IIITD&M KANCHEEPURAM

DBMS

CS2008

Guest Room Availability & Management System (GRAMS)

Team 15:
Srinath CS22B1008
Akash CS22B1084
Ganesh CS22B2008
Rahul CS22B2042
Ritesh CS22B2043

I . System Requirements Specification (SRS)

1 Introduction

GRAMS is a web-based application designed to streamline guest room management within our institute. This document outlines the requirements for the Guest Room Availability Management System (GRAMS), providing a comprehensive understanding of the system's functionalities and constraints.

1.1 Purpose

The purpose of our project (GRAMS) is to modernize and optimize the process of managing guest room reservations, availability, billing, and related tasks for our institution. By providing a comprehensive and user-friendly platform, the system aims to streamline operations, enhance efficiency, and improve the overall guest experience.

1.2 Scope

The system will provide administrators with a user-friendly interface to manage room inventory, reservations, guest information, and billing. Additionally, guests will be able to make reservations and payments conveniently through the system.

1.3 Glossary

SRS: System Requirements Specifications
GRAMS: Guest Room Availability & Management System
GUI: Graphics User Interface

2 Functional Requirements

2.1 Admin Features

(a) **Dashboard**:

- Display real-time statistics on room occupancy, reservations, and revenue.
- Provide quick access to essential features and reports.

b. Room Management

- Manage rooms by creating, updating, or removing entries with details including room number, type, capacity, and price.
- Adjust room availability based on dates and occupancy to meet demand effectively.

c. Reservation Management

- Manage reservations by creating, modifying, or cancelling bookings.
- Easily search and filter reservations by guest name, date, or room type for streamlined management.
- Access reservation history and upcoming bookings for easy tracking.

d. Guest Management

- Maintain guest profiles with contact information, preferences, and booking history.
- Manage guest check-ins and check-outs.
- Allow administrators to manage user accounts with different access levels.

e. Billing Management

- Generate and manage bills for reservations, including options for additional charges.
- Track payment status and view transaction history conveniently.
- Offer flexibility with additional charges to accommodate various billing needs.

f. Availability Management

- Display a visual representation of room availability for easy reference.
- Allow administrators to block or release rooms for maintenance or special events.
- Optionally, set limitations on guest stay duration.

g. Reporting & Analytics

- Generate reports on occupancy rates, revenue, and guest demographics.
- Customize report parameters and export options.
- Analyze trends in room occupancy, booking patterns, and revenue generation.
- Forecast future demand based on historical data.

h. Email Notifications

- Sends automated emails for reservation confirmations, reminders, and billing notifications.

2.2 User Features

a. User Registration/Login

- Allow users to create accounts or login as guests.
- Ensure secure authentication and password management.

b. Room Reservation

- Search for available rooms based on dates, room types, and preferences.
- View room details, images, and amenities.
- Reserve rooms and specify booking details (e.g., check-in/out dates, special requests).

c. Booking Management

- View, modify, and cancel reservations.
- Receive email confirmations and reminders.

d. Payment Processing

- Secure online payment gateway integration.
- Provide multiple payment options and currencies.

e. Feedback and Ratings

- Allow guests to provide feedback and ratings after their stay.
- Administer guest satisfaction surveys.

3 Non-Functional Requirements

a. Performance

- The system should handle concurrent user sessions without performance degradation.
- Response times for critical operations should be within milliseconds.
- Database queries should be optimized for efficiency.

b. Security

- Implement role-based access control (RBAC) to restrict access to sensitive features and data.
- Encrypt sensitive data such as passwords and payment information.
- Regularly audit and monitor system activity for security breaches.

c. Reliability

- Ensure system availability of at least 99.9
- Implement automated backups and disaster recovery mechanisms.

d. Usability

- Design a user-friendly GUI with intuitive navigation and clear labeling.
- Provide context-sensitive help and tooltips for complex operations.
- Support multi-platform compatibility (web, mobile).

4 System Constraints

- The system must be developed using MySQL as the backend database.
- The frontend should be developed using HTML5, CSS3 (Tailwind CSS), and JavaScript frameworks (React, NEXT.js).
- The backend should be developed using Node.js & Express.js.
- Compatibility with modern web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge is required.

5 Use Cases

5.1 Make Reservation

Actors:

Guest

Description:

This use case describes the process by which a guest makes a reservation for a room in the institution.

Preconditions:

- The guest has access to the Guest Room Availability & Management System.
- The guest has logged into their account (optional).

Basic Flow:

- The guest selects the option to make a new reservation.
- The system presents the guest with a search interface to specify reservation details such as check-in date, check-out date, room type, and any other preferences.
- The guest enters the desired reservation details and submits the request.
- The system checks the availability of rooms based on the provided criteria.
- If rooms are available, the system confirms the reservation and provides a booking confirmation to the guest.
- The guest receives a confirmation email with details of the reservation.

Alternate Flows:

- If no rooms are available for the specified dates or criteria, the system notifies the guest and allows them to modify their search criteria.

Postconditions:

- A new reservation is created in the system with the specified details.
- The room availability status is updated accordingly.

5.2 Manage Reservations & Room Inventory

Actors:

Administrator

Description:

This use case describes the process by which an administrator manages the inventory of guest rooms in the institution.

Preconditions:

- The administrator has access to the administrative interface of the Guest Room Availability & Management System.

Basic Flow:

- The administrator logs into the administrative interface of the system.
- Administrator views a list of reservations.
- Administrator modifies or cancels reservations as needed.
- System updates reservation status and sends notifications to guests if necessary.
- The system presents the administrator with options to manage room inventory.
- The administrator selects the option to add, edit, or delete rooms.
- If adding or editing rooms, the administrator enters or modifies the room details such as room number, type, capacity, rate, and availability.
- The system updates the room inventory based on the administrator's actions.

Alternate Flows:

