

HOTEL AUTOMATION SOFTWARE

By:

Jigir Shah (CE-100)

Jinish Shah (CE-101)

Ketul Shah (CE-102)

Mitul Shah (CE-103)

LAB 1: REQUIREMENT SPECIFICATION

DATABASE REQUIREMENTS:

Database should be capable to store followings:

- Name
- Room type
- Arrival time
- Advance paid
- Approximate duration
- Food Details
- Token number
- Unique Identification Number

SYSTEM REQUIREMENTS:

➤ **R1: SYSTEM SHOULD PROVIDE FACILITY TO CHECK AVAILABILITY OF ROOM.**

- **DESCRIPTION:** Provides facility to check availability for various types of the rooms like AC, non AC, single or double.

Input : Details of room types as above.

Output: Display details for room or an apology message if room is not available.

➤ **R2: SYSTEM SHOULD PROVIDE FACILITY TO ALLOCATE A ROOM.**

- **DESCRIPTION:** If room is available then allocate the room as per customer's choice and update database by adding arrival time and other personal details and assign him a unique token number for further usage.

Present State: The person has already passed through the authentication process.

R2.1: System should provide facility to allocate a room to guest.

Input : Type of room, arrival time, approximate stay, advance paid and other personal details.

Output: Unique token number.

➤ **R3: SYSTEM SHOULD PROVIDE FACILITY TO GENERATE OCCUPANCY RATIO.**

- **DESCRIPTION:** Occupancy ratio is ration of occupied room divided by total rooms in hotel and according to occupancy ratio, if it's high than increase the rate by some amount else decrease or keep rates as it is.

Input : The month/duration for which you want to find occupancy ratio.

Output: Occupancy ratio and changes in tariffs.

➤ **R4: SYSTEM SHOULD PROVIDE FACILITY TO ACCEPT FOOD ORDER FROM GUEST.**

- **DESCRIPTION:** Provide Menu of Various types of Food available and maintain quantity as per orders.

R4.1: System should provide list of available items.

R4.2: System should generate the bill on basis of food type and quantity and add them to particular user's account known by token number.

Input: Food type and quantity.

Output: Bill for the order.

R4.3: System should keep track of total food available and do some action if quantity is below average.

Input: Inventory of food.

Output: Generate report for all type of food.

➤ **R5: SYSTEM SHOULD GENERATE FINAL BILL AT CHECK-OUT TIME.**

DESCRIPTION: Here the final bill will contain the amount payable by guest according to duration of stay and room type which also includes the restaurant billing amount and special discount if applicable.

Input : Token Number.

Output: Display Payable Amount.

➤ **R6: SYSTEM SHOULD ASK GUEST TO REGISTER IN 'FREQUENT GUEST' PROGRAM.**

DESCRIPTION: At check out time the guest can enroll himself for this program which will be helpful to him in future for getting special discounts.

R6.1: System should provide Unique Identification Number if guest is interested in program.

Input: User is interested in the program.

Output: Unique identification number.

LAB 2: SOFTWARE REQUIREMENTS ANALYSIS & SPECIFICATIONS (SRS)

1 INTRODUCTION:

The following are the specifications of the software required for "HOTEL AUTOMATION".

1.1 Purpose:

This software is used to control and manage Room Reservation System of 5-star hotel. This is stand-alone software used by Receptionists and some authenticated Hotel Staff only.

1.2 Scope:

- The software name is "HOTEL AUTOMATION SOFTWARE".
- The software maintains information about room types, availability of room, restaurant details.
- It also provides facility to store database of Guest, database of Restaurant and other information.
- So, this software is useful for Hotel to perform its work very easily and in efficient way.

1.3 Definitions, Acronyms & Abbreviations:

| Term/Abbreviation | Definition |
|-------------------------------|--|
| Occupancy Ratio | This is the ratio of occupied room and total rooms. So, that we can Increment or Decrement Rates of the room. |
| Frequent Guest Program | This is the special program only for those guest who are frequent and facilitate them with Special Discount. |

1.4 References:

We referred Textbook Pressman for some kind of understanding about various terms and also referred Rajib Mall and Internal Discussion of Team Members.

1.5 OverView:

This software provides facility to reserve rooms as well it also has facility to change room, it also has facility to give Special Discount to Frequent Guest.

This software has mainly three Databases as Customer's Personal Details, Restaurant Database and Database of Frequent Guest Information. So, We can Change, Update or Delete our Database Items as per requirement.

So, This is just Simple Overview of "Hotel Automation Software".

2 THE OVERALL DESCRIPTION

Here we describe the general factors that affect the Product and its requirements.

2.1 Product Perspective

The product is partially independent and self-contained. It requires the following interfaces:

2.1.1 System Interfaces:

None

2.1.2 Interfaces:

The medium of interaction between the software product and its users will be only terminal of the computers connected in LAN.

2.1.3 Hardware Interfaces:

The system will need a well configured printer to print the final Bill at check-out time.

2.1.4 Software Interfaces:

- 1) Any windows-based operating system.
- 2) MS Access 2007 for Database.
- 3) JAVA Language for coding system.

2.1.5 Communication Interfaces:

None

2.1.6 Memory Constraints:

At least 64MB RAM and up to 1GB space on hard disk.

2.1.7 Operations:

There are mainly two operations Required from the user side-

- Necessary insertions, updations and deletions are to be done by the Authenticated user.
- To view the information, necessary queries are to be fired by the Hotel Manager on the specified databases.

2.1.8 Site Adaptation Requirements:

None

2.2 Product Functions:

- Provide facility to log-in for enabling authorized access to the system.

- User can be able to add/modify/delete information about room, food and database of guest.

2.3 User Characteristics:

All users should have basic knowledge of computer and this software. There is no need to have any special knowledge.

2.4 Constraints:

Following are the constraints to limit the developer's Option:

- Reliability requirements
- Safety and security considerations

Unauthorized users should not be able to access or update the information as it is to be kept confidential for a Hotel only.

- Interface to the application

2.5 Assumptions and Dependencies:

This includes any changes that can affect the requirement of the software.

- If the database version or database software (Oracle, SQL) is not available with the user, the connectivity parameters need to be changed accordingly.

2.6 Apportioning of Requirements:

None (No requirement can be delayed).

3 SPECIFIC REQUIREMENTS

This section contains all the software requirements at a level of detail sufficient to enable designers to design and testers to test the system efficiently.

3.1 External Interfaces:

We don't require any external interface here.

3.2 Functions:

It defines the specific actions to be taken by the Software during accepting, processing and generating the outputs.

These specific actions are:

- The system shall do validity checks on the inputs:
It is important as the system is providing the facility to manage a room and enter personal details of guest. (e.g.: to prevent user to provide invalid values).
- The system shall check exact sequence of operation:
The sequence is important to maintain as the system needs to do calculations step by step for occupancy ratio and generating final bill at check-out time.
- The system shall generate responses to abnormal situations:

In the cases of overflow, invalid data type and error handlings, appropriate error messages and notifications need to be given.

- The system shall check the relationship of input and output.

3.3 **Performance Requirements:**

These are the numerical requirements (static and dynamic) for enhancing the performance of the software.

Static:

- No. of terminals required: No of guests + 3 (for Receptionist, Restaurant Manager, Hotel Manager and for each guest).
- No. of simultaneous users: Again same as number of terminals and at least three.
- Amount of information to handle: Mostly the amount of data would be in MB's.

Dynamic:

- The checking availability should be performed in 3-4 second.
- Generating a bill also should be performed in some seconds.
- Almost 70% of all functions should be done in less than 10 second.

3.4 Logical database required:

1. Customer Personal Database.

- This will be used very frequently and updating database will be performed whenever new guest comes.
- The attributes are – Name, Token Number, Arrival time and Room type.
- Here Token number will be used as primary key.

2. Restaurant Database.

- This database will be used whenever any guest orders food.
- The attributes are – Food quantity, Food type and Token number.
- Token number will be used as primary key.

3. Room Database.

- This Database will be used to store data about every room in hotel
- The attributes are – Room No., Room Type, Room Status, Room Rate, Guest Name(Foreign Key).

4. Database for 'Frequent guest' program.

- This database will be used whenever guest wants to participate in 'FREQUENT GUEST PROGRAM'.
- The attributes are – Name, Unique Identification Number.

- Unique Identification will be used as primary key.

4.5 Design Constraints:

None (As this software design is not affected by any hardware limitations).

4.6 Software System Attributes:

These are the attributes which are the implicit requirement of the system.

