**Lab Exercise:**  
Research different types of internet connections and list their pros and cons.

Solution:

1. Broadband:  
 Pros: Fast, reliable, always-on connection.  
 Cons: Availability depends on location.  
  
 2. Fiber:  
 Pros: Very high speed, low latency.  
 Cons: Expensive, limited availability.  
  
 3. Satellite:  
 Pros: Available in remote areas.  
 Cons: High latency, weather-dependent.  
  
 4. Mobile Data (4G/5G):  
 Pros: Portable, flexible.  
 Cons: Data limits, network coverage issues.

**Lab Exercise:**Research and create a diagram of how data is transmitted from a client to a server over the internet.

Solution:

Steps of Data Transmission:  
 1. Client sends a request (HTTP) to the server via browser.  
 2. Request passes through network layers (Application → Transport → Network → Data Link → Physical).  
 3. Server receives the request, processes it, and sends back a response.  
 4. Response passes through the same layers in reverse and reaches the client.  
  
 Diagram (simplified representation):  
 Client → ISP → Internet Backbone → Web Server → Response back to Client

## **Types of Internet Connections**

Lab Exercise:  
Research different types of internet connections and list their pros and cons.

Solution:

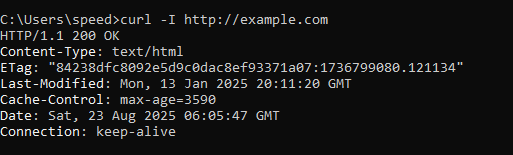
1. Broadband:  
 Pros: Fast, reliable, always-on connection.  
 Cons: Availability depends on location.  
  
 2. Fiber:  
 Pros: Very high speed, low latency.  
 Cons: Expensive, limited availability.  
  
 3. Satellite:  
 Pros: Available in remote areas.  
 Cons: High latency, weather-dependent.  
  
 4. Mobile Data (4G/5G):  
 Pros: Portable, flexible.

Cons: Data limits, network coverage issues.

**Protocols**

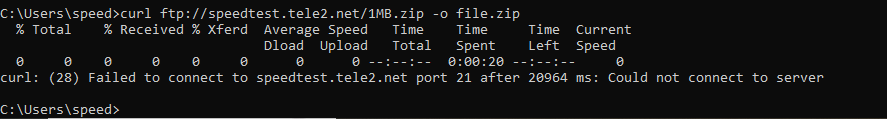
Lab Exercise:   
Simulate HTTP and FTP requests using command line tools (e.g., curl).

Solution: Example using curl (HTTP):



curl http://example.com

Example using curl (FTP):



The FTP command (syntax written, even if failed)

Mention in your report:

“FTP test server could not be reached due to connectivity restrictions. The correct curl command syntax is shown above.”

curl ftp://speedtest.tele2.net/1MB.zip -o file.zip  
  
**Application Security**

Lab Exercise:  
Identify and explain three common application security vulnerabilities. Suggest possible solutions.

Solution:

1. SQL Injection:  
 Vulnerability: Attackers insert malicious SQL queries.  
 Solution: Use prepared statements.  
  
 2. Cross-Site Scripting (XSS):  
 Vulnerability: Injection of malicious scripts in web pages.  
 Solution: Validate and sanitize inputs.  
  
 3. Weak Passwords:  
 Vulnerability: Easy to guess or brute-force.  
 Solution: Enforce strong password policies, use MFA.

## **Software Applications and Its Types**

Lab Exercise:  
Identify and classify 5 applications you use daily as either system software or application software.

| **Application Used Daily** | **Category (System/Application Software)** | **Reason** |
| --- | --- | --- |
| Windows 10 | System Software | It controls the computer and manages hardware. |
| Google Chrome | Application Software | Used for browsing the internet. |
| MS Word | Application Software | Used for creating and editing documents. |
| Antivirus (e.g., Avast) | System Software | Protects the system from malware and viruses. |
| WhatsApp (Desktop/Mobile) | Application Software | Used for messaging and communication. |

In this practical, I identified five software applications I use daily and classified them as either system software or application software. Windows 10 and Antivirus software are system software as they manage and protect the system. Chrome, MS Word, and WhatsApp are application software because they help me perform specific user tasks."

Lab Exercise: Design a basic three-tier software architecture diagram for a web application.

Solution:

Three-Tier Architecture:

1. Presentation Layer (User Interface – Web browser)

2. Business Logic Layer (Server-side scripts – Python, PHP, Java)

3. Data Layer (Database – MySQL, MongoDB)  
  
 Diagram Representation:  
 [Client UI] ↔ [Application Server] ↔ [Database]

## 

## **Layers in Software Architecture**

### **1. Presentation Layer (User Interface)**

* This is the **front-end** that the user interacts with.
* Includes mobile app or website screens.
* Functions:  
  + Displays menu, restaurant details, and offers.
  + Allows users to select food items and place orders.
  + Provides features like login, cart, and payment options.

### **2. Business Logic Layer (Application Layer)**

* Acts as the **middle layer** that processes requests from the user.
* Functions:  
  + Validates user login details.
  + Applies business rules such as discounts, taxes, and delivery charges.
  + Coordinates between user actions and database operations.
  + Ensures correct flow from order placement to confirmation.

### **3. Data Access Layer (Database Layer)**

* This is the **back-end** that stores and manages data.
* Functions:  
  + Stores restaurant information, food items, user details, and past orders.
  + Retrieves menu and restaurant data when requested.
  + Updates the database after every successful order.
  + Ensures secure storage of sensitive data like payment details.

In this case study, the three layers work together as follows:

* The **Presentation Layer** shows the app interface.
* The **Business Logic Layer** manages order processing and rules.
* The **Data Access Layer** stores and provides data.

This separation improves **scalability, security, and maintainability** of the software system.

## **Software Environments**

Lab Exercise:  
Explore different types of software environments (development, testing, production).

**1. Development Environment**

* **Purpose:** Used by programmers to build and write code.
* **Tools:** IDEs (Visual Studio Code, Eclipse), compilers, and local servers.
* **Example:** A student writes and runs a small Python or Java program on their computer.

### **2. Testing Environment**

* **Purpose:** Used to test the software before it is released.
* **Tools:** Testing frameworks (JUnit, Selenium), test servers.
* **Example:** After coding, a team tests a website for bugs like broken links or incorrect login validation.

### **3. Production Environment**

* **Purpose:** The final live environment where the end-users access the software.
* **Tools:** Cloud servers (AWS, Azure), monitoring tools.
* **Example:** The actual live e-commerce website (like Amazon) where customers place real orders.

### **Conclusion:**

* **Development** → where code is written.
* **Testing** → where quality is checked.
* **Production** → where real users use the software.  
   Each environment is essential to ensure that the software is reliable, bug-free, and user-friendly.

## **Types of Software**

Lab Exercise:   
Types of Software  
To list software used daily and classify them into System Software, Application Software, and Utility Software.

I observed and listed the software I use regularly on my computer and mobile.

**System Software:**  
Windows 11 (Operating System)

**Application Software:**  
MS Word (Document editing)

Google Chrome (Web browsing)

WhatsApp (Messaging/Communication)

**Utility Software:**

Antivirus (Protects the system from viruses)

WinRAR (File compression and extraction)

Lab Exercise:  
Report on Types of Application Software and Their Role in Productivity

To explore the different types of application software and understand how they help users in daily life and professional work.

1. Word Processing Software

Example: MS Word, Google Docs

Use: Writing, editing, and formatting documents.

Productivity Impact: Saves time, improves document quality, and makes sharing easier.

2. Spreadsheet Software

Example: MS Excel, Google Sheets

Use: Data analysis, financial calculations, graphs.

Productivity Impact: Helps in quick calculations, automation with formulas, and better decision-making.

3. Presentation Software

Example: MS PowerPoint, Google Slides

Use: Creating slideshows for teaching, business, or training.

Productivity Impact: Improves communication and makes learning or presentations more engaging.

4. Database Software

Example: MySQL, MS Access

Use: Storing and managing large sets of information.

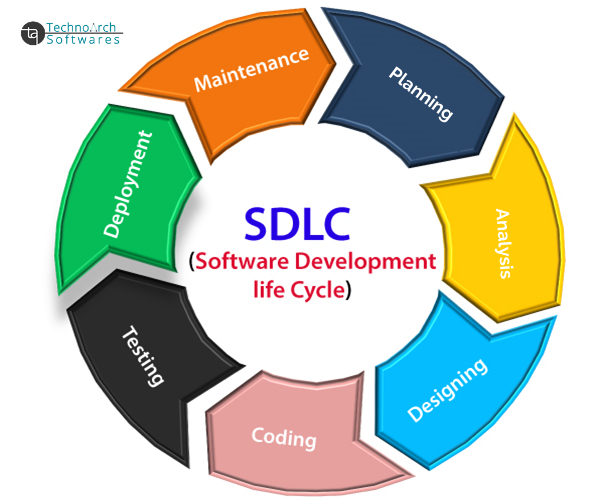
Productivity Impact: Makes searching, retrieving, and managing data faster and more reliable.  
  
5. Communication Software

Example: WhatsApp, Gmail, Zoom

Use: Messaging, emailing, video calls.

Productivity Impact: Increases collaboration, reduces travel time, and speeds up work processes.  
  
Application software directly improves productivity by automating tasks, organizing data, supporting collaboration, and simplifying work. Different software types are used in daily life, education, and businesses to save time and improve efficiency.

## **Software Development Process**

Lab Exercise:  
Create a flowchart representing the Software Development Life Cycle (SDLC).

## **Software Requirement**

Lab Exercise: Write a requirement specification for a simple library management system.

Lab Exercise: Requirement Specification for a Library Management System

Objective:

To prepare requirement specifications for a simple Library Management System (LMS).

Requirements:  
1. Functional Requirements:

User registration/login.

Add, search, update, and delete books.

Issue and return books.

Generate reports of issued/returned books.

2. Non-Functional Requirements:

Easy to use.

Secure login.

Fast search response.

Scalable for more users/books.

3. System Requirements:

Hardware: 2 GB RAM, 10 GB storage.

Software: Windows/Linux, MySQL, Java/Python/PHP.

The LMS helps manage books and users efficiently, ensuring smooth library operations.

## **Software Analysis**

Lab Exercise:   
Perform a functional analysis for an online shopping system.

To analyze and list the core functions of an online shopping system.

Functional Requirements:

1. User Management

User registration and login.

Profile management.

2. Product Management

Search and browse products.

View product details.

Add/update/delete products (for admin).

3. Shopping Cart

Add items to the cart.

Update or remove items.

View cart summary.

4. Order Management

Place orders.

View order history.

Cancel/return orders.

5. Payment & Checkout

Multiple payment options (credit/debit, UPI, COD).

Generate invoice/receipt.

6. Customer Support

FAQs and helpdesk.

Contact customer service.

The functional analysis shows how an online shopping system supports users, sellers, and admins to perform activities like browsing, ordering, paying, and managing products.

## **System Design**

Lab Exercise:   
Design a Basic System Architecture for a Food Delivery App

Objective:

To design a simple system architecture that explains how a food delivery application works.

System Architecture Components:

1. Presentation Layer (User Interface):

Mobile App / Website for customers.

Restaurant dashboard for order management.

Delivery partner app for tracking deliveries.

2. Application Layer (Business Logic):

Handles order placement and processing.

Matches the delivery partner with the customer location.

Manages payments, discounts, and notifications.

3. Data Layer (Database & Storage):

Stores customer profiles, restaurant details, and menus.

Stores order history and delivery status.

Secures payment transactions and feedback.

Workflow:

1. Customer browses restaurants and selects food.

2. Order is placed → Restaurant receives notification.

3. Restaurant prepares food → Delivery partner assigned.

4. The delivery partner picks up and delivers to the customer.

5. System updates status in real-time and stores history.

This architecture ensures smooth communication between customers, restaurants, delivery partners, and the system, making food ordering fast and reliable.

## 

## **Maintenance**

Lab Exercise:

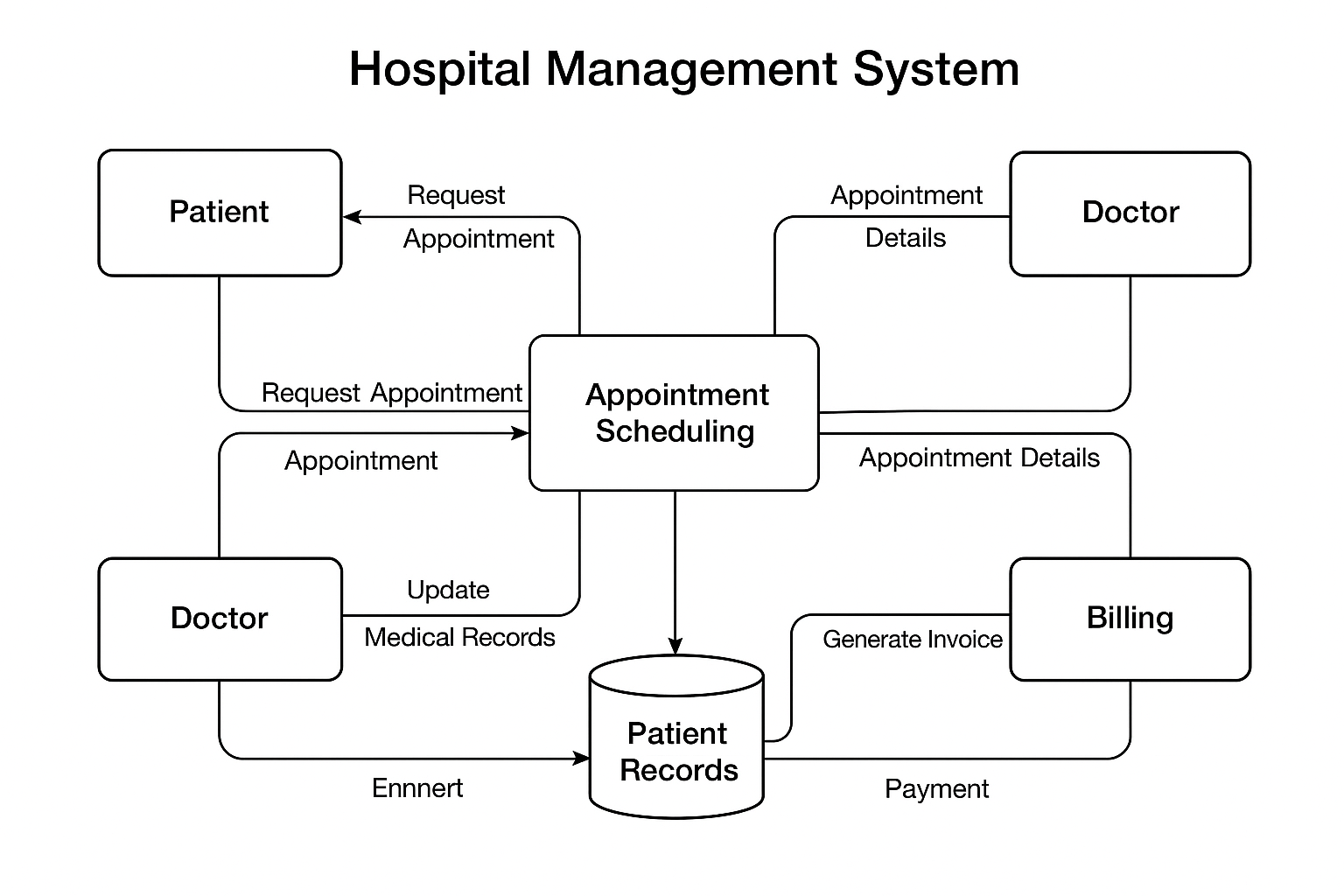
Document a real-world case where a software application required critical maintenance.

To understand how software maintenance plays a critical role in ensuring system availability, performance, and security.

### **Case Study – WhatsApp Outage (October 2022)**

1. **Background:**
   * WhatsApp, one of the most widely used messaging applications, experienced a **global outage** in October 2022.
   * Billions of users were unable to send or receive messages for more than 2 hours.
2. **Problem:**
   * The issue was traced back to a **software update/configuration problem** in WhatsApp’s backend servers.
   * Incorrect configuration caused server processes to fail, leading to downtime across multiple regions.
3. **Critical Maintenance Performed:**
   * Engineers had to **roll back the faulty update** and reconfigure the affected servers.
   * Monitoring systems were upgraded to detect such failures faster.
   * Preventive maintenance steps were added to reduce downtime risk in the future.
4. **Impact:**
   * Millions of users were affected, including businesses that depend on WhatsApp for communication.
   * The outage highlighted the importance of **critical maintenance and quick recovery** in real-time applications.
5. **Lessons Learned:**
   * Always test software updates in a **staging/test environment** before deploying to production.
   * Critical applications must have a **disaster recovery plan**.
   * Continuous monitoring and **automated alerts** are essential.

## **DFD (Data Flow Diagram)**

Lab Exercise:  
Create a DFD for a hospital management system.  


## **Flow Chart**

Lab Exercise: Draw a flowchart representing the logic of a basic online registration system.

Solution: