 (Multicast addressing, used for sending data to multiple hosts simultaneously)

•Class E:240-255.0.0.0 to 255.255.255.255 (Experimental, reserved for research)

Task 12.WHAT IS VPN & ITS ADVANTAGES ?

A Virtual Private Network (VPN) is a secure, encrypted connection that creates a private tunnel over the public internet. It encrypts your data, masks your IP address, and hides your location, enhancing privacy and security while browsing online. VPNs offer benefits like improved privacy, access to geo-blocked content, and protection on public Wi-Fi networks.

Benefits of using a VPN:

•Enhanced Privacy: VPNs encrypt your internet traffic, making it difficult for third parties, including your Internet Service Provider (ISP), to monitor your online activity.

•IP Address Masking: VPNs mask your IP address, making it harder for websites and trackers to identify your location or track your browsing habits.

•Access to Geo-Blocked Content: By connecting to a VPN server in a different location, you can bypass geo-restrictions and access content that may be blocked in your region.

•Security on Public Wi-Fi: VPNs encrypt your data on public Wi-Fi networks, protecting it from potential eavesdroppers.

•Work Remotely: VPNs allow employees to securely access corporate networks and resources from remote locations, according to Check Point Software

Task13.TYPES OF VPN ?

Types of VPN include:

•Site-to-site VPN

•Remote access VPN

•Cloud VPN

•SSL VPN

•Double VPN

VPNs, or Virtual Private Networks, can be broadly categorized into three main types: remote access, site-to-site, and cloud VPNs. Each type serves a different purpose and is tailored for specific needs.

Here's a more detailed breakdown:

1. Remote Access VPN:

•Purpose: Enables individual users, such as remote employees, to securely access a private network from outside the office.

•How it works: A VPN client on the user's device establishes a secure, encrypted tunnel to the private network, allowing access to resources like files, applications, and the corporate intranet.

•Example: A consultant working from home connecting to their company's network.

2. Site-to-Site VPN:

•Purpose : Connects multiple geographically dispersed networks, creating a secure, encrypted link between them.

•How it works: This type is often used by companies with branches in different locations to enable communication and resource sharing across their networks.

•Example: A company with offices in different cities connecting their networks to share data and applications.

3. Cloud VPN:

•Purpose: Offers a VPN service hosted in the cloud, combining the benefits of both remote access and site-to-site VPNs.

•How it works: Users connect to the cloud VPN service, which then provides secure access to the private network, often without the need for complex infrastructure setup.

•Example: A company using a cloud-based VPN to enable secure access to their applications for employees and customers.

Additional VPN Types and Protocols:

•SSL VPN: Uses the SSL/TLS protocol to create secure tunnels, often offering a user-friendly experience with web browser access.

•OpenVPN: A widely used open-source VPN protocol known for its strong security and flexibility.

•IPsec: An Internet Protocol Security protocol that provides encryption and authentication at the IP layer.

•L2TP/IPsec: Combines the Layer 2 Tunneling Protocol (L2TP) with IPsec for enhanced security.

•Wire Guard: A newer, more efficient, and faster VPN protocol.

•PPTP: Point-to-Point Tunneling Protocol, a simpler protocol but less secure compared to others.

Task14. WHAT IS NODE & LINK ?

In networking, a node refers to a point or device within a network that can communicate with other nodes. A link is the connection or medium that allows nodes to communicate with each other, enabling the transfer of information.

Nodes:

•Nodes are devices or points within a network that can send and receive data.

•Examples of nodes include computers, routers, switches, and other network devices.

•Nodes are the fundamental building blocks of a network, acting as connection points.

Links:

•Links are the physical or logical connections between nodes in a network.

•Examples of links include wires, cables, Wi-Fi, and other communication channels.

•Links enable the transmission of data between nodes, forming the network's communication pathways.

In essence: Nodes are the devices or points that exist within the network, while links are the ways these devices connect and communicate with each other.

Task15.WHAT IS NETWORK TOPOLOGY ?

In networking, topology refers to the arrangement of devices and connections within a network, determining how data flows between them. It encompasses both the physical layout of cables and devices and the logical way data moves through the network. Understanding network topology is crucial for network design, troubleshooting, and optimization.

Key aspects of network topology:

•Physical Topology:The actual arrangement of devices (computers, servers, routers, switches) and cables in a network.

•Logical Topology: The way data flows between devices, regardless of their physical location, according to TechTarget.

•Common types:

Bus: All devices connect to a single cable (bus).

Star: Devices connect to a central hub or switch.

Ring: Devices connect in a circular path.

Mesh: Devices are interconnected with multiple paths.

Tree: Devices are arranged in a hierarchical structure.

Hybrid: A combination of two or more topologies.