3.6.1 Reliability: Backups would be available to retain any data lost unintentionally (or due to connection breakdown).

3.6.2 Availability: No need to provide any breakpoint but restarting facility should be there.

3.6.3 Security: Checking the data integrity for critical variables, keeping login accounts for authorized users.

3.6.4 Maintainability: None

3.6.5 Portability:

It's high as-

- 100 % code is host independent.
- 0% components are host dependent.

4.7 Organizing the specific requirements:

Careful consideration is given on the organization of requirements, so that it is optimal for understanding.

3.7.1 System Mode:

None (no dependence on mode of operation)

3.7.2 User Class:

Different sets of functions are provided for different class of authorized users like:

- Insertions, Deletions, and Updations to be performed by Authenticated user only.
- Firing queries and viewing of details should also be done by Authenticated user.

3.7.3 Objects:

Objects share attributes and services. Object considered here is:

- Guest Object:
They share attributes like name, token number, room type and other relevant details.

They share functions such as Insertion, Deletion, Updation and Viewing.

3.7.4 Feature:

None (since requirements are not organized on the basis of external features required).

3.7.5 Stimulus:

None (since requirements are not organized on the basis of stimuli).

3.7.6 Response:

None (since requirements are not organized on the basis of response).

3.7.7 Functional Hierarchy:

None (since requirements are not organized into a hierarchy of functions depending on common inputs and outputs).

3.8 Additional Components:

None

5. CHANGE MANAGEMENT:

Here our software is for HOTEL AUTOMATION so we need requirement change in the tariffs in the hotel according to occupancy rate and if the new rooms are added then update database accordingly.

6. DOCUMENT APPROVAL:

Approver's Name: Miss Shruti Shroff

Date: 23-01-2010

Signature:

7. SUPPORTING INFORMATION:

None (no index and appendices are required).

LAB 3: COST APPROXIMATION

TOTAL INPUTS: 4

1. Allocate room.
2. Accept food order.
3. Generate final bill.
4. Provide unique identification number.

TOTAL INQUERIES: 4

1. Checking availability of room.
2. List of available food items.
3. Monthly data for occupancy ratio.
4. Quantity of available food.

TOTAL OUTPUT: 7

1. Apology message.
2. Token number.
3. Occupancy Ratio.
4. List of available items.
5. Bill for type of food and quantity.
6. Payable amount for guest.
7. Identification number.

External Interface: 0

Interface Files: 3

1. Customer Personal Database.
2. Restaurant Database.
3. Room Database
4. Database for 'Frequent guest' program.

Factors of DI:

- Data Communication - 0
- Distributed data communication - 0
- Performance - 4
- Configuration - 0
- Transaction - 3
- On-line data entry - 5
- End-user efficiency - 4
- On-line update - 4
- Complex processing - 1
- Reusability - 0
- Installation Ease - 0
- Operational Ease - 3
- Multiple sites - 0
- Change requirement – 3

Total: 27

- **$FP = UFP * TCF$**

- **$TCF = 0.65 + 0.01 * DI$**

$$= 0.65 + 0.01 * 27$$

$$= 0.92$$

- **$UFP = \text{Inputs} * 4 + \text{Outputs} * 5 + \text{Inquiries} * 4 + \text{Internal Files} * 10 + \text{External interfaces} * 10$**

