
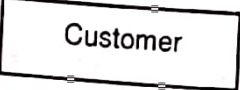
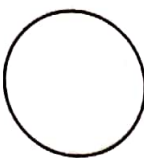



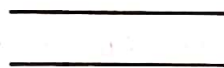
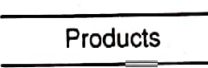


2.1 DATA FLOW DIAGRAMS (DFDs)

- A Data Flow Diagram (DFD) is a tool that depicts the flow of data through a system and the work or processing performed by that system.
- Data flow diagrams are the most commonly used way of documenting the process of current and required systems. As their name suggests they are a pictorial way of showing the flow of data into/within the system, around the system and out of a system.
- DFD is a graphical representation of flow of data within a system. Unlike, flowcharts, DFDs do not give detailed descriptions of modules but graphically describe data and how the data interact with the system.
- The DFD enable us to visualize how the system operates, its final output and the implementation of the system as a whole including modification if any.
- The Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system. It enables us to represent the processes in the information system from the viewpoint of data.
- The DFD lets us to visualize how the system operates, what the system accomplishes and how it will be implemented, when it is refined with further specification.
- Data flow diagrams are used by systems analysts to design information-processing whole organizations. A data-flow diagram also known as bubble chart or work flow diagram.
- A DFD maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, to show data inputs, outputs, storage points and the routes between each destination.
- IEEE defines DFD as, "a diagram that depicts data sources, data storage, and processes performed on data as nodes, and logical flow of data as links between the nodes."
- DFD accomplishes the following objectives:
 1. A DFD represents system data in a hierarchical manner and with required levels of detail.
 2. A DFD depicts processes according to defined user requirements and software scope.
- DFD focus on the process that transforms incoming data flows (input) into outgoing data flows (output).

Symbols used in DFD: Following table shows symbols used in DFDs with example.

| Sr. No. | Name | Symbol | Description | Example |
|---------|--------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| 1. | Entity |  | An entity represents the source of data as input to the system. They are also the destination of system data. A system comes into action based on input signal or data it receives from an 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. | |
| 2. | Process |  | The process causes some transformation to data. 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. In other words, we can say that process transforms the input data into output data. |  |
| 3. | Data Flow |  | A data element that goes into a process as input or out of a process as output is called as data flow. These represent the flow of data to or from a process. In other words, we can say that data flow represents the movement of data from one component to the other. |  |
| 4. | Data Store |  | This is a place where data is stored either temporarily or permanently. Data elements stored in a data file are called data store. Data stores represent stores of data within the system, for example, computer files or databases. |  |

2.1.1 Types of DFDs

- There are two types of DFDs, both of which support a top-down approach to systems analysis as given below:
 - Logical DFDs** are implementation-independent and describe the system, rather than how activities are accomplished. The logical DFD specify the various logical processes performed on data i.e., type of operations performed.
 - Physical DFDs** shows how the system will be implemented. A physical DFD specify who does the operations whether it is done manually or with a computer and also where it is done.
- Fig. 2.1 shows basic notations in DFDs.

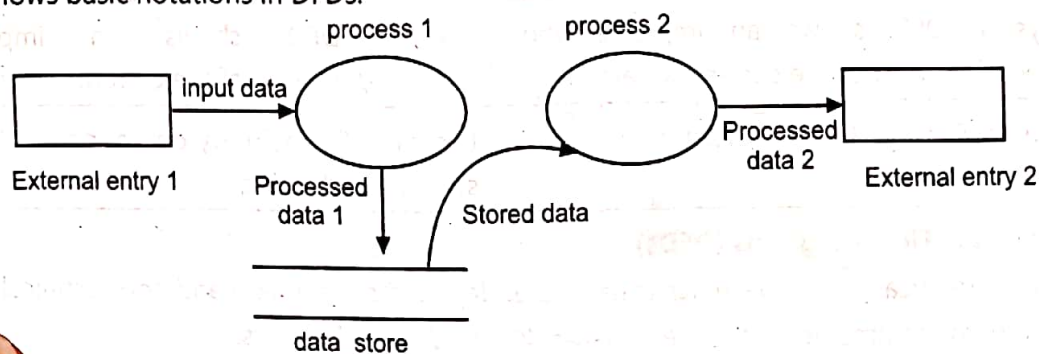


Fig. 2.1

- Physical DFD to issue a book from library is shown in Fig. 2.2.

