Module 1 –Overview of IT Industry

1) What is a Program?

- A program is a set of instructions written in a programming language that tells a computer how to perform a specific task or solve a problem. It can range from simple tasks, like adding two numbers, to complex operations, like running an entire website or managing a database.

2) Write a simple "Hello World" program in two different programming languages of your choice. Compare the structure and syntax.

- **C Language: -**

```
#include < stdio .h >
     int main () {
          printf (“Hello, World! \n”);
           return 0;
     }
```

- **Java Language: -**

```
public class Main {
    public static void main (String [] args) {
          System.out.println(“Hello, World!”);
    }
}
```

<u>Comparison</u>

- **Syntax:**

C:

- Simpler, procedural syntax.
- The function main () is used for execution.
- printf () is the function for printing text.
- No need for object-oriented concepts like classes or objects.

Java:

- Object-oriented syntax.
- Every Java program must be inside a class.
- Uses System.out.println() for printing text, which is part of the Java standard library.
- Java uses public static void main (String [] args) for the entry point
- method, which is more verbose compared to C.

Structure:

C:

- Starts with including necessary libraries (#include <stdio.h>).
- The program has a main () function which is the entry point.
- The printf () function is used to print the output.
- The program terminates with a return 0; to indicate successful execution.

Java:

- Requires a class definition (public class HelloWorld), as Java is object-oriented
- The main () method is defined inside the class, and it is the entry point for Java applications.
- The program uses System.out.println() to print the output.

3) Explain in your own words what a program is and how it functions.

- A program is a set of instructions or commands that a computer follows to perform a specific task.
- A program is written in a programming language (like Python, C, or Java). The instructions in the program are translated into machine code that the computer understands. Once the program is executed, the computer follows these instructions to produce the desired outcome, like showing a result or performing an action.

What is Programming?

4) What are the key steps involved in the programming process?

- **Key Steps in Programming:**

1. **Problem Definition**: Understand the problem.
2. **Planning**: Design the solution.
3. **Writing Code**: Implement the solution.
4. **Testing**: Fix bugs and ensure it works.
5. **Deployment**: Make the program available.
6. **Maintenance**: Update and fix issues as needed.

Types of Programming Languages?

5) What are the main differences between high-level and low-level programming languages?

- **Abstraction:**
  High-Level: Easier for humans to understand.
  Low-Level: Closer to machine code, harder to understand.
- **Ease of Use:**
  High-Level: Simple to write and use.
  Low-Level: Complex and detailed.
- **Portability:**
  High-Level: Works on many devices.
  Low-Level: Works on specific devices.
- **Control:**
  High-Level: Less control over hardware.
  Low-Level: More control over hardware.
- **Speed:**
  High-Level: Slower execution.
  Low-Level: Faster execution.

Software Applications and Its Types:

6) Identify and classify 5 applications you use daily as either system software or application software.

- Operating System (Windows / macOS) – System software:

Manages hardware resources and provides a platform for running application software.

- Web Browser (e.g., Google Chrome) – Application Software:
  Allows browsing and accessing websites.

- Microsoft Office Suite (e.g., Word, Excel) – Application Software:
  Used for productivity tasks like word processing and data analysis.

- Media Player (e.g., VLC) – Application Software:
  Plays audio and video files.

- File Explorer (e.g., Windows File Explorer)-Software System:
  Manages and organizes files and folders on the computer.

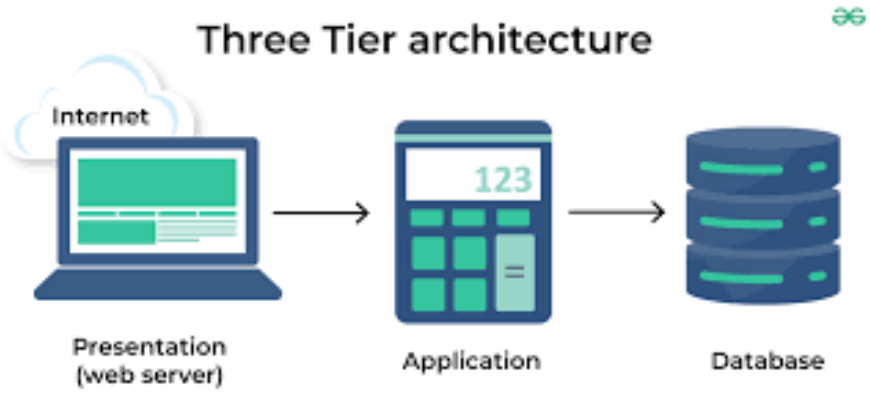
7) What is the difference between system software and application software?

- System Software:
  - **Purpose**: Manages hardware and provides the environment for running application software.
  - **Examples**: Operating systems (Windows, macOS), device drivers, antivirus software.
  - **Function**: Controls system resources like memory, processing, and hardware components.

