Software Engineering - CS4443

Instructor: Dr. Manish Singh

Software Requirement Analysis

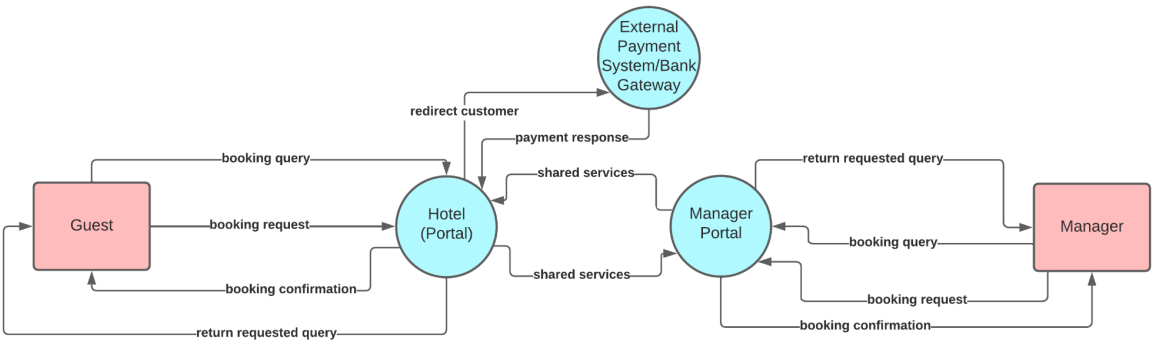
Group 08

Sharath Chandra S	ES18BTECH11016
Akashdeep Singh Kalra	CS18BTECH11003
Rohit Reddy K	ES18BTECH11010
Ravi Teja N	CS18BTECH11032
Seetha Sahithi	ES18BTECH11015

Software Requirement Analysis (SRA) for Hotel Reservation System (HRS)

Context Diagram:

A Context Diagram is something that defines the boundary between the software part and all its external entities such as inputs, outputs, sinks, etc., and hence gives the basic structural idea of the software. The following diagram shows the context diagram of our project which is a Hotel Reservation System:

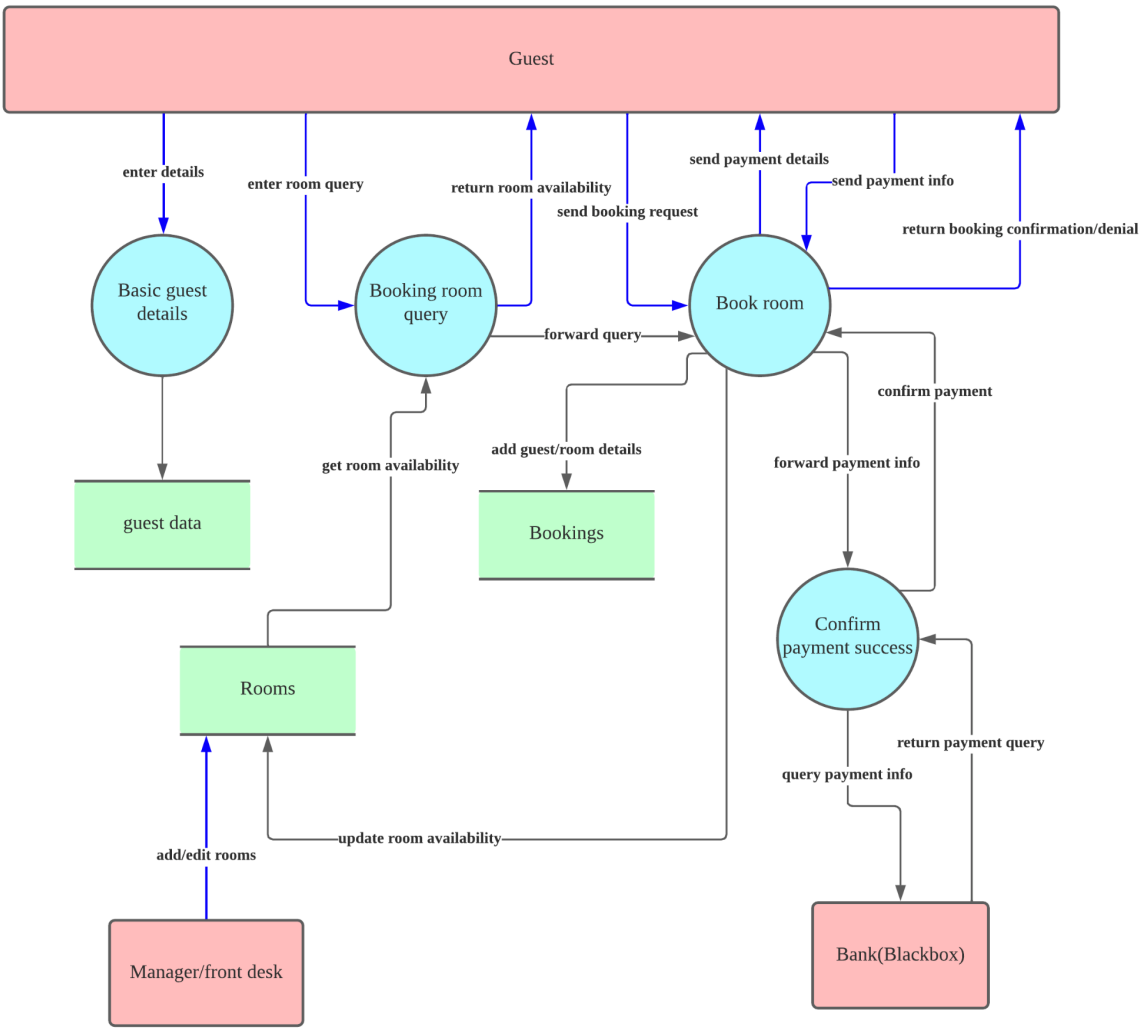


From the above diagram, one can see that there are two portals namely Hotel Portal and Manager Portal. From Hotel Portal, customers or Guests perform their required activities such as booking queries, booking requests get the responses for booking confirmation, return requested queries, etc., The Manager Portal works for Manager to perform certain operations such as booking queries, booking requests, etc., and gets the response from Manager Portal for return requested queries and booking confirmation. There definitely are certain shared services between Hotel Portal and Manager Portal. For payments, the External Payment System/Bank Gateway and Hotel Portal exchange information about payment responses and redirect the payment information for a specific customer.

Data Flow Diagram (DFD):

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.

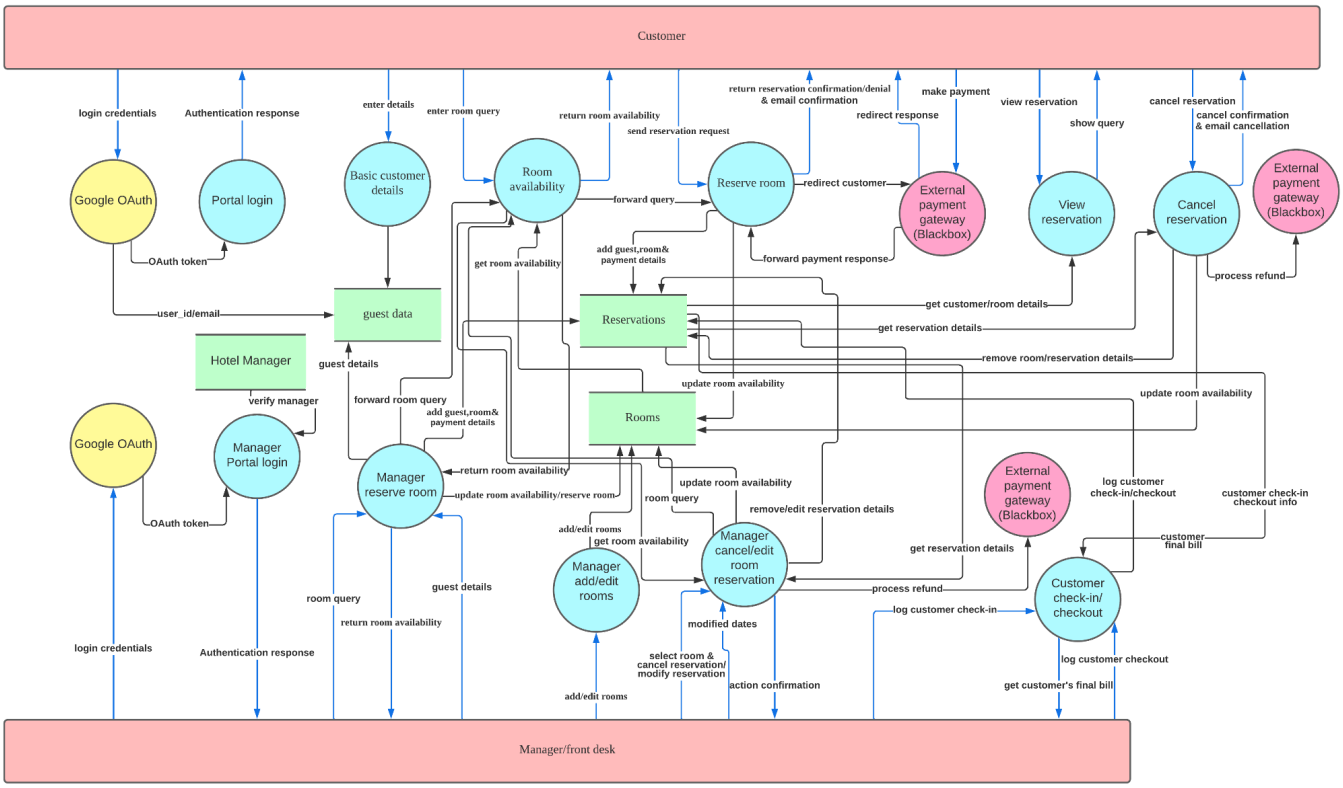
1. First DFD:



In the above DFD, the blue coloured arrows represent human-machine interaction and the black coloured arrows represent machine-machine interaction.
This DFD describes that the application has two portals: Manager Portal which is basically the front desk or receptionist and Customer Portal (Guest). The working procedure of the application goes in such a way that the customer can enter their details then they can query the room availability according to their needs, and if they decide to book from the available rooms, they're provided with payment information, and after they make the payment to the provided address, they share the reference id, and after

that, the payment is verified from the banks' server, on successful payment the customer is provided with the booking information.
On the manager side of things, the manager can add/edit room information like price, suite, capacity, etc.

2. Second DFD(We plan to use this):



Note: for a clearer view of the above click here to open the pdf.

In the above DFD, the blue coloured arrows represent human-machine interaction and the black coloured arrows represent machine-machine interaction.
The above shown DFD handles the use cases and exceptions in a more detailed manner. After the user has been authenticated, they're prompted to enter their details(if it's their first login), then they can query the room availability according to their needs, then if they decide to reserve from the available rooms they're redirected to an external payment gateway, which on successful payment gives some response.
Customers also have an option to view their existing reservations or cancel them, on successful cancellation of a reservation, a refund is processed.

On the manager side of things, after login (login is verified from Hotel Manager database), they can reserve a room for offline reservations, and they can cancel/modify

the reservations, they can also add/remove rooms or edit room information like price, suite, and capacity, etc. On a customer's visit to the hotel, the manager can log their check-in/checkout and get the customer's expenditure during the stay.

Why the Second DFD is better than the First DFD:

1) Google Authentication and Portal login: There is google authentication for user and manager portal login purposes in the final DFD. Then we can get and store user information in the database for further purposes.
2) Here, the customer can query his booking and even cancel it.
3) This DFD handles payment in a better way, as it redirects the customer to an external payment gateway instead of them having to make payment to the bank and sharing the payment reference id.
4) Managers can reserve rooms for offline customers who want to book their rooms by visiting the hotel directly.
5) Managers can modify the reservations(cancel or change the dates).
6) Managers can log the customer's check-in and checkout dates, and get information about the customer's expenditure during the stay.