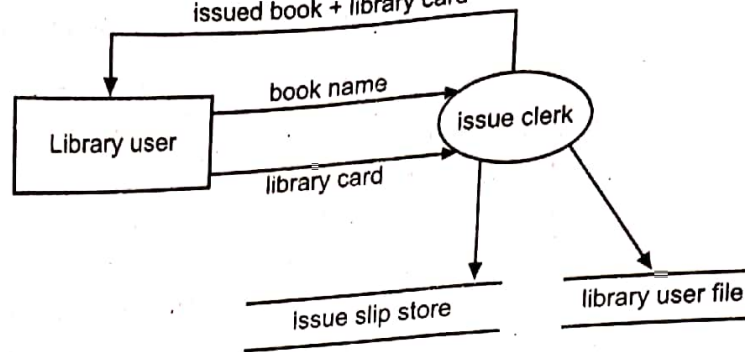


Fig. 2.2

- Logical DFD to issue a book from library in Fig. 2.2 as shown in Fig. 2.3.

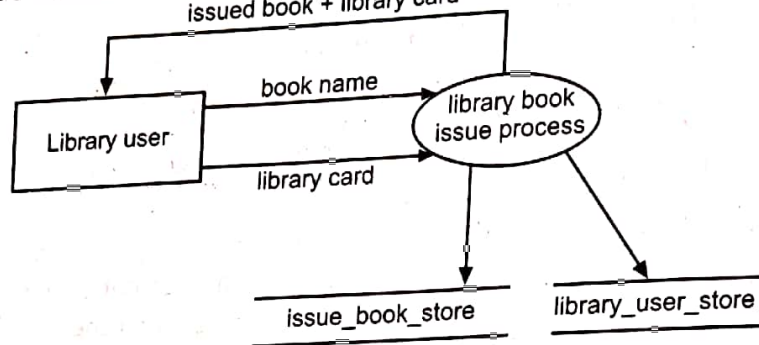


Fig. 2.3

- Here, library user will give book name and library card, after successful issue his name and book name will be stored in issue_book store and library_user_store contain all the books issued to the user.

Comparison between Physical and Logical DFDs:

| Sr. No. | Physical DFD | Logical DFD |
|---------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| 1. | The DFD that shows "what is going on" is a physical DFD. | The DFD that shows "how it is going on" is a logical DFD. |
| 2. | Physical DFDs are more detailed in nature. | Logical DFDs are more abstract in nature. |
| 3. | Physical DFDs are more realistic and implementation oriented. | Logical DFDs are more logical in format. |
| 4. | Physical DFDs show physical component and system like people, documents, reports etc. | Logical DFDs show work done without referring to order processing etc. |
| 5. | Physical DFD shows an implementation dependent view of the current system. | Logical DFD shows an implementation independent view of the current system. |
| 6. | Physical DFDs just for visualization. | Logical DFD help to get a clear idea of what system is achieving. |

Advantages of Data Flow Diagrams (DFDs):

1. A simple graphical technique which is easy to understand by technical and non-technical users.
2. It is useful for communicating current system knowledge to the users.
3. A DFD is used as the part of system documentation file.