In networking, topology refers to the layout or structure of a network, defining how its components are interconnected and how data flows between them. It encompasses both the physical arrangement of devices and the logical pathways for data transmission.

Key aspects of network topology:

•Physical Topology: Describes the physical arrangement of devices and cables, including the wiring and layout of network components.

•Logical Topology: Defines how data travels through the network, regardless of the physical arrangement of devices.

•Types of Topologies: Common topologies include bus, star, ring, mesh, and tree, each with its own advantages and disadvantages.

•Impact on Performance: Topology choices significantly influence network performance, scalability, and reliability.

•Optimization: Understanding topology is crucial for optimizing network performance, reducing congestion, and ensuring smooth data transmission.

Task16. DIFFERENT TYPES OF NETWORK TOPOLOGY ?

Network topology refers to the arrangement of nodes (devices) and links (connections) in a network. Common types include point-to-point, bus, star, ring, mesh, tree, and hybrid. Each topology offers different advantages and disadvantages regarding performance, scalability, and cost.

Types of Network Topologies:

•Point-to-Point: A direct connection between two nodes, offering simplicity and dedicated bandwidth, but limited scalability.

•Bus: Nodes connect to a single cable (the "bus"), which can be simpler and cheaper to implement, but performance can degrade with more nodes.

•Star: All nodes connect to a central hub or switch, making it easy to troubleshoot and add new nodes, but the hub's failure can disrupt the entire network.

•Ring: Nodes connect in a circular fashion, where data travels in one direction, potentially offering higher speeds but can be vulnerable to a single point of failure.

•Mesh: Nodes are interconnected with multiple redundant paths, providing high reliability and fault tolerance, but can be more complex and expensive to implement.

•Tree: A hierarchical structure with a central node branching out to others, combining the reliability of the bus topology with the scalability of the star topology.

•Hybrid: Combines two or more of the above topologies to leverage their individual strengths.

Task17. WHAT IS EXTENDED BUS TOPOLOGY & TREE TOPOLOGY ?

An extended bus topology combines multiple bus networks into a larger network, while a tree topology organizes devices in a hierarchical structure, with a central root node and branching connections, resembling a tree.

Extended Bus Topology:

•Description: Instead of a single bus, multiple bus networks are connected using a central hub or switch, effectively creating a larger network.

•Advantages: Offers scalability, allowing for easy expansion by adding new bus networks.

•Disadvantages: If the central hub or switch fails, the entire network can be affected.

Tree Topology:

•Description: Devices are arranged in a hierarchical structure, with a central root node and branching connections, similar to a tree.

•Advantages: Offers a scalable and organized network design, making it easier to manage and troubleshoot. It can also support a large number of nodes.

•Disadvantages: More complex to configure and maintain compared to other topologies. It can also have slower performance due to data passing through multiple nodes.

Key Differences:

•Structure: Extended bus topology is a network of multiple bus networks, while tree topology has a hierarchical structure with a central root node.

•Scalability: Both are scalable, but tree topology is designed for larger networks and easier expansion.

•Complexity: Tree topology is more complex to manage and troubleshoot due to its hierarchical structure.

Task18. WHAT IS THE USE OF A ROUTER & HOW ITS DIFFERENT FROM GATEWAY ?

A router and a gateway are both network devices that facilitate communication between different parts of a network, but they have distinct roles. A router primarily handles data within similar networks, choosing the best path for data packets to travel. A gateway, on the other hand, connects dissimilar networks, often acting as a bridge between different protocols or formats.

Here's a more detailed explanation:

Router:

•Purpose: Routers are designed to forward data packets between different networks or network segments. They determine the optimal path for data to travel based on their routing tables.

•Function: Routers typically operate at the network layer (Layer 3 of the OSI model). They read the destination IP address of a data packet and then use their routing table to determine the next hop or destination.

•Example: A home Wi-Fi router connected to your modem can be seen as a router, as it directs data between your devices and the internet.

Gateway:

•Purpose: Gateways connect different networks that may use different protocols or have different addressing schemes. They act as a translator or converter between these networks.

•Function: Gateways can operate at various levels of the OSI model, including layers 4 and above. They may convert data formats, protocols, or addressing schemes to ensure compatibility between networks.

•Example: A gateway might be used to connect a company's internal network (with one set of protocols) to the internet (using another set of protocols).

Key Differences:

•Network Type: Routers work within similar networks, while gateways connect dissimilar networks.

•Protocol Conversion: Gateways often involve protocol conversion, while routers primarily focus on path selection within a network.

•OSI Layer: Routers typically operate at Layer 3, while gateways may operate at higher layers.

In essence: A router is like a traffic cop within a network, directing traffic between different roads (network segments), while a gateway is like a customs officer, allowing traffic to pass between different countries (networks) with different rules.

Task19.Explain SMTP Protocol with diagram.

Simple Mail Transfer Protocol (SMTP) is an application layer protocol used for exchanging email messages between servers. It is essential in the email communication process and operates at the application layer of the TCP/IP stack.

To send an email, the client opens a TCP connection to the SMTP server. The server, which is always listening on port 25, initiates the connection as soon as it detects a client. Once the TCP connection is established, the client sends the email across the connection

Types of SMTP Protocol

The SMTP model supports two types of email delivery methods: end-to-end and store-and-forward.

•End-to-end delivery is used between organizations. In this method, the email is sent directly from the sender's SMTP client to the recipient's SMTP server without passing through intermediate servers.

•Store-and-forward is used within organizations that have TCP/IP and SMTP-based networks. In this method, the email may pass through several intermediate servers (Message Transfer Agents, or MTAs) before reaching the recipient.

With end-to-end delivery, the SMTP client waits until the email is successfully copied to the recipient's SMTP server before sending it. This is different from the store-and-forward method, where the email might stop at multiple intermediate servers before reaching its destination. In store-and-forward systems, the sender is notified as soon as the email reaches the first server, not the final destination.

Task20.Differentiate between OSI and TCP/IP

Differences Between OSI Model and TCP/IP Model

The OSI (Open Systems Interconnection) Model and the TCP/IP (Transmission Control Protocol/Internet Protocol) Model are two frameworks used to understand how data moves through networks. While they both help in organizing network communication, they have distinct structures and purposes. Understanding these differences is essential for anyone learning about or working with computer networks.

| **Parameters** | **OSI Model** | **TCP/IP Model** |
| --- | --- | --- |
| **Full Form** | OSI stands for Open Systems Interconnection | TCP/IP stands for Transmission Control Protocol/Internet Protocol |
| **Layers** | It has 7 layers | It has 4 layers |
| **Usage** | It is low in usage | It is mostly used |
| **Approach** | It is vertically approached | It is horizontally approached |
| **Delivery** | Delivery of the package is guaranteed in OSI Model | Delivery of the package is not guaranteed in TCP/IP Model |
| **Replacement** | Replacement of tools and changes can easily be done in this model | Replacing the tools is not easy as it is in OSI Model |
| **Reliability** | It is less reliable than TCP/IP Model | It is more reliable than OSI Model |
| **Protocol Example** | Not tied to specific protocols, but examples include HTTP (Application), SSL/TLS (Presentation), TCP (Transport), IP (Network), Ethernet (Data Link) | HTTP, FTP, TCP, UDP, IP, Ethernet |
| **Error Handling** | Built into Data Link and Transport layers | Built into protocols like TCP |
| **Connection Orientation** | Both connection-oriented (TCP) and connectionless (UDP) protocols are covered at the Transport layer | TCP (connection-oriented), UDP (connectionless) |

Conclusion

In conclusion, while both the OSI Model and TCP/IP Model are essential for understanding network communication, they differ in their structure and practical application. The OSI Model provides a theoretical framework with seven layers, emphasizing clear separation of functions, while the TCP/IP Model, with its four layers, reflects the protocols used on the internet today. Each model offers unique insights into how data is transmitted across networks, catering to different aspects of network design, management, and troubleshooting.

Task21.HTTP and HTTPs?

HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) are both used for transmitting data over the web, but HTTPS is the secure version. HTTPS encrypts the data, making it more secure, while HTTP transmits data in plain text, making it vulnerable to eavesdropping and tampering.

HTTP:

•Unsecure: Data is sent in plain text, meaning anyone with access to the network can read it.

•Default port: Typically uses port 80.

•Stateless: Each request is independent, requiring no authentication or identification.

HTTPS:

•Secure: Uses encryption (TLS/SSL) to protect data in transit, ensuring confidentiality.

•Default port: Typically uses port 443.

•Requires authentication: Requires a handshake before delivering data.

In essence: HTTPS provides a layer of security that HTTP lacks. HTTPS encrypts the data exchanged between a web browser and a web server, making it more difficult for malicious actors to intercept or tamper with the data. This is crucial for transmitting sensitive information like login credentials, credit card details, or personal data.

Task22.What is Low Level Design and High level Design.. Explain

High-Level Design (HLD) provides a broad overview of a system's architecture and functionality, while Low-Level Design (LLD) focuses on the detailed implementation of individual components and algorithms. HLD establishes the system's overall structure and components, whereas LLD specifies how those components will work together at a granular level.

High-Level Design (HLD):

•Focus: Overall system architecture, functionality, and top-level components.

•Scope: A broad, high-level view of the system, including modules, interactions, and data flow.

•Purpose: To define the system's structure, components, and their relationships, providing a roadmap for development.