- Application Software:
  - **Purpose**: Performs specific tasks or solves particular problems for the user.
  - **Examples**: Word processors (Microsoft Word), web browsers (Chrome), media players (VLC).
  - **Function**: Allows users to perform tasks like document creation, web browsing, or multimedia playback.

8) Design a basic three-tier software architecture diagram for a web application.

➢



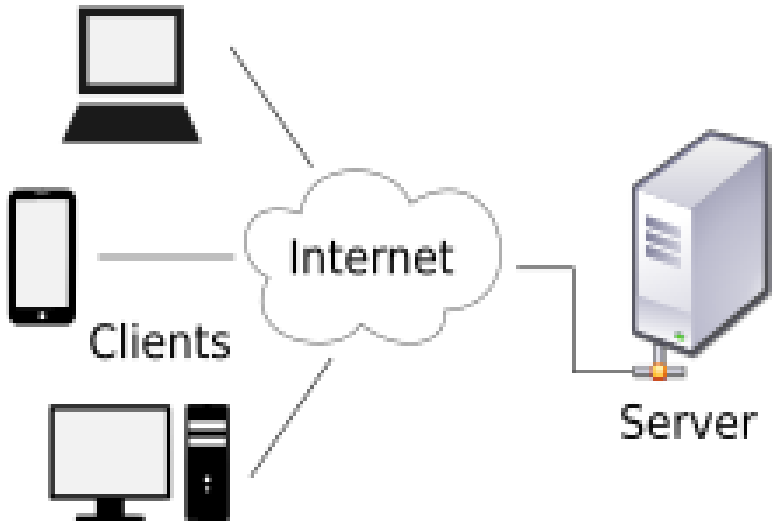
9) What is the significance of modularity in software architecture?

➢ Modularity in software architecture means breaking a system into smaller, independent parts (modules). It makes the system easier to maintain, update, scale, and understand. Each module can be developed and tested separately, improving efficiency and flexibility.

World Wide Web & How Internet Works:

10) Research and create a diagram of how data is transmitted from a client to a server over the internet.

➢



11) Describe the roles of the client and server in web communication.

- In web communication, a “client” is the device (like a web browser) that initiates a request for information or services from a “server,” which is the computer that stores and provides the requested data.

Network Layers on Client and Server:

12) Explain the function of the TCP/IP model and its layers.

- The TCP/IP model is a group of protocols that control how data is sent over the internet.

Application Layer: Manages user applications (e.g., web browsers, email).

Transport Layer: Ensures reliable data transfer (using TCP or UDP).

Internet Layer: Routes data packets to the correct destination (using IP).

Network Access Layer: Handles the physical transmission of data (e.g., Ethernet, Wi-Fi).

Client and Servers:

13)Explain Client Server Communication.

- Client-server communication is when a client (like a computer or phone) asks a server for information, and the server sends the requested data back.
- How it's work:
  1. The client sends a request to the server.
  2. The server processes the request.
  3. The server performs the required actions, such as retrieving data or running a program.
  4. The server sends a response back to the client.

Types of Internet Connections:

14)Research different types of internet connections (e.g., broadband, fiber, satellite) and list their pros and cons.

1. Broadband (Cable) Internet

- **Pros:** Fast speeds, reliable, good for multiple devices.
- **Cons:** Can be expensive, speed may drop during busy hours, not available everywhere.

2. Fiber-Optic Internet

- **Pros:** Very fast, reliable, low delays, future-proof.
- **Cons:** Limited availability, expensive installation.

3. DSL (Digital Subscriber Line)

- **Pros:** Affordable, available in many areas, doesn't tie up the phone line.
- **Cons:** Slower speeds, less reliable, speed drops with distance from the provider.

4. 5G Home Internet

- **Pros:** Fast speeds, low delays, becoming widely available.
- **Cons:** Coverage can be limited, signal can be blocked indoors, expensive.

15) How does broadband differ from fiber-optic internet?

➢ Here's the comparison in a simple table:

Feature	Broadband	Fiber-Optic Internet
Definition	General term for high-speed internet	Specific type of broadband using light
Speed	Varies (DSL, cable, satellite)	Very fast (up to 10 Gbps or more)
Reliability	Varies (less stable with DSL/cable)	Very reliable and stable
Cost	Generally more affordable	Typically more expensive
Availability	Widely available (DSL, cable, etc.)	Limited availability in rural areas
Best For	General use (browsing, streaming)	High-speed needs (gaming, heavy data)

Protocol:

16) What are the differences between HTTP and HTTPS protocols?

Feature	HTTP	HTTPS
Definition	Hypertext Transfer Protocol	Hypertext Transfer Protocol Secure
Security	Unsecured	Secured
Encryption	No encryption	Encrypted using SSL/TLS

Port	Uses port 80	Uses port 44

Application Security:

17) Identify and explain three common application security vulnerabilities. Suggest possible solutions.

1. SQL Injection
What: Attackers manipulate SQL queries.
Solution: Use **prepared statements** and **input validation**.
2. XSS (Cross-Site Scripting)
What: Malicious scripts injected into web pages.
Solution: **Sanitize input** and use **CSP**.

