******

***University of mustansiriyah /College of Education***

***Computer Science Department***

***Software Engineering 3rd Class***

### Topic:

**Chapter 4**

**Formal Requirements**

**منهج المتطلبات**

### **Analysis Model**

* 1. **Analysis Model Objectives and Elements of Analysis Model**
  2. **Data Model**
  3. **Creating an Entity/Relationship Model**
  4. **Creating a Data Flow Model**

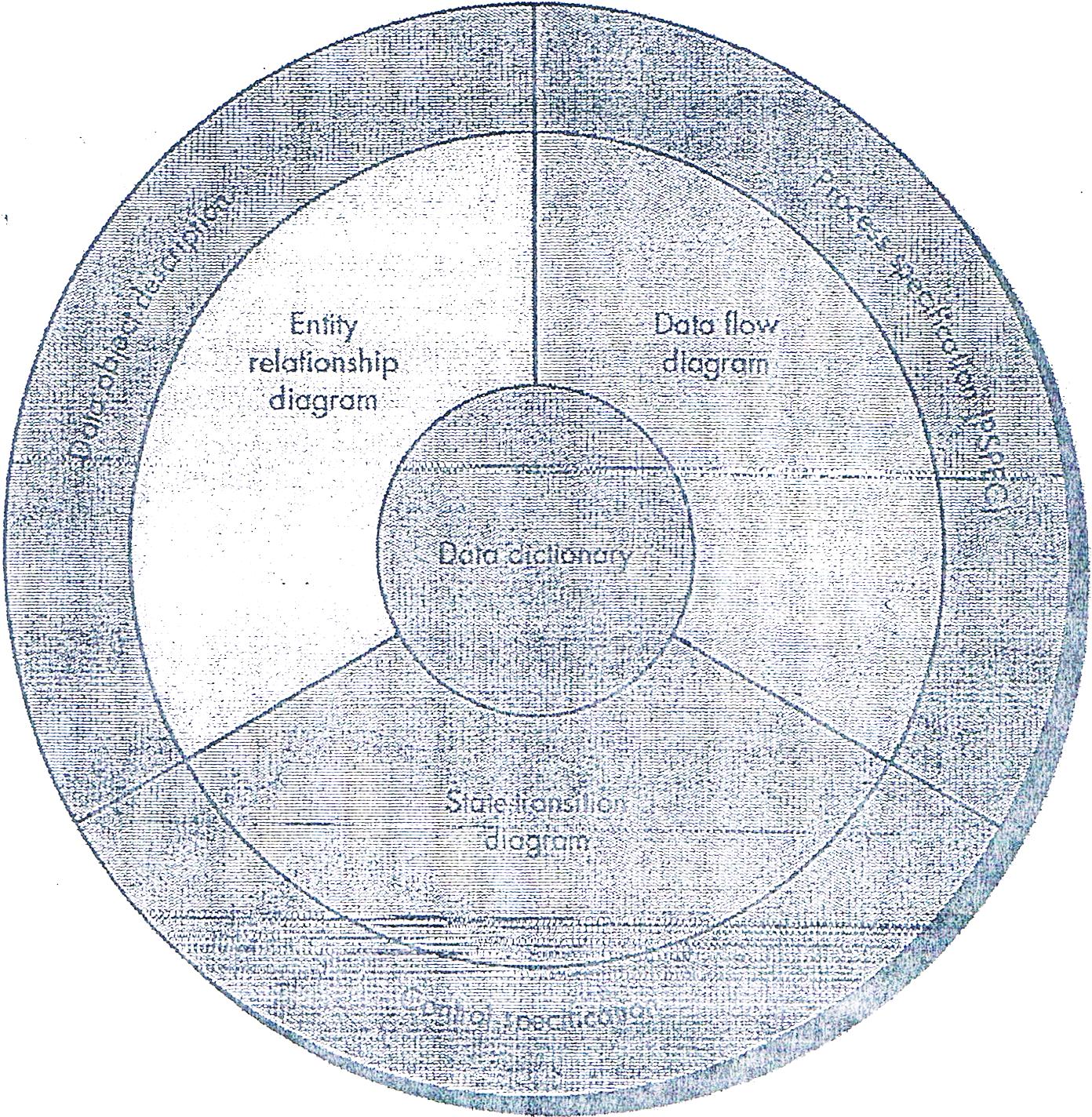
4.1 Analysis Model

The *analysis model,* actually a set of models, is the first technical representation of a system. Over the years many methods have been proposed for analysis modeling, however, two now dominate. The first, *structured analysis* is a classical modeling method and is described in this chapter. The other approach, *object oriented analysis.*

## 4.2 Analysis Model Objectives:

The analysis model must achieve three primary objectives:

1. to describe what the customer requires,
2. to establish a basis for the creation of a software design, and
3. to define a set of requirements that can be validated once the software is built. To accomplish these objectives, the analysis model derived during structured analysis takes the form illustrated in Figure (4.1)



## Figure 4.1: The structure of the analysis model

**Elements of Analysis Model (Components for structure of the analysis model)**

### 1- Data dictionary

2- Entity relation diagram (ERD)

3- Data flow diagram (DFD)

4- State transition diagram (STD)

At the core of the model lies the *data dictionary:* is an organized listing of all data elements that ''are pertinent to the system, with precise, rigorous definitions so that both user and system analyst will have a common understanding of inputs, outputs, components of stores and [even] intermediate calculations.

