

## **EXPERIMENT NO.:04**

**Date of Performance:**

**Date of Submission:**

**Aim:** Structured Data Flow Analysis

### **Software Used: Star UML**

Theory: Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user.

**It has following attributes –**

- It is graphic which specifies the presentation of application.
- It divides the processes so that it gives a clear picture of system flow.
- It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
- It is an approach that works from high-level overviews to lower-level details.

### **Structured Analysis Tools**

**During Structured Analysis, various tools and techniques are used for system development. They are –**

- Data Flow Diagrams
- Data Dictionary
- Decision Trees
- Decision Tables
- Structured English
- Pseudo code

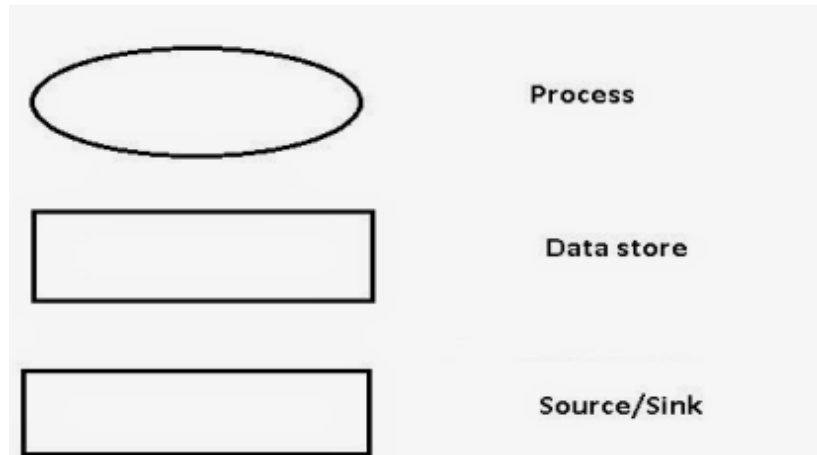
### **Data Flow Diagrams (DFD)**

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

- It shows the flow of data between various functions of system and specifies how the current system is implemented.
- It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
- Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.
- It gives an overview of what data a system processes, what transformations are

performed, what data are stored, what results are produced and where they flow.

### **Symbols used in DFD**





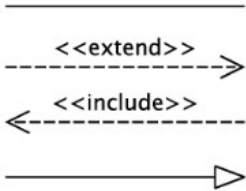
### **Used Case Diagram**

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

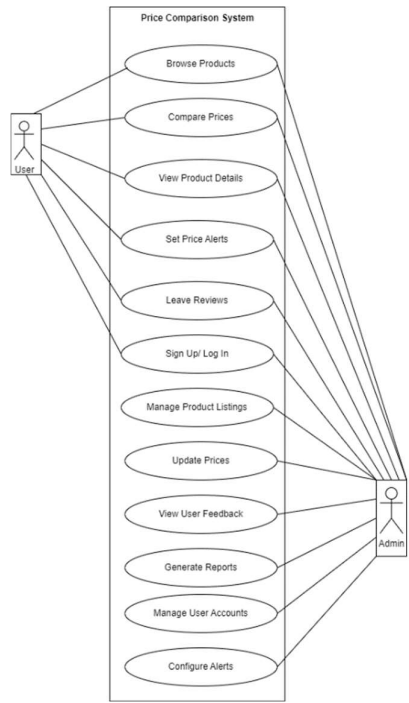
- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors) achieve
- The scope of your system

#### **UML use case diagrams are ideal for:**

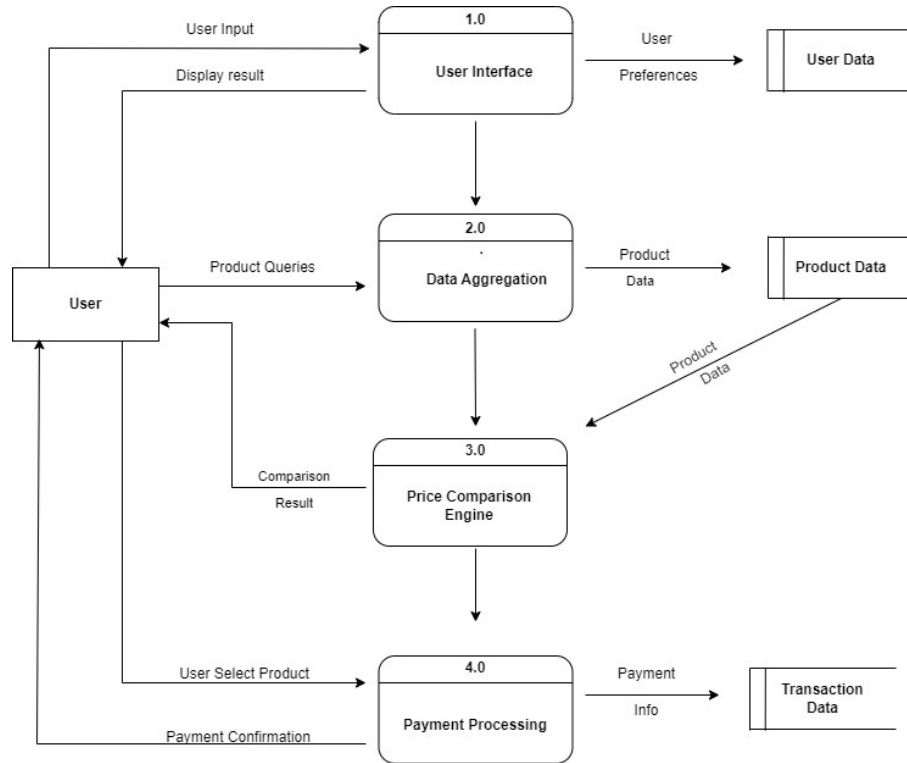
- Representing the goals of system-user interactions
- Defining and organizing functional requirements in a system
- Specifying the context and requirements of a system
- Modeling the basic flow of events in a use case

Symbol	Reference Name
	Actor
	Use case
	Relationship

Use Case Diagram



## Data Flow Diagram



**Conclusion:** Thus we performed Structured Data Flow Analysis

**Sign and Remark:**

R1	R2	R3	Total Marks	Signature
(5)	(5)	(5)	(15)	