4...

Analysis and Design Tools

Learning Objectives...

- To understand the concept of Analysis and Design.
- To learn about Decision Tree and Table.
- To study Data Flow Diagram and Data Dictionary in detail.
- To learn the concept of Input and Output Design, Pseudocode.

4.1 INTRODUCTION TO ANALYSIS AND DESIGN

- System analysis and design is a process used in software engineering and business systems to understand how a system functions and how it can be improved or newly created.
- System analysis involves examining a business situation with the intent of improving it through better procedures and methods. It includes identifying what the system is supposed to do, who uses it, what data it handles, and the processes involved.
- System design is the phase where the findings from the analysis are used to create a blueprint for building or modifying the system. It focuses on how the system will operate, including hardware, software, network, user interface, and data design.

4.1.1 Decision Tree

- A **Decision Tree** helps us in making decisions by visually outlining various choices and their potential outcomes in a tree-like structure. It is commonly used in machine learning for tasks such as classification and prediction, where it helps determine the best course of action or category based on given input data.
- A Decision Tree helps in making decisions by displaying various options and their relationships in a tree-like format. It begins with a main question at the root node,

which represents the entire dataset, and branches out based on different features or conditions in the data.

- **Root Node:** The starting point of the tree that represents the complete dataset and the first decision to be made.
- **Branches:** Paths that connect nodes, showing the flow from one decision or condition to the next.
- **Internal Nodes:** Points within the tree where specific decisions are made based on data features or conditions.
- **Leaf Nodes:** The final nodes in the tree, representing the outcome, decision, or prediction.

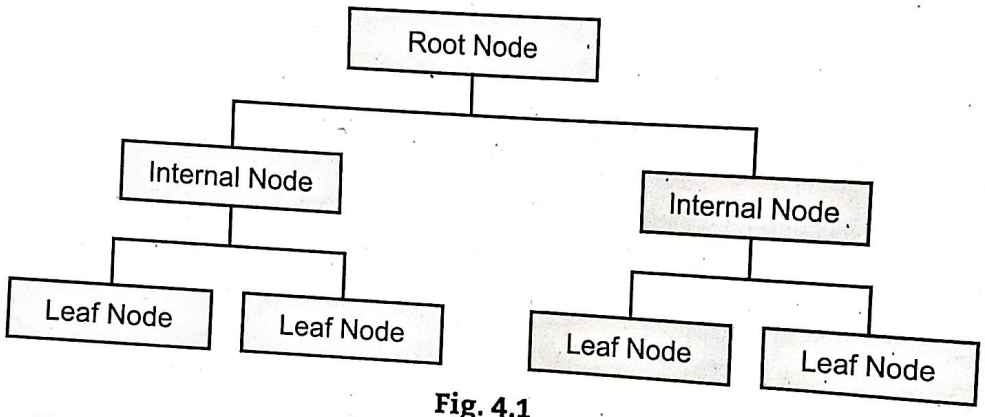


Fig. 4.1

How Decision Trees Work?

1. **Start with the Root Node:** It begins with a main question at the root node which is derived from the dataset's features.
2. **Ask Yes/No Questions:** From the root, the tree asks a series of yes/no questions to split the data into subsets based on specific attributes.
3. **Branching Based on Answers:** Each question leads to different branches:
   - If the answer is yes, the tree follows one path.
   - If the answer is no, the tree follows another path.
4. **Continue Splitting:** This branching continues through further decisions helps in reducing the data down step-by-step.
5. **Reach the Leaf Node:** The process ends when there are no more useful questions to ask leading to the leaf node where the final decision or prediction is made.

- Let's look at a simple example to understand how it works. Imagine we need to decide whether to drink coffee based on the time of day and how tired we feel. The tree first checks the time:

1. **In the morning: It asks "Tired?"**

- If yes, the tree suggests drinking coffee.
- If no, it says no coffee is needed.

2. **In the afternoon: It asks again "Tired?"**

- If yes, it suggests drinking coffee.
- If no, no coffee is needed.

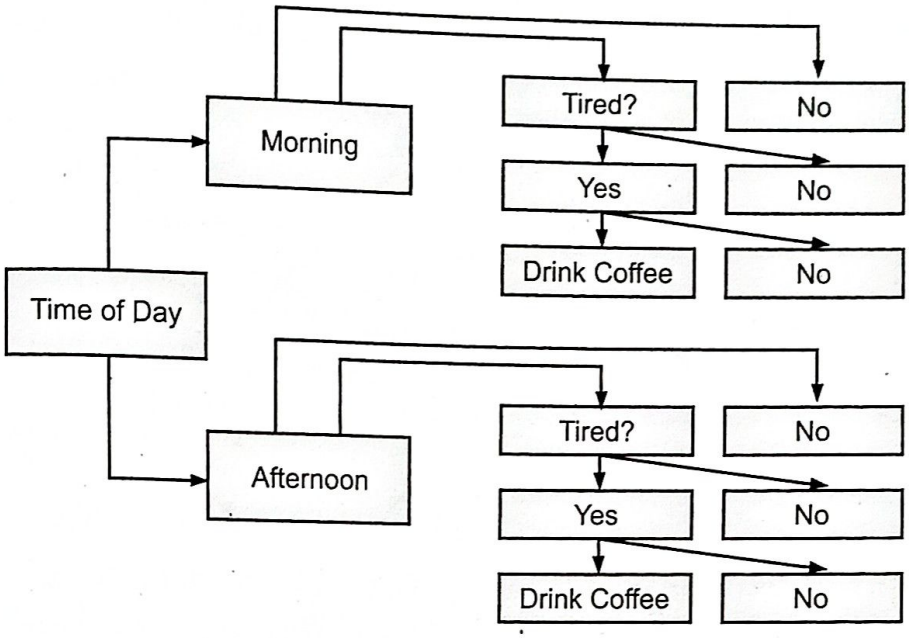


Fig. 4.2

For Example, Decision Tree:

Let's understand decision trees with the help of an example:

Day	Weather	Temperature	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5.	Rainy	Mild	High	Strong	Yes
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

- Decision trees are upside down. This means the root is at the top. Then this root is split into various nodes. They are nothing but a bunch of if-else statements in layman terms. It checks if the condition is true and then it goes to the next node attached to that decision.

- In the fig. 4.3 the tree will first ask, what is the weather? Is it sunny, cloudy, or rainy? If yes then it will go to the next feature which is humidity and wind. It will again check if there is a strong wind or weak. If it's a weak wind and it's rainy, the person may go and play.

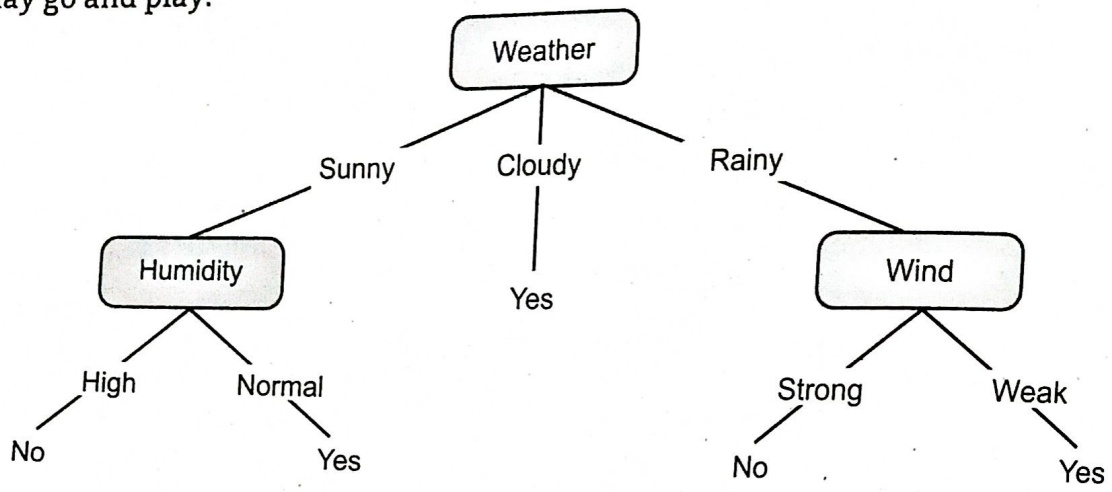


Fig. 4.3

- Did you notice anything in the fig. 4.3 flowchart? We see that if the weather is cloudy then we must go to play. Why didn't it split more? Why did it stop there?
- To answer this question, we need to know about few more concepts like entropy, information gain, and Gini index. But in simple terms, I can say here that the output for the training dataset is always yes for cloudy weather. Since there is no disorderliness here, we don't need to split the node further.
- The goal of machine learning is to decrease uncertainty or disorders from the dataset and for this, we use these trees.
- Now you must be thinking how do I know what should be the root node? What should be the decision node? When should I stop splitting? To decide this, there is a metric called "Entropy" which is the amount of uncertainty in the dataset.

Advantages:

1. Easy to read and understand.
2. Useful for decision-making with clear conditions.

Disadvantages:

1. Can become very large with many conditions.
2. Prone to errors if over-complicated.

4.1.2 Decision Table

- A decision table is a precise way of modelling logic. Each possible combination of conditions is considered in turn and what action is required.
- A decision table is a tool for documenting complicated logic which is a part of some business problem. The aim is to state all combinations of conditions and outcomes which result.

- The decision table should be thought of as two parts the conditions, which each have a True or False outcome the possible outcomes.
- The table then form a grid where the outcome can be shown for all possible combinations of conditions.

Decision Table in Test Designing:

Table 4.1

Blank Decision Table				
CONDITIONS	STEP 1	STEP 2	STEP 3	STEP 4
Condition 1				
Condition 2				
Condition 3				

Table 4.2

Decision Table Combinations				
CONDITIONS	STEP 1	STEP 2	STEP 3	STEP 4
Condition 1	Y	Y	N	N
Condition 2	Y	N	Y	N
Condition 3	Y	N	N	Y

Example 1: How to Make Decision Base Table for Login Screen.

Specification:

- Let's take the example of a login screen of any application. The condition states that if the user provides the correct username and password the user will be redirected to the homepage. If any of the input is wrong, an error message will be displayed on the screen.

Decision Table for Login Screen:

Conditions	Rule 1/TC1	Rule 2/TC2	Rule 3/TC3	Rule 4/TC4
Email Id (Input)	T	T	F	F
Password (Input)	T	F	T	F
Action(Output)	H	E	E	E

- In the above example,

T stands for correct Email id/password.
F stands for incorrect Email id/password
H stands for Home screen to be displayed to the user
E stands for the Error message to be displayed to the user
TC stands for the Test case.

- Now let's discuss the actions or output of the above decision table as per the provided inputs (email id and password).
  - **Test Case 1:** The email id and password both are correct. Hence, the user should be directed to Testsigma's Homepage.
  - **Test Case 2:** The email id was correct, but the password was wrong. Hence, an error message should be shown to the user mentioning "Incorrect Password".
  - **Test Case 3:** The email id was wrong, but the password was correct. Hence, an error message should be shown to the user mentioning "Incorrect Email Id".
  - **Test Case 4:** The email id and password both are incorrect. Hence, an error message should be shown to the user mentioning "Incorrect Email Id".
- In the above example, you can see how all possible conditions or test cases have been included, the testing team can refer to this decision table and create their test cases so that upcoming bugs can be found at the testing level.
- Let's check another example of how a decision table can be used in case of an upload screen:

Example 2: How to Make Decision Table for Upload Screen

- Let's now take another example, we consider the decision table and test scenarios for an Upload screen. There is a dialogue box that will ask the user to upload a file.

Specification:

The file should be in .png format.

The size of the file should be less than 25 kb.

The file resolution must be 132 * 170px.

Now, let's create a decision table for the upload screen keeping the above specification in mind:

Decision Table for Upload Screen:

Conditions	Rule 1/TC1	Rule 2/TC2	Rule 3/TC3	Rule 4/TC4	Rule 1/TC5	Rule 2/TC6	Rule 3/TC7	Rule 4/TC8
Format (Input)	.png	.png	.png	.png	Not .png	Not.png	Not.png	Not.png
Size (Input)	<25kb	<25kb	>=25kb	>=25kb	<25kb	<25kb	>=25kb	>=25kb
Resolution (Input)	= 132*170px	!= 132*170px	=132*170px	!= 132*170px	=132*170px	!= 132*170px	=132*170px	!= 132*170px
Action (Output)	Upload the .png file successfully.	Error Message displaying resolution mismatched.	Error message displaying size mismatch	Error message displaying size mismatched	Error Message displaying format mismatched.	Error Message displaying format mismatched.	Error Message displaying format mismatched	Error Message displaying format mismatched.

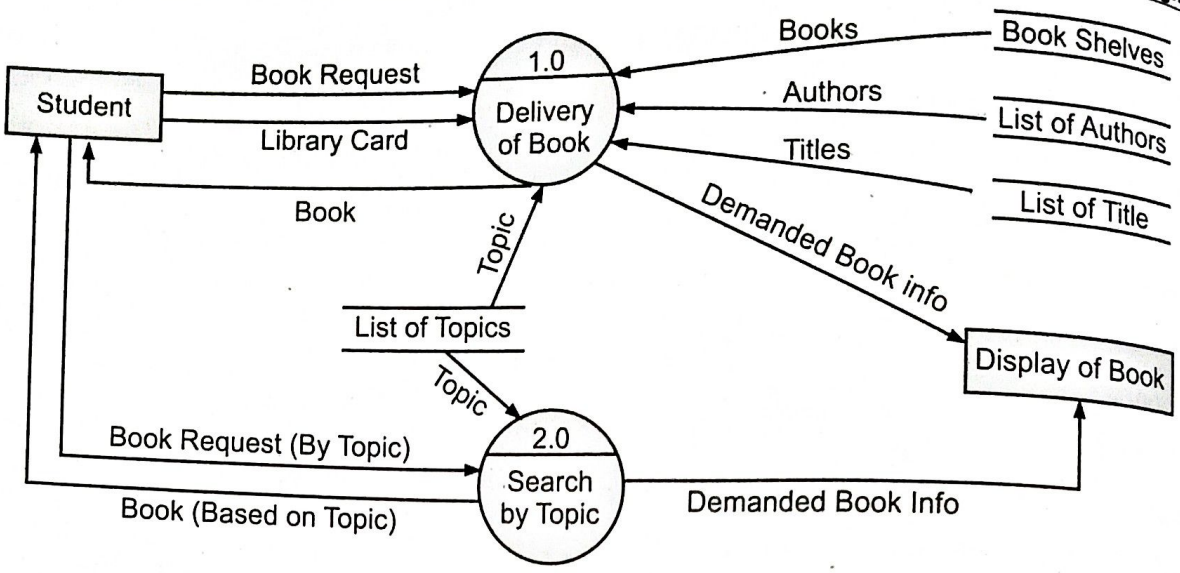


Fig. 4.19: Level 1

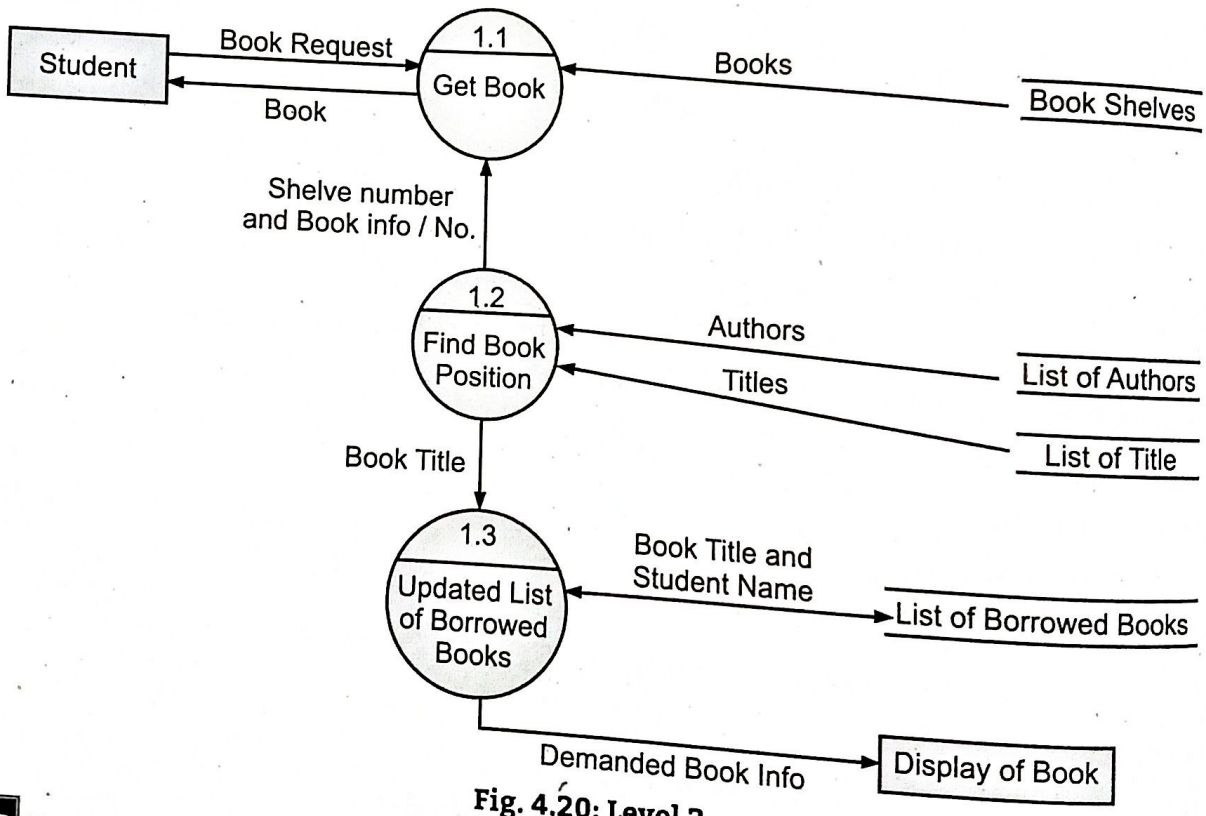


Fig. 4.20: Level 2

4.6.3 Data Flow Diagram (DFD) for a Student Attendance System

External Entities:

- Student
- Faculty
- Admin

Process:

- 0.0 Student Attendance System

Data Flows:

- **Student → Student Attendance System:** Student Detail, Attendance Report.
- **Faculty → Student Attendance System:** Take Attendance, Student Detail.
- **Admin → Student Attendance System:** Add Course/Subject/Batch.
- **Student Attendance System → Faculty:** Attendance Report.
- **Student Attendance System → Admin:** Attendance Report.

DFD Level 1: Student Attendance System

Processes:

- 1.0 Student Registration
- 2.0 Course Master
- 3.0 Subject Master
- 4.0 Batch Schedule
- 5.0 Student Attendance
- 6.0 Email Attendance

External Entities:

- Student
- Faculty
- Admin

Data Stores/Flows:

- **Student → 1.0 Student Registration:** Student Details.
- **1.0 → 5.0 Student Attendance:** Student Details.
- **Admin → 2.0 Course Master:** Add Course.
- **2.0 → 3.0 Subject Master:** Course.
- **Admin → 3.0 Subject Master:** Add Subject.
- **3.0 → 4.0 Batch Schedule:** Subject.
- **Admin → 4.0 Batch Schedule:** Add Batch.
- **4.0 → 5.0 Student Attendance:** Batch.
- **Faculty → 5.0 Student Attendance:** Take Attendance.
- **5.0 → Admin:** Attendance Report.
- **5.0 → 6.0 Email Attendance:** Attendance Detail.
- **6.0 → Student:** Email Attendance Report.

- Based on the above decision table data, we can develop eight separate test cases to assure comprehensive coverage for the given specification.
  - **Test Case 1:** Click on upload to add a file with the format of '.png,' type, with a file size of less than 25kb, and a resolution of 132*170 pixels. The expected outcome is that the photo will upload successfully.
  - **Test Case 2:** Click on upload to add a file with the format of '.png' type, with a file size of less than 25kb, and a resolution of more than 132*170 pixels. The expected outcome is that it should display an error message text with the file resolution mismatch string and the user shouldn't be able to upload the file.
  - **Test Case 3:** Click on upload to add a file with the format of '.png' type, with a file size of more than 25kb, and a resolution of 132*170 pixels. The expected outcome is that it should display an error message text with the file size incorrect string and the user shouldn't be able to upload the file.
  - **Test Case 4:** Click on upload to add a file with the format of '.png' type, with a file size of more than 25kb, and a resolution of less than 132*170 pixels. The expected outcome is that it should display an error message text with the file size incorrect string and the user shouldn't be able to upload the file.

    (**Note:** Even if the resolution is not equal to 132*170px, it should give a size incorrect error message as size would be the first validation and the size is wrong in this case.)
  - **Test Case 5:** Click on upload and select a file with a format of other than '.png' type, with a file size of less than 25 kb, and a resolution of 132*170 pixels. The anticipated outcome is an error message should be displayed mentioning there is a format mismatch and the user shouldn't be allowed to upload the file.
  - **Test Case 6:** Click on upload and select a file with a format other than '.png,' type and a file size of less than 25 kb, and a resolution of more than 137*177 pixels. The anticipated outcome is an error message should be displayed mentioning there is a format mismatch and the user shouldn't be allowed to upload the file.

    (**Note:** Even if the resolution is not equal to 132*170px, it should give a format mismatch error based on the first format validation and it is incorrect.)
  - **Test Case 7:** Click on upload to add a file in a format other than '.png' type, with a file size equal to 25kb, and a resolution of 137*177 pixels. The expected outcome is an error message should be displayed mentioning there is a format mismatch and the user shouldn't be allowed to upload the file.

    (**Note:** Even if the size is not less than 25 kb, it should give a format mismatch error based on the first format validation and it is incorrect.)

- **Test Case 8:** Click on upload and select a file with a format other than '.png' type with a file size of more than 25 kb, and a resolution of less than 137*177 pixels. The expected outcome is an error message should be displayed mentioning there is a format mismatch and the user shouldn't be allowed to upload the file.

  (**Note:** Even if the size is not less than 25kb and the resolution is not equal to 132*170px, it should give a format mismatch error based on the first format validation and it is incorrect.)

- From the above two examples we can conclude that:

In order to find the number of all possible conditions for the decision table, you can use the 2^n formula where n denotes the number of inputs; in example-1 there is the number of inputs is 2 (one is the Email id and the second is the Password).

Number of possible test conditions = $2^{\text{Number of input conditions}}$

Number of possible test conditions = $2^2 = 4$

Hence, in the case of example 1 we have covered four test cases.

Similarly in case of example 2, on a high level there are 3 different types of inputs such as the file format, the size and the file resolution.

Applying the same 2^n formula:

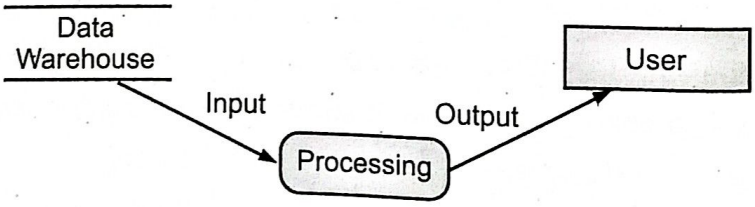
Number of possible test conditions = $2^{\text{Number of input conditions}}$

Number of possible test conditions = $2^3 = 8$

Hence, in case of example 2 we have covered eight test cases.

4.2 DATA FLOW DIAGRAM (DFDs)

- Data Flow Diagram (DFD) is a graphical representation of data flow in any system. It is capable of illustrating incoming data flow, outgoing data flow and store data. The DFD depicts both incoming and outgoing data flows and provides a high-level overview of system functionality. It is a relatively simple technique to learn and use, making it accessible for both technical and non-technical stakeholders.
- Data Flow Diagram can be represented in several ways. The Data Flow Diagram (DFD) belongs to structured-analysis 1 tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.



Note: Arrows signity flow of data.

Fig. 4.4

Characteristics of Data Flow Diagram (DFD):

- Below are some characteristics of Data Flow Diagram (DFD):

1. **Graphical Representation:** Data Flow Diagram (DFD) use different symbols and notation to represent data flow within system. That simplifies the complex system into understandable visual elements. This makes them easier to interpret by both technical and non-technical stakeholders.
2. **Problem Analysis:** Data Flow Diagram (DFDs) is very useful in understanding a system and can be effectively used during analysis. Data Flow Diagram (DFDs) is quite general and is not limited to problem analysis for software requirements specification.
3. **Abstraction:** DFDs abstract away the implementation details and focus on the data flow and processes within a system. They provide a high-level overview and omit unnecessary technical information.
4. **Hierarchy:** Data Flow Diagram (DFD) provides a hierarchy of a system. High-level diagram i.e. 0-level diagram provides an overview of entire system while lower-level diagram like 1-level DFD and beyond provides a detailed data flow of individual process.

Data Flow Diagram Symbols and Notation used in DFD:

- The two common systems of symbols are named after their creators (Yourdon & De Marco or Gane & Sarson). Graphically the main difference between these two in terms of drawing data flow diagram is how processes look. In Yourdon & De Marco system processes are represented by circles, while in Gane & Sarson processes are represented by squares with rounded corners.
- There are four basic symbols to represent a data flow diagram.

1. **External entity:**
   - External entities are objects outside the system with which system communicates. These are sources and destinations of the system inputs and outputs. They are also known as terminators, sinks, sources or actors.

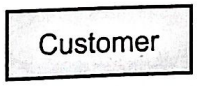


Fig. 4.5

2. **Process:**
   - A process receives input data and process output data with a different form or content. Every process has a name that identifies the function it performs. Process can be as simple as collecting input data and saving in the database or it may be as complex as producing monthly sales report of any particular product in any selected region.

- The symbol of process is a circle or rectangle with rounded corners.

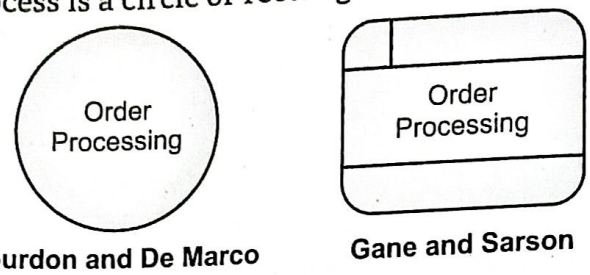


Fig. 4.6

3. Data flow:

- Data flow is the path for data to move from one part of the system to another. It may be a single data element or set of data element. The symbol of data flow is the arrow. The arrow shows the flow direction.

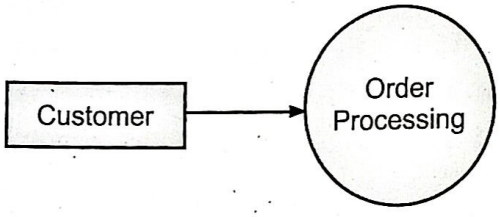


Fig. 4.7

4. Data Store:

- Data store are repositories of data in the system. They sometimes also referred as files. Each data store receives a simple label such as Orders.

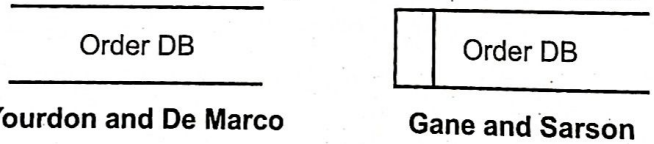


Fig. 4.8

- The following data flow diagram symbols comparison table shows a clear difference between these two.

Table 4.3: DDS Symbol

Notation	De Macro and Yourdon	Gane and Sarson
External Entity		
Process		
Data Store		
Data Flow		

4.2.1 Types of DFDs

- DFDs can be classified into two main types, each focusing on a different perspective of system design:

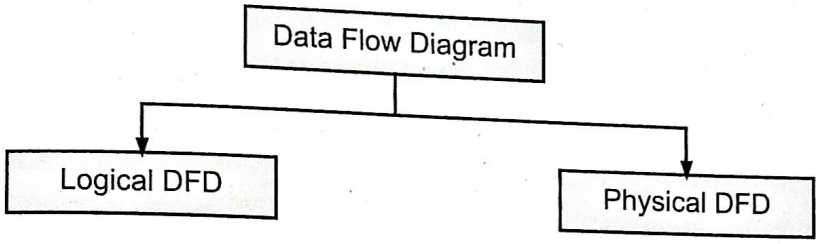


Fig. 4.9

1. **Logical Data Flow Diagram (DFD):**

- The Logical Data Flow Diagram mainly focuses on the system process. It illustrates how data flows in the system. Logical Data Flow Diagram (DFD) mainly focuses on high level processes and data flow without diving deep into technical implementation details.
- Logical DFDs is used in various organizations for the smooth running of system. Like in a Banking software system, it is used to describe how data is moved from one entity to another.

2. **Physical Data Flow Diagram:**

- Physical data flow diagram shows how the data flow is actually implemented in the system. In the Physical Data Flow Diagram (DFD), we include additional details such as data storage, data transmission, and specific technology or system components. Physical DFDs are more detailed and provide a closer look at the actual implementation of the system, including the hardware, software, and physical aspects of data processing.

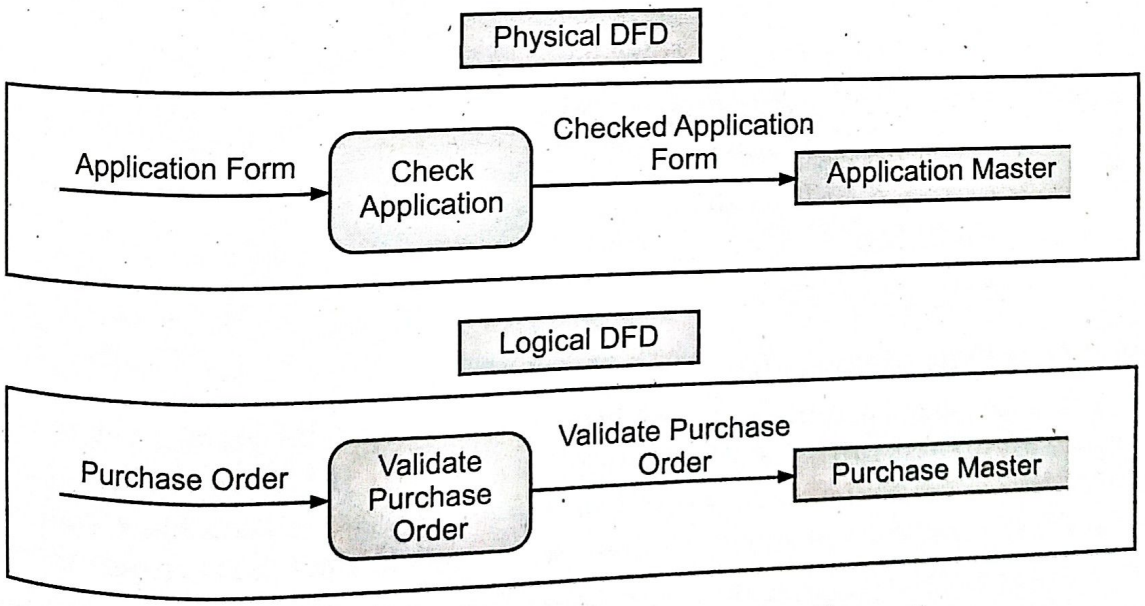


Fig. 4.10: Logical and physical data flow diagram

Table 4.4

Logical DFD	Physical DFD
1. Logical DFD tells how the business operates	1. Physical DFD tells how the system will be implemented (or how the current system operates).
2. The processes represent the business activities.	2. The processes represent the programs, program modules, and manual procedures.
3. The data stores represent the collection of data regardless of how the data are stored.	3. The data stores represent the physical files and databases, manual files.
4. It show business controls.	4. It show controls for validating input data, for obtaining a record, for ensuring successful completion of a process, and for system security.

4.2.2 Levels of DFDs

- DFDs are categorized into various levels, with each level providing different degrees of detail. The levels are numbered from 0 and onward. The higher the level, the more detailed the diagram becomes. The following are the four levels of DFDs:

1. 0-Level Data Flow Diagram (DFD):

- Level 0 DFD is the highest-level diagram, representing the system as a single process with its interactions with external entities. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes. It is also known as the Context Diagram, which abstracts the system's operations and shows how data enters and leaves the system.

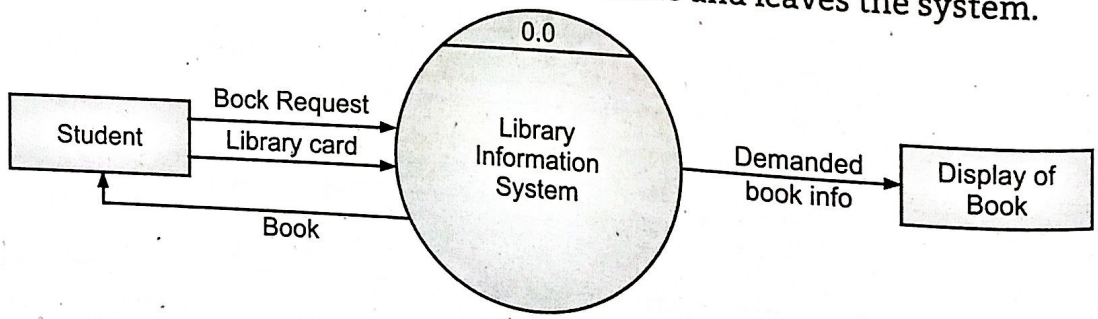


Fig. 4.11

2. 1-Level Data Flow Diagram (DFD):

- Level 1 DFD provides a more detailed view of the system by breaking down the major processes identified in the level 0 Data Flow Diagram (DFD) into sub-processes. Each sub-process is depicted as a separate process on the level 1 Data Flow Diagram (DFD). The data flows and data stores associated with each sub-process are also shown.
- Level 1 DFD provides a more detailed view of the system, focusing on key functional aspects. The Context Diagram from Level 0 is expanded into multiple bubbles/ processes.

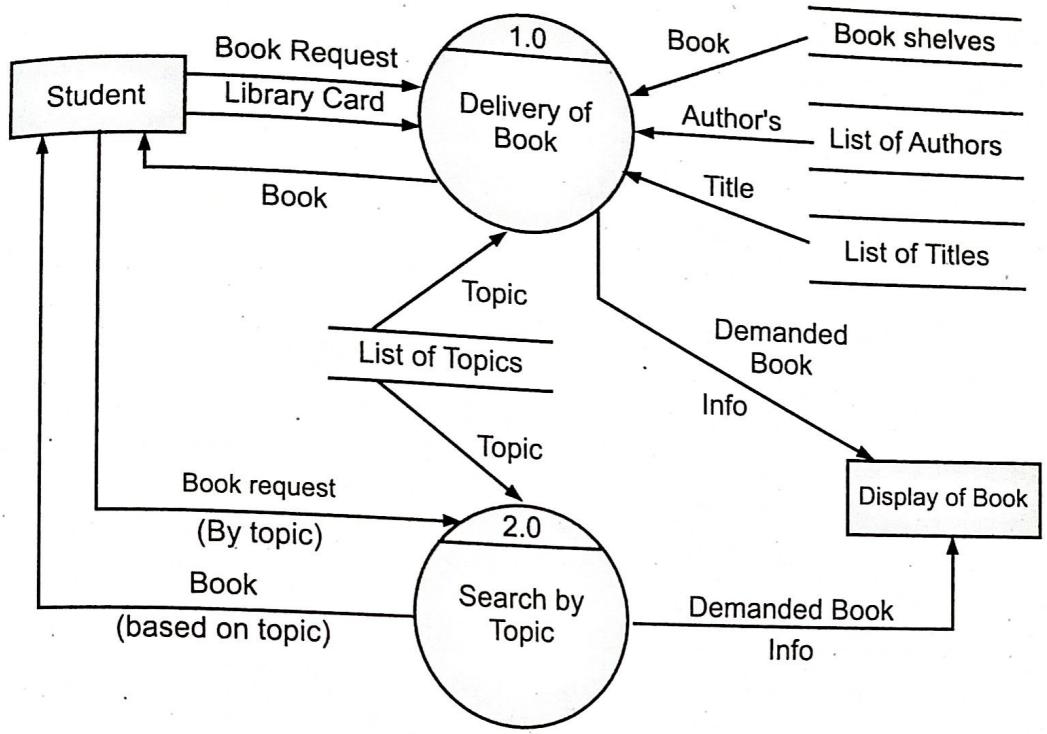


Fig. 4.12: Level 1 DFD

3. 2-Level Data Flow Diagram (DFD):

- Level 2 DFD further breaks down the sub-processes from Level 1 DFD into additional sub-processes, providing an even more detailed view. This level is useful when dealing with specific requirements or parts of the system that need a closer examination of their processes and interactions.

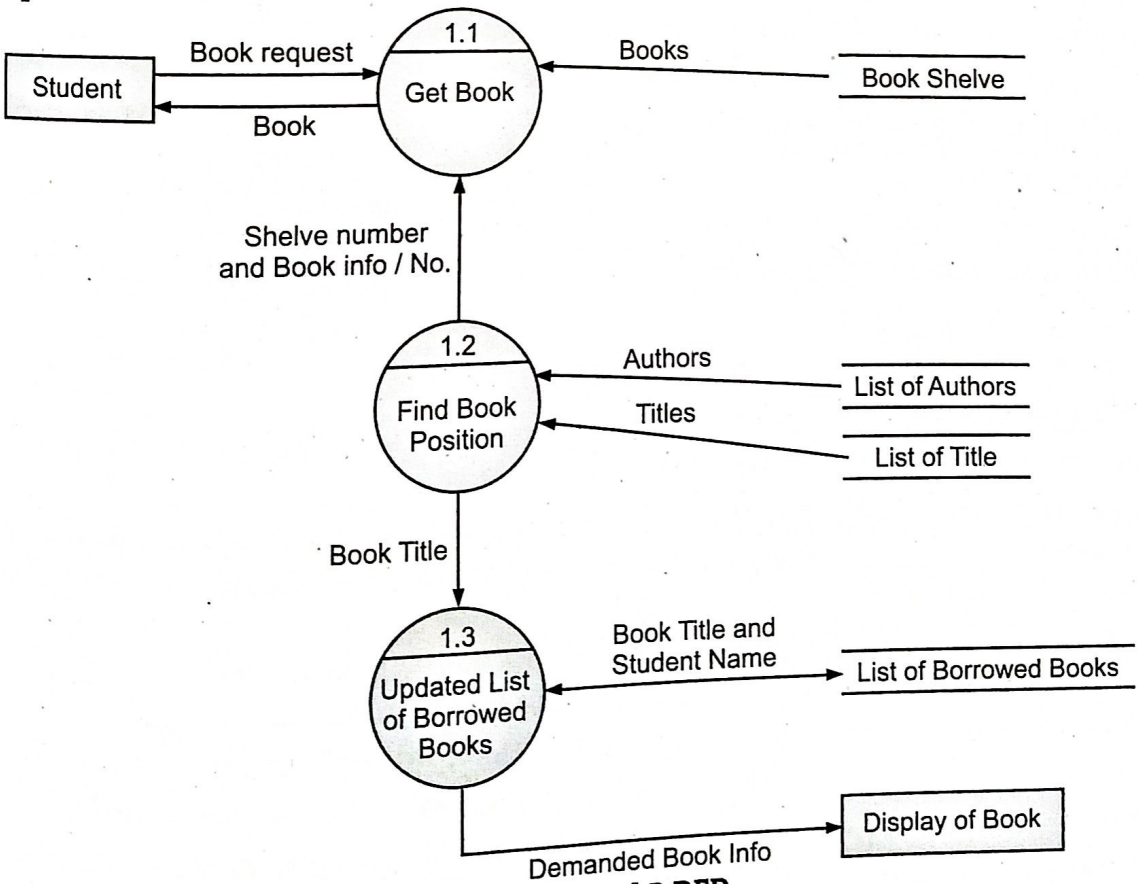


Fig. 4.13: Level 2 DFD

4. 3-Level Data Flow Diagram (DFD)

- 3-Level is the most detailed level of Data Flow Diagram (DFDs), which provides a detailed view of the processes, data flows, and data stores in the system. This level is typically used for complex systems, where a high level of detail is required to understand the system. It includes detailed descriptions of each process, data flow, and data store, and is usually used when there is a need for a comprehensive understanding of the system.

Advantages of using Data Flow Diagrams (DFD):

- Following are the Advantage of Data Flow Diagram (DFD):
  1. **Easy to understand:** DFDs are graphical representations that are easy to understand and communicate, making them useful for non-technical stakeholders and team members.
  2. **Improves system analysis:** DFDs are useful for analyzing a system's processes and data flow, which can help identify inefficiencies, redundancies, and other problems that may exist in the system.
  3. **Supports system design:** DFDs can be used to design a system's architecture and structure, which can help ensure that the system is designed to meet the requirements of the stakeholders.
  4. **Enables testing and verification:** DFDs can be used to identify the inputs and outputs of a system, which can help in the testing and verification of the system's functionality.
  5. **Facilitates documentation:** DFDs provide a visual representation of a system, making it easier to document and maintain the system over time.

Disadvantages of using Data Flow Diagram (DFD):

- Following are the Disadvantage of Data Flow Diagram (DFD):
  1. **Can be time-consuming:** Creating DFDs can be a time-consuming process, especially for complex systems.
  2. **Limited focus:** DFDs focus primarily on the flow of data in a system, and may not capture other important aspects of the system, such as user interface design, system security, or system performance.
  3. **Can be difficult to keep up-to-date:** DFDs may become out-of-date over time as the system evolves and changes.
  4. **Requires technical expertise:** While DFDs are easy to understand, creating them requires a certain level of technical expertise and familiarity with the system being analyzed.

4.3 DATA DICTIONARY

- A **Data Dictionary (DD)** is a centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format. It defines each data element used in a system and helps ensure consistency and clarity in understanding and managing data.

(d) If an employee has at least 15 but less than 30 years of service, 2 extra days are given.

(e) These 2 days are also provided for employees of age 45 or more.

(f) These 2 extra days cannot be combined with the extra 5 days.

8. The decision table below shows the discount policy of an airline. Convert the decision table in to decision tree to illustrate the decision rules.

	R1	R2	R3	R4	R5	R6
Is the passenger <2 ?	Y	N	N	N	N	N
Is the passenger between 2 and 18 ?	–	Y	N	N	N	N
Is the flight on Monday or Friday	–	–	Y	N	N	N
Is the flight inside Germany	–	–	–	N	Y	Y
Will the passenger stay more than 6 days ?	–	–	–	–	Y	N
100% discount	X	–	–	–	–	-
40% discount	–	X	–	–	–	-
25% discount	–	–	–	X	–	-
20% discount	–	–	–	–	X	X
10% discount	–	–	–	–	X	-
0% discount	–	–	X	–	–	-

9. Discuss the advantages of using decision tables in system design.
10. Explain the components of a DFD with a diagram.
11. Describe the importance of DFD in system analysis.
12. Draw and explain a Level 0 and Level 1 DFD for a
    (a) Library Management System.
    (b) Student Admission System
    (c) Hospital Management System
    (d) Airline Reservation System
13. Differentiate between physical DFD and logical DFD with examples.
14. Explain various levels of DFD with suitable examples.
15. What is a Data Dictionary?
16. List two contents of a data dictionary.
17. Explain the purpose and elements of a data dictionary.
18. Describe how a data dictionary supports DFDs.
19. Explain the key elements of a data dictionary with examples.
20. List and explain the advantages and disadvantages of maintaining a data dictionary.
21. Explain the principles of good input design.
22. Describe the types of output and methods for designing user-friendly outputs.

***

12. Define Level 1 DFD.
13. Name any two elements of a data dictionary.
14. What is the role of aliases in a DD?
15. Mention one advantage and one disadvantage of a DD.
16. What is input design?
17. Give one example of output design.
18. What is pseudocode?
19. Write a pseudocode to find the largest of three numbers.

Q.II Answer the following questions in detail:

1. Explain the difference between system analysis and system design with examples.
2. Discuss the importance of analysis and design in the development life cycle.
3. Differentiate between decision tree and decision table.
4. Create the decision table and decision tree for a login screen that asks which contains User id and Password. The condition here is that the user will be redirected to the homepage if he enters the correct user name and password, and an error message will be displayed if the input is wrong.
5. Create decision table for an upload screen. There is a dialogue box that will ask the user to upload a photo with the following conditions:
   (a) The file must be in the .jpg format.
   (b) The file size must be less than 32 kb.
   (c) The image resolution must be 137*177.
   (d) If any one of the above conditions fails, system will display corresponding error messages about the issue. If all conditions are satisfied, the photo will be uploaded successfully.
6. Create the decision table for the following conditions:
   (a) If there is a working-day, it's not a holiday and weather is Rainy Then, Go to office.
   (b) If there is not a working-day but it's not a holiday and weather is not Rainy Then, Go to office.
   (c) If there is not a working-day, it's a holiday and weather is also Rainy Then, Watch TV.
   (d) If there is not a working-day, it's a holiday and weather is not Rainy Then, Go to picnic.
7. Create decision tree for the following case study:
   (a) The number of vacation days depends on age and years of service. Every employee receives at least 22 days. Additional days are provided according to the following criteria:
   (b) Only employees younger than 18 or at least 60 years, or employees with at least 30 years of service will receive 5 extra days.
   (c) Employees with at least 30 years of service and also employees of age 60 or more, receive 3 extra days, on top of possible additional days already given.

12. Which of the following is an element of a data dictionary?
    (a) Data Types (b) Data Flows
    (c) User Manuals (d) Network Layout
13. One advantage of a data dictionary is:
    (a) Increases redundancy
    (b) Reduces clarity
    (c) Helps ensure consistency in data
    (d) Prevents documentation
14. One disadvantage of a data dictionary is:
    (a) Easy to maintain
    (b) May require additional maintenance effort
    (c) Reduces system complexity
    (d) Helps testing
15. The main objective of input design is:
    (a) To write user manuals
    (b) To determine data entry methods
    (c) To reduce screen size
    (d) To create system memory
16. A good output design should be:
    (a) Clear and easy to understand
    (b) Hard to interpret
    (c) Complex and detailed
    (d) Always in tabular form

Answers

1. (b)	2. (c)	3. (b)	4. (d)	5. (b)	6. (c)	7. (a)	8. (c)
9. (b)	10. (a)	11. (b)	12. (a)	13. (c)	14. (b)	15. (d)	16. (a)

Practice Questions

Q.I Answer the following questions in short.

1. What is system analysis?
2. Define system design.
3. Why is analysis important before design?
4. What is a decision tree?
5. List any two uses of decision trees in system analysis.
6. What is a decision table?
7. What is a DFD?
8. Mention any two symbols used in DFD.
9. Name two types of DFDs.
10. What is the difference between logical and physical DFD?
11. What is a context diagram?

3. A decision tree is best used for:
   (a) Describing programming languages
   (b) Analyzing hierarchical decisions
   (c) Displaying images
   (d) Sending emails
4. In a decision tree, each leaf node represents:
   (a) A new process
   (b) A decision to be made
   (c) An input
   (d) A final outcome or action
5. A decision table helps in:
   (a) Storing financial data
   (b) Representing complex logic in tabular form
   (c) Creating charts
   (d) Data compression
6. The condition stubs in a decision table represent:
   (a) Actions to be taken
   (b) Rules
   (c) Inputs or conditions
   (d) Output files
7. What does a DFD represent?
   (a) Flow of data in a system
   (b) The physical layout of hardware
   (c) Testing reports
   (d) Class hierarchy
8. Which of the following is not a type of DFD?
   (a) Physical DFD
   (b) Logical DFD
   (c) Abstract DFD
   (d) Both A and B are valid types
9. The highest level of DFD is called:
   (a) Level 1 DFD
   (b) Context Diagram
   (c) Flowchart
   (d) Pseudocode
10. A Level 1 DFD breaks down the system into:
   (a) Sub-processes
   (b) Inputs and Outputs only
   (c) Final screens
   (d) Class diagrams
11. A data dictionary contains:
   (a) Code for a project
   (b) Definitions and formats of data elements
   (c) Software bugs
   (d) Algorithms

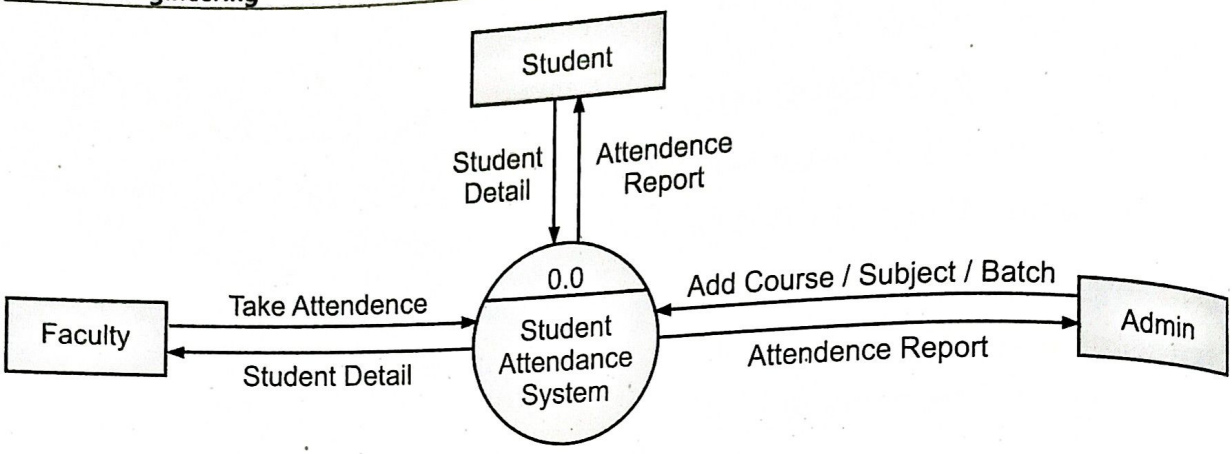


Fig. 4.21: Level 0

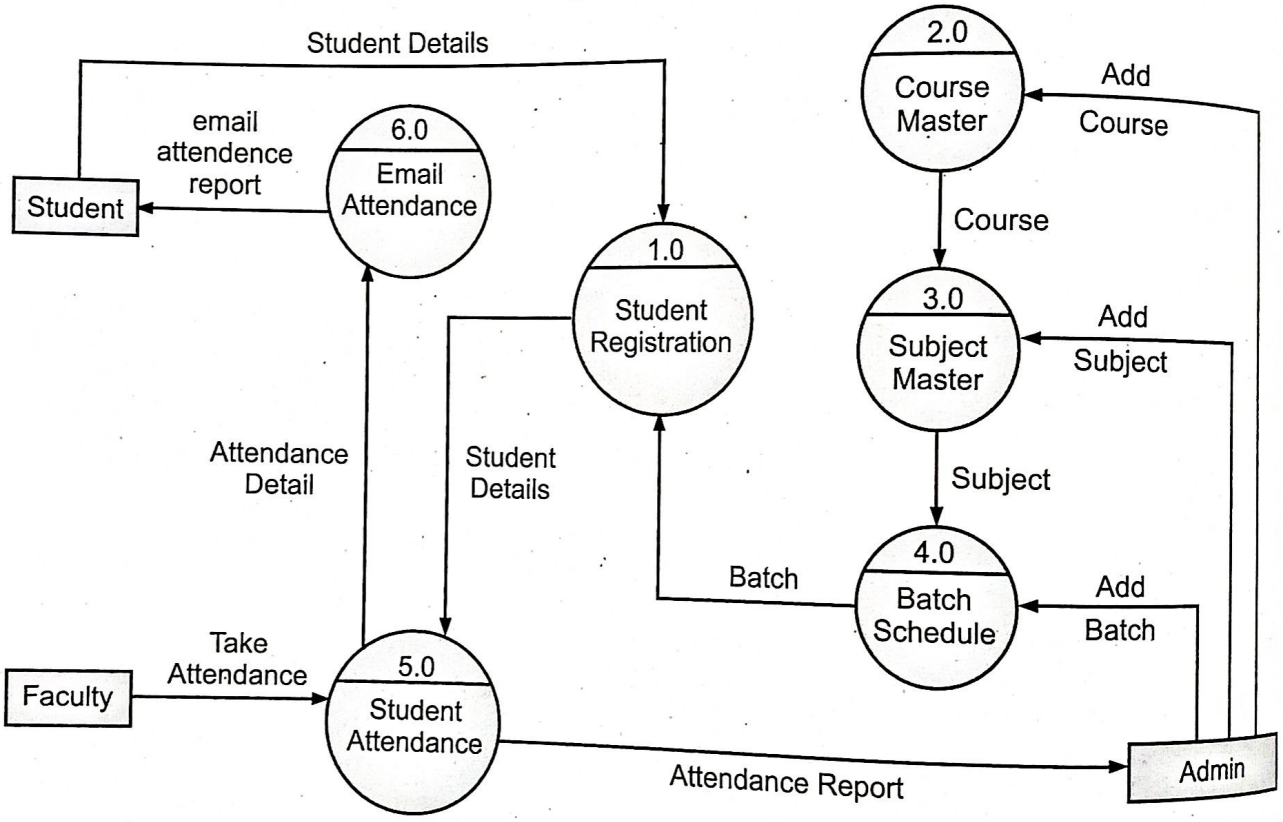


Fig. 4.22: Level 1

Check Your Understanding

1. What is the purpose of system analysis?
   (a) Writing code
   (b) Identifying what a system should do
   (c) Testing the software
   (d) Creating user manuals
2. System design involves:
   (a) Analyzing errors
   (b) Creating advertisements
   (c) Describing how the system will work
   (d) Cost estimation

4.6.2 Data Flow Diagram (DFD) for a Library Management System

Level 0 (Context Diagram)

Process:

- Delivery of Book

External Entities:

- Student
- Library Card System Info

Data Flows:

- **Student → Delivery of Book:** Book request.
- **Delivery of Book → Library Card System Info:** Book request.
- **Library Card System Info → Delivery of Book:** Display of book.

Level 1 DFD

Process 1: 1.0 Delivery of Book

- **Inputs:**
  - Book request (from Student).
  - Titles (from Process 2.0 Search Topic).
- **Outputs:**
  - Books, Authors, List of Titles (to Books entity and List of Titles store).
- **Data Stores/Entities Accessed:**
  - Books.
  - List of Titles.

Process 2: 2.0 Search Topic

- **Inputs:**
  - Base on topic (from external source/entity).
- **Outputs:**
  - Titles (to Delivery of Book process).
  - Demanded book info (to Delivery of Book process).
- **Data Stores Accessed:**
  - Books (based on topic).

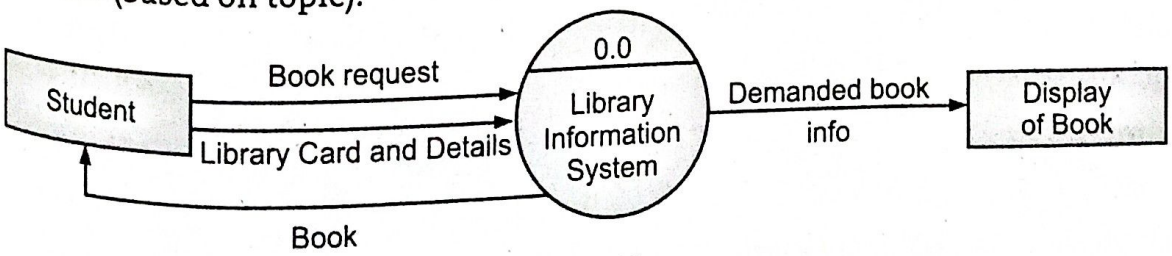


Fig. 4.18

4.3.1 Elements of Data Dictionary

The common elements of a Data Dictionary include:

1. **Data Element Name:** The unique name identifying the data item.
2. **Alias or Synonyms:** Alternate names for the data element (if any).
3. **Description:** A brief explanation of the data element's purpose or meaning.
4. **Data Type:** The type of data (e.g., integer, float, string, date).
5. **Length:** The maximum number of characters or digits allowed.
6. **Default Value:** The initial value if no other value is supplied.
7. **Allowed Values (Domain):** The set or range of valid values the data can take.
8. **Source:** Where the data originates from (input, calculated, etc.).
9. **Usage:** How and where the data is used in the system.
10. **Relationships:** Dependencies or associations with other data elements.

4.3.2 Advantages and Disadvantages of Data Dictionary

Advantages of Data Dictionary:

1. **Improves Data Consistency:** Ensures uniform definitions and uses of data elements.
2. **Eases Communication:** Provides a common understanding among developers, analysts, and users.
3. **Facilitates Maintenance:** Simplifies system updates by providing clear information about data.
4. **Supports Documentation:** Acts as part of system documentation, useful during audits and training.
5. **Enhances Data Quality:** Helps detect and prevent inconsistencies and errors.

Disadvantages of Data Dictionary:

1. **Time-Consuming to Create and Maintain:** Requires effort to establish and regularly update.
2. **Complexity for Small Projects:** May be too detailed or unnecessary for smaller systems.
3. **Dependency:** Teams may overly rely on it and neglect other documentation or system logic.
4. **Initial Cost and Training:** Involves training and resource investment during early adoption.

4.4 INPUT AND OUTPUT DESIGN

4.4.1 Input Design

1. **Form Design:**

Form design focuses on how data entry screens are created to collect information from users efficiently and accurately.

Principles of Effective Form Design:

- Clear field labels and instructions.
- Logical grouping of related fields.
- Use of input controls like dropdowns, checkboxes, date pickers.
- Highlighting required fields.
- Immediate error feedback for invalid inputs.

Best Practices:

- Keep it simple and concise.
- Align labels and fields properly.
- Use appropriate field sizes.
- Provide examples or tooltips.

2. Web Forms Design

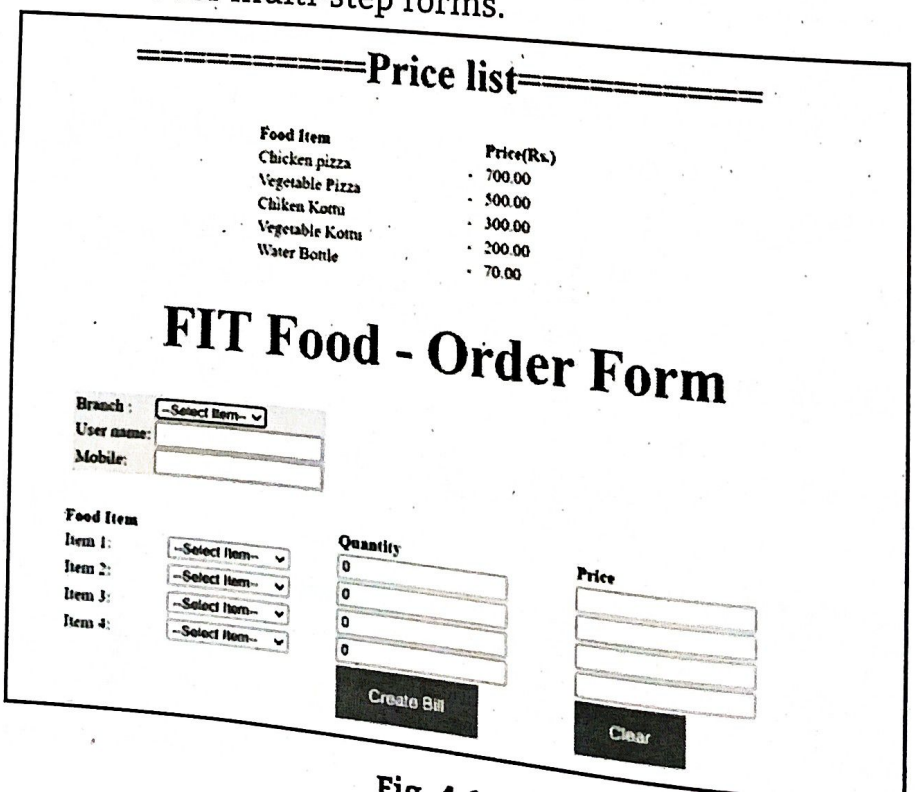
- Web form design involves creating forms for online data collection, optimized for usability, accessibility, and performance.

Key Considerations:

- **Responsive Design:** Forms should work across devices (desktop, tablet, mobile).
- **Security:** Input validation to prevent SQL injection, XSS attacks.
- **Accessibility:** Support screen readers, keyboard navigation, and use ARIA labels.
- **Performance:** Use AJAX for real-time validation and dynamic loading.

Example Features:

1. Auto-fill suggestions.
2. Captcha for spam protection.
3. Progress indicators for multi-step forms.



4.4.2 Output Design

1. **Characteristics of Good Output Design:**

- Good output design ensures that the information provided to users is useful, timely, and easy to interpret.

Key Characteristics:

1. **Clarity:** Information should be clearly formatted and labelled.
2. **Relevance:** Only essential information should be displayed.
3. **Timeliness:** Output should be available when needed.
4. **Accuracy:** Data must be reliable and correct.
5. **Accessibility:** Easy for all users to access and understand.

2. **Output Technologies:**

- Various technologies are used to present outputs to users:

Types of Output Technologies:

1. **Visual Outputs:** Screens, dashboards, graphs.
2. **Printed Outputs:** Reports, invoices, summaries.
3. **Audio Outputs:** Alarms, spoken messages (used in special systems).
4. **Electronic Outputs:** Emails, SMS alerts, downloadable files (PDF, CSV).

3. **Output Design Objectives:**

- The main goal is to provide meaningful, accurate, and actionable information to the end users.

Objectives:

- Support business operations and decision-making.
- Improve readability and presentation.
- Reduce information overload.
- Ensure secure and authorized access to outputs.
- Integrate with input and processing modules.

4. **Report Design:**

- Reports are structured formats of output used for formal presentation of information.

Types of Reports:

1. **Detail Reports:** Transaction-wise output (e.g., daily sales).
2. **Summary Reports:** Aggregated data (e.g., monthly revenue).
3. **Exception Reports:** Highlight abnormalities (e.g., failed transactions).
4. **Dashboard Reports:** Visual reports with charts, KPIs.

Design Principles:

- Use headers and footers appropriately.
- Group related information.
- Ensure alignment and spacing.
- Include timestamps and page numbers.

5. Screen Design:

- Screen design refers to how information and controls are arranged on a computer or mobile screen for interaction.

Principles of Good Screen Design:

- Consistency in layout, fonts, and navigation.
- Visibility of important information and buttons.
- Feedback after user actions (like form submission).
- Error messages those are informative and non-technical.
- Minimalism to avoid clutter.

Elements:

- Navigation menus.
- Input fields.
- Data displays (tables, charts).
- Action buttons (Submit, Cancel).

Product Details:

Table 4.5

Sr.	Product ID	Product Description	HSN/ SAC	Rate ₹	Quantity	Amount ₹	Discount %	Amount ₹
1.	PBT001	Dell Inspiron 1050	84713010	25,000	2	50,000	2%	49,000
2.	PBT002	Lenovo 5125-I	84713010	28,000	2	56,000	3%	54,320
3.	PBT004	Logitech Mouse Wireless	847330	1,200	2	2,400	2%	2,352
4.	PBT005	Logitech Keyboard	847330	2,500	2	5,000	3%	4,850
5.	PBT014	Zebronics Keyboard	847330	2,300	2	4,600	10%	4,140
				Total	10	1,18,000	3,338	1,14,662

4.5 PSEUDOCODE

Definition:

- Pseudocode is a textual representation of an algorithm, written in a structured, language-independent way to express logic clearly.

Purpose:

- To plan and understand algorithms before coding.
- To bridge the gap between requirements and implementation.
- To enhance communication between technical and non-technical teams.

Basic Syntax Elements:

- START/END
- INPUT, OUTPUT
- IF, ELSE, ENDIF
- FOR, WHILE, DO, ENDWHILE
- Variable assignments: x ← 10

For Example,

```
START
  INPUT num1, num2
  IF num1 > num2 THEN
     PRINT "num1 is greater"
  ELSE
     PRINT "num2 is greater"
  ENDIF
END
```

Advantages of Pseudocode:

1. Easier to understand than actual code.
2. Quick to write and modify.
3. Focuses on logic, not syntax.
4. Useful for teaching, design, and code reviews.

Limitations:

1. Not standardized.
2. Cannot be executed by a machine.
3. Needs conversion to code before use.

4.6 CASE STUDIES

4.6.1 Case Study : Hotel Management System

1. **Introduction:**

- A hotel management system is designed to streamline the administrative and operational tasks of a hotel. It helps in managing reservations, check-ins, check-outs, billing, room allocation, and customer service, all through a centralized software interface. This case study focuses on automating a mid-sized hotel's day-to-day operations to improve efficiency, accuracy, and customer satisfaction.

Data Stores:

- Customer Details
- Room Information
- Booking Records
- Payment Records

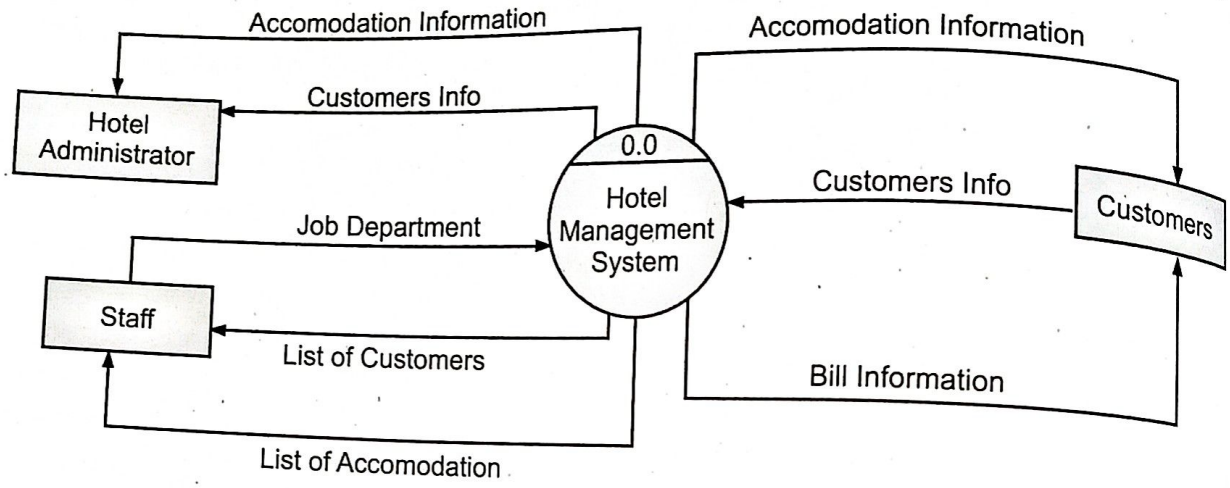


Fig. 4.16: level 0

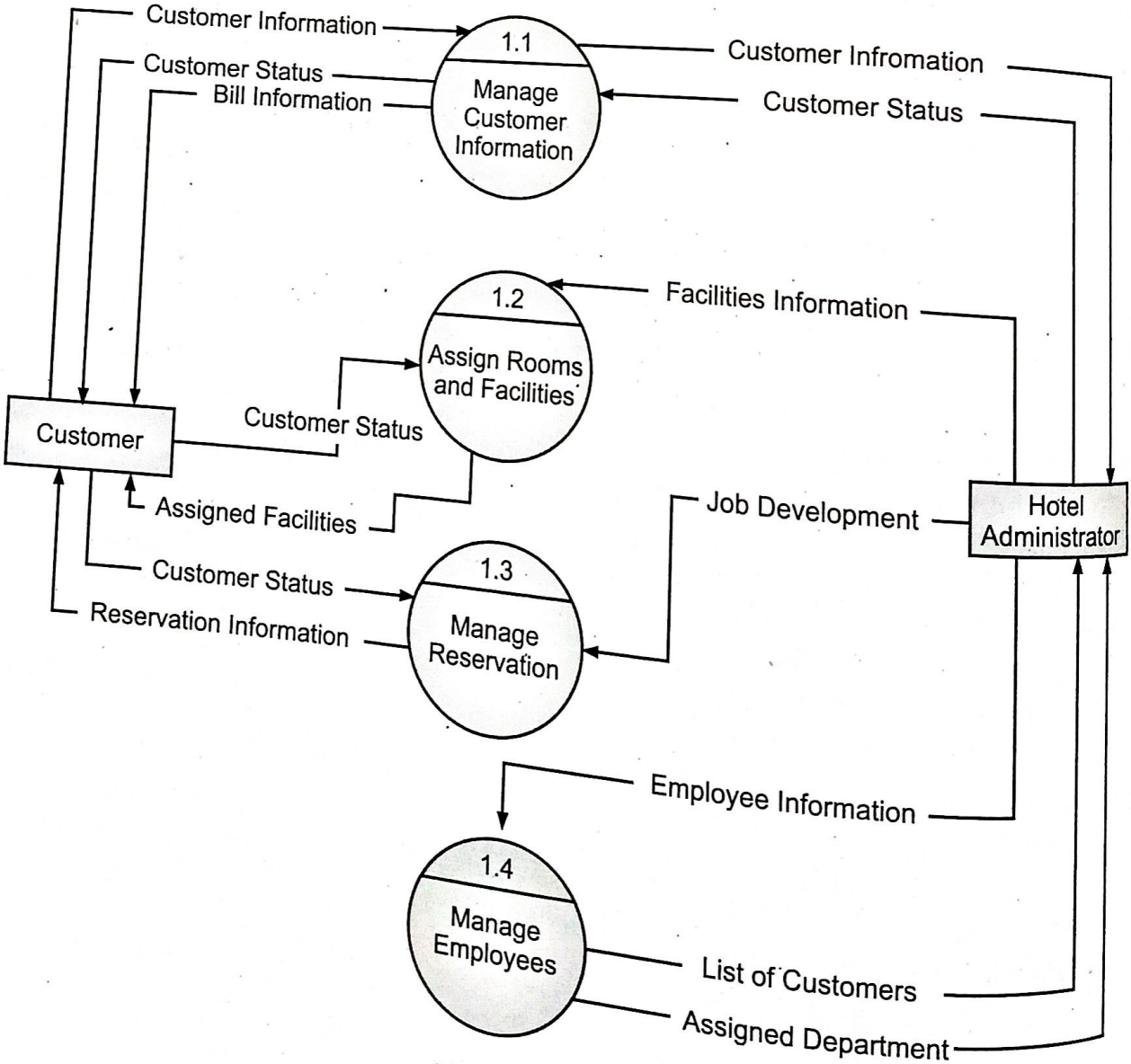


Fig. 4.17: Level 1

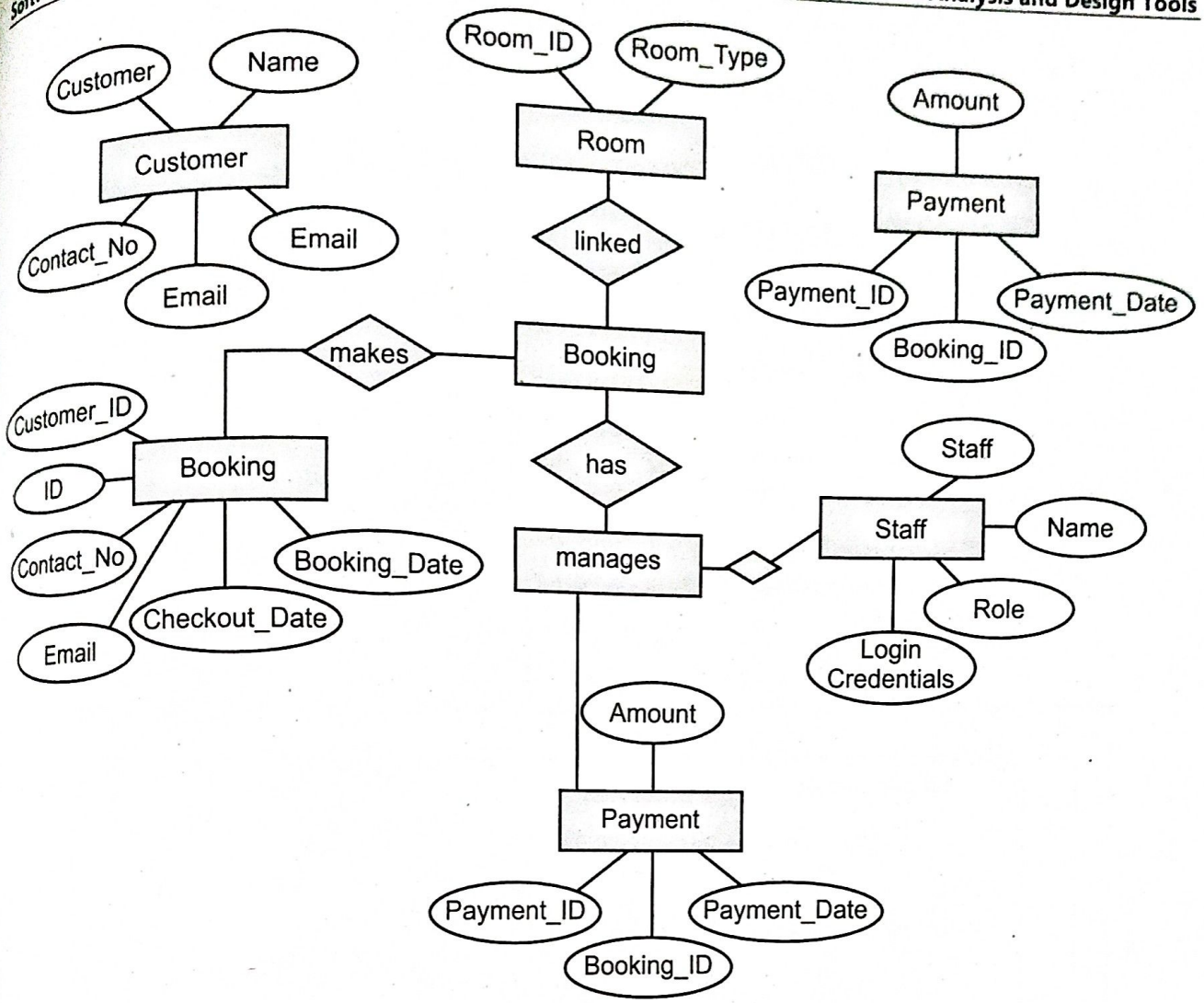


Fig. 4.15

5. **Data Flow Diagram (DFD):**

Level 0 (Context Diagram):

- Shows the Hotel Management System as a single process interacting with Customer, Staff, and Database.

Level 1 DFD:

- **Process 1: Manage Booking**
  - **Input:** Booking request from customer.
  - **Output:** Booking confirmation, Room update.
- **Process 2: Manage Check-in/Check-out**
  - **Input:** Customer arrival/departure.
  - **Output:** Updated room status.
- **Process 3: Manage Billing**
  - **Input:** Check-out details.
  - **Output:** Bill generation, payment receipt.
- **Process 4: Generate Reports**
  - **Input:** Admin request.
  - **Output:** Sales, occupancy, staff reports.

2. **Existing System:**

- The existing system is mostly manual and paper-based. Reservations are recorded in registers, room allocations are done by the front desk manually, and billing is handled using spreadsheets. This leads to multiple issues such as:
  - Double-booking of rooms.
  - Manual calculation errors in billing.
  - Difficulty in retrieving customer records.
  - Lack of real-time availability updates.
  - Inefficient report generation.

3. **Proposed System Plan:**

- The proposed system is a computer-based Hotel Management System with the following features:
  - Room booking and cancellation.
  - Customer check-in and check-out tracking.
  - Billing and invoicing automation.
  - Staff management.
  - Daily, weekly, and monthly reports.
  - User access control for admin and receptionist.
  - Integration with payment gateways for online payments.

Technologies Used:

- **Frontend:** HTML, CSS, JavaScript.
- **Backend:** PHP/Python.
- **Database:** MySQL.

Goals:

- Reduce manual work and paperwork.
- Eliminate booking conflicts.
- Automate billing and reporting.
- Provide a user-friendly interface for staff.

4. **ER Diagram:**

Entities:

- **Customer:** Customer_ID, Name, Address, Contact_No, Email.
- **Room:** Room_ID, Room_Type, Room_Status, Room_Rate.
- **Booking:** Booking_ID, Customer_ID, Room_ID, Booking_Date, Checkin_Date, Checkout_Date.
- **Payment:** Payment_ID, Booking_ID, Amount, Payment_Mode, Payment_Date.
- **Staff:** Staff_ID, Name, Role, Login_Credentials.

Relationships:

- One customer can have many bookings.
- One booking is linked to one room.
- Each booking has one payment.
- Staff manages bookings and payments.