Or *data dictionary* a repository that contains descriptions of all data objects consumed or produced by the software

Three different diagrams surround the the core:.

1- The *entity relationship diagram* (ERD) depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data object noted in the ERD can be described using a data object description.

2- The *data flow diagram* (DFD) : is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble chart.,

**Serves two purposes:**

(1) to provide an indication of how data are transformed as they move through the system and (2) to depict the functions (and sub functions) that transform the data flow. The DFD provides additional information that is used during the analysis of the information domain and serves as a basis for the *modeling of function.* A description of each function presented in the DFD is contained in a *process specification* (PSPEC).

DFD are used to represent data and the processes that manipulate it.

1. *State transition diagram (STD ):* By studying the STD, a software engineer can determine the behavior of the system and, more important, can ascertain whether there are "holes" in the specified behavior. STD is a behavioral model

The STD indicates how the system behaves as a consequence of external events. To accomplish this, the STD represents the various modes of behavior (called *states)* of the system and the manner in which transitions are made from state to state. The STD serves as the basis for behavioral modeling. Additional information about the control aspects of the software is contained in the *control specification* (CSPEC).

Data ob ject Description: The attributes of each data object noted in the ERD can be described using a data object description.

In other word: incorporates the data object and all of its attributes.

Process Specification (PSPEC): is used to describe all flow model processes that appear at the final level of refinement. The content of the process specification can include narrative text, a *program design language* (PDL) description. of the process algorithm, mathematical equations, tables, diagrams, or charts.

The control specification (CSPEC): represents the behavior of the system in two different ways. The CSPEC contains a state transition diagram that is a sequential specification of behavior. It can also contain a program activation table-a combinatorial specification of behavior.

CSPEC is used to indicate (1) how the software behaves when an event or control signal is sensed and (2) which processes are invoked as a consequence of the occurrence of the event.



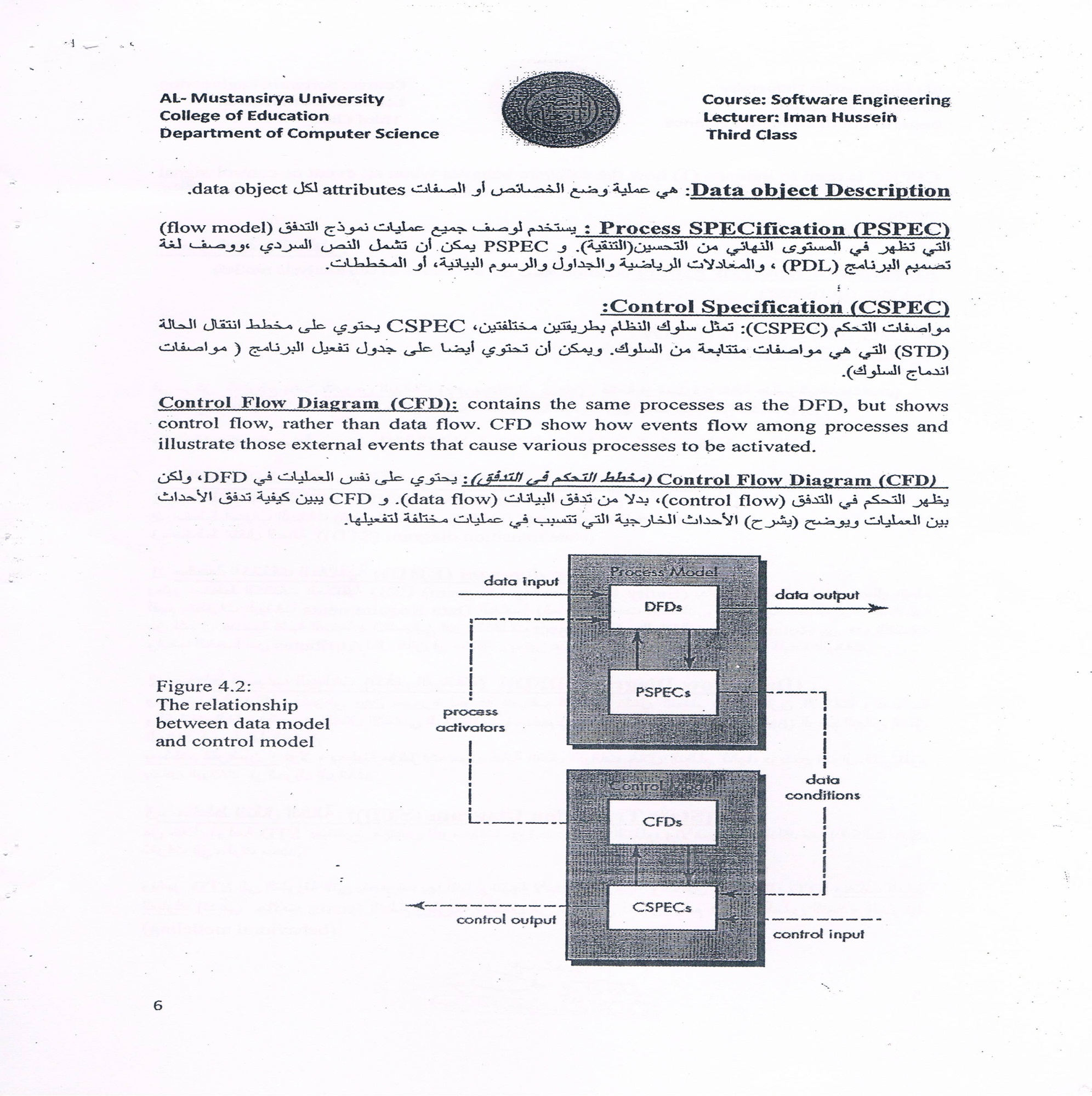
Elements of Analysis Model (Components for structure of the analysis model)

#### Data dictionary

* 1. Entity relation diagram (ERD)
  2. 3- Data flow diagram(DFD)

4- State transition diagram (STD)

Control Flow Diagram (CFD): contains the same processes as the DFD, but shows control flow, rather than data flow. CFD show how events flow among processes and illustrate those external events that cause various processes to be activated.



**Figure 4.2 : The relationship between data model and control model**

* 1. **4.3 Data Model** :

The data model consists of three interrelated pieces of information:

1- Data object (inpu.t and output from a system)

2- Attributes that describe the data object

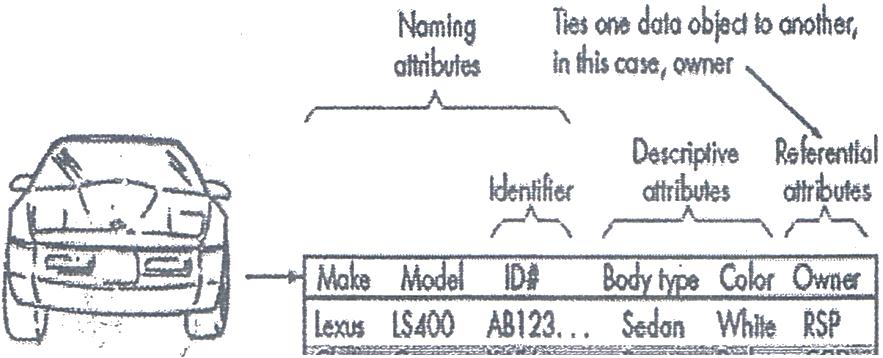
3- Relationships that connect data objects to one another.

1- Data object : A data object is a representation of almost any composite information that must be understood by software. By composite information, we mean something that has a number of different properties or attributes. A data object can be an external entity (e.g., anything that produces or consumes information), a thing (e.g., a report or a

display), an occurrence (e.g., a telephone call) or event (e.g., an alarm), a role (e.g., salesperson), an organizational unit (e.g., accounting department), a place (e.g., awarehouse), or a structure (e.g., a file). A data object encapsulates data only-there is no reference within a data object to operations that act on the data.

1. Attributes: Attributes define the properties of a data object and take on one of three different characteristic$. They can be used to (I) name an instance of the data object, (2) describe the instance, or (3) make reference to another instance in another table.

- In addition, one or more of the attributes must be defined as an identifier-that is, the identifier attribute becomes a "key" when we want to find an instance of the data object. In some cases, values for the identifier(s) are unique, although this is not a requirement.



,,.,

- - ;=- - • --'

BiWN 750il *W65...* Coupe While IJL

Ford Taurus Q12A45... Sedon Blue » BLF Blf

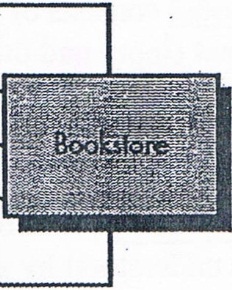
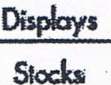
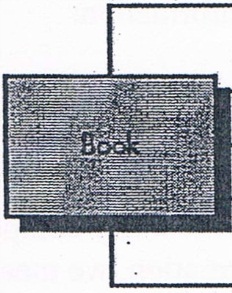
1. Relationships:

Data objects are connected to one another in different ways. Relationships define a set of object/relationship pairs that define the relevant relationships. It is important to note that object/relationship pairs are bidirectional. That is, they can be read in either direction. A bookstore orders books or books are ordered by a bookstore.

There are two basic points in the Relationships be aware of:

**1- Cardinality 2- Modality**

-



Sells

Returns

1- Cardinality: the specification of the number of occurrences of one [object] that can be related to the number of occurrences of another [object]. Cardinality is usually expressed as simply 'one' or 'many.'

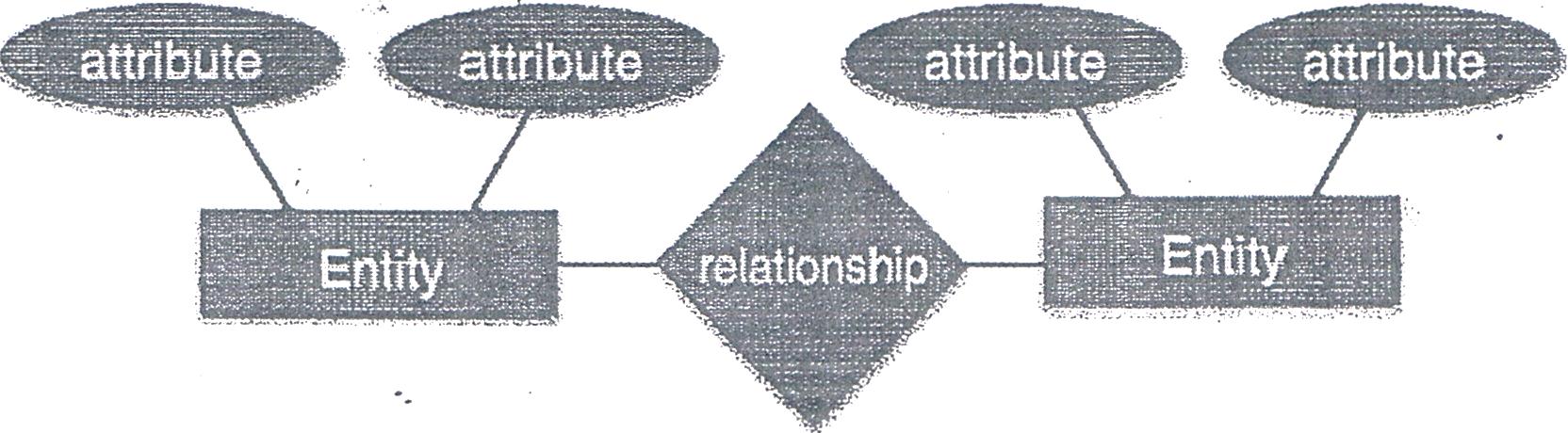
Taking into consideration all combinations of 'one' and 'many' two [objects] can be related as:

* + One-to-one (1:1) An occurrence of [object] 'A' can relate: to one and only one occurrence of [object] 'D,' and an occurrence of 'D' can relate to only one occurrence of 'A.'
  + One-to-many (l: N)-One occurrence of [object] 'A' can relate to one or many occurrences of [object] 'B,' but an occurrence of 'B' can relate to only one occurrence of 'A.' For example, a mother can have many children, but a child can have only one mother.
  + Many-to-many (M: N)--An occurrence of [object] 'A' can relate to one or more occurrences of 'B,' while an occurrence of 'B' can relate to one or more occurrences of 'A.' For example, an uncle can have many nephews, while a nephew can have many uncles.

2- Modality: The modality of a relationship is 0 if there is no explicit need for the relationship to occur or the relationship is optional. The modality is 1 if an occurrence of the relationship is mandatory. ·

* 1. Creating an Entity/Relationship Diagram

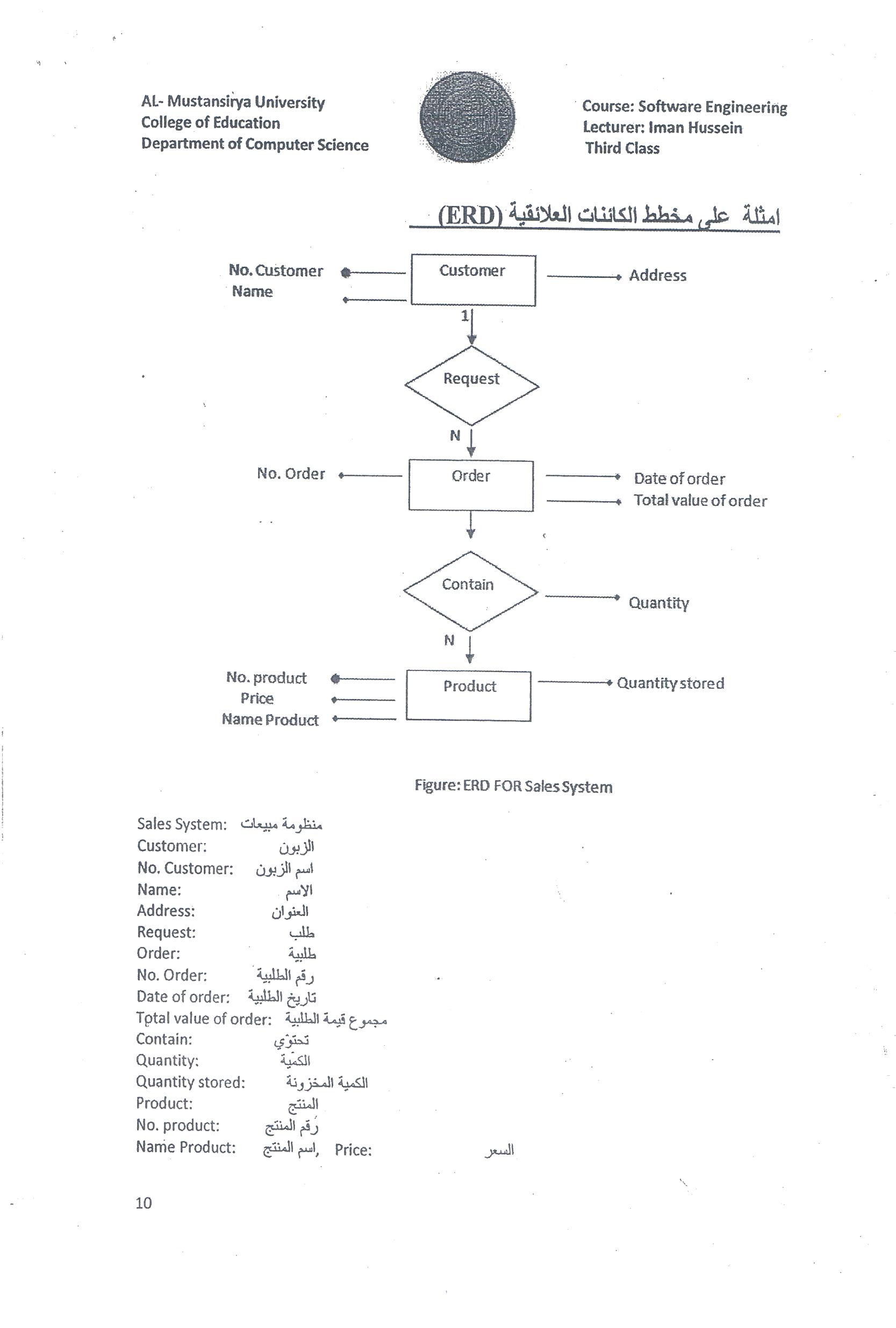
The entity/relationship diagram enables a software engineer to fully specify the data objects that are input and output from a system, the attributes that define the properties of these objects, and their relationships. Like most elements of the. Analysis model, the ERB is constructed in an iterative manner. The following approach is taken:

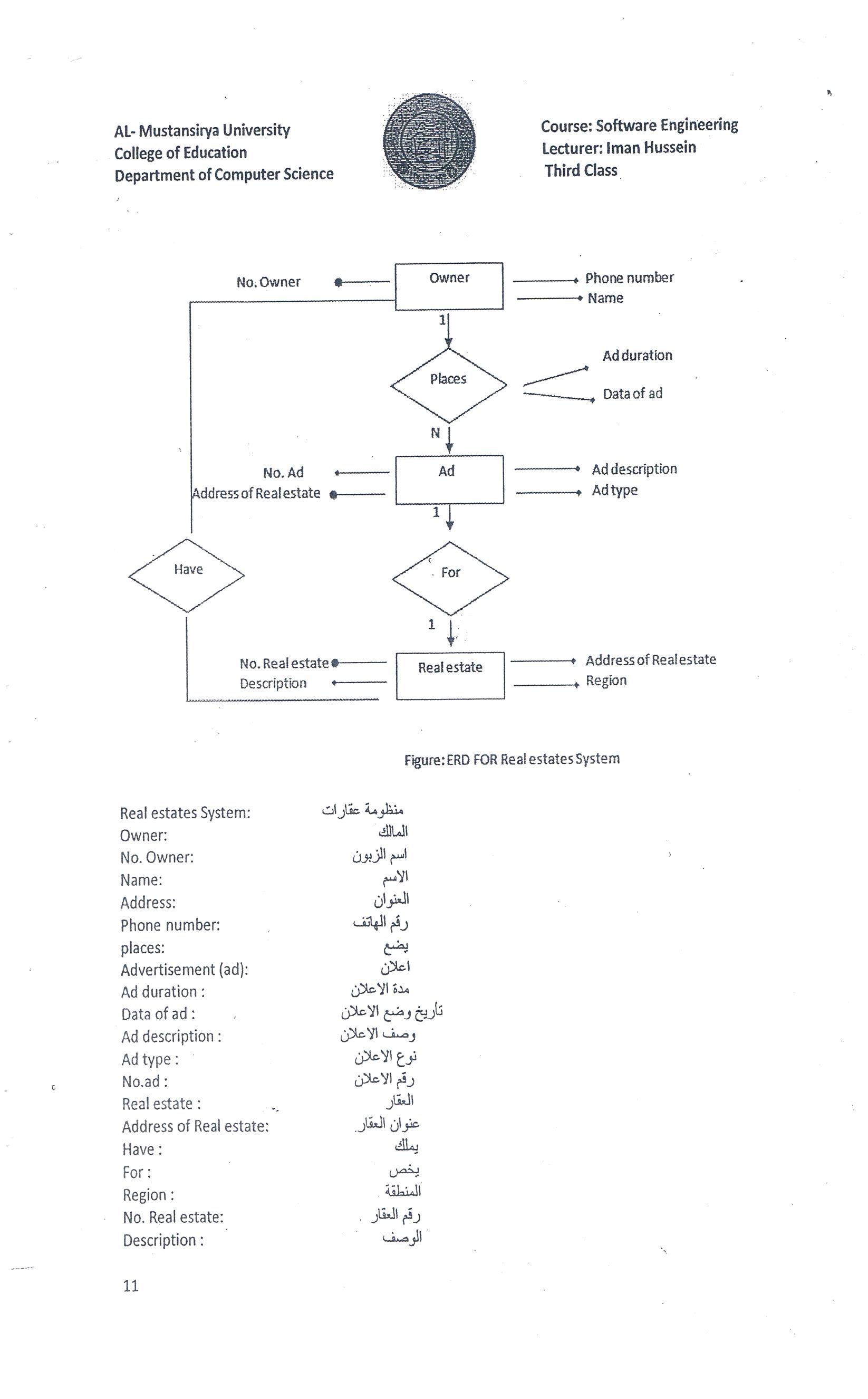


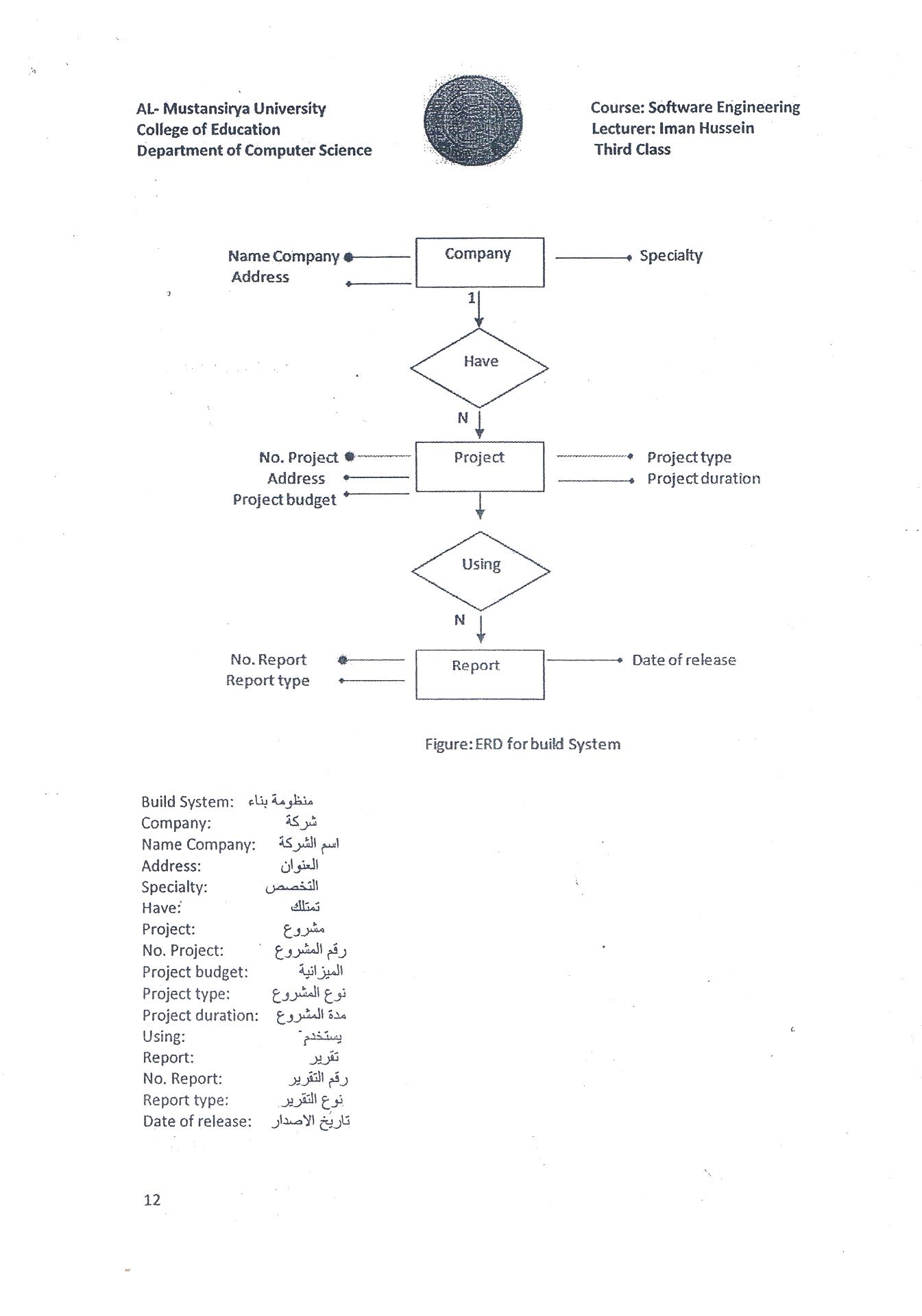
Key: مفتاح

Attribute الخاصية :

