

2.0 Introduction

System analysis is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all components work efficiently to achieve their results.

The various tasks in the system analysis include the following:

- **Requirement gathering:** Gathering requirements from stakeholders, including guests, hotel staff, and management, to understand their needs and expectations from the system.
- **Functional analysis:** Identifying the functions and features required for the system to meet the gathered requirements.
- **Data analysis:** Analyzing the data required for the system, such as guest information, room details, and billing information.
- **Process modeling:** Creating process models to understand how the system works, including how guests make reservations, check in, check out, and request services.
- **Use case analysis:** Identifying and analyzing the use cases of the system, including scenarios such as guest reservations, check-in, check-out, and room service.
- **Performance analysis:** Evaluating the performance of the system, including its response time, reliability, and scalability.
- **Security analysis:** Analyzing the security of the system, including its vulnerability to hacking, data breaches, and other security threats.
- **User interface analysis:** Evaluating the usability of the system's user interfaces, including the web and mobile applications used by guests and hotel staff.
- **Testing:** Conducting testing to identify and resolve any bugs or issues with the system's functionality.

2.1 Functional Requirements

Functional requirements are a key aspect of system analysis, defining the features and capabilities required for the system to meet the needs of its users. For a Hotel Management System, the following functional requirements may be necessary:

- **User registration and authentication:** The system must allow users to create accounts, providing basic information such as name, email, and phone number. The system must also authenticate users to ensure that only authorized users can access the system.
- **Reservation management:** The system must allow guests to make, modify, and cancel reservations. It must provide information on room availability, rates, and booking confirmations.

- **Check-In and Check-Out:** The system must manage the check-in and check-out processes, assigning rooms to guests and updating room status accordingly.
- **Payment processing:** The system must process payments for reservations and other services, allowing guests to pay using various payment methods, such as credit cards or digital wallets. The system must also handle payment disputes and issue refunds when necessary.
- **Room service management:** The system must allow guests to request room services, and hotel staff to manage and fulfill these requests.
- **Housekeeping and maintenance:** The system must manage housekeeping schedules and maintenance requests, ensuring that rooms are cleaned and maintained properly.
- **Customer support:** The system must provide customer support to guests, allowing them to report any issues or problems with the system, such as reservation disputes or service concerns.

2.2 Non-Functional Requirements

Non-functional requirements are the quality requirements that stipulate how well the web application does what it must do:

- **Performance:** The system must be able to handle a large volume of reservations and service requests, processing them quickly and efficiently. The system must also handle high traffic and user loads, without slowing down or crashing.
- **Availability:** The system must be available 24/7, with minimal downtime or disruptions. This is important for ensuring that users can access the system when they need to, without any delays or interruptions.
- **Reliability:** The system must be reliable, with minimal errors or bugs. The system should also be able to recover from any errors or failures quickly and automatically, without causing any significant disruptions or downtime.
- **Security:** The system must be secure, protecting user information and payment data from unauthorized access or breaches. This may include implementing encryption, firewalls, and other security measures to prevent hacking and other attacks.
- **Usability:** The system must be easy to use, with a user-friendly interface that is easy to navigate and understand. This is important for ensuring that users can use the system effectively, without any confusion or frustration.
- **Compatibility:** The system must be compatible with various devices and operating systems, allowing users to access the system from their preferred devices, such as smartphones, tablets, and laptops.
- **Scalability:** The system must be scalable, able to handle an increasing number of users and service requests as the system grows. This is important for ensuring

that the system can continue to meet user needs and demands, even as the user base expands.

2.3.0 Data Flow Diagram (DFD) of Hotel Management System

A Data Flow Diagram (DFD) is a pictorial representation of the movement of data between external entities and the processes and data stores within a system. The context level DFD provides a high-level overview of the system. It shows the system as a single process and the interactions with external entities.

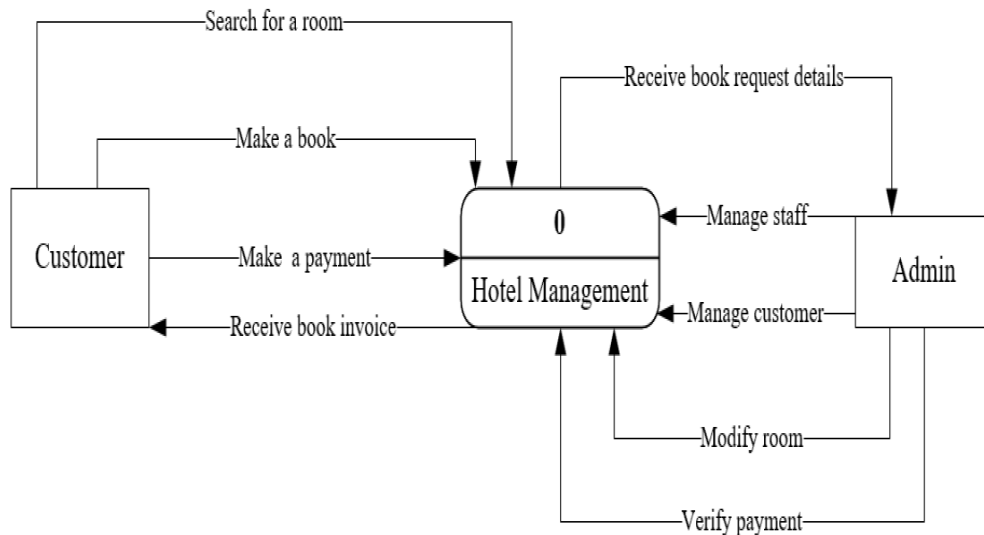


Fig 3: Zero-Level DFD

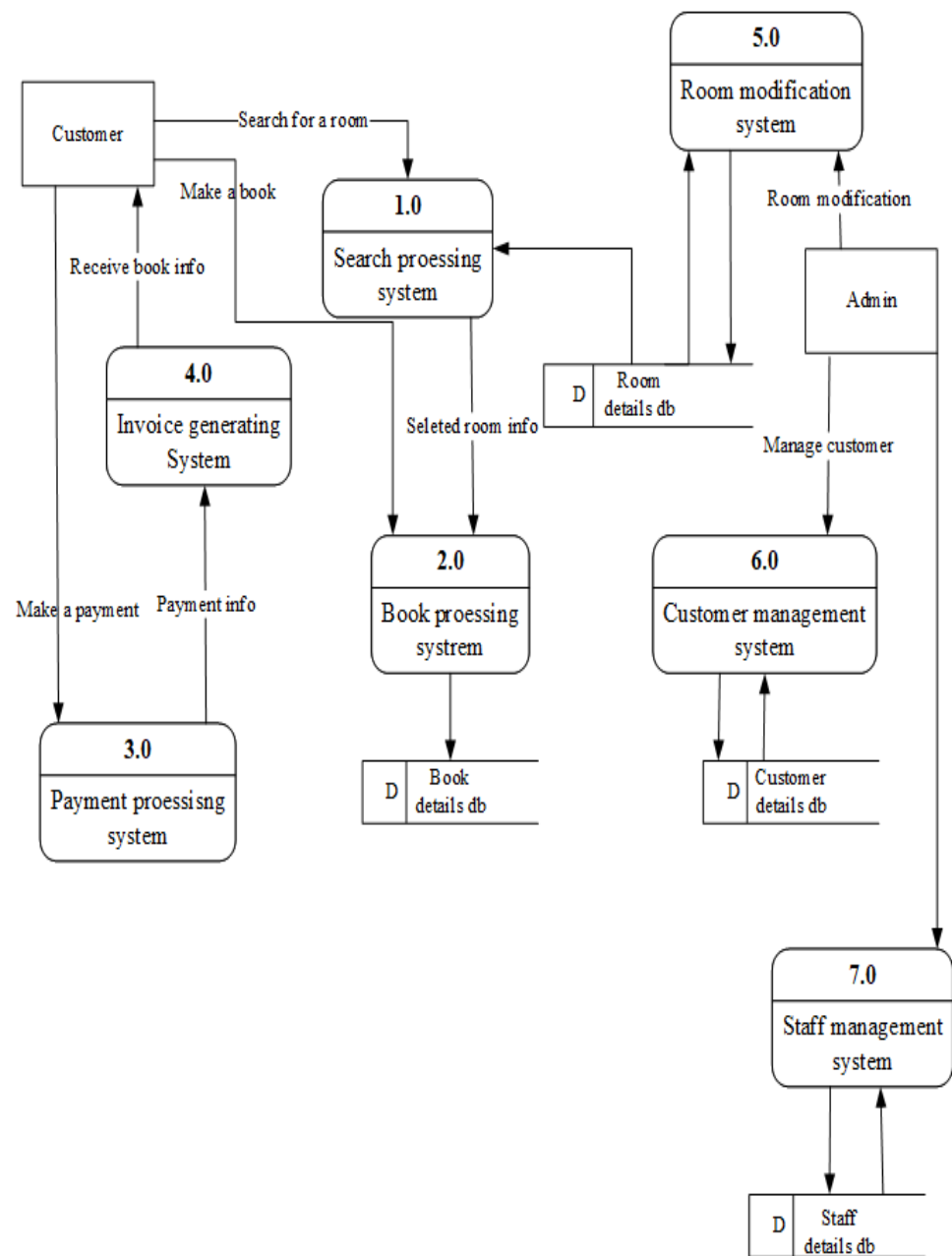


Fig 4: Level-0 DFD

2.3.1 Level-1 DFD

The Level 1 DFD provides a more detailed breakdown of the major processes in the system, illustrating how data flows between them.

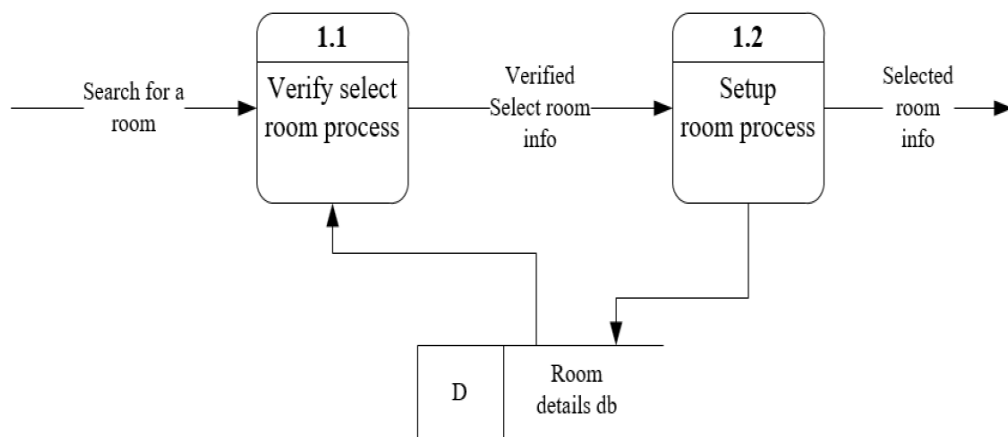


Fig 5: Level-1 DFD of Search Processing System

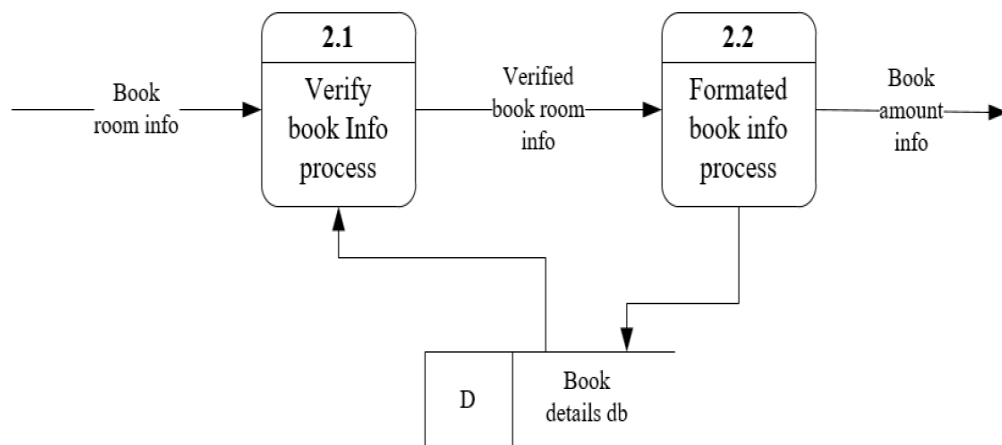


Fig 6: Level-1 DFD of Book Processing System

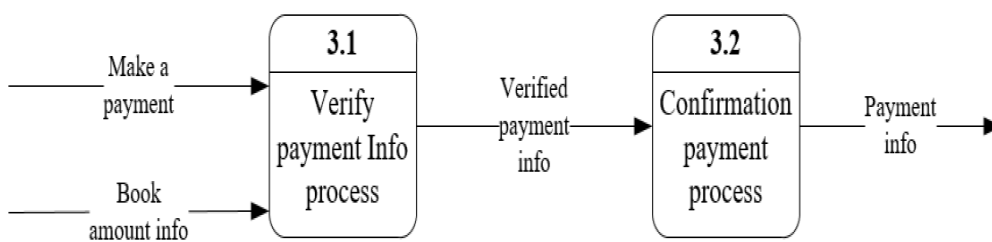


Fig 7: Level-1 DFD of Payment Processing System

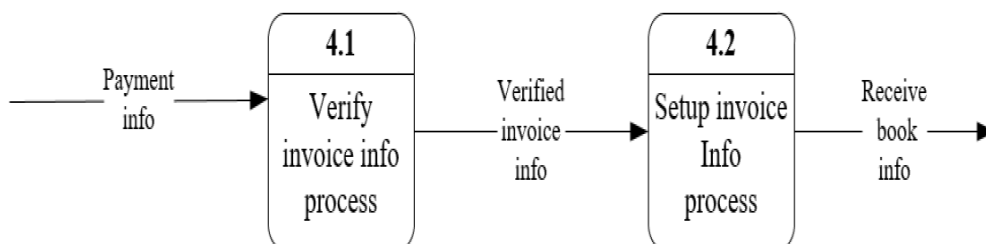


Fig 8: level-1 DFD of Invoice Generating System

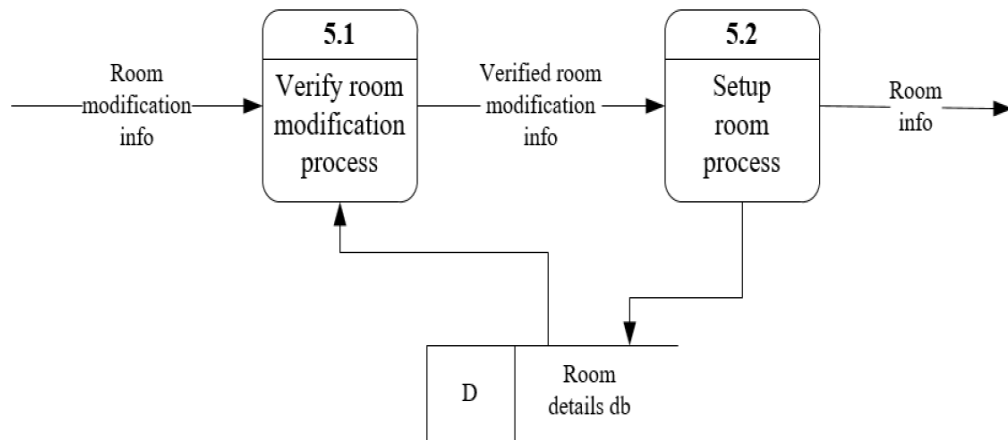


Fig 9: Level-1 DFD of Room Modification System

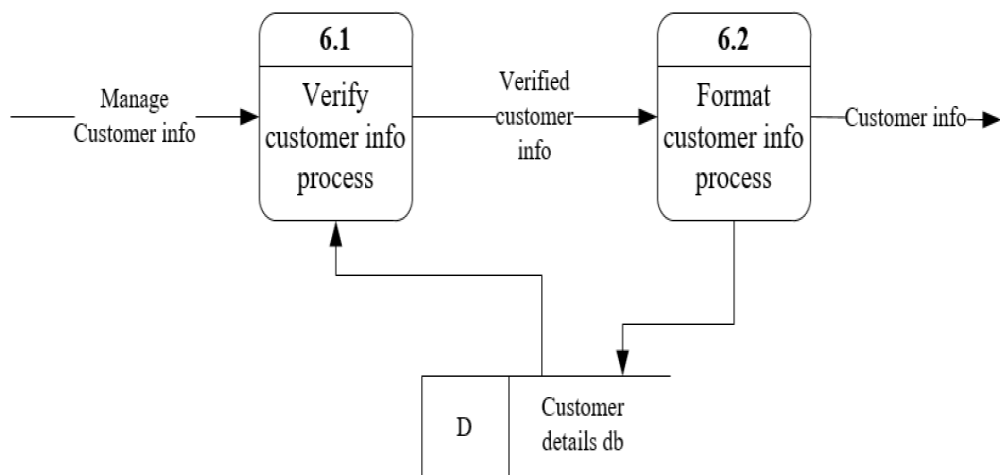


Fig 10: Level-1 DFD of Customer Management System

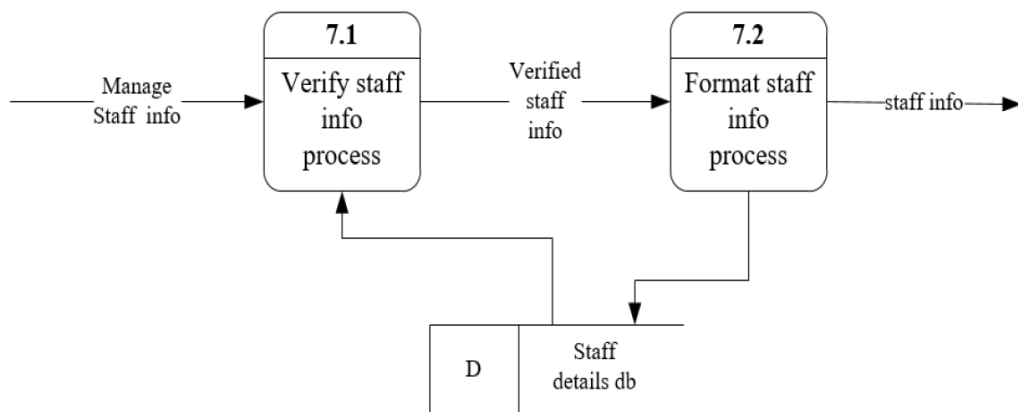


Fig 11: Level-1 DFD of Staff Management System

2.4 Decision Tree

A decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of a test, and each terminal node holds a class label. Decision trees method for defining complex relationships by describing decisions and avoiding problems in communication.

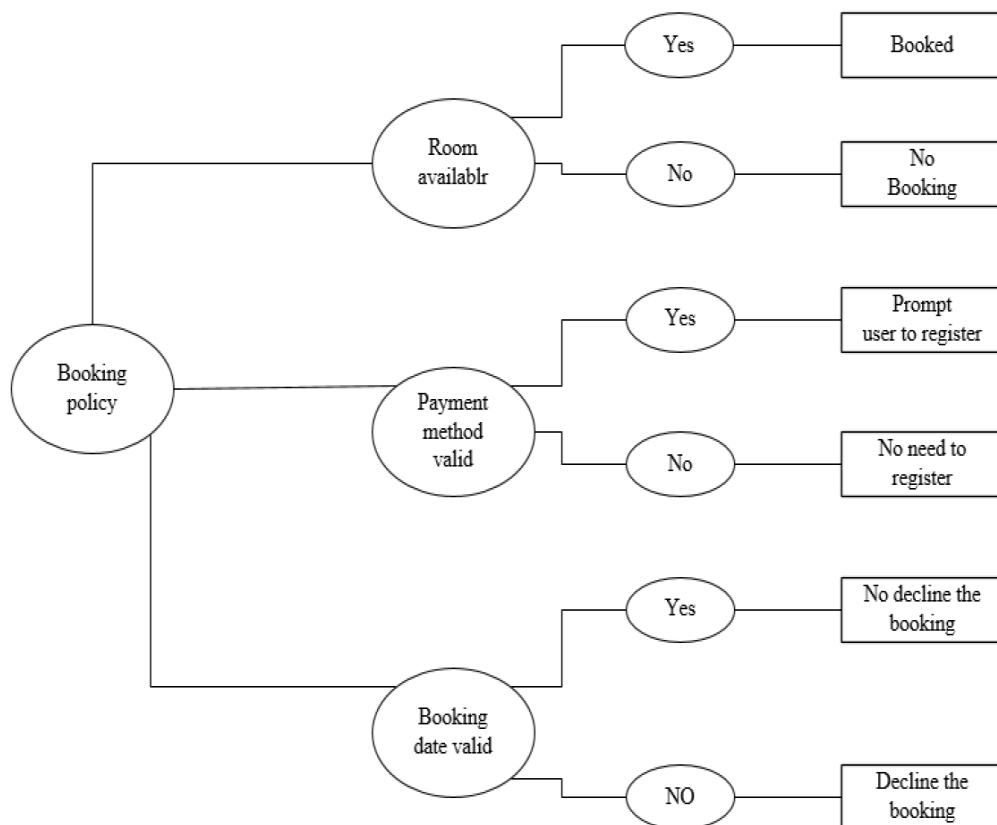


Fig 12: Decision Tree of Booking Policy

2.5 Decision Table

Decision tables are brief visual representation for specifying which actions to perform depending on given conditions. They are algorithms whose output is a set of actions. It is an excellent tool to use in both testing and requirement management.

Condition Stub	Condition Entries				
	1	2	3	4	5
Is the room available?	Yes	Yes	Yes	No	No
Is the booking date valid?	Yes	Yes	No	Yes	Yes
Is the request register?	Yes	No	Yes	Yes	Yes
Is the payment method valid?	Yes	Yes	Yes	Yes	No
Action Stub	Action Entries				
	1	2	3	4	5
Book the room	X	–	–	–	–
Decline the booking	–	–	X	X	X
Prompt user to register	–	X	–	–	–
Valid payment method	–	–	–	–	X

Table 1: Decision Table of Booking Room

2.6 Entity-Relationship (E-R) Diagram

An Entity-Relationship Diagram (ERD) is a visual representation of the data and relationships within a database. It shows how entities (objects or concepts) relate to each other within the system.

