System Analysis and Design – Detailed Explanation (Unit-Wise)

Unit - I: System Concepts and Methodology

1. System Analysis Fundamentals

1.1 Introduction to System

A **system** is a collection of interrelated components that work together to achieve a common goal. It takes **input**, processes it, and produces **output**.

✅ **Key Characteristics of a System:**

- **Input:** Raw data or information entered into the system.
- **Process:** The operations performed on the input.
- **Output:** The final result or information generated.
- **Control:** Ensures the system functions as intended.
- **Feedback:** Provides information on the system's performance.

✅ **Example:**
A **Banking System** takes customer transactions (input), processes them, and provides account balance details (output).

1.2 System Analysis and Design

System Analysis refers to the process of studying and understanding a system’s components, interactions, and operations.
System Design is the phase where the system’s structure, data flow, and user interface are created based on the analysis.

✅ **Key Differences:**

System Analysis	System Design
Studies the existing system.	Creates a new or modified system.
Identifies problems.	Provides solutions.
Focuses on **"What"** to do.	Focuses on **"How"** to do it.
Defines requirements.	Specifies the architecture.

✅ **Example:**

- **Analysis:** Identify customer needs and issues in an online shopping website.
- **Design:** Create the layout, database, and navigation flow of the website.

1.3 Need for System Analysis and Design

System Analysis and Design is essential for:

- **Problem Identification:** Helps in identifying and fixing issues in existing systems.

- **Efficiency Improvement:** Improves speed, accuracy, and functionality.
- **Cost-effectiveness:** Reduces operational costs by optimizing processes.
- **User Satisfaction:** Ensures the system meets user needs and preferences.

✅ **Example:**
A hospital management system undergoes analysis to identify issues with patient data management and design better patient record handling.

1.4 Types of Systems

There are several types of systems based on **functionality** and **structure**:

1. **Physical System:** Tangible systems like hardware, machines, etc.
2. **Abstract System:** Conceptual systems such as mathematical models.
3. **Open System:** Interacts with its environment (e.g., a library system).
4. **Closed System:** No interaction with the environment (e.g., a clock).
5. **Deterministic System:** Predictable output (e.g., calculator).
6. **Probabilistic System:** Output may vary (e.g., weather forecasting system).

1.5 Role of the System Analyst

A **System Analyst** is responsible for analyzing, designing, and implementing information systems.

✅ **Key Responsibilities:**

- **Requirement Analysis:** Identifies user needs.
- **Designing Solutions:** Creates system models and prototypes.
- **Documentation:** Prepares technical documentation.
- **Testing and Validation:** Ensures system accuracy and reliability.
- **Communication:** Acts as a bridge between **users** and **developers**.

✅ **Example:**
In a **Payroll System**, the system analyst collects employee data, identifies calculation methods, and designs the salary processing flow.

---

2. System Development Strategies

2.1 SDLC (System Development Life Cycle)

The **SDLC** is a step-by-step process used for building, deploying, and maintaining systems.

✅ **Phases of SDLC:**

1. **Requirement Analysis:** Identify system requirements.
2. **System Design:** Define architecture and components.
3. **Implementation:** Coding and testing the system.
4. **Testing:** Detect and fix errors.
5. **Deployment:** Make the system operational.
6. **Maintenance:** Regular updates and fixes.

✅ **Example:**
Developing a **Library Management System** using the SDLC ensures structured development and efficient implementation.

2.2 Difference between System Analysis and System Design

System Analysis	System Design
Defines system requirements.	Creates the system architecture.
Focuses on understanding the problem.	Focuses on creating the solution.
Involves data collection and study.	Involves creating models and structures.
"What" to do	**"How" to do**

2.3 Need for Structured Analysis and Design

Structured analysis ensures:

- **Clarity:** Clear representation of system requirements.
- **Consistency:** Standardized process with fewer errors.
- **Efficiency:** Reduces complexity by using modular design.
- **Better Communication:** Helps developers, testers, and users understand the system flow.

✅ **Example:**
A **Hotel Management System** uses structured analysis to define how customers book rooms, check-in, and check-out.

2.4 Structured Analysis Development Method (SSADM)

SSADM is a systematic approach used to analyze and design information systems.