**امثلة على المخطط الكائنات العلائقية (ERD**

****





**4.5 Creating a Data Flow Model**

A few simple guidelines can aid immeasurably during derivation of a data flow diagram:

1. the level 0 data flow diagram should depict the software/system as a single bubble;
2. primary input and output should be carefully noted;
3. Refinement should begin by isolating candidate processes, data objects, and stores to be represented at the next level;
4. All arrows and bubbles should be labeled with meaningful names;
5. Information flow continuity must be maintained from level to level, and
6. One bubble at a time should be refined. There is a natural tendency to overcomplicate the data flow diagram.

**DFD Components**

DFD can represent Source, destination, storage and flow of data using the following set of components -

**Data Flow**

Entity

**Data Store**

* **Entities** - Entities are source and destination of information data.

Entities are represented by rectangles with their respective names.

* **Process** - Activities and action taken on the data are represented by Circle or Round-edged rectangles.
* **Storage** - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.
  + **Data Flow** - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of



Levels of DFD

(On line shopping system) نظام التسوق عبر الانترنيت

* Level 0 - Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.

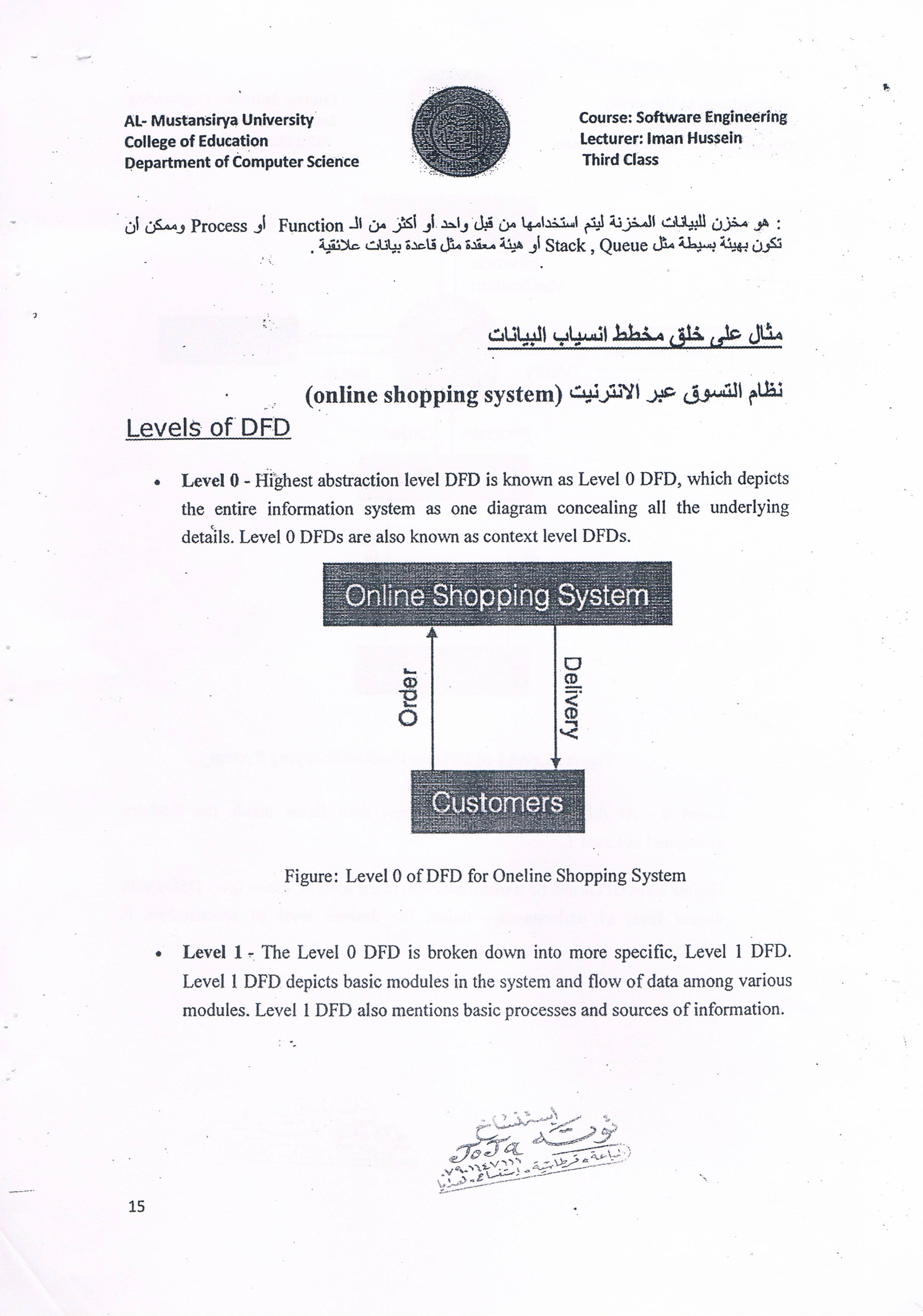


Figure: Level 0 of DFD for One line Shopping System

* Level 1- The Level 0 DFD is broken down into more specific, Level 1 DFD.

Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level IDFD also mentions basic processes and sources of information.

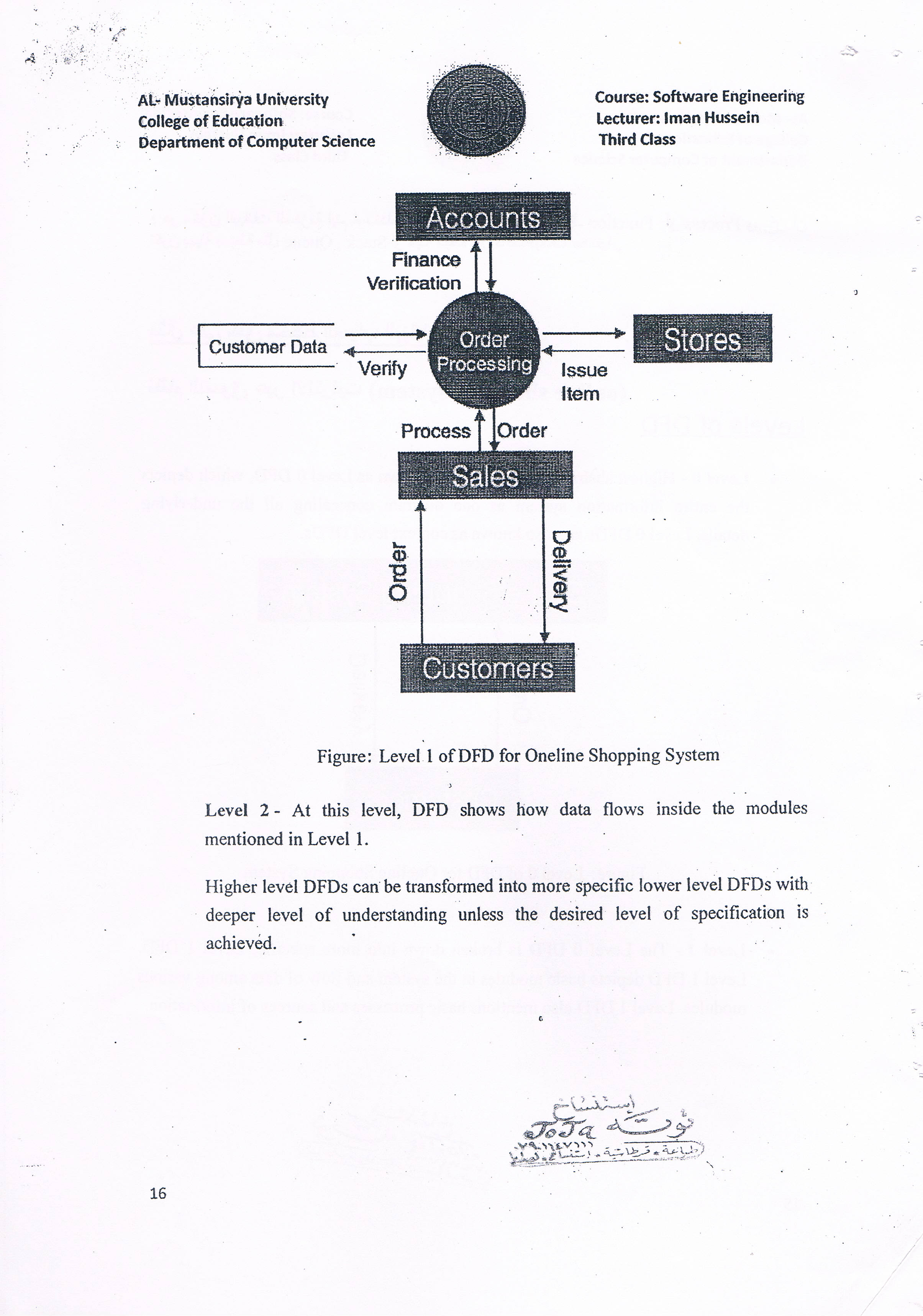


Figure: Level 1 of DFD for One line Shopping System

Level 2 - At this level, DFD shows how data flows inside the modules mentioned in Level I .

Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.