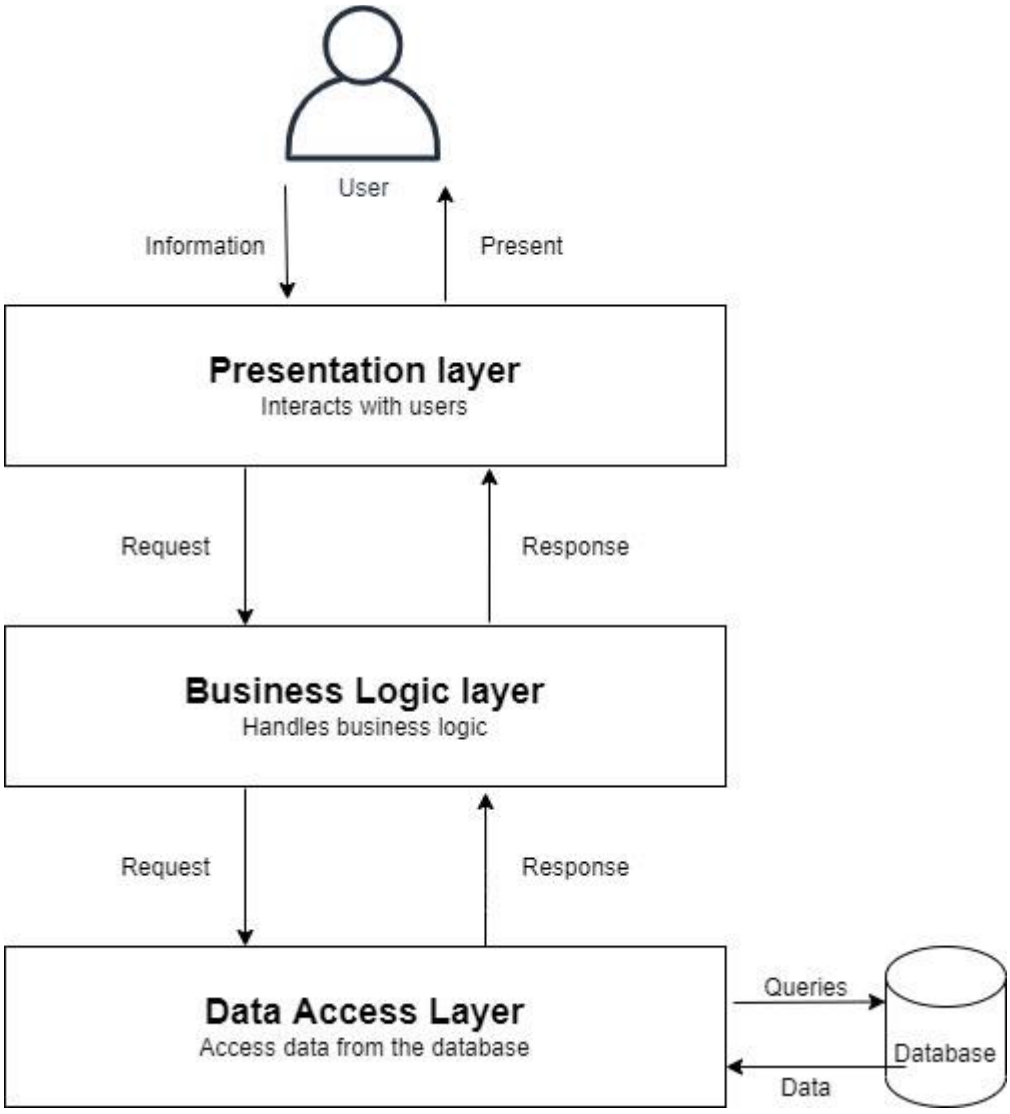
3. CSRF (Cross-Site Request Forgery)
What: Trick users into unwanted actions.
Solution: Use **anti-CSRF tokens** and re-authentication.

18) What is the role of encryption in securing applications?

- **Data Privacy**: Keeps sensitive data unreadable to unauthorized users.
- **Secure Communication**: Protects data during transfer (e.g., HTTPS).
- **Data Integrity**: Ensures data isn't altered in transit.
- **Authentication**: Verifies the identity of users and systems.
- encryption ensures **confidentiality**, **integrity**, and **authenticity**, keeping data safe from unauthorized access and tampering.

Layers in Software Architecture:

19)Create a case study on the functionality of the presentation, business logic, and data access layers of a given software system.



20) Why are layers important in software architecture?

- Layers in software architecture help by:

Separation of tasks: Each layer focuses on one job (e.g., UI, logic, data). **Easier maintenance**: Changes in one layer don't affect others.
Better scalability: Layers can be scaled separately.
Faster testing: Layers can be tested individually.
Improved security: Sensitive tasks are isolated.
Better teamwork: Teams can work on different layers without overlap.

Software Environments:

21) Explain the importance of a development environment in software production.

- A **development environment** is vital in software production because it allows developers to:

1. **Write and test code** without affecting production.
2. **Debug** and fix issues early.
3. Use **version control** to manage code changes.
4. Increase **efficiency** with pre-configured tools.
5. **Prevent bugs** from reaching users by catching them early.

- It ensures smooth and error-free software development.

Source code:

22) What is the difference between source code and machine code?

Aspect	Source Code	Machine Code
Definition	Human-readable code	Binary code understood by CPU
Purposes	Written by developers	Executed by the computer's CPU
Example	Print ("Hello, World!") (Python)	10111010 01101101 (binary)
Readability	Easy for humans to read	Unreadable to humans

Github and Introduction:

23) Why is version control important in software development?

- Version control is important in software development because it:

1. **Tracks changes**: Keeps a history of code changes.
2. **Enables collaboration**: Allows multiple developers to work together without conflicts.
3. **Supports rollback**: Lets you revert to previous stable versions.
4. **Acts as a backup**: Protects code from loss.
5. **Maintains code integrity**: Ensures consistency across updates.
6. **Supports branching**: Enables feature development without affecting the main code.

Student Account in Github:

24) What are the benefits of using Github for students?

- GitHub benefits students by:

1. **Version Control**: Tracks code changes and prevents errors.
2. **Collaboration**: Enables teamwork on projects.
3. **Portfolio**: Showcases coding skills to employers.
4. **Learning Git**: Provides experience with a key industry tool.
5. **Cloud Storage**: Stores projects safely online.
6. **Issue Tracking**: Manages tasks and bugs.
7. **Community**: Connects with other developers and open-source projects.
8. **Documentation**: Helps document and organize projects.

Types of Software:

25) Create a list of software you use regularly and classify them into the following categories: system, application, and utility software.

System Software:

- **Operating System**: Windows, macOS

- **Device Drivers**: Graphics, printer drivers

Application Software:

- **Web Browser**: Google Chrome
- **Word Processor**: Microsoft Word
- **Email Client**: Outlook
- **Media Player**: VLC
- **Photo Editing**: Photoshop

Utility Software:

- **Antivirus**: Avast
- **File Compression**: WinRAR
- **Backup**: Acronis True Image
- **Disk Cleanup**: CCleaner

26) What are the differences between open-source and proprietary software?

Aspect	Open-Source Software	Proprietary Software
Access to Source Code	Available for public use and modification	Source code is closed and not available
Cost	Usually free or low-cost	depending on the software

GIT and GITHUB Training:

27) How does GIT improve collaboration in a software development team?

- Git improves collaboration by:

1. **Tracking Changes**: Keeps a record of all code changes.
2. **Branching**: Allows multiple developers to work on different features at the same time.
3. **Merging**: Combines changes from different branches and resolves conflicts.
4. **Distributed**: Each developer has a local copy to work offline and sync later.
5. **Code Reviews**: Facilitates code review through pull requests.
6. **CI Integration**: Automatically tests and deploys code to ensure stability.

- In short, Git helps teams work independently, track progress, and integrate changes safely.

28) Write a report on the various types of application software and how they improve productivity.

- Application software helps users perform specific tasks efficiently, improving productivity across various fields. This report highlights different types of application software and their productivity benefits.

1. Productivity Software

Examples: Microsoft Word, Excel, PowerPoint, Google Docs

Impact:

- **Document Creation**: Simplifies writing and editing.
- **Data Management**: Spreadsheets help in analysis and calculations.
- **Communication**: Email clients organize and speed up communication.

2. Design and Media Software

Examples: Adobe Photoshop, Premiere Pro, Audacity

Impact:

- **Creative Work**: Enables efficient design, video, and audio production.
- **Time-saving**: Streamlines editing and enhances creativity in media projects.

3. Web Browsers

Examples: Google Chrome, Mozilla Firefox

Impact:

- **Information Access**: Facilitates quick and easy access to online resources.
- **Research**: Improves research efficiency through fast browsing and resource management.

4. Database Management Software

Examples: Microsoft Access, MySQL, Oracle

Impact:

- **Data Organization**: Efficiently stores, retrieves, and manages large amounts of data.
- **Automation**: Automates data processing tasks, reducing manual efforts.