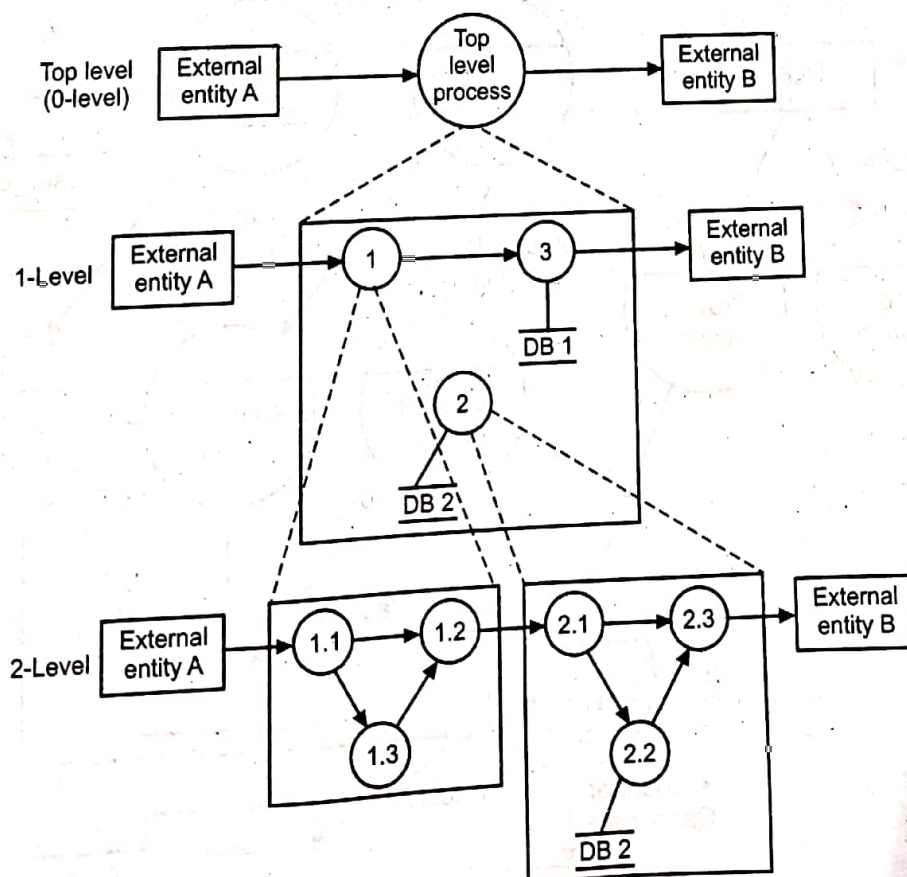
4. A DFD explains the logic behind the data flow within the system.
5. DFDs can provide a detailed representation of system components.

Disadvantages of Data Flow Diagrams (DFDs):

1. Data flow diagram undergoes lot of alteration before going to users, so makes the process little slow.
2. The DFD takes a long time to create.
3. Physical considerations are left out in DFD.
4. It make the programmers little confusing concerning the system.
5. Different DFD models have different symbols like in Gane and Sarson process is represented as rectangle where as in DeMarco and Yourdan symbol it is represented as eclipse.

2.1.2 Levels of DFDs

- DFD supports a top-down approach for analysis. There are various levels of DFD, which provide details about the input, processes, and output of a system.
- Note that the level of detail of process increases with increase in level(s). However, these levels do not describe the system's internal structure or behavior. These levels are listed below:
 1. **Level 0 DFD:** This shows an overall view of the system. Level a DFD is also known as context diagram.
 2. **Level 1 DFD:** This elaborates level a DFD and splits the process into a detailed form.
 3. **Level 2 DFD:** This elaborates level 1 DFD and displays the process(s) in a detailed form.
 4. **Level 3 DFD:** This elaborates level 2 DFD and displays the process(s) in a detailed form.
- Levelled DFD are drawn in top down approach, i.e. at top level 0 level DFD or context in made then this 0 level DFD is expanded to level 1 DFD from (expanding and dividing) the main process bubble of 0 level DFD.
- At level 1 the bubbles are also called as the sub processes. To draw level 2 DFD, these sub processes are further expanded and divided individually in 3 to 5 bubbles and so on. A detailed illustration of levelling is shown in Fig. 2.4.



Example of DFD:

- To understand various levels of DFD, let us consider an example of **Banking System**.

Level 0 DFD:

- The level 0 DFD (also known as the context level DFD) is the simplest DFD. The outermost level (level 0) is concerned with how the system interacts with the outside world. This level basically represents the input and output of the entire system.
- The Level 0 DFD depicts the entire Banking system as a single process. There are various tasks performed in a Bank such as Transaction processing, Pass book entry, Registration, Demand draft creation, Online help and so on.
- The data - flow indicates that these tasks are performed by both the user (customer) and the bank. When the user performs a transaction, the bank verifies whether the user is registered in the Bank.

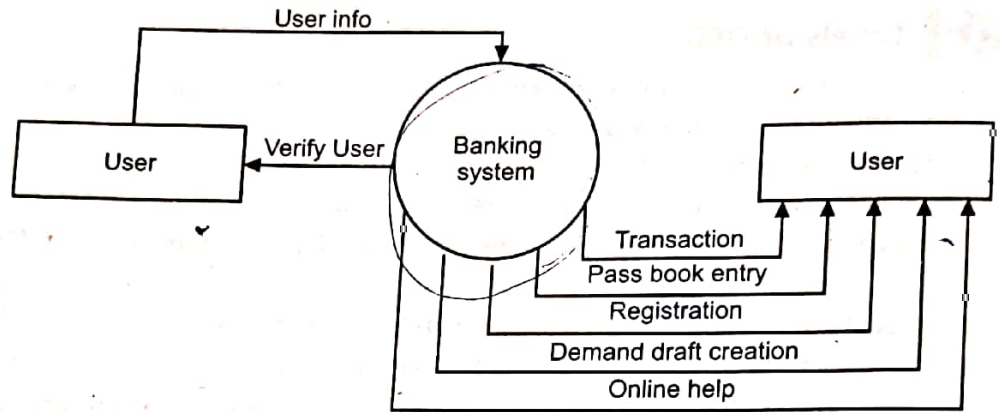


Fig. 2.5

Level 1 DFD:

- The Level 1 DFD is expanded in Level 1 DFD. In Level 1 DFD, the 'user' entity is related to several processes in the Bank, which include 'register', 'user support', and 'provide cash'. Transaction can be performed only if the user is already registered in the Bank.

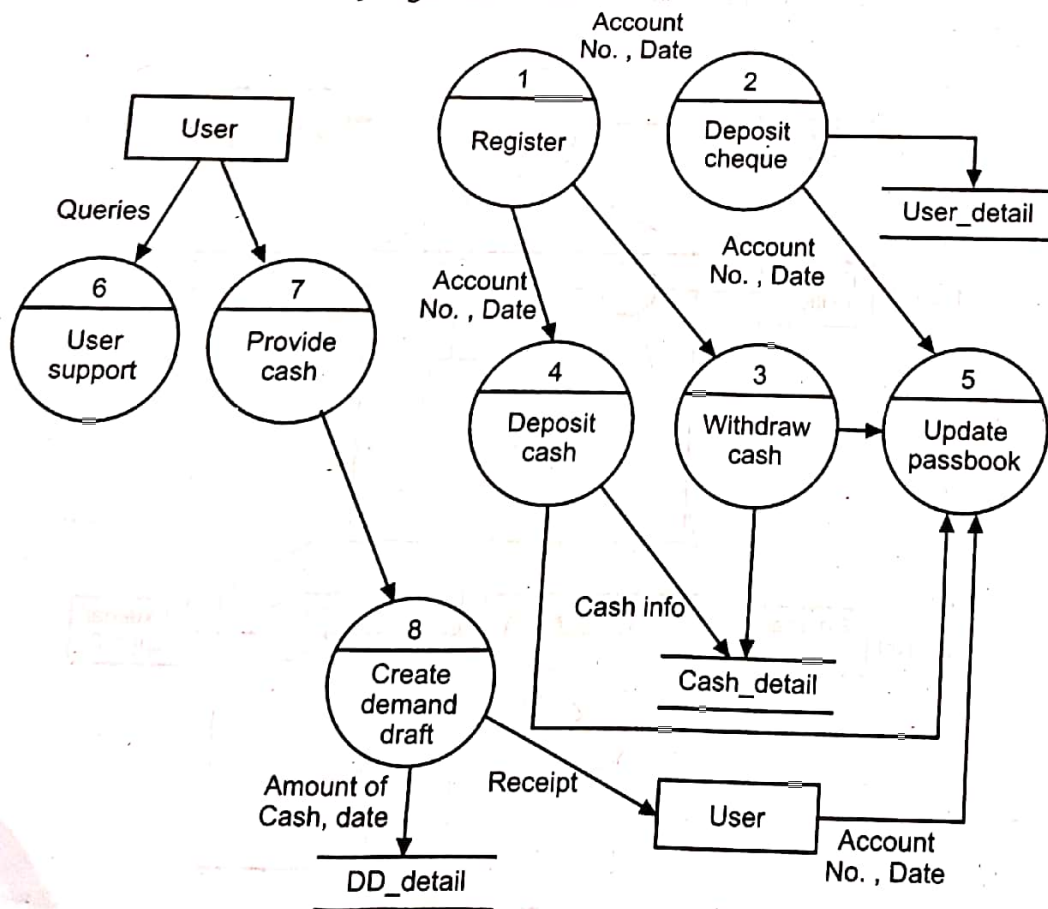


Fig. 2.6

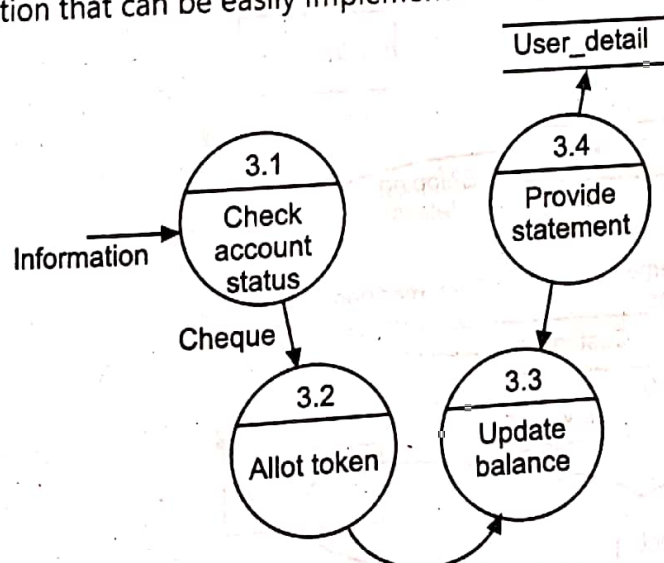
- Once, the user is registered, he can perform a transaction by the processes, namely, 'deposit cheque', 'deposit cash' and 'withdraw cash'. If the user is performing transaction 'deposit cheque', the user needs to provide a cheque to the bank. The user's information such as name, address, and account number is stored in 'user-detail' data store, which is a database.
- If cash is to be deposited and withdrawn, then the information about the deposited cash is stored in 'cash-detail' data store.
- The user can get a demand draft created by providing cash to the bank. It is not necessary for the user to be registered in that bank to have a demand draft. The details of amount of cash and date are stored in 'DD-detail' data store.
- Once, the demand draft is prepared its receipt is provided to the user. The 'user support' process helps users by providing answers to their queries related to the services available in the bank.

Level 2 DFD:

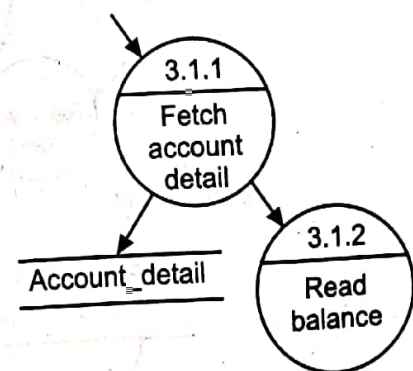
- Level 1 DFD can be further refined into Level 2 DFD for any process of a banking system that has detailed tasks to perform. For instance, Level 2 DFD can be prepared to deposit a cheque, deposit cash, withdraw cash, provide user support, and to create a demand draft.
- However, it is important to maintain the continuity of information between the previous levels (Level 0 and Level 1) and Level 2 DFD. As mentioned earlier, the DFD is refined until each process performs a simple function, which is easy to implement.
- Let us consider the 'withdraw cash' process to illustrate Level 2 DFD, (See Fig. 2.7 (a)). The information collected from Level 1 DFD acts as an input to Level 2 DFD.
- To withdraw cash, the bank checks the status of balance in the user's account ('check account status' process) and then allots a token ('allot token' process). After the user withdraws cash, the balance in user's account is updated in the 'user-detail' data store and a statement is provided to the user.

Level 3 DFD:

- If a particular process of Level 2 DFD requires elaboration, then this level is further refined into Level 3 DFD. Let us consider the process 'check account status' to illustrate Level 3 DFD, (See Fig. 2.7 (b)).
- To check the account status, the bank fetches the account detail ('fetch account detail') from the 'account-detail' data store. After fetching the details, the balance is read ('read balance') from the user's account.
- Note that the requirements engineering process of DFDs continues until each process performs a function that can be easily implemented as an individual program component.



(a) Level 2 DFD to Withdraw Cash



(b) Level 3 DFD to Check Account Status

Fig. 2.7

1. DFD for Online Shopping System.

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.

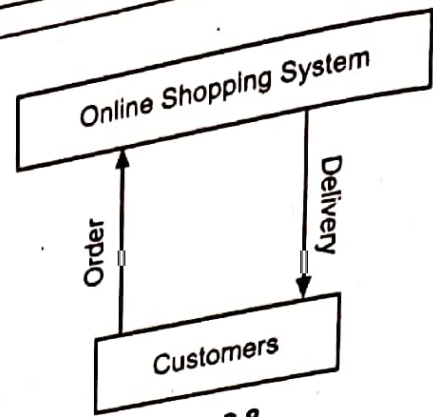


Fig. 2.8

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 1 DFD also mentions basic processes and sources of information.

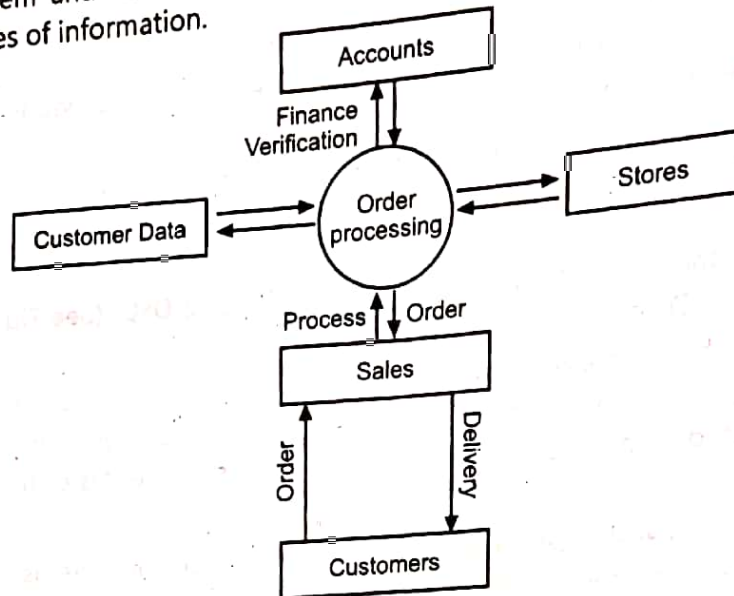


Fig. 2.9

Level 2: At this level, DFD shows how data flows inside the modules mentioned in Level 1. 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.

2. DFDs for Book Ordering System.

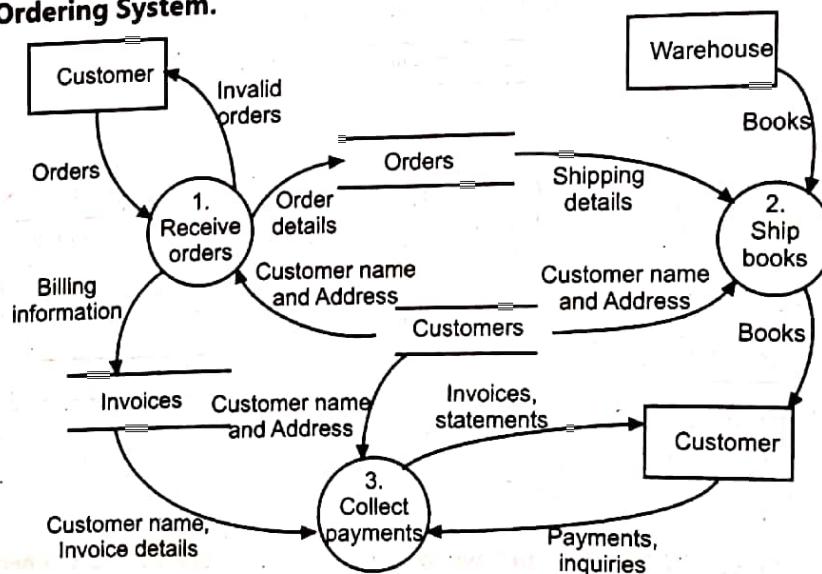
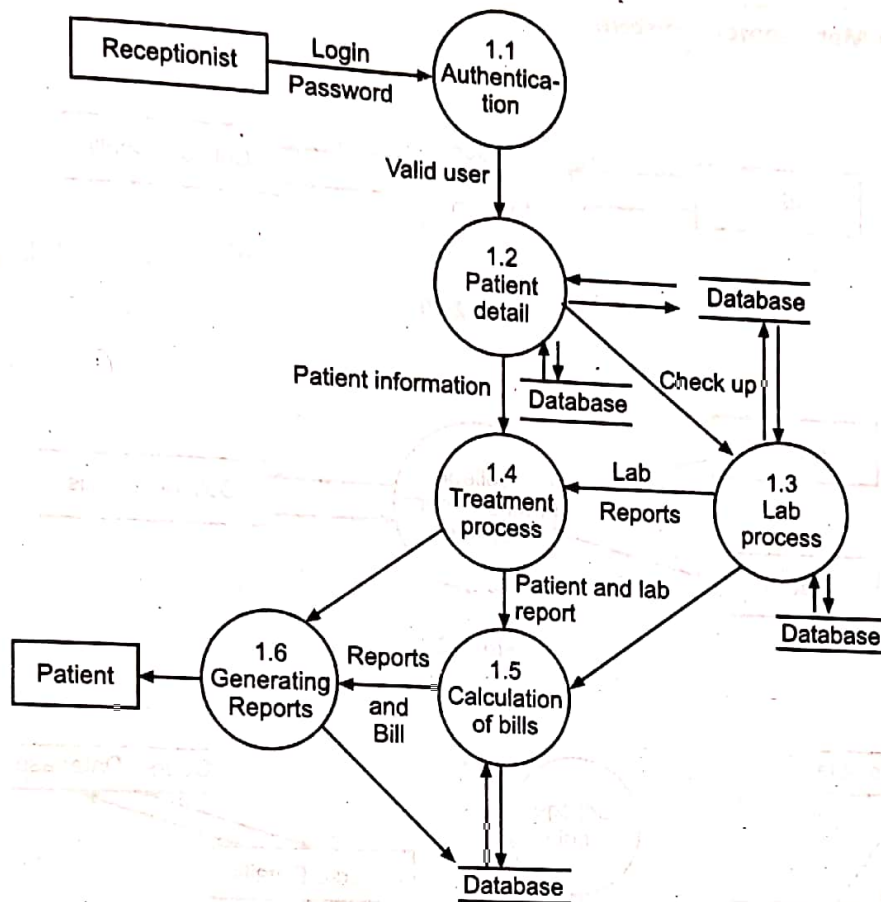


Fig. 2.10

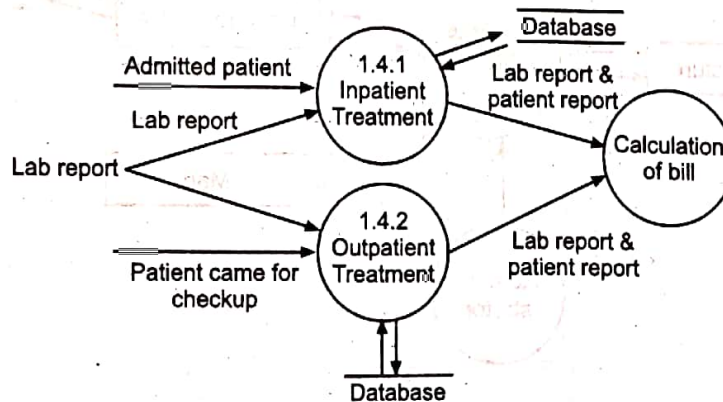
5. DFD for Hospital Management System.

```
graph LR; Admin[Administrator/user] -- Authentication --> HMS((Hospital Management System)); HMS -- "Lab reports, Bill statement" --> Admin; HMS <--> DB[Hospital database];
```

DFD Level 1:



DFD Level 2 and 3:



Scanned with CamScanner

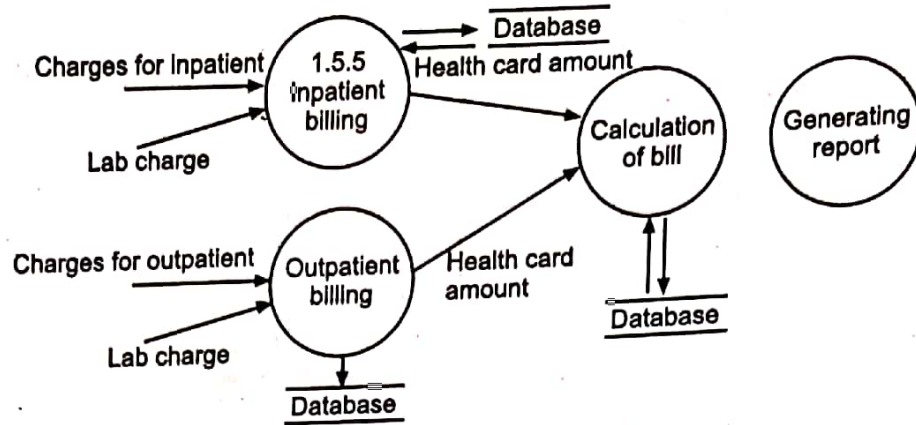


Fig. 2.19

6. DFD for College Management System.

DFD Level 0:

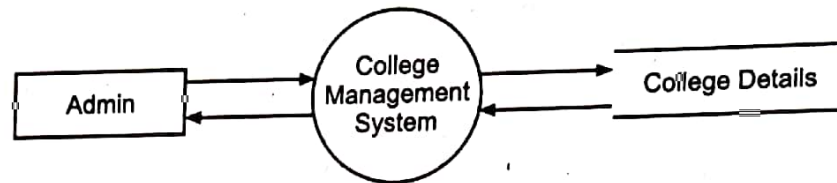


Fig. 2.20

DFD Level 1:

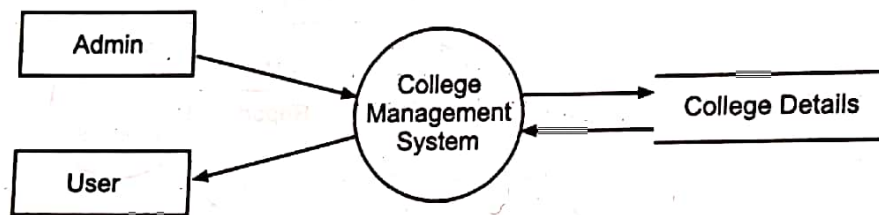


Fig. 2.21

DFD Level 2:

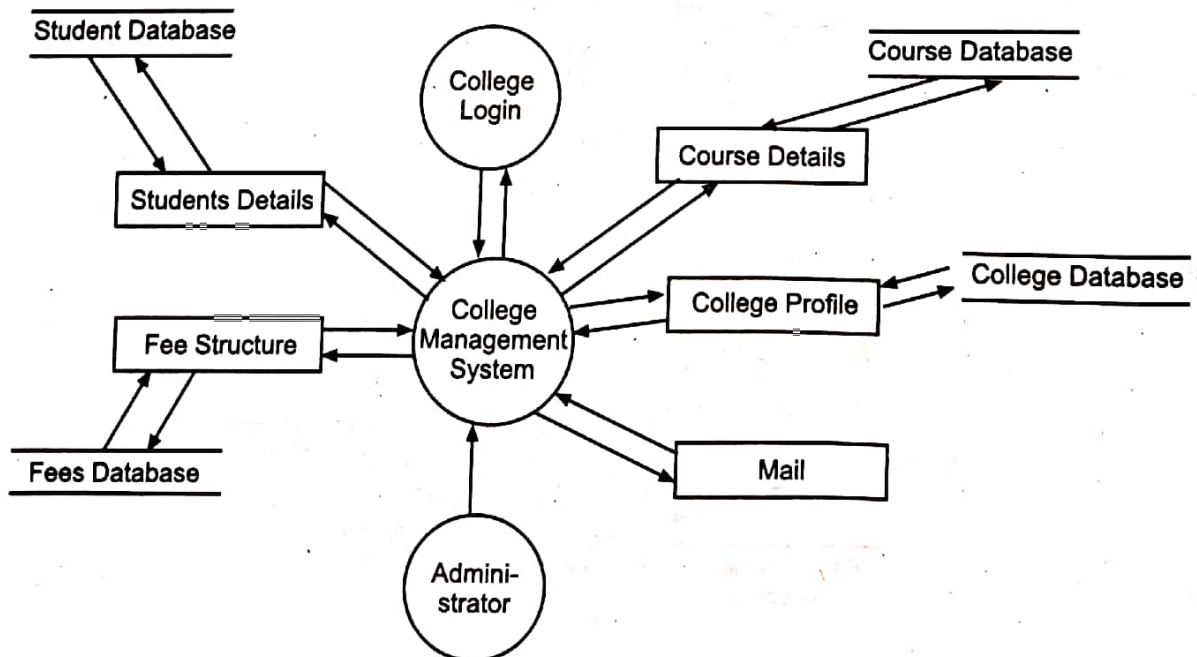


Fig. 2.22