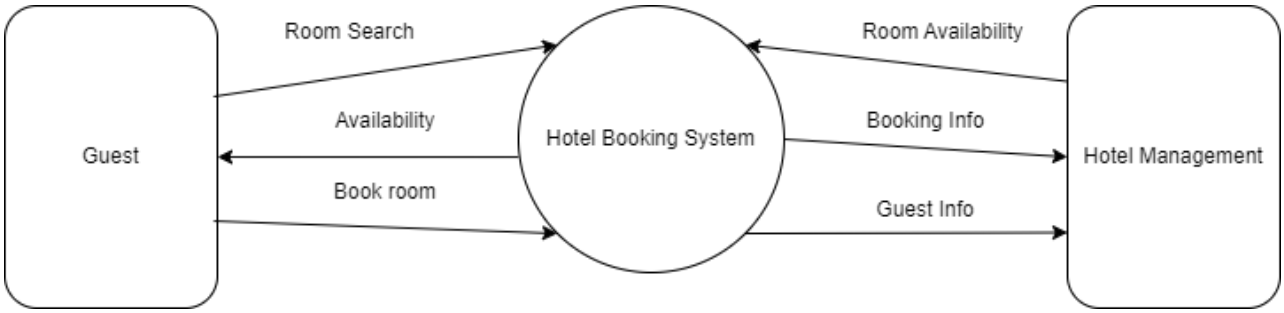
Data Flow Diagrams



A **Data Flow Diagram (DFD)** is a graphical tool used to visualise how data moves through an information system. It represents the flow of data between processes, data stores, and external entities, showing how information enters, transforms, and exits the system.

What is a Data Flow Diagram?

A **Data Flow Diagram** provides a structured way to analyse and understand the flow of information within a system by focusing on:

1. **Processes**: The operations or functions within the system that transform incoming data into outputs.
2. **Data Stores**: Locations where data is stored for future use (e.g., databases or files).
3. **External Entities**: Sources or destinations for data that interact with the system but are not part of it (e.g., a customer).
4. **Data Flows**: Represent the movement of data between processes, data stores, and external entities.

DFDs are typically created at different levels of detail, ranging from a broad overview (Level 0 DFD) to more detailed layers (Level 1, Level 2, etc.), which dive deeper into each process.

Why Use a Data Flow Diagram?

For a hotel booking system, a DFD is essential to:

- **Identify Data Movement**: Show how data moves within the booking, room availability, and payment processes.
- **Streamline System Design**: Clarify the inputs and outputs, making it easier to design an efficient and user-friendly system.

- **Improve Communication**: Simplify complex system interactions, helping stakeholders understand data flow without requiring deep technical knowledge.

Steps to Create a Data Flow Diagram for a Hotel Booking System

Step 1: Define the System Boundary

First, identify the boundaries of the system. For a hotel booking system, these boundaries might include:

- **External Entities**: Individuals or systems interacting with the booking system, such as **Customers**, **Payment Gateways**, and **Hotel Management**.
- **Main Processes**: Key activities within the system, such as **Search Room Availability**, **Make a Booking**, **Process Payment**, and **Update Booking Status**.
- **Data Stores**: Where the system stores essential information, like **Customer Data**, **Room Data**, **Booking Data**, and **Payment Records**.

Step 2: Create a Level 0 (Context) Diagram

The Level 0 DFD provides a high-level view of the hotel booking system, showing only the primary processes and data flows. In this level, the entire booking system is represented as a single process with data flows connecting to external entities.

For the hotel booking system, the Level 0 DFD might look like this:

- **External Entities**:
    - **Customer**: Searches for rooms, makes bookings, and receives booking confirmation.
    - **Payment Gateway**: Processes payments for bookings made by the customer.
    - **Hotel Management**: Views and manages booking information.
- **Primary Process (Level 0)**:
    - **Hotel Booking System**: The central system where all customer interactions and booking processes happen.

Step 3: Create a Level 1 DFD

A Level 1 DFD breaks down the main process into sub-processes, showing a more detailed flow of data between them. For a hotel booking system, the Level 1 DFD might include these sub-processes:

1. **Search Room Availability**: Customer sends a search request, and the system retrieves available rooms from the **Room Data** store.

2. **Make a Booking**: The customer selects a room and confirms the booking. The system checks room availability and updates **Booking Data**.
3. **Process Payment**: The system forwards payment information to an external **Payment Gateway** and, upon approval, stores the payment record.
4. **Send Confirmation**: Once payment is confirmed, the system sends a booking confirmation to the customer and updates **Booking Data**.

Each sub-process in the Level 1 DFD will have data flows connecting it to data stores and external entities.

Step 4: Identify and Connect Data Flows

Define data flows that connect processes, data stores, and external entities. Data flows should have descriptive names to indicate the type of data being transferred. For the hotel booking system, example data flows might include:

- **Availability Request** and **Available Rooms** between **Customer** and **Search Room Availability** process.
- **Booking Request** and **Booking Confirmation** between **Customer** and **Make a Booking** process.
- **Payment Details** sent from **Make a Booking** process to the **Payment Gateway**.
- **Confirmation** from **Process Payment** process to **Customer**.

Step 5: Create a Level 2 DFD (Optional)

If needed, create a Level 2 DFD to provide even more detail by breaking down specific processes in the Level 1 DFD. For instance, within the **Process Payment** process, you could show the steps for validating payment information, communicating with the payment gateway, and confirming payment.

Example DFD for a Hotel Booking System

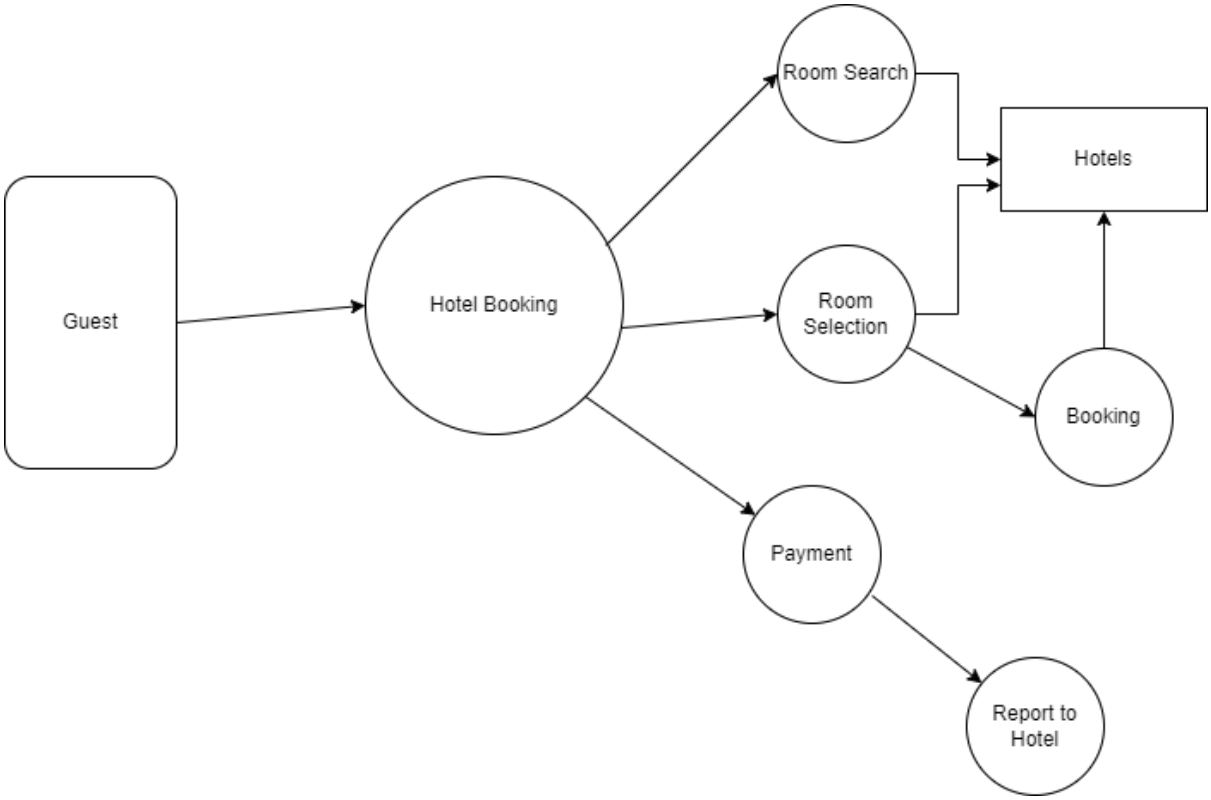
Here's how a DFD might look at different levels for the hotel booking system:

Level 0 DFD (Context Diagram)

- **Customer** ⇆ **Hotel Booking System** (data flows include “Search Request,” “Booking Request,” “Confirmation”).
- **Payment Gateway** ⇆ **Hotel Booking System** (data flows include “Payment Request” and “Payment Confirmation”).
- **Hotel Management** ⇆ **Hotel Booking System** (data flow includes “Booking Data” access).

Level 1 DFD (Detailed Overview)

1. **Search Room Availability**
    - Inputs: **Search Request** from Customer
    - Outputs: **Available Rooms** to Customer
    - Data Store: **Room Data**
2. **Make a Booking**
    - Inputs: **Booking Request** from Customer, **Available Rooms** from Room Data
    - Outputs: **Booking Confirmation** to Customer
    - Data Stores: **Booking Data**, **Customer Data**
3. **Process Payment**
    - Inputs: **Payment Details** from Customer
    - Outputs: **Payment Confirmation** to Customer
    - External Entity: **Payment Gateway**
    - Data Store: **Payment Records**
4. **Send Confirmation**
    - Inputs: **Booking Confirmation** and **Payment Confirmation**
    - Outputs: **Confirmation Message** to Customer
    - Data Store: **Booking Data**



DFD Best Practices

When creating a DFD, keep in mind the following best practices:

- **Use Clear Labels**: Label each process, data flow, data store, and external entity clearly to avoid confusion.
- **Define Data Flows Precisely**: Each data flow should represent a specific type of data (e.g., "Search Request" rather than just "Request").
- **Avoid Too Much Detail Early On**: Start with a high-level view (Level 0), then add more details in subsequent levels if necessary.
- **Validate with Stakeholders**: Ensure the DFD aligns with how stakeholders envision the system's data flow and processes.

Conclusion

A Data Flow Diagram (DFD) is a powerful tool for visualising the movement of data within a system. By outlining processes, data stores, and relationships with external entities, a DFD helps clarify how the system will handle processes. Creating a DFD involves progressively detailing data flows from a high-level view (Level 0) to more granular levels (Level 1, Level 2), providing a comprehensive map of system interactions that guide the database design and software development process.