✅ **Phases of SSADM:**

1. **Feasibility Study:** Identify whether the project is viable.
2. **Requirements Analysis:** Collect and analyze user needs.
3. **Logical Design:** Create DFDs and ER diagrams.
4. **Physical Design:** Design the database, user interfaces, and processes.

✅ **Example:**
In a **Payroll System**, SSADM ensures proper flowchart design, data dictionary creation, and output verification.

2.5 System Prototype Method (SPM)

System Prototyping involves creating a **working model** of the system before final development.

✅ **Advantages of Prototyping:**

- **User Feedback:** Helps gather user input early.
- **Error Detection:** Detects flaws in the early stages.
- **Time-saving:** Speeds up development.

- **Cost-effective:** Reduces rework costs.

✅ **Example:**
An **Online Shopping System** can create a prototype of the product search, cart, and payment sections before full-scale development.

Unit - II: System Tools and Techniques

1. Fact-Finding Techniques

1.1 Interview

A direct conversation between the system analyst and stakeholders to gather requirements.

✅ **Types of Interviews:**

- **Structured:** Pre-defined set of questions.
- **Unstructured:** Free-flow conversation.
- **Semi-Structured:** Combination of both.

✅ **Example:**
Interviewing **hotel staff** to gather requirements for the **Hotel Management System**.

1.2 Questionnaire

A **set of questions** designed to collect information from multiple respondents.

✅ **Advantages:**

- Covers a large audience.
- Cost-effective.
- Easy to analyze.

✅ **Example:**
Sending questionnaires to **library users** to understand their needs.

1.3 Record Review

Analyzing **existing documents and records** to gather information.

✅ **Example:**
Reviewing **payroll records** to design a new salary calculation system.

1.4 Observation

Directly observing system users to understand their activities.

✅ **Example:**
Observing **cashiers** in a supermarket to study the **billing process**.

2. System Flowchart

A graphical representation of the system's processes and data flow.

✅ **Types of System Flowcharts:**

1. **Input/Output Flowchart**
2. **Process Flowchart**
3. **Decision Flowchart**

✅

System Analysis and Design – Detailed Explanation (Unit-Wise) – Continued

---

Unit - III: System Design

1. Code Design

1.1 Objectives of Code Design

Code design refers to creating **efficient, organized, and readable** code for system operations.

✅ **Objectives:**

- **Efficiency:** Code should execute quickly and accurately.
- **Readability:** Easy to read and understand by other developers.
- **Modularity:** Divide the code into reusable components.
- **Error Handling:** Ensure proper validation and exception handling.

✅ **Example:**
In an **Online Shopping System**, the code design ensures proper validation for payment gateways and user authentication.

1.2 Principles of Code Design

When designing code, the following principles should be followed:

1. **Modularity:** Divide the program into smaller modules for better management.
2. **Reusability:** Write reusable functions to avoid redundancy.
3. **Clarity:** Use meaningful variable names and comments.
4. **Consistency:** Follow a consistent coding style.
5. **Error Handling:** Include proper exception handling.

✅ **Example:**
In a **Library System**, separate modules for **book management**, **user management**, and **transaction handling** ensure modularity.

1.3 Types of Codes

In system design, different coding systems are used for **data representation** and **processing**.

✅ **Types of Codes:**

- **Alphanumeric Codes:** Combination of letters and numbers (e.g., AB1234).
- **Binary Codes:** Use 0s and 1s for data representation.
- **Error Detection Codes:** Used for error detection (e.g., parity bits).
- **Mnemonic Codes:** Abbreviations representing operations (e.g., ADD, SUB).
- **Barcodes:** Visual representation of data using lines and spaces.

✅ **Example:**
In a **Payroll System**, employee IDs are coded using **alphanumeric codes** like EMP001, EMP002, etc.

---

2. Form Design

2.1 Objectives of Form Design

Form design refers to creating **user interfaces** for data entry.

✅ **Objectives:**

- **User-friendliness:** Easy to use with clear instructions.
- **Data Accuracy:** Prevents errors through validation.
- **Efficiency:** Allows quick and efficient data entry.
- **Consistency:** Uniform appearance and layout.

