EXPERIMENT - 3

Software Engineering Lab

Aim

To perform the function oriented diagram: Data Flow Diagram (DFD), ER Diagram and Structured chart.

Syeda Reeha Quasar

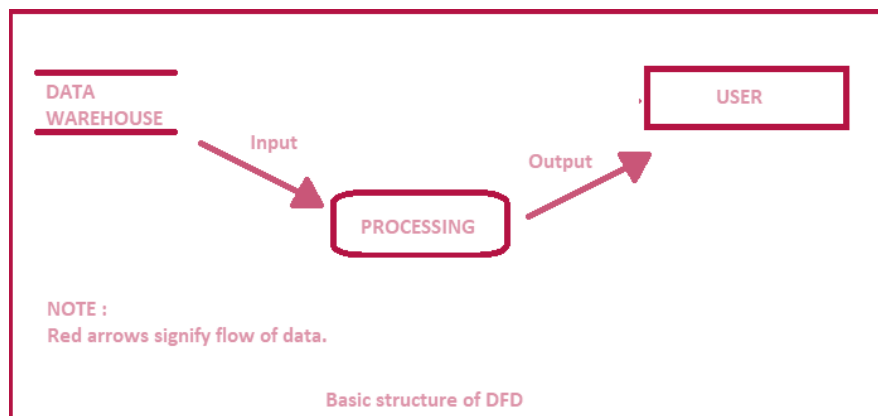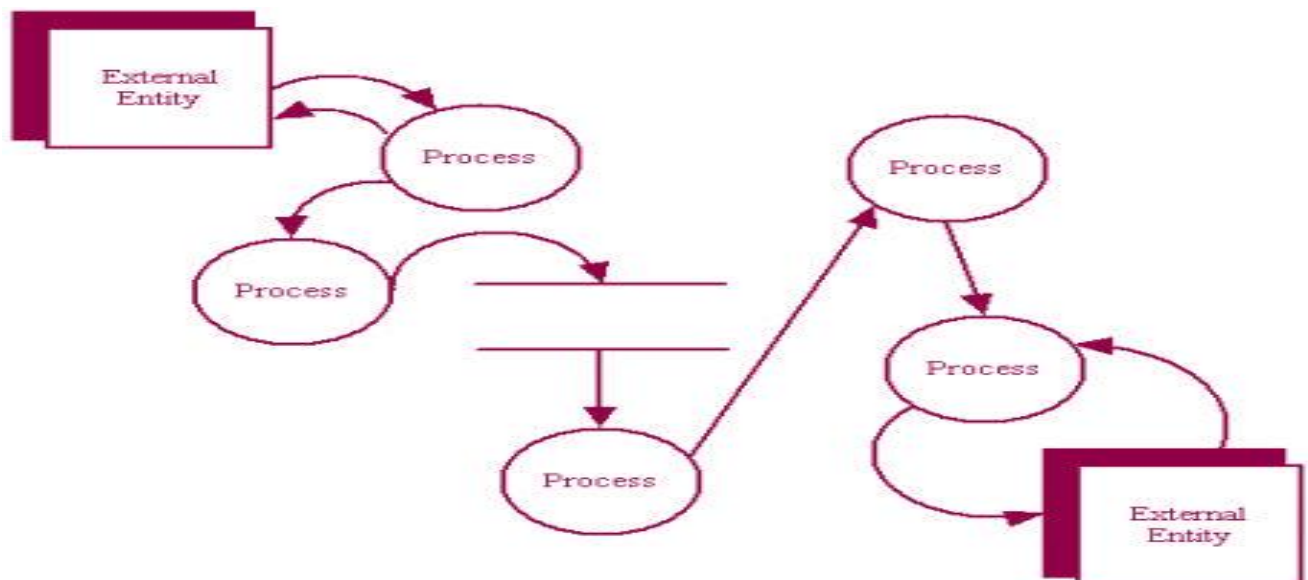14114802719

4C7

EXPERIMENT – 3

Aim:

To perform the function-oriented diagram: Data Flow Diagram (DFD), ER Diagram and Structured chart.

Theory:

Data flow diagrams are versatile diagramming tools. With only four symbols, data flow diagrams can represent both physical and logical information systems. The four symbols used in DFD representation are data flows, data stores, processes, and sources / sinks (or external entities).

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs.

Basic structure of DFD

Symbols of DFD:

| Name	Symbol	Description	Example
Entity		Used to represent people and organizations outside the system. They either input information to the system, accept output information from the system or both	Customer
Process		These are actions that are carried out with the data that flows around the system. A process accepts input data and produces data that it passes on to another part of the DFD	Verify Order
Data Flow		These represent the flow of data to or from a process	Customer Details
Data Store		This is a place where data is stored either temporarily or permanently	Products

Data Flow Diagram Layers

Draw data flow diagrams in several nested layers. A single process node on a high-level diagram can be expanded to show a more detailed data flow diagram. Draw the context diagram first, followed by various layers of data flow diagrams.
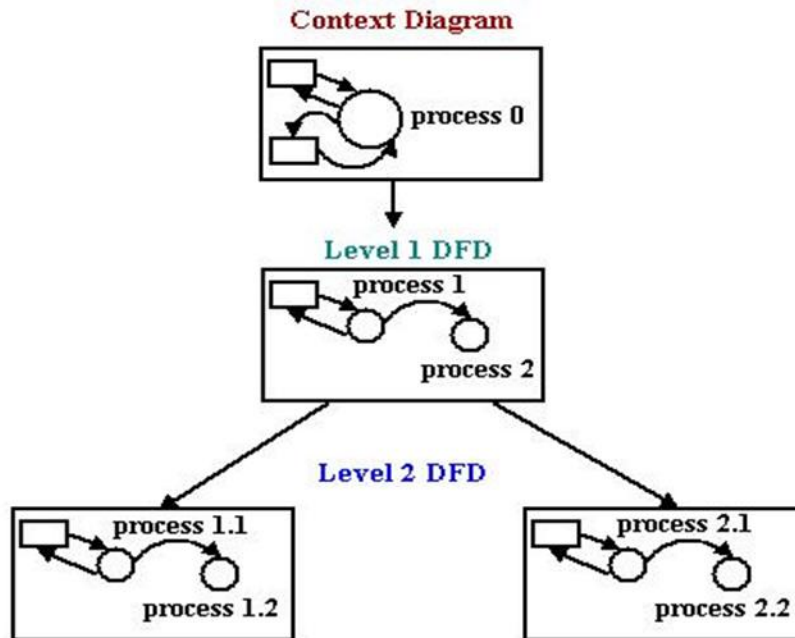
ER Diagram:

An entity relationship model, *also called an* entity-relationship (ER) diagram, *is a* graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.

Relationships Between Entities

A relationship is how the data is shared between entities. There are three types of relationships between entities:

1. One-to-One

One instance of an entity (A) is associated with one other instance of another entity (B). For example, in a database of employees, each employee name (A) is associated with only one social security number (B).
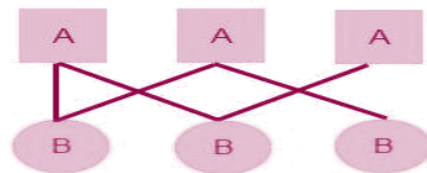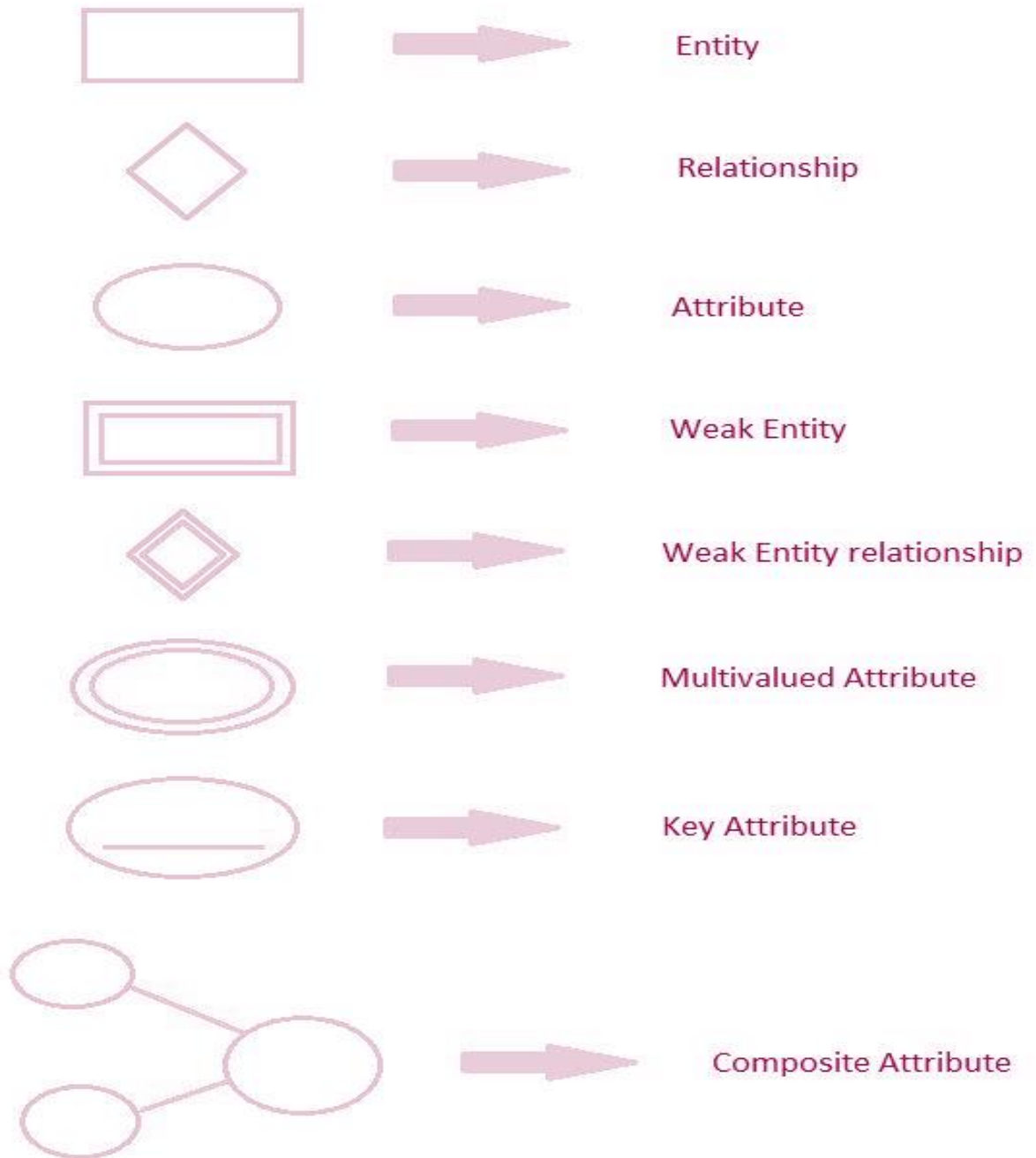
2. One-to-Many

One instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A. For example, for a company with all employees working in one building, the building name (A) is associated with many different employees (B), but those employees all share the same singular association with entity A.
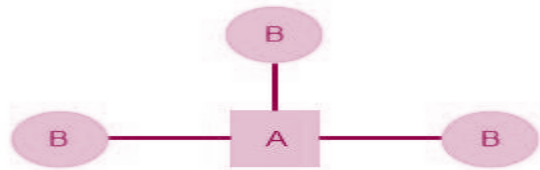
3. Many-to-Many

One instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity A. For example, for a company in which all of its employees work on multiple projects, each instance of an employee (A) is associated with many instances of a project (B), and at the same time, each instance of a project (B) has multiple employees (A) associated with it.

Symbols used to draw ER diagram:

▭	Entity
◇	Relationship
⬭	Attribute
▭ (double rectangle)	Weak Entity
◇ (double diamond)	Weak Entity relationship
⬭ (double ellipse)	Multivalued Attribute
⬭ (underlined)	Key Attribute
⬭–⬭ (connected ellipses)	Composite Attribute

A **Structure Chart** (SC) in software engineering is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The lines represent the connection and or ownership between activities and sub activities as they are used in organization charts. The tree structure visualizes the relationships between modules.

Performance Instructions:

To draw DFD

1. Identify various processes, data store, input, output etc. of the system and analyse it.

2. Use processes at various levels to draw the DFDs.

To draw ER Diagram

1. Identify all entities and their attributes.

2. Identify weak entities, attributes and their identifying relationship.

3. Design ER diagram according to norms.
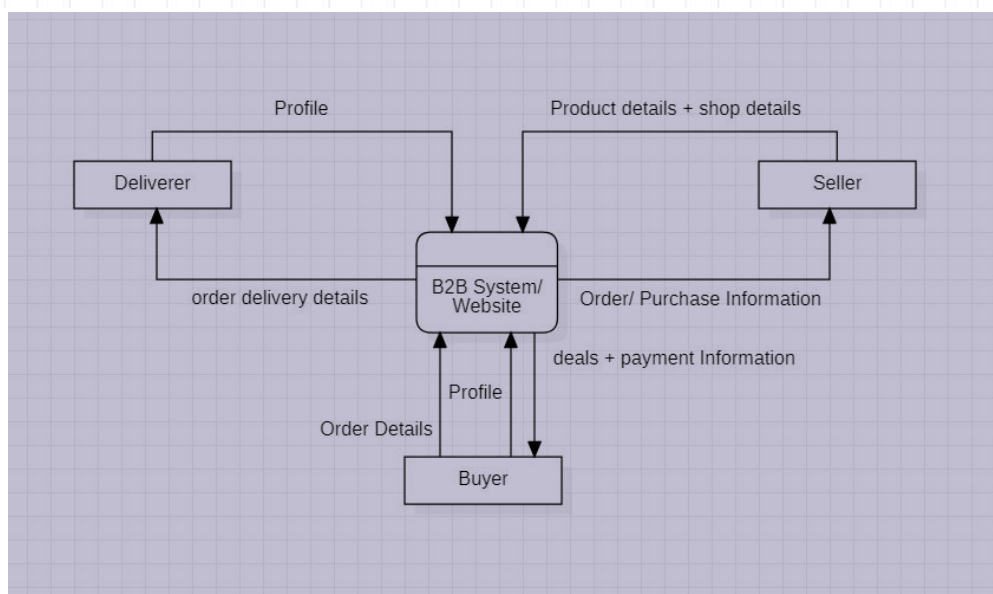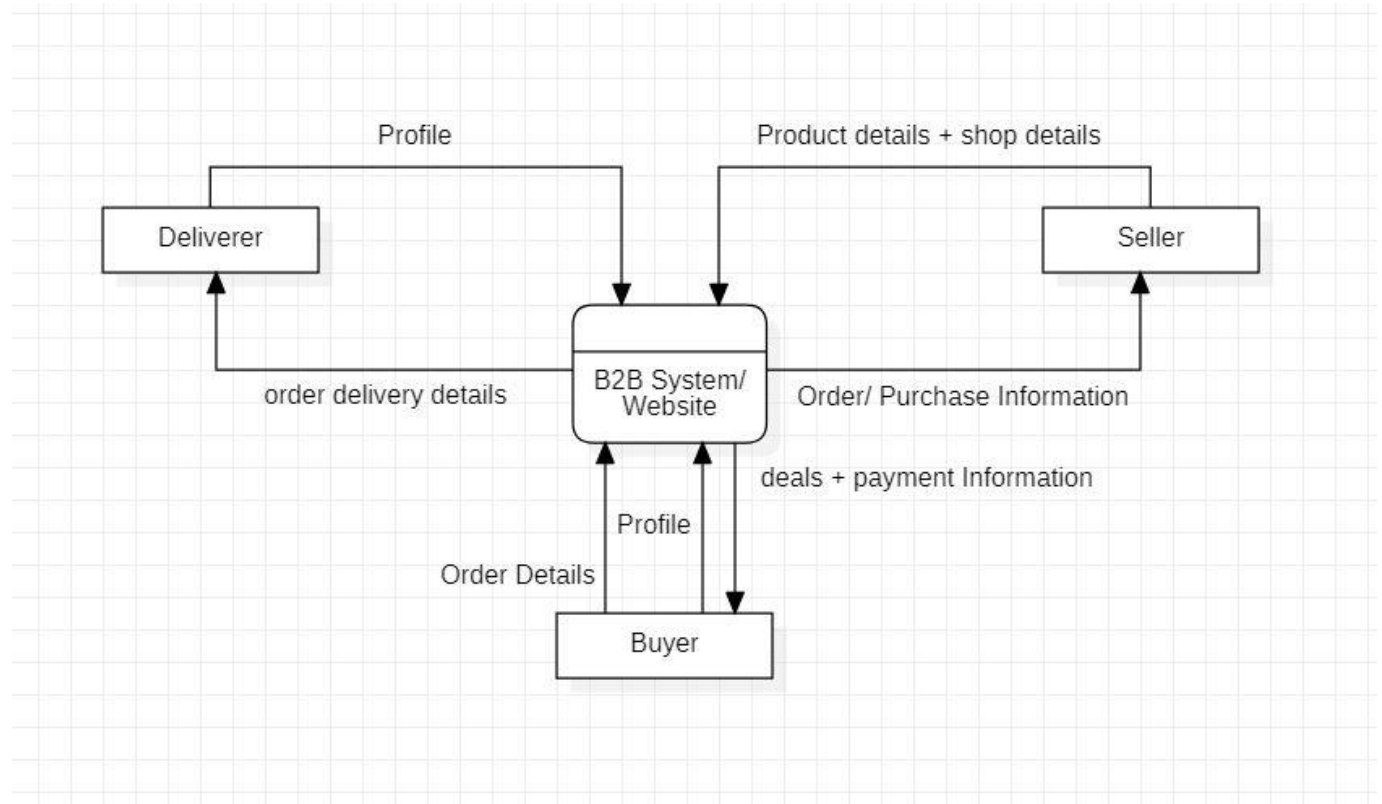
4. Imply cardinalities on diagram.
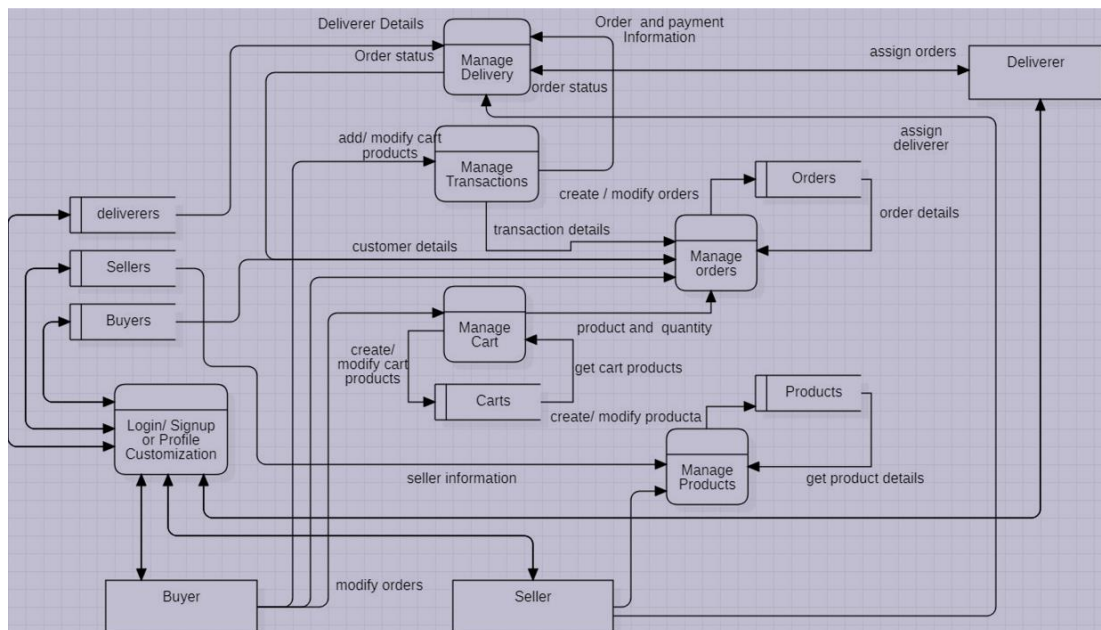
To draw structured Chart Diagram

1. Identify various modules, input, output etc. of the system.

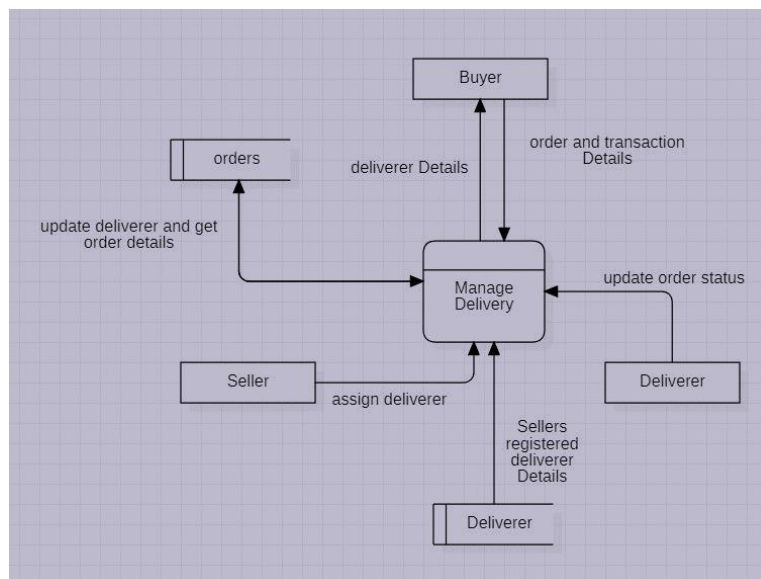2. Draw structured chart diagram describing it in form of levels.
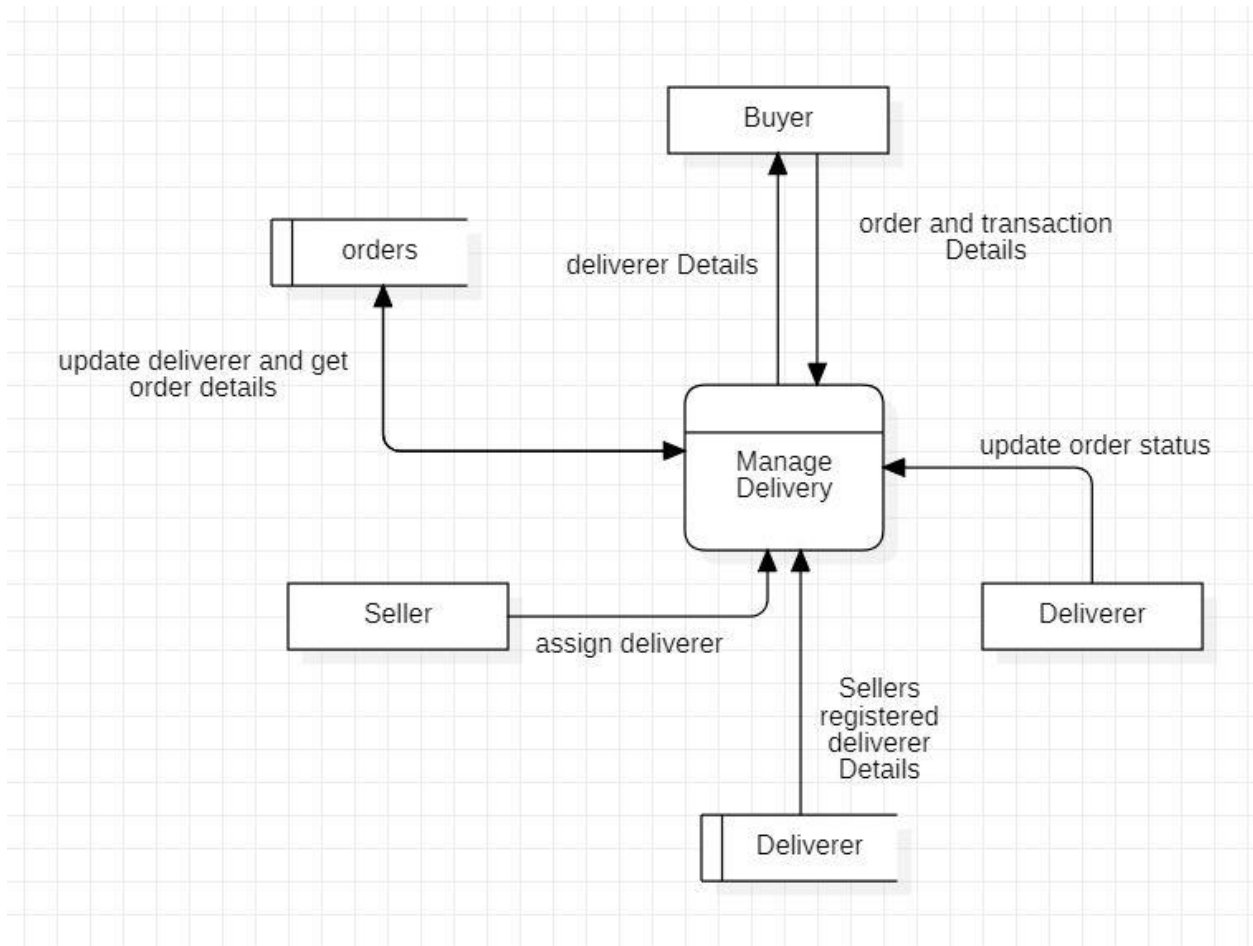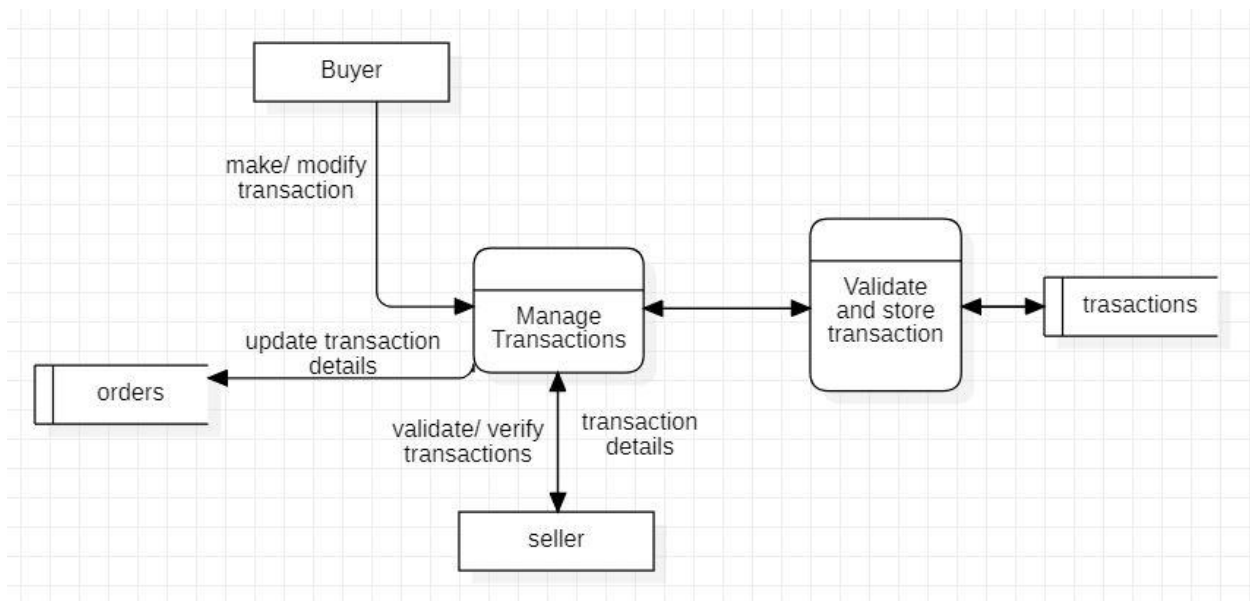
OUTPUT:

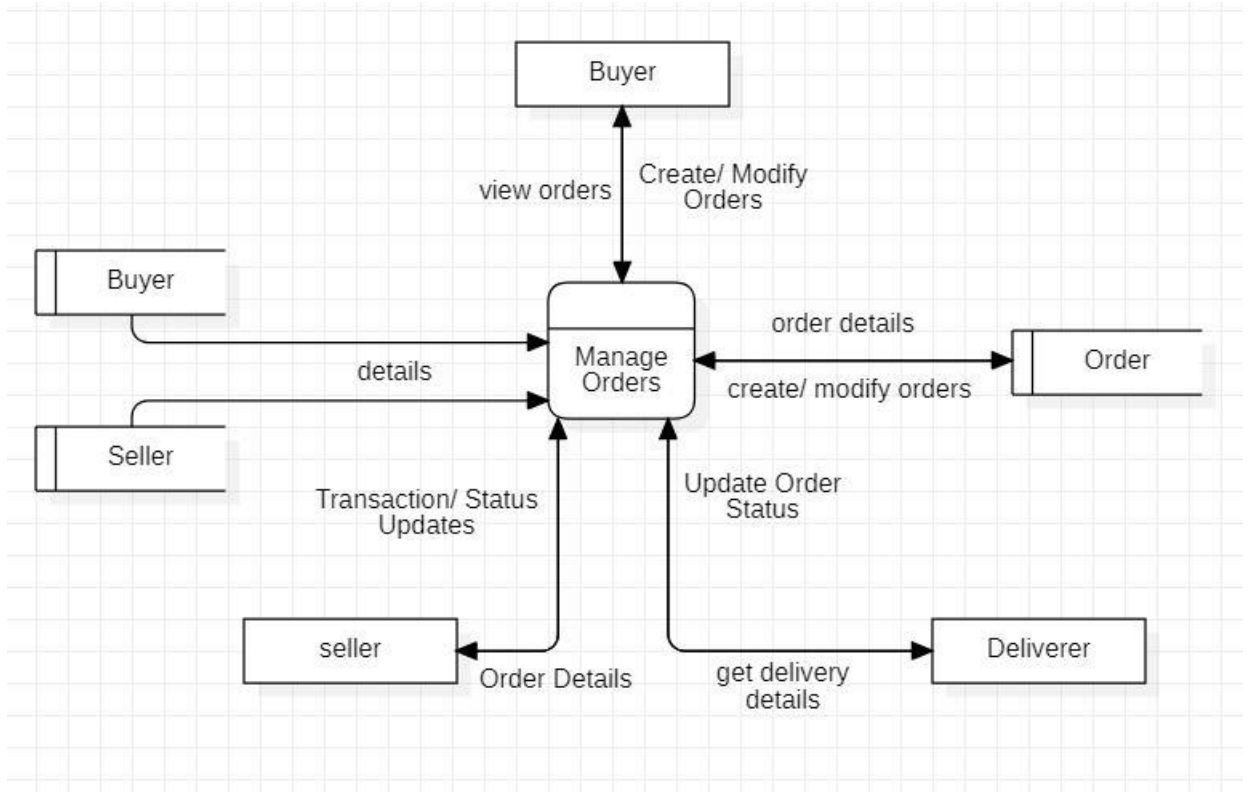## DFD (Data Flow Diagram):

Level 0:

Level 1:

Level 2.1:

Level 2.2:

Level 2.3:

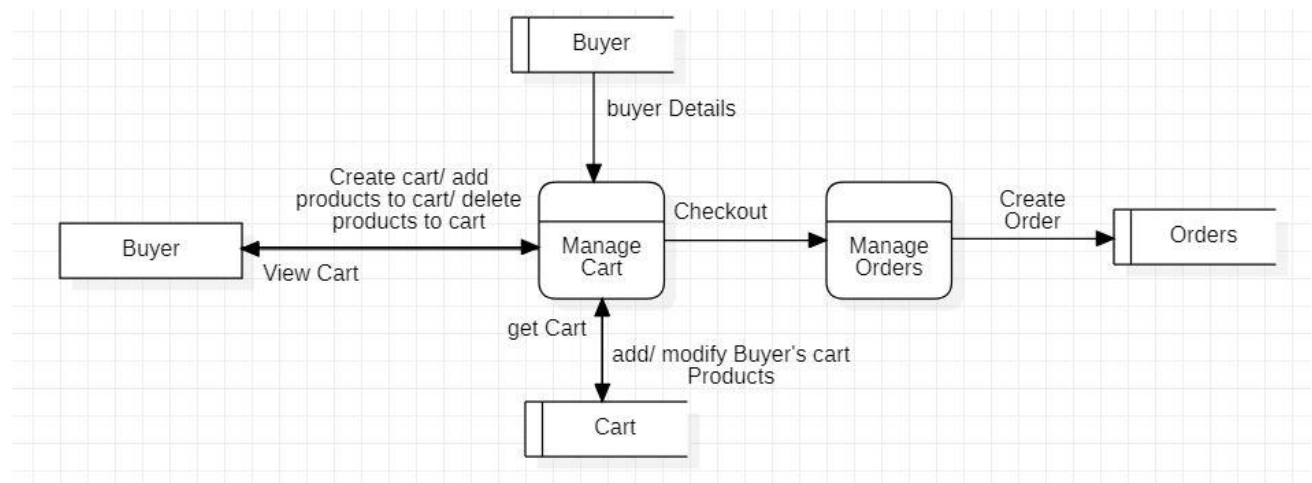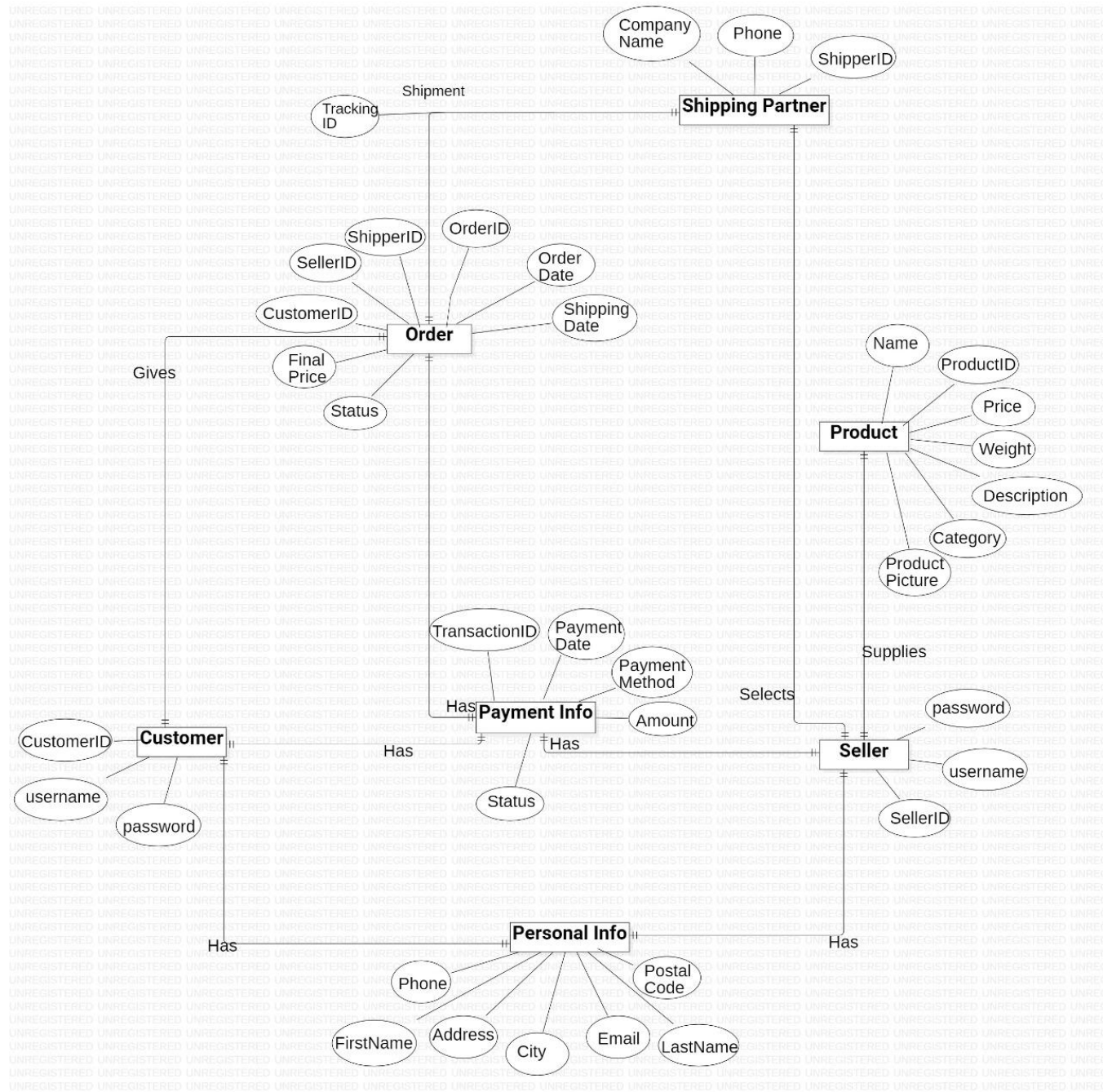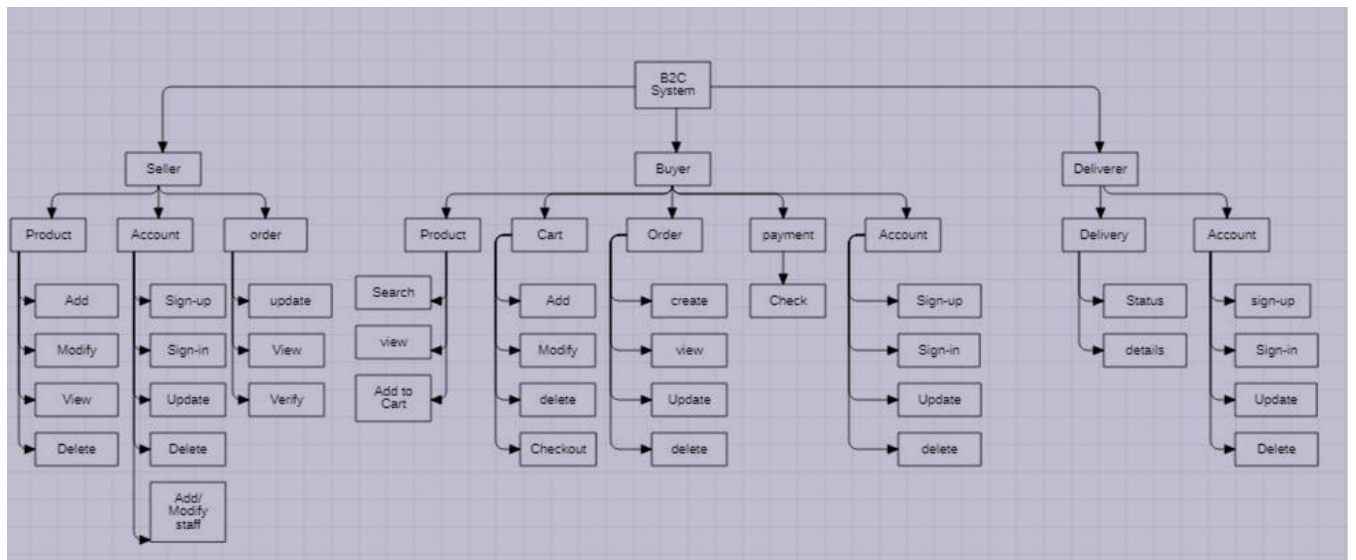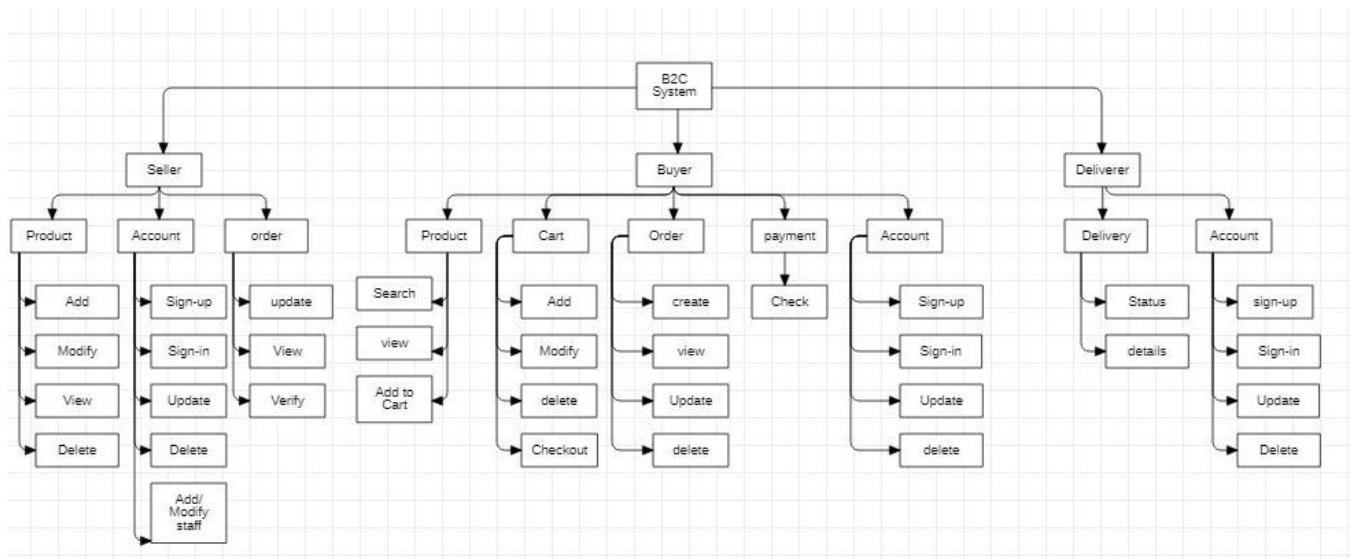## Level 2.4:
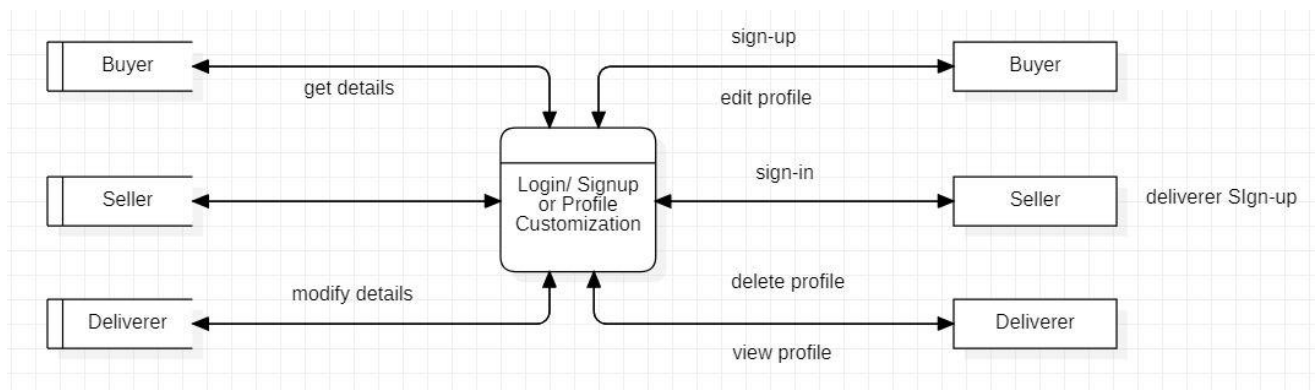
Level 2.5:
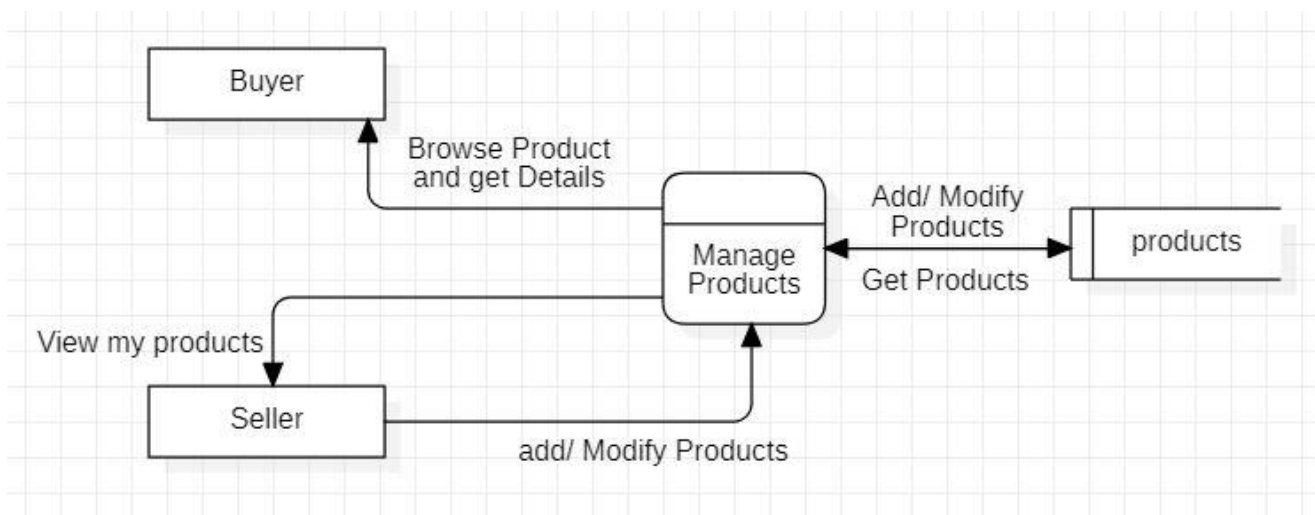
Level 2.6:

ER (Entity Relation) Diagram:

Structured Diagram:

Viva Questions

1. Define DFD? What are different levels of DFD?

Ans.

Data flow diagrams are versatile diagramming tools. With only four symbols, data flow diagrams can represent both physical and logical information systems. The four symbols used in DFD representation are data flows, data stores, processes, and sources / sinks (or external entities).

0-level DFD:
It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble

1-level DFD:
In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system

2-level DFD:
2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

2. Describe symbols used for constructing DFDs?

Ans.

The four symbols used in DFD representation are data flows, data stores, processes, and sources / sinks (or external entities).

3. Distinguish between a data flow diagram and a flow chart with example?

Ans.

The main difference between DFD and Flowchart is that DFD is a graphical diagram that represents the data flow of a system while a flowchart is a graphical diagram that represents the sequence of steps to solve a problem.

4. Explain structured chart diagram?

Ans.

A Structure Chart (SC) in software engineering is a chart which shows the breakdown of a system to its lowest manageable levels.

5. Describe symbols used for constructing structured chart diagram?

Ans.

Module - It represents process or subroutine or task. A control module branches to more than one sub-module. Library Modules are re-usable and invokable from any module.

Condition - It is represented by small diamond at the base of module. It depicts that control module can select any of sub-routine based on some condition.

Jump - An arrow is shown pointing inside the module to depict that the control will jump in the middle of the sub-module.

Loop - A curved arrow represents loop in the module. All sub-modules covered by loop repeat execution of module.

Data flow - A directed arrow with empty circle at the end represents data flow.

Control flow - A directed arrow with filled circle at the end represents control flow.

6. Explain ER diagram?

Ans.

An entity relationship model, *also called an* entity-relationship (ER) diagram, *is a* graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems.

7. What is entity? Explain strong and weak entity?

Ans.

Entity : An entity is an object or component of data. An entity is represented as rectangle in an ER diagram. The entity set which does not have sufficient attributes to form a primary key is called as Weak entity set. An entity set that has a primary key is called as Strong entity set.

8. Explain structured chart diagram?

Ans.

Attributes are the properties which define the entity type. They are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.