$$= 4 * 4 + 7 * 5 + 4 * 4 + 4 * 10 + 0 * 10$$

$$= 16 + 35 + 16 + 40 + 0$$

$$= 107$$

- **$FP = 107 * 0.92$**

$$= 98.44$$

- **$LOC = 60 * FP$**

$$= 60 * 98.44$$

$$= 5906.4$$

- **$\text{Effort} = 2.4 * (KLOC)^{1.05}$**

$$= 2.4 * (5.9064)^{1.05}$$

$$= 15.49 \text{ PM}$$

- **$T_{dev} = 2.5 * (Effort)^{0.38}$**

$$= 2.5 * (15.49)^{0.38}$$

$$= 7.08 \text{ months}$$

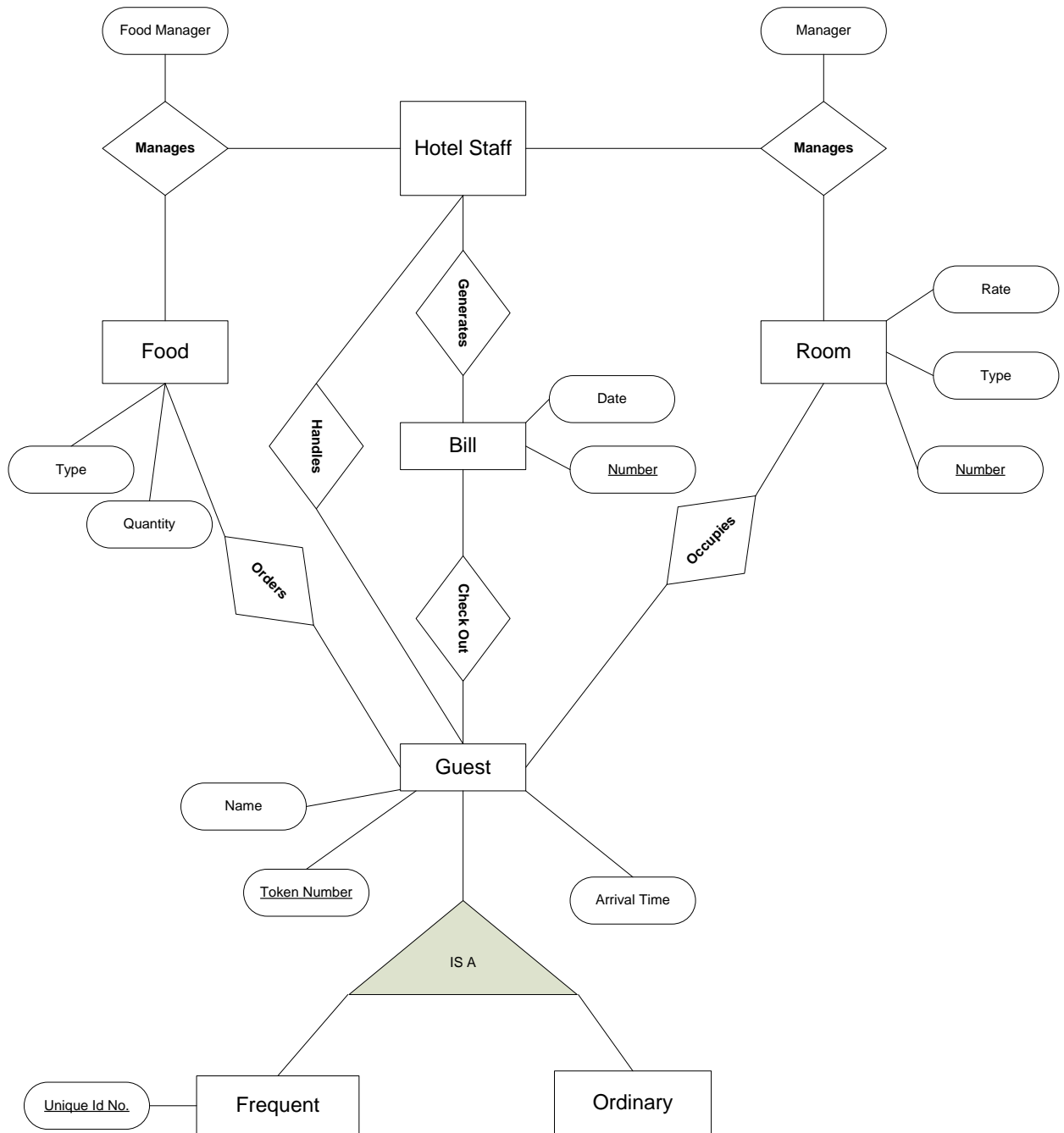
$$= \mathbf{7 \text{ months}}$$

- **$Cost = T_{dev} * Cost \text{ per month}$**

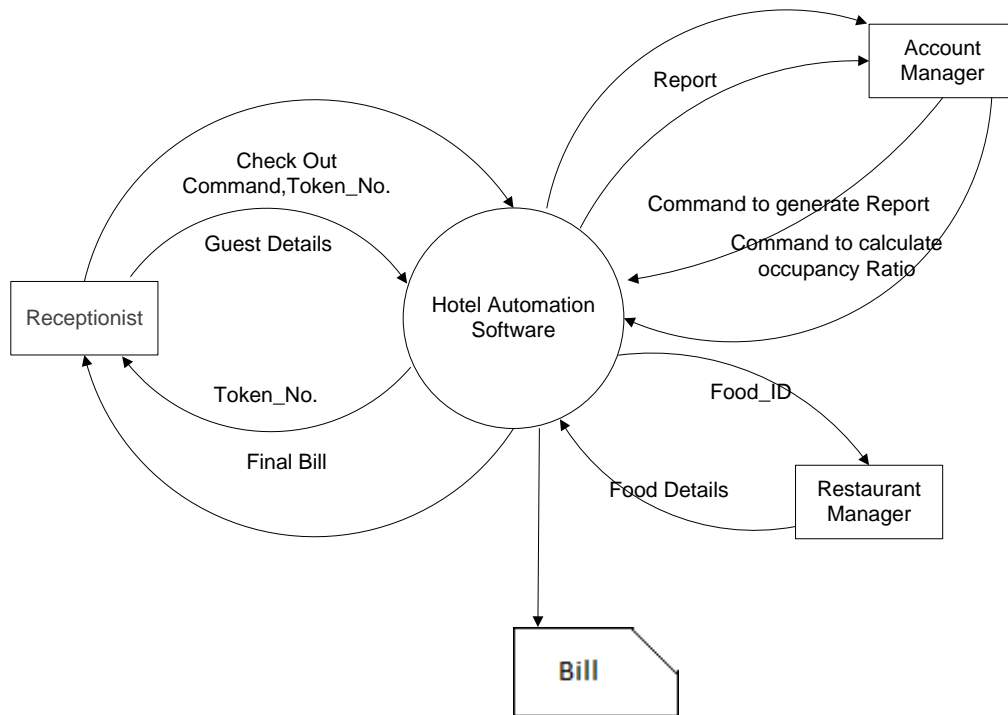
$$= 7 * 15,000$$

$$= \mathbf{1, 05,000/-}$$

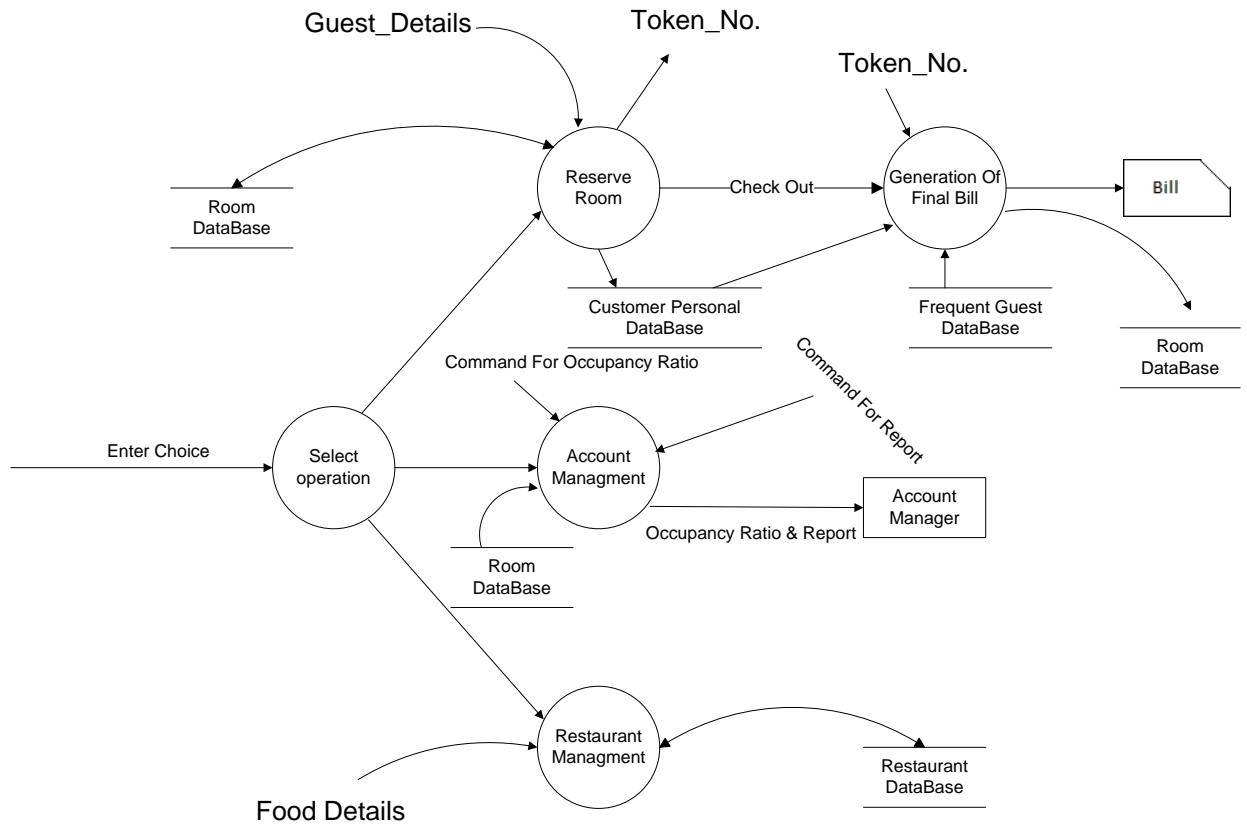
E-R DIAGRAM:



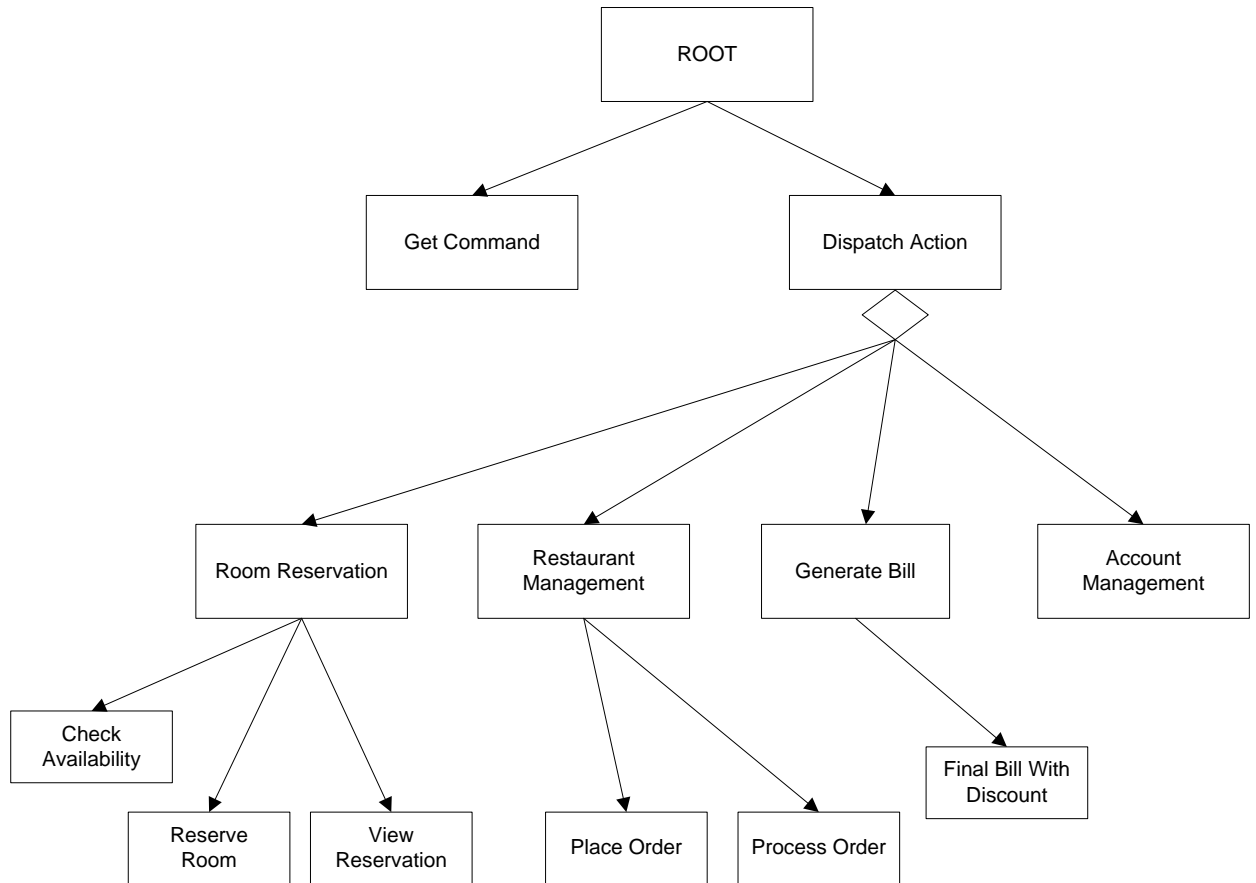
DFD LEVEL 0:



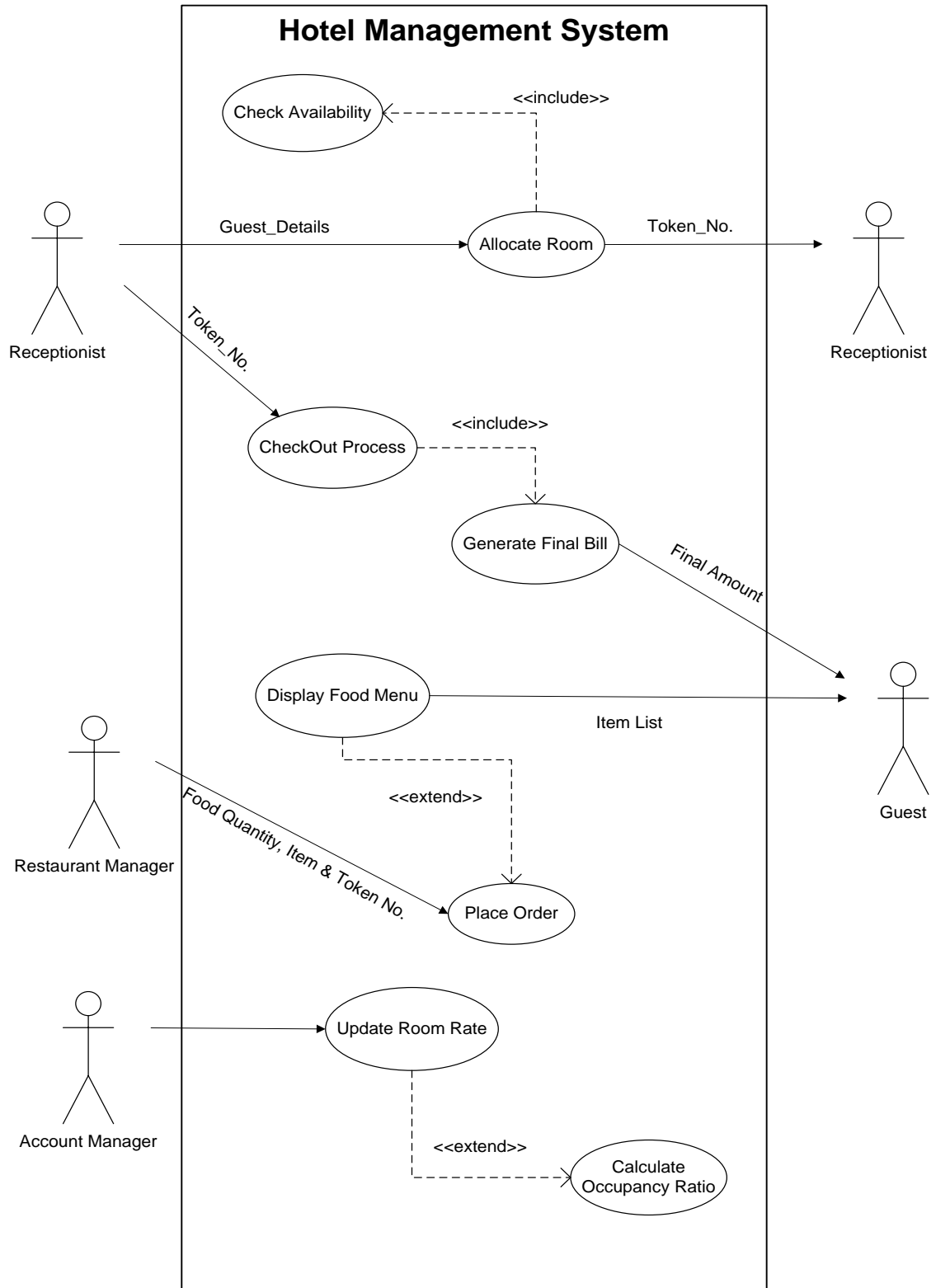
DFD LEVEL 1:



STRUCTURE CHART:



USE CASE:



USE CASE DESCRIPTION:

1. Allocate Room

- Actors : Receptionist
- Description: This Use-case is for allocating a room to some guest. First Checking would be performed for availability of particular room type. And room will be allocated if it's available and accordingly database would be updated. One Unique ID will be generated dedicated to that guest.
- Basic Flow:
 - R: Sends command for checking availability of some type of room.
 - S: Displays Status and room number if available else an apology message.
 - R: If Guest wants than allocate room.
 - S: Waits for entering Guest Details.
 - R: Enters Details.
 - S: Updates Database and Provides Token_No.

2. Check Out Process

- Actors : Receptionist, Guest
- Description: This Use-case is for the process when some user wants to checkout from your Hotel. First we have to generate final bill for that user and have to apply discounts if applicable. The final bill will also include the restaurant bill.
- Basic Flow:
 - R: Sends Checkout Command.
 - S: Asks for Token_No. of that Guest.
 - R: Provides Token_No.
 - S: Checks weather there's any pending restaurant bill and weather guest is frequent or not. If guest is among frequent guests then give him discounts. Finally display total amount to pay to Guest.

3. Place Order

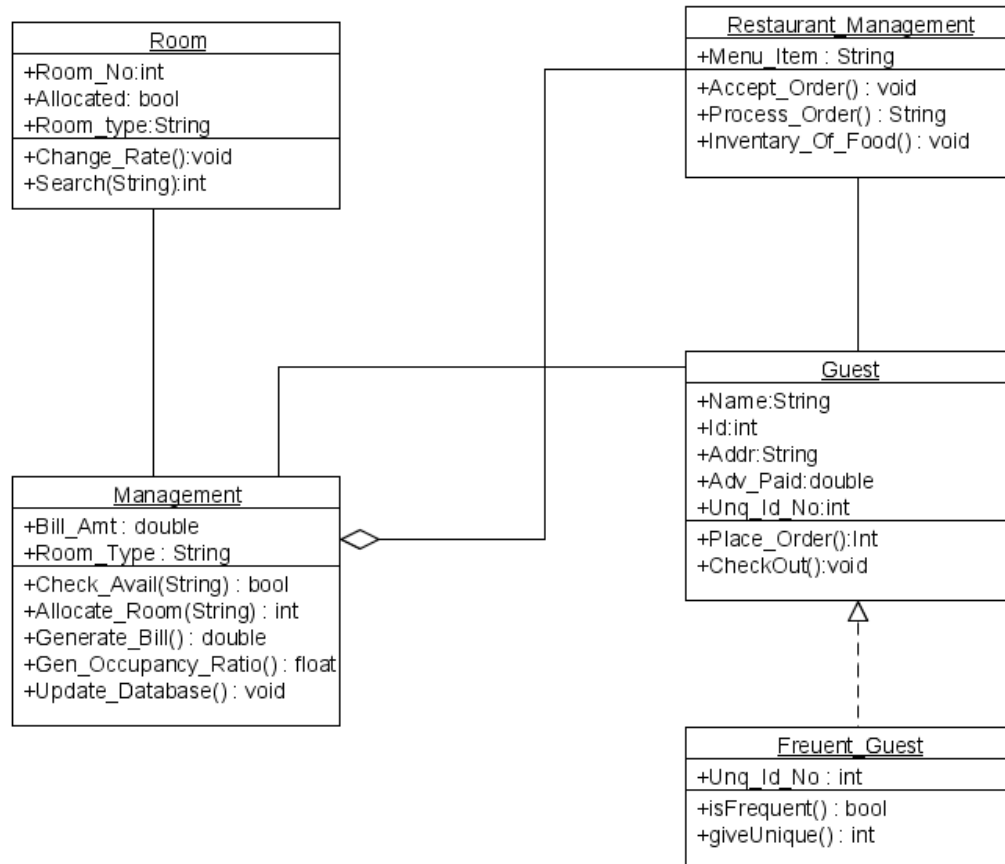
- Actors: Restaurant Manager, Guest
- Description: This Use-case is used for ordering some item in restaurant .Whenever Guest wants to order something he informs Hotel manager and then updates the system about food quantity and type of food ordered. The system calculates amount to be paid and adds it to that guest's account.
- Basic Flow:
 - G: Asks for items available to hotel manager.
 - HM: Displays the menu.
 - G: Places the order.
 - HM: Enter Food details in system.