✅ **Example:**
A **Hotel Booking Form** collects customer details, check-in and check-out dates, and payment details.

2.2 Types of Forms

1. **Data Entry Form:** Used for entering new data (e.g., customer registration).
2. **Inquiry Form:** Retrieves existing data (e.g., customer search form).
3. **Transaction Form:** Used for processing transactions (e.g., payment form).
4. **Report Form:** Displays output data (e.g., sales report form).

✅ **Example:**
In an **Inventory System**, a **data entry form** is used to add new products, while a **report form** displays the current stock.

2.3 Guidelines for Form Design

- **Clarity:** Use clear labels and instructions.
- **Grouping:** Group related fields together.
- **Navigation:** Provide clear navigation buttons (Next, Previous).
- **Validation:** Ensure data validation (e.g., email format check).
- **Consistency:** Use consistent fonts, colors, and alignments.

✅ **Example:**
In a **Login Form**, fields like **username** and **password** should be clearly labeled with validation for correct input format.

2.4 Form Design Steps

1. **Requirement Analysis:** Identify what data needs to be collected.
2. **Design Layout:** Create a rough sketch of the form.
3. **Field Placement:** Place fields logically (e.g., name, address, phone).
4. **Validation:** Add validation checks.
5. **User Testing:** Test the form for usability.

✅ **Example:**
In an **Online Shopping System**, the form design includes sections for **user details**, **shipping address**, and **payment information**.

---

3. Input Design

3.1 Objectives of Input Design

Input design focuses on creating **efficient and accurate data entry interfaces**.

✅ **Objectives:**

- **Accuracy:** Prevent incorrect data entry.
- **Efficiency:** Minimize data entry time.
- **User-friendliness:** Make it easy to understand and use.
- **Consistency:** Maintain a uniform style across all inputs.

✅ **Example:**
In a **Payroll System**, the input design includes fields for **employee ID**, **hours worked**, and **salary details**.

3.2 Data Capture

Data capture involves **collecting and entering** data into the system.

✅ **Methods of Data Capture:**

- **Manual Data Entry:** User manually enters the data.
- **Scanners:** Capture data using barcodes or QR codes.
- **Magnetic Strip Cards:** Used for capturing credit/debit card data.
- **RFID Tags:** Automatically capture data from tagged objects.

✅ **Example:**
A **Library Management System** uses **barcode scanners** to capture book details.

3.3 Data Validation

Data validation ensures that only **correct and valid data** is entered into the system.

✅ **Validation Techniques:**

- **Range Check:** Ensures values fall within a specific range.
- **Format Check:** Ensures data format is correct (e.g., email format).
- **Presence Check:** Ensures no field is left empty.
- **Consistency Check:** Compares data with related fields.

✅ **Example:**
In an **Online Registration Form**, validation ensures that **email format** is correct and **phone number** contains 10 digits.

---

4. Output Design

4.1 Objectives of Output Design

Output design focuses on creating **clear and accurate displays or reports**.

✅ **Objectives:**

- **Accuracy:** Ensure output correctness.
- **Clarity:** Make the output easy to read and understand.
- **Relevance:** Display only relevant information.
- **Timeliness:** Provide output when needed.

✅ **Example:**
In a **Payroll System**, the output design generates **salary slips** with employee details.

4.2 Principles of Output

- **Clarity:** Easy to read and interpret.
- **Consistency:** Uniform format and layout.
- **Accuracy:** Display accurate and reliable data.
- **Timeliness:** Display information promptly.

✅ **Example:**
An **Online Shopping System** displays **order confirmation details** immediately after purchase.

4.3 Types of Output

1. **Internal Output:** Used by system users (e.g., reports for management).
2. **External Output:** For external users (e.g., customer invoices).
3. **Hard Copy Output:** Printed reports or documents.
4. **Soft Copy Output:** Displayed on screen (e.g., email receipts).

✅ **Example:**
In a **Stock Management System**, **internal output** shows stock levels, while **external output** generates customer invoices.

4.4 Output Media

Output can be presented in different forms:

- **Screen Display:** Real-time information display.
- **Printed Reports:** Hard copy reports for record-keeping.
- **Electronic Files:** Digital output (PDF, Excel, etc.).
- **Audio/Visual Output:** Alerts, notifications.