5. Communication Software

Examples: Slack, Zoom, Skype

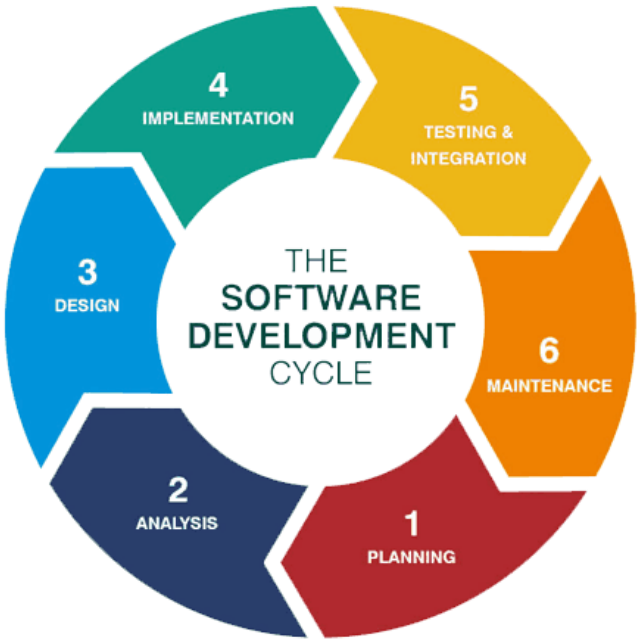
Impact:

- **Team Collaboration**: Enhances communication, allowing instant messaging, video calls, and file sharing.
- **Remote Work**: Supports virtual meetings and teamwork, boosting productivity.

29) What is the role of application software in businesses?

- **Automating Tasks**: Tools like accounting and CRM systems save time and reduce mistakes.
- **Improving Communication**: Email and messaging apps make team and client communication faster.
- **Managing Data**: Database software organizes and analyzes business data efficiently.
- **Supporting Decisions**: Analytics tools provide data insights for better decision-making.

30) Create a flowchart representing the Software Development Life Cycle (SDLC).



31) What are the main stages of the software development process?

- **Planning**: Decide on the project goals and what's needed to complete it.
- **Analysis**: Understand the user needs and gather requirements.

- **Design**: Plan how the software will work and look (structure and features).
- **Implementation**: Write the code to create the software.
- **Testing & Integration**: Test the software for problems and make sure all parts work together.
- **Maintenance**: Fix any issues, update features, and provide support after the software is released.

Software Requirement:

32) Write a requirement specification for a simple library management system.

- **User Management**: Register, log in, and manage user profiles. Admins can manage users.
- **Book Management**: Admins add/update/delete books. Users can search books by title, author, or genre.
- **Issue/Return**: Users borrow/return books; admins track transactions.
- **Search & Catalog**: Users search books by title, author, or genre, and view availability.
- **Reports**: Admins generate reports on books and users.

Non-Functional: Secure, user-friendly, fast performance.

33) Why is the requirement analysis phase critical in software development?

- proper requirement analysis is essential for building software that meets user expectations, stays within budget, and is delivered on time.

Software Analysis:

34) Perform a functional analysis for an online shopping system.

1. **User Account**: Register, log in, and manage profile details (personal info, passwords, payment methods).

2. **Product Search & Browsing**: Search for products, filter by categories, and view detailed product info.
3. **Shopping Cart**: Add/remove items, adjust quantities, and view a cart summary.
4. **Order Checkout**: Proceed to checkout, select shipping, and make secure payments.
5. **Payment Processing**: Multiple payment options (cards, wallets) with secure transactions.
6. **Shipping**: Choose delivery methods and track orders.
7. **Customer Support**: Access help for returns, exchanges, and refunds.
8. **Reviews**: Rate and review products.

- This covers the essential functions of an online shopping system.