•Examples: Identifying main components (e.g., web server, database), their interactions, and data flow between them.

•Analogies: HLD is like the blueprint of a building, outlining the overall structure and layout before construction.

Low-Level Design (LLD):

•Focus: Detailed implementation of individual components, algorithms, data structures, and code-level details.

•Scope: A granular view of how each component works, including classes, interfaces, methods, and their interactions.

•Purpose: To define the implementation details of each component, ensuring efficiency, maintainability, and scalability.

•Examples: Designing specific classes, their methods, interfaces, and algorithms, including data structures.

•Analogies: LLD is like the detailed engineering drawings of a building, showing the specific components and how they fit together.

Key Differences:

Feature High-Level Design (HLD) Low-Level Design (LLD)

Scope System-wide, architectural Component-level, implementation

Abstraction High, abstract Low, detailed

Focus Overall structure, functionality Specific components, algorithms

Timing Early stages of development Later stages of development

Tools UML, architecture diagrams UML, coding details, algorithms

Example Identifying modules, interfaces Designing classes, methods

Task23.What is SRS (Software Requirement Specification).

A Software Requirements Specification (SRS) is a detailed document that outlines the requirements for a software system. It's a crucial document in software development that defines what the software should do and how it should behave. It's a binding contract between the developers, testers, and stakeholders, ensuring everyone is on the same page regarding the software's intended functionality and performance.

Purpose of an SRS:

•Clarifies Requirements: The SRS ensures that all stakeholders have a clear understanding of the software's intended functionality, features, and non-functional requirements like performance and security.

•Guides Development: It serves as a roadmap for the development team, guiding them on how to build the software to meet the specified requirements.

•Enables Testing: Testers use the SRS to create test cases and verify that the software functions as intended.

•Minimizes Misunderstandings: By clearly defining requirements, the SRS reduces the risk of misunderstandings and rework during development.

•Serves as a Reference: The SRS can be used as a reference for future maintenance and modifications of the software.

Key Components of an SRS:

•Introduction: Provides context, scope, and purpose of the document.

•Functional Requirements: Describes the specific functions and features the software will provide.

•Non-Functional Requirements: Specifies aspects like performance, usability, security, and reliability.

•Use Cases: Illustrates how users will interact with the software through various scenarios.

•Constraints: Defines any limitations or restrictions on the development process.

•Assumptions: Lists any assumptions made during the development process.

•Interface Specifications: Details how the software will interact with other systems or users.

Benefits of using an SRS:

•Improved communication: Ensures everyone involved in the project has a clear understanding of the requirements.

•Reduced development costs: By minimizing misunderstandings and rework, the SRS can help save time and money.

•Higher quality software: By providing a clear guide for development and testing, the SRS helps ensure that the software meets the required quality standards.

•Easier maintenance: The SRS can serve as a valuable reference when modifying or maintaining the software.

In essence, the SRS is a critical document that helps ensure the success of a software project by providing a clear and comprehensive specification of the software's requirements.

SDLC MCQ

1.

A feasibility study using the SDLC model is conducted to

determine whether or not the project is technically possible

determine whether the proposal is financially viable

Both a and b

None of the above

ANS :: C- Both a & b

2.

A well-documented life cycle model aids in the detection of what during the development phase?

Inconsistencies

Redundancies

Omission

All of the above

ANS : D- All of the above

3.

How many lines of code does the Build & Fix Model suit for programming exercises?

100-200

300-400

600-700

Above 800+

ANS :A-100-200

4.

In which life cycle does regression testing play a significant role?

Waterfall model

V model

Iterative model

All of the above

ANS : D- All of the above

5. What determines if the project should go forward?

feasibility assessment

opportunity identification

system evaluation

program specification

ANS :A- feasibility assessment

6.

What is the most significant disadvantage of employing the RAD Model?

Developers/designers that are highly specialized and skilled are required.

Component reusability is improved.

Encourages client/customer input.

Increases component reusability.

ANS :A- Developers/designers that are highly specialized and skilled are required.

7.

Which of the following developmental models is incremental?

Prototyping, V model, Agile

Prototyping, RAD, Agile, RUP

Prototyping, V model, RAD, Agile, RUP

All of the above

ANS :D - All of the above

8.

Which of the following is an Agile development characteristic?

Shared code ownership

Test-Driven Development

Implement the simplest solution to meet today's problem

Continual feedback from customer

All of the above

ANS :;D- All of the above

9.

Which of the following steps in the SDLC framework are valid?

Requirement Gathering

Software Design

System Analysis

All of the above

ANS :D- All of the above

10.Who is in charge of system development, staffing, budgeting, and reporting, as well as ensuring that deadlines are met?

Project managers

Network engineers

Graphic designers

Systems analysts

ANS :A- Project managers