✅ **Example:**
In an **Online Examination System**, results are shown on **screen display** and sent as **email PDFs**.

---

5. System Implementation and Testing

5.1 Training

Training prepares **users and staff** to use the new system effectively.

✅ **Types of Training:**

- **On-the-Job Training:** Hands-on training during work.
- **Classroom Training:** Theoretical and practical sessions.
- **Online Training:** Remote learning using tutorials.

✅ **Example:**
In a **Library System**, staff is trained to **manage book records**.

✅

System Analysis and Design – Detailed Explanation (Unit-Wise) – Continued

---

Unit - IV: Case Studies

1. Case Study: Stock (Inventory) Management System

1.1 Context Level DFD

The **Context Level Data Flow Diagram (DFD)** shows the entire system as a **single process** with external entities and data interactions.

✅ **Components:**

- **External Entities:** Users interacting with the system (e.g., Supplier, Manager).
- **Process:** The entire system represented as one process.
- **Data Flows:** Information moving between entities and the system.
- **Data Store:** Where data is saved (e.g., Inventory Database).

✅ **Example:**

```css
[Supplier] → [Stock Management System] → [Inventory Database]
```

- The supplier sends stock details.
- The system updates the inventory database.
- The manager views stock reports.

1.2 First Level DFD

The **First Level DFD** breaks down the **main system** into smaller processes.

✅ **Processes:**

1. **Stock Entry:** Adding new stock items.
2. **Stock Update:** Modifying stock details.
3. **Stock Report:** Generating inventory reports.
4. **Stock Removal:** Removing sold or expired items.

✅ **Data Flows:**

- **Input:** Supplier sends stock data.
- **Output:** Reports are generated and sent to the manager.

1.3 Second Level DFD

The **Second Level DFD** further **details individual processes**.

✅ **Processes:**

1. **Add Stock:**
   - Verify product details.
   - Add quantity and save to the database.
2. **Update Stock:**
   - Retrieve existing stock.
   - Modify details.
3. **Generate Report:**
   - Filter stock based on date, category, etc.
   - Generate and display the report.

✅ **Example:** In an **Inventory System**, the second-level DFD shows detailed steps for **stock addition, removal, and report generation**.

---

2. Case Study: Hotel Management System

2.1 Context Level DFD

Shows the entire **Hotel Management System** as a single process interacting with external entities.

✅ **Entities:**

- **Customer:** Makes reservations.
- **Receptionist:** Manages bookings and customer records.
- **Manager:** Generates reports and oversees operations.

✅ **Data Flow:**

```css
[Customer] → [Hotel Management System] → [Booking Database]
```

- The customer books a room.
- The receptionist manages the booking.
- The manager generates reports.

2.2 First Level DFD

Breaks down the **Hotel Management System** into major processes.

✅ **Processes:**

1. **Room Booking:** Handles reservations.
2. **Check-in/Check-out:** Manages customer arrival and departure.
3. **Billing:** Generates invoices.
4. **Report Generation:** Creates revenue and occupancy reports.

✅ **Data Flow:**

- **Input:** Customer details and booking information.
- **Output:** Room availability and invoices.

2.3 Second Level DFD

Further details the **individual processes**.

✅ **Processes:**

1. **Room Booking:**
    - Search for available rooms.
    - Confirm reservation.
2. **Check-in/Check-out:**
    - Verify customer details.
    - Update room status.
3. **Billing:**
    - Calculate charges.
    - Generate and print the bill.

✅ **Example:**
In a **Hotel System**, the second-level DFD shows detailed steps for **booking, check-in, and generating bills**.

---

3. Case Study: Library Management System

3.1 Context Level DFD

Shows the entire **Library Management System** as a **single process**.

✅ **Entities:**

- **Librarian:** Manages books and members.
- **Member:** Borrows and returns books.
- **Administrator:** Generates reports.

✅ **Data Flow:**

```css
[Member] → [Library Management System] → [Book Database]
```

- The member borrows/returns books.
- The librarian updates the records.
- The administrator generates reports.

3.2 First Level DFD

Breaks down the **Library System** into core processes.