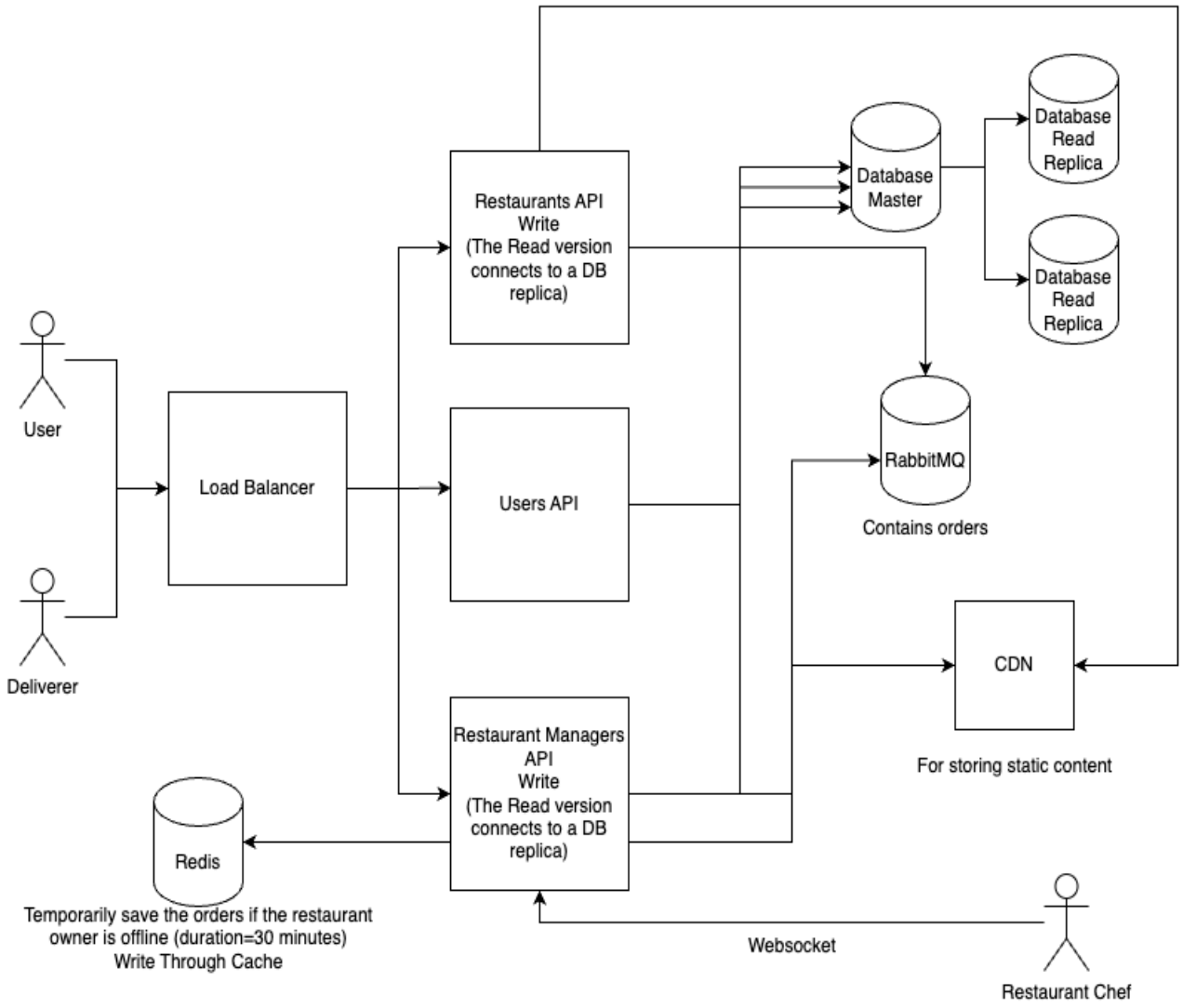
35)What is the role of software analysis in the development process?

- **Software analysis** is important because:
- **Clarifies Requirements**: It helps understand what users need from the software.
- **Identifies Problems Early**: It finds issues before they become bigger problems.
- **Defines Project Scope**: It prevents unexpected changes during development.
- **Guides Design**: It provides a clear plan for developers to follow.
- **Manages Risks**: It highlights potential risks to address early.
- software analysis ensures the project stays on track and meets user needs.

System Design:

36) Design a basic system architecture for a food delivery app.

-



37) What are the key elements of system design?

- System design includes:

1. **Requirements**: Understanding system needs.
2. **Architecture**: Structuring components and their interactions.
3. **Data Design**: Organizing data storage and access.
4. **Scalability**: Ensuring the system grows with demand.
5. **Security**: Protecting against threats.
6. **Performance**: Optimizing efficiency.
7. **Fault Tolerance**: Ensuring reliability and recovery.
8. **Testing**: Validating functionality.

- These elements ensure the system is functional, scalable, and secure.

Software Testing:
38) Develop test cases for a simple calculator program.

- **Addition**:

- 5 + 3 = 8
- 5 + (-3) = 2

- **Subtraction**:

- 10 - 4 = 6
- 3 - 5 = -2

- **Multiplication**:

- 4 * 3 = 12
- 7 * 0 = 0

- **Division**:

- 10 / 2 = 5
- 5 / 0 = Error

- **Edge Cases**:

- 1000000 + 1 = 1000001
- 10 - 10 = 0
- 5.5 + 2.3 = 7.8

- **Invalid Input**:

- "a" + 5 = Error
- Empty input = Error

39) Why is software testing important?

- Software testing is important because it:

1. Ensures quality by finding bugs.
2. Improves reliability and performance.
3. Enhances user experience.
4. Reduces costs by catching issues early.
5. Ensures the software meets requirements.
6. Boosts security by identifying vulnerabilities.

- It helps deliver better, more reliable software.

Maintenance:

40) Document a real-world case where a software application required critical maintenance.

- Case: E-Commerce Platform Maintenance

Problem:

XYZ Corp's online store faced slow checkout, payment failures, and poor performance during peak traffic.

Cause:

- Outdated software and database issues.
- Incompatible payment gateway.
- System couldn’t handle traffic spikes.

Fixes:

- Optimized code and database.

- Updated payment gateway integration.
- Scaled infrastructure for higher traffic.

Outcome:

- Faster website and checkout.
- Fewer payment issues.
- Increased customer satisfaction and sales recovery.

41) What types of software maintenance are there?

- **Corrective**: Fixing bugs.
- **Adaptive**: Updating for new environments.
- **Perfective**: Improving performance and adding features.
- **Preventive**: Preventing future issues.

42) What are the key differences between web and desktop applications?

Feature	Web Application	Desktop Application
Platform	Runs in a browser.	Installed on a specific device.
Access	Requires internet connection.	Can run offline.
Updates	Automatically updated.	Requires manual updates.

43) What are the advantages of using web applications over desktop applications?

- Here's the comparison in table form:

Advantages	Explanation
Accessibility	Access from any device with internet.
Platform Independence	Works across different operating systems.
Automatic Updates	Updates are applied automatically.
No Installation	No need for installation on each device.
Centralized Management	Easier to manage on a central server.
Collaboration	Easier real-time collaboration and sharing.

Designing:

44) What role does UI/UX design play in application development?

- UI/UX design is essential in app development because it ensures the app is visually appealing, easy to navigate, and user-friendly.

- **UI Design** focuses on the visual elements, layout, and consistency, making the app aesthetically pleasing and intuitive.
- **UX Design** focuses on optimizing the user experience, ensuring the app is easy to use, efficient, and enjoyable.

- Together, they improve user satisfaction, enhance accessibility, boost usability, and provide a competitive edge.

Mobile Application:

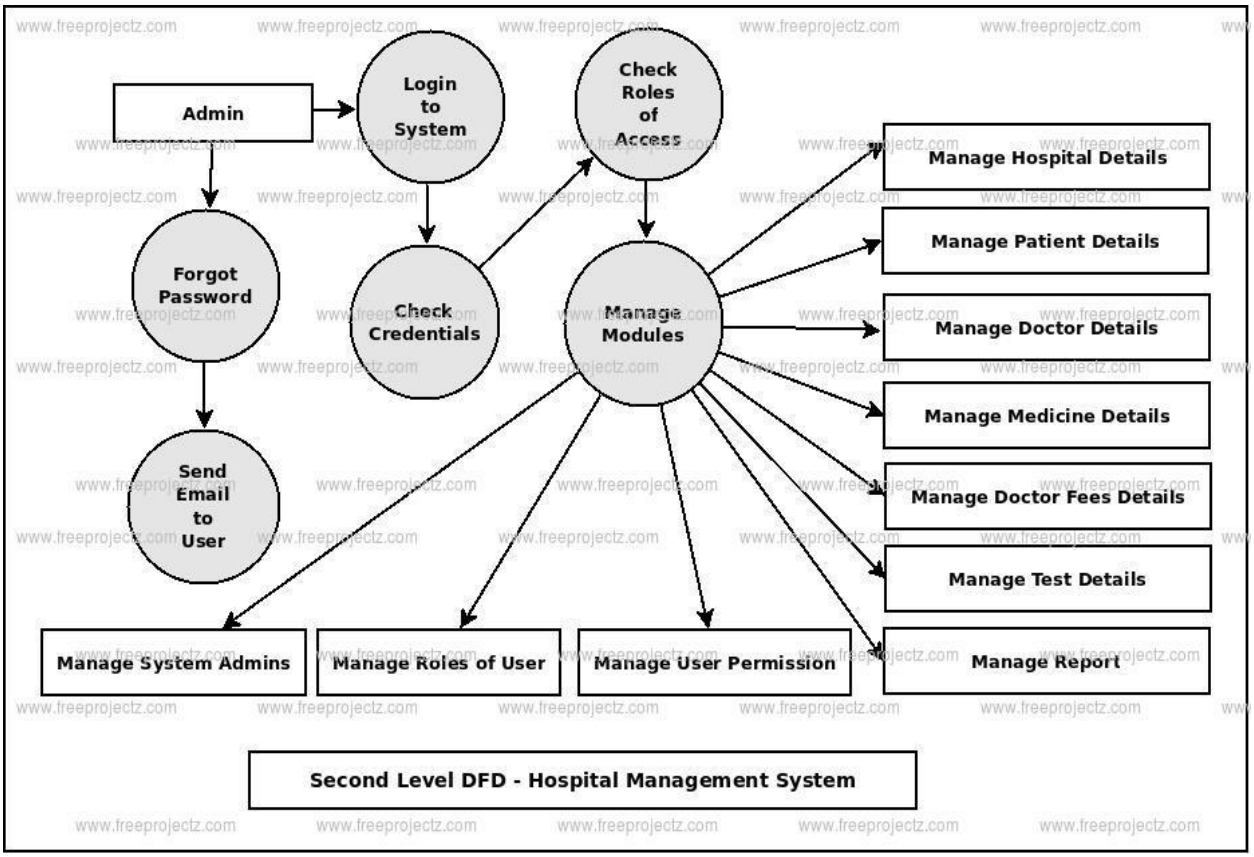
45) What are the differences between native and hybrid mobile apps?

-

Aspect	Native Apps	Hybrid Apps
Development	Platform-specific (iOS, Android)	Single codebase for multiple platforms
Performance	Faster and more efficient	Slower due to web view
User Experience	Better UX, seamless	May feel less integrated
Cost	Higher development cost	Lower development cost

DFD (Data Flow Diagram):

46) Create a DFD for a hospital management system.



47) What is the significance of DFDs in system analysis?

- DFDs help visualize how data flows through a system, making it easier to understand and design. They simplify complex systems, improve communication among team members, and help identify problems. Overall, DFDs make system analysis clearer and more efficient.

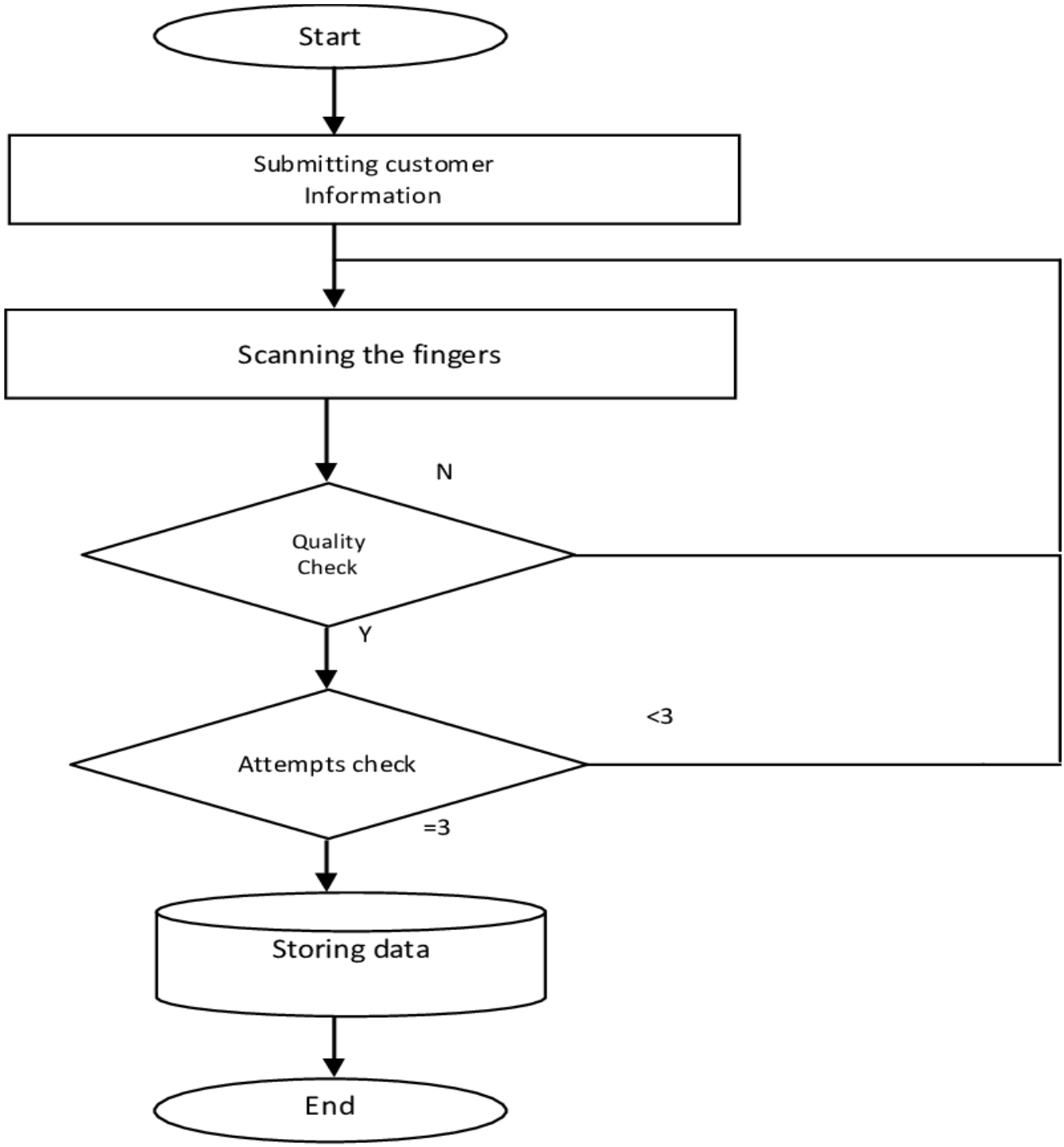
Desktop Application:

48) What are the pros and cons of desktop applications compared to web applications?

Feature	Desktop Apps	Web Apps
Performance	Faster, more resource-efficient	Slower, browser-dependent
Installation	Requires installation	No installation needed
Updates	Manual updates	Automatic updates
Security	More control over security	More vulnerable to online threats
Maintenance	Harder to maintain	Easier to maintain centrally
Accessibility	Device-specific	Accessible from any device
Customization	Highly Customizable	Limited by browser capabilities

Flow Chart:

49) Draw a flowchart representing the logic of a basic online registration system.



50)How do flowcharts help in programming and system design?

- Flowcharts help in programming and system design by:
- **Visualizing Logic:** Simplifying complex processes and showing the flow of operations.
- **Clarifying Steps:** Breaking down tasks into clear, understandable steps.
- **Debugging:** Identifying errors or inefficiencies in the system by following the flow.