- S: Calculates the amount and add it to that guest's account.

4. Update Room Rate

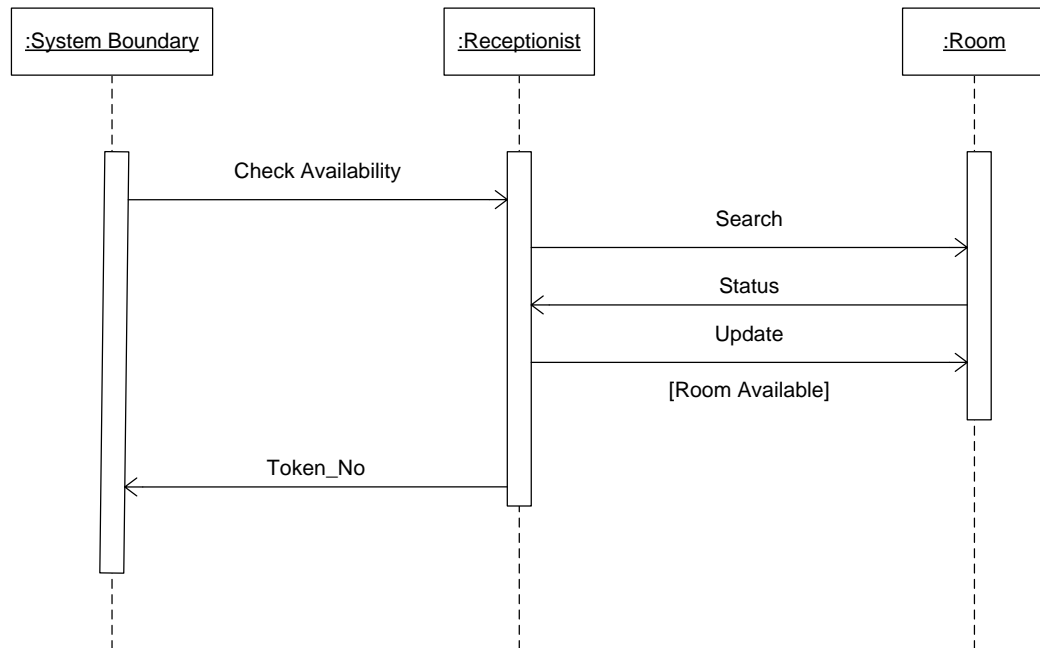
- Actors: Accountant
- Description: This use case is for changing the room rates. Here we are changing it according to occupancy ratio which is $\text{Occupied Rooms} / \text{Total Rooms}$. If it's high than increase the rates else decrease rates.
- Basic Flow:
 - A: Command to update room rate.
 - S: Displays Current Occupancy Ratio and asks whether to increase or decrease rate.
 - A: Analysis data and decides what to do.
 - S: According to input from accountant it updates tariff of the rooms.

CLASS DIAGRAM:

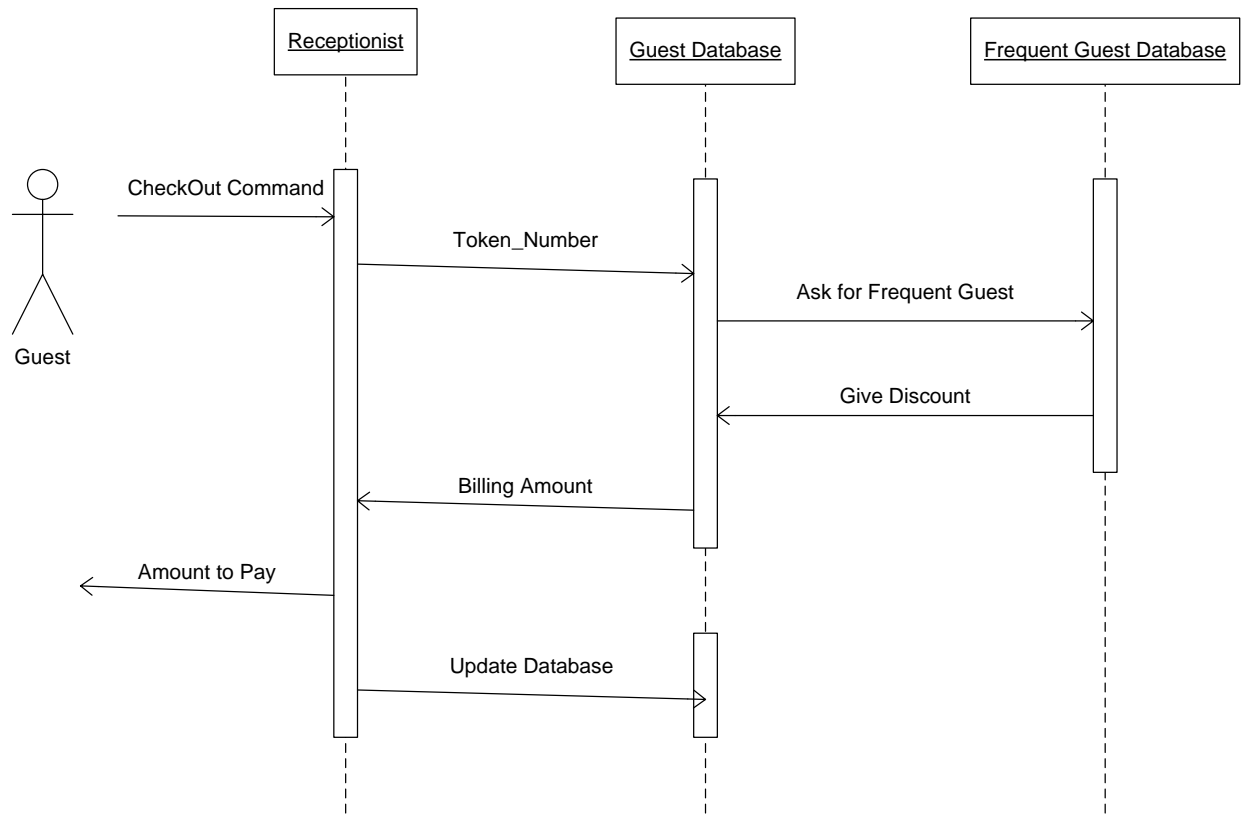


SEQUENCE DIAGRAM:

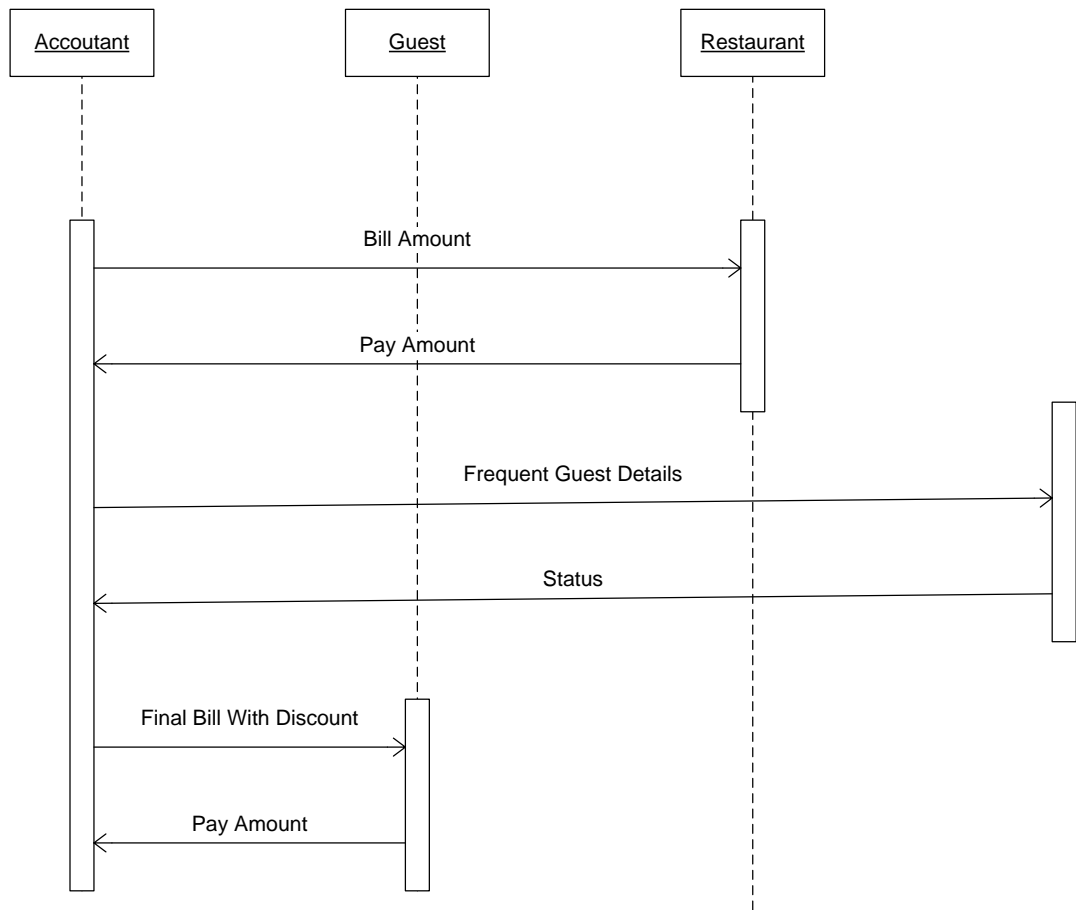
1. ALLOCATE ROOM



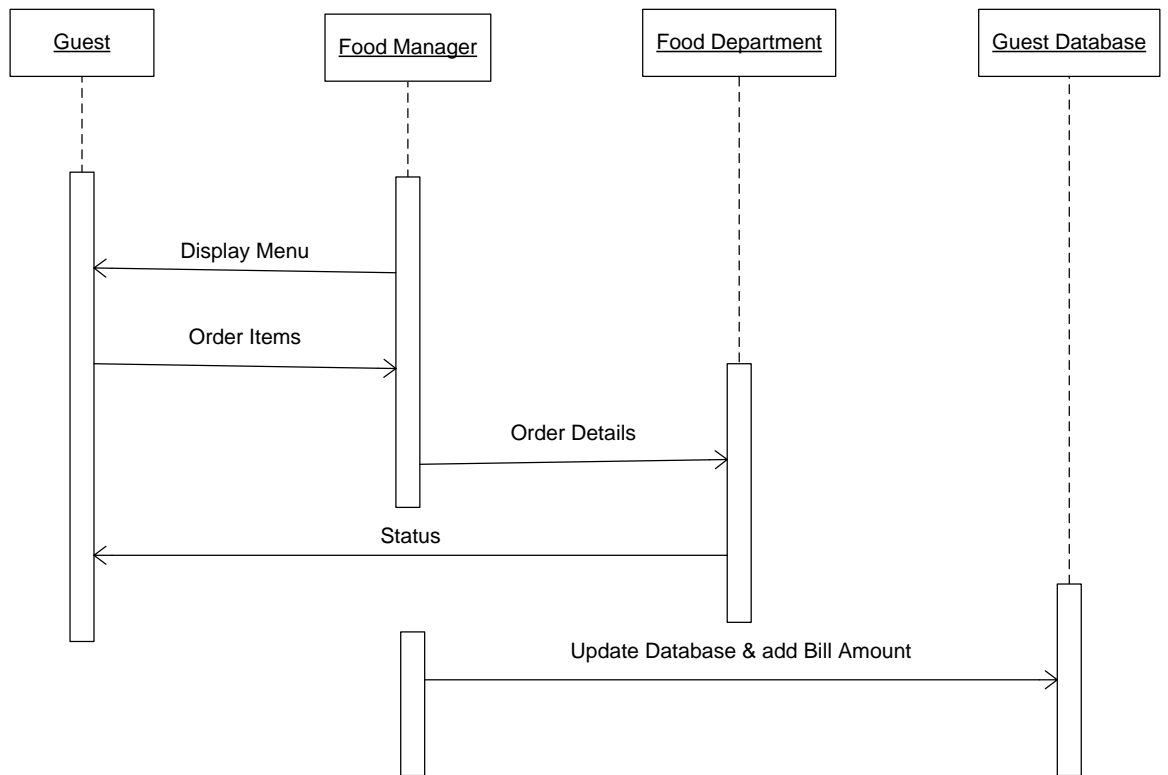
2. CHECK OUT PROCESS:



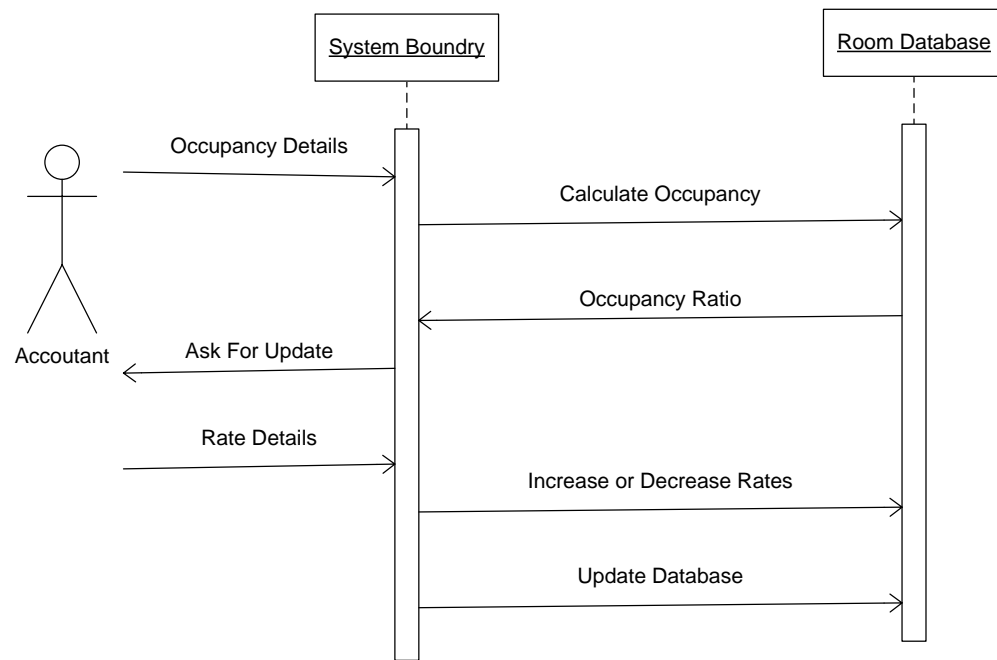
3. GENERATION OF FINAL BILL:



4. PROCESS ORDER:

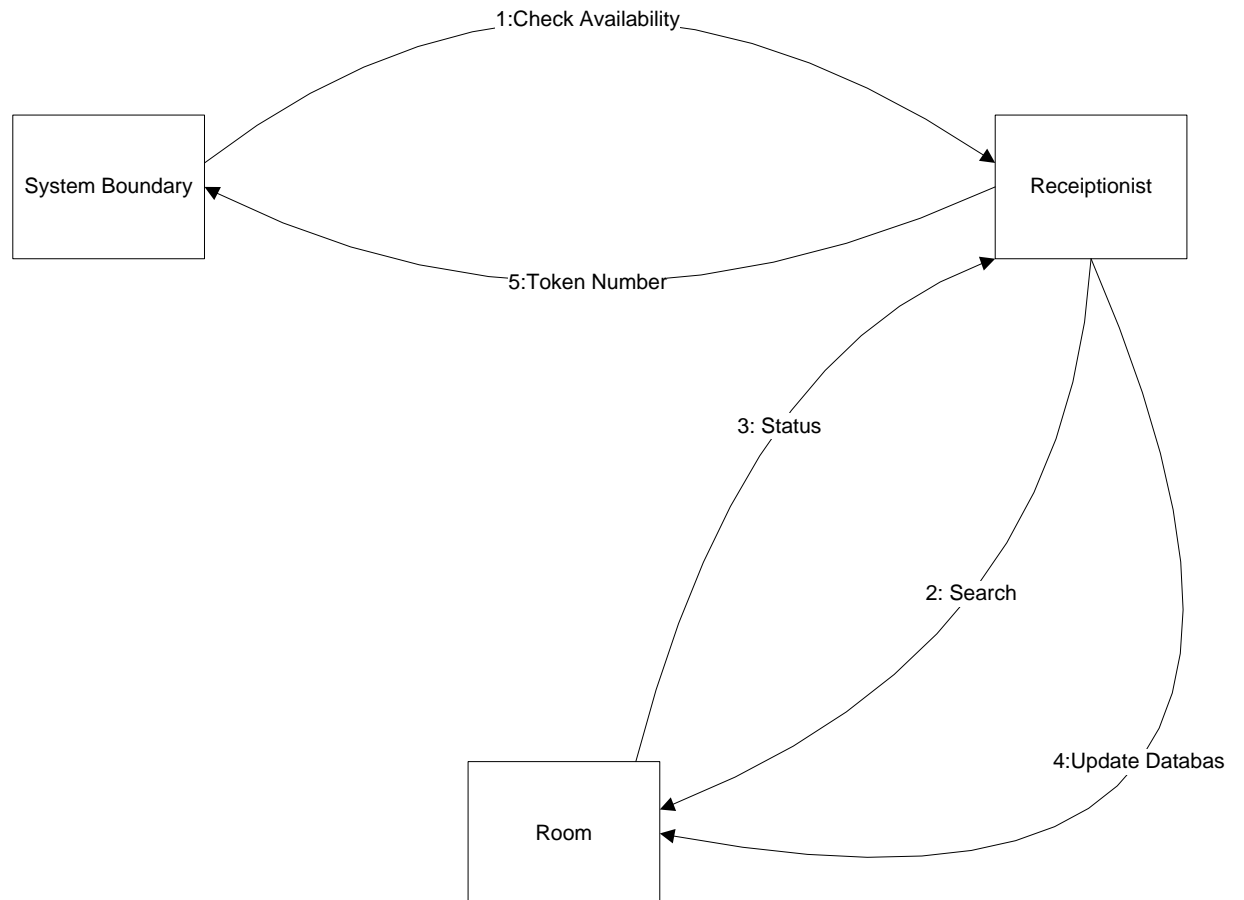


5. UPDATE ROOM RATE:

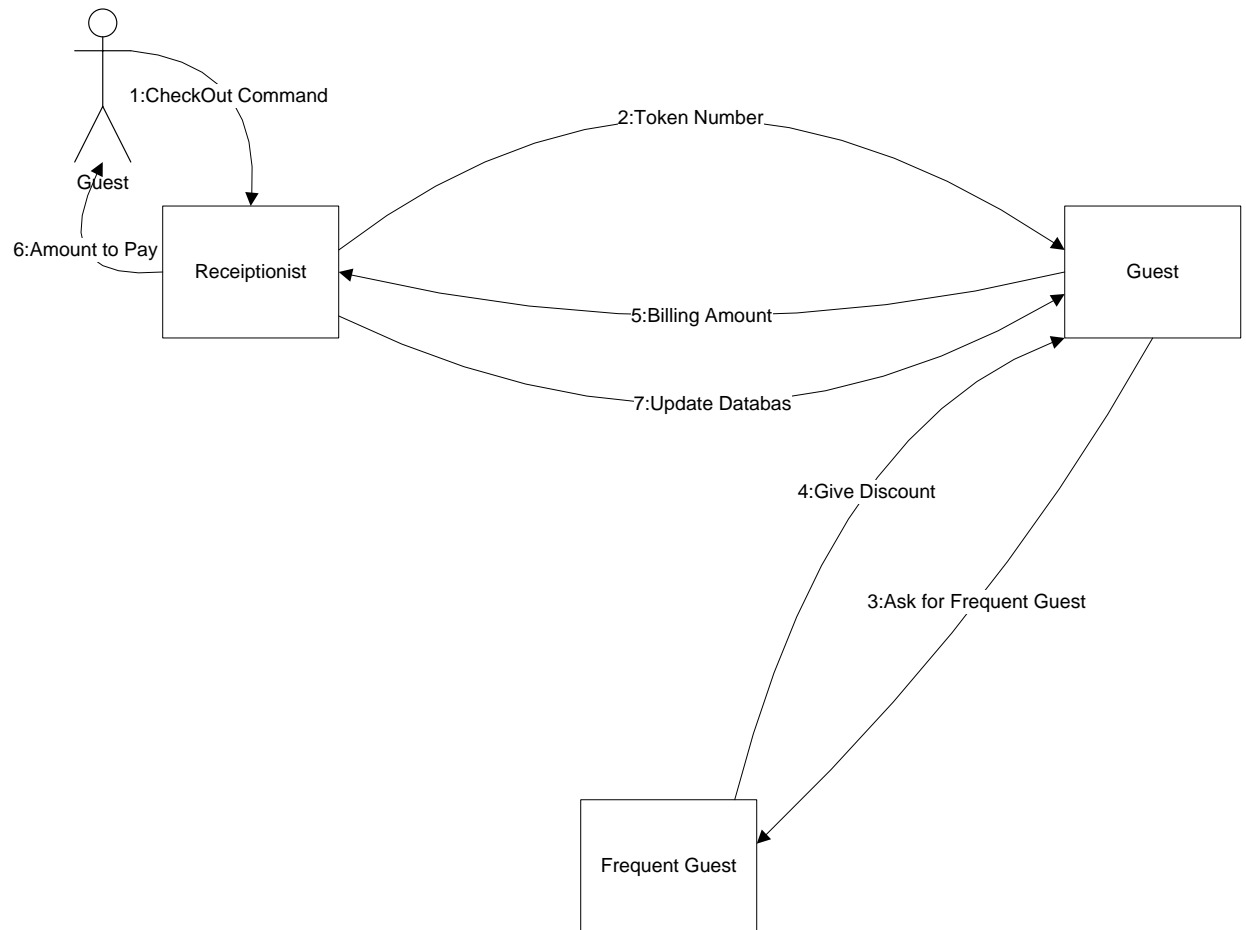


COLLABORATION DIAGRAMS

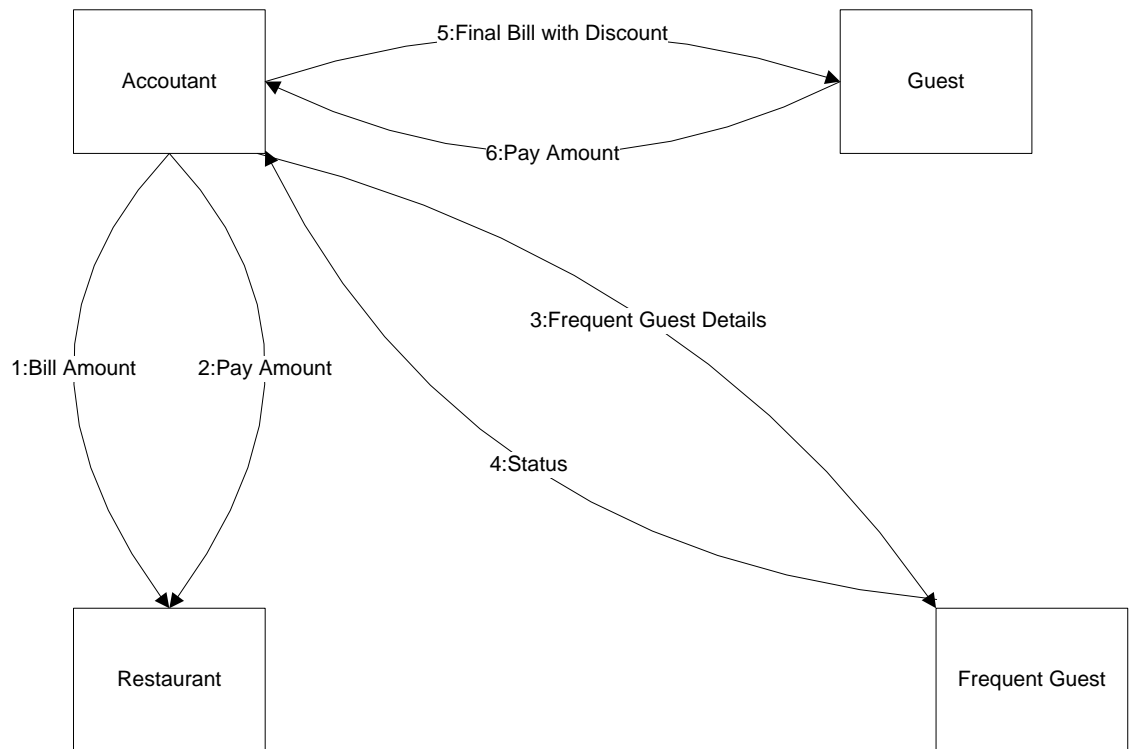
1. ALLOCATE ROOM:



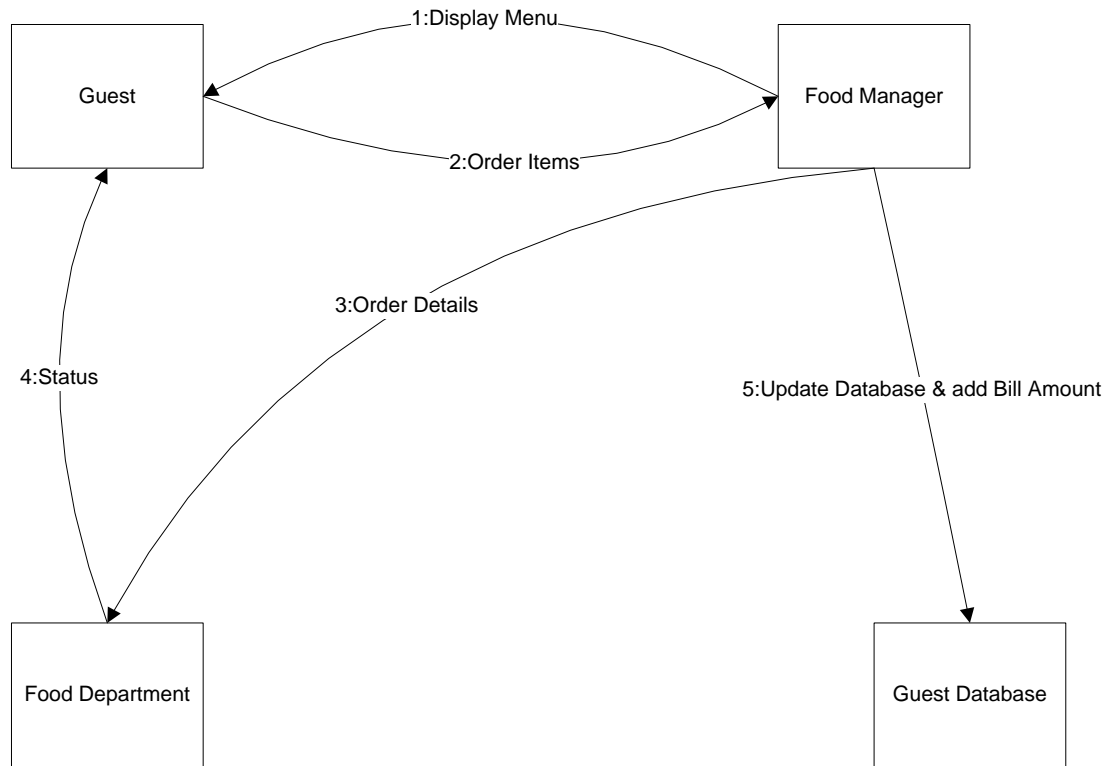
2. CHECK OUT PROCESS:



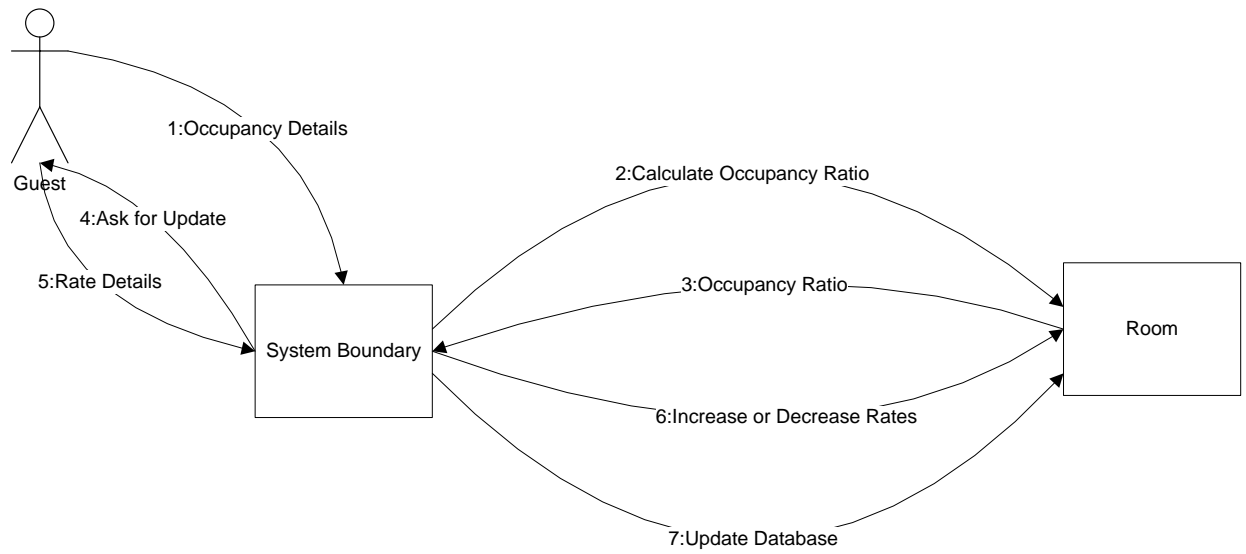
3. GENERATION OF FINAL BILL:



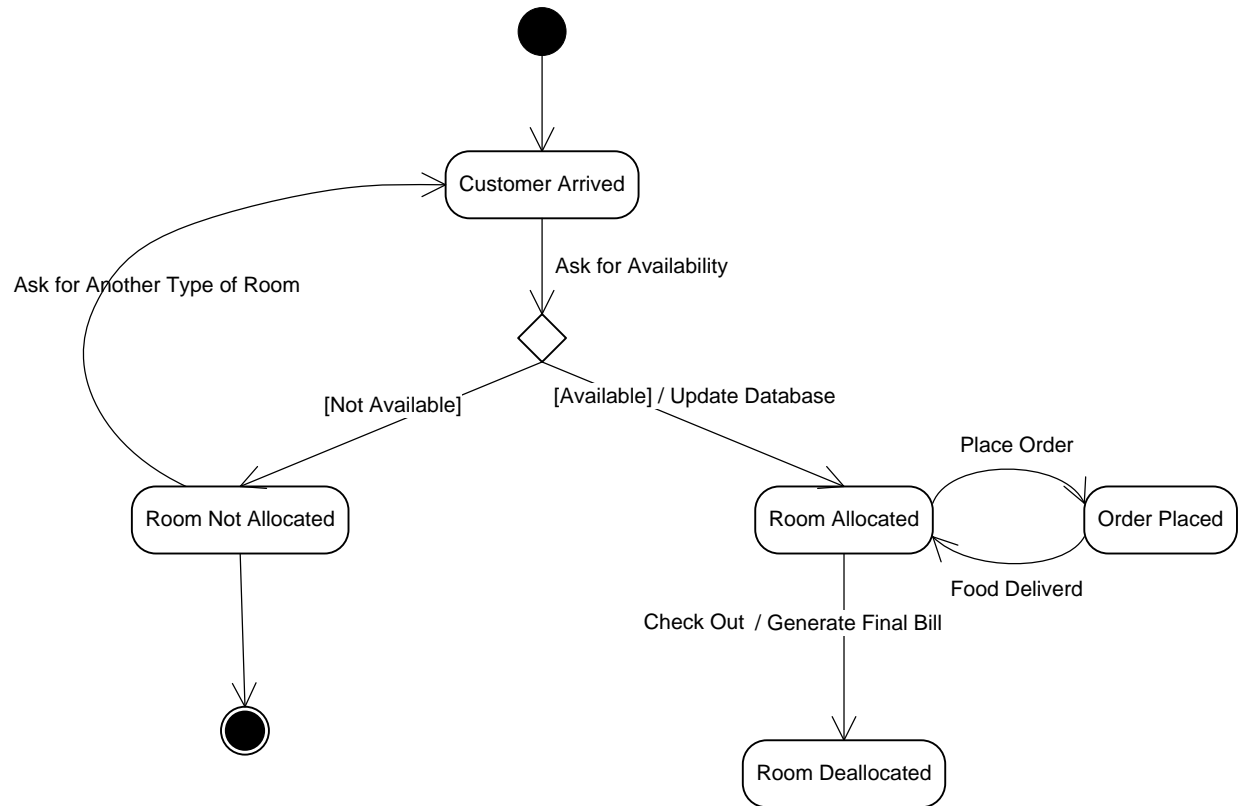
4. PROCESS ORDER:



5. UPDATE ROOM RATE:



STATE CHART



ACTIVITY DIAGRAM