✅ **Processes:**

1. **Book Management:** Adding and removing books.
2. **Member Management:** Handling membership records.
3. **Borrow/Return:** Managing book loans and returns.
4. **Report Generation:** Displaying library status.

✅ **Data Flow:**

- **Input:** Book details, member information.
- **Output:** Borrowing records, reports.

3.3 Second Level DFD

Detailed breakdown of **individual processes**.

✅ **Processes:**

1. **Add Book:**
    - Enter book details.
    - Store in the database.
2. **Borrow Book:**
    - Check member validity.
    - Update book status.
3. **Return Book:**
    - Verify return date.

   - Update stock.
4. **Generate Report:**
   - Filter by date, category, etc.

✅ **Example:**
In a **Library System**, the second-level DFD shows detailed steps for **adding books, borrowing, and returning**.

---

4. Case Study: Online Examination System

4.1 Context Level DFD

Shows the entire **Online Examination System** as a single process.

✅ **Entities:**

- **Student:** Takes exams.
- **Examiner:** Sets and evaluates the exam.
- **Administrator:** Manages the system.

✅ **Data Flow:**

```css
[Student] → [Online Exam System] → [Result Database]
```

- The student takes the exam.
- The examiner evaluates the results.
- The system stores the result.

4.2 First Level DFD

Breaks the system into **core processes**.

✅ **Processes:**

1. **Exam Registration:** Students register for exams.
2. **Exam Conduction:** Administering the exam.
3. **Evaluation:** Automatic/manual result generation.
4. **Result Display:** Showing final scores.

✅ **Data Flow:**

- **Input:** Student details and answers.
- **Output:** Exam results.

4.3 Second Level DFD

Detailed breakdown of **individual processes**.

✅ **Processes:**

1. **Student Registration:**
   - Enter details.
   - Store in the database.
2. **Exam Conduction:**
   - Display questions.
   - Record answers.
3. **Evaluation:**
   - Check answers.
   - Assign marks.
4. **Result Display:**
   - Generate scores.
   - Show on the student dashboard.

✅ **Example:**
In an **Online Exam System**, the second-level DFD shows detailed steps for **registration, conducting exams, and displaying results**.

---

5. Case Study: Online Shopping System

5.1 Context Level DFD

Shows the **Online Shopping System** as a **single process**.

✅ **Entities:**

- **Customer:** Purchases products.
- **Admin:** Manages inventory and sales.
- **Payment Gateway:** Handles payments.

✅ **Data Flow:**

```css
[Customer] → [Online Shopping System] → [Payment Database]
```

- The customer places an order.
- The system processes the payment.
- The admin manages orders and inventory.

5.2 First Level DFD

Breaks the system into **core processes**.

✅ **Processes:**

1. **User Registration:** Create customer accounts.
2. **Product Management:** Add and update products.
3. **Order Processing:** Handle orders and payments.
4. **Invoice Generation:** Generate bills.

✅ **Data Flow:**

- **Input:** Customer details, product info.
- **Output:** Order confirmation, invoice.

✅

System Analysis and Design – Unit-Wise Summary

Unit - I: System Concepts and Methodology

1. System Analysis Fundamentals

- **System:** A collection of interrelated components that work together to achieve a goal.
- **System Analysis:** The process of studying and understanding the existing system to identify improvements.
- **System Design:** The process of creating a new system or modifying an existing one.
- **Types of Systems:**
  - **Physical:** Tangible systems (e.g., machines).
  - **Abstract:** Conceptual systems (e.g., mathematical models).
  - **Open:** Interacts with the environment.
  - **Closed:** Does not interact with the environment.
- **System Analyst Role:**
  - Gathers requirements.
  - Designs solutions.
  - Acts as a bridge between users and developers.

---

2. System Development Strategies

- **SDLC (System Development Life Cycle):**
  - **Phases:**
    1. Requirement Analysis
    2. System Design
    3. Implementation
    4. Testing
    5. Deployment
    6. Maintenance
- **Difference between System Analysis and Design:**
  - **Analysis:** Defines requirements (**What** to do).
  - **Design:** Creates the system (**How** to do it).
- **Structured Analysis Development Method (SSADM):**
  - A systematic approach with phases:
    1. Feasibility Study
    2. Requirements Analysis
    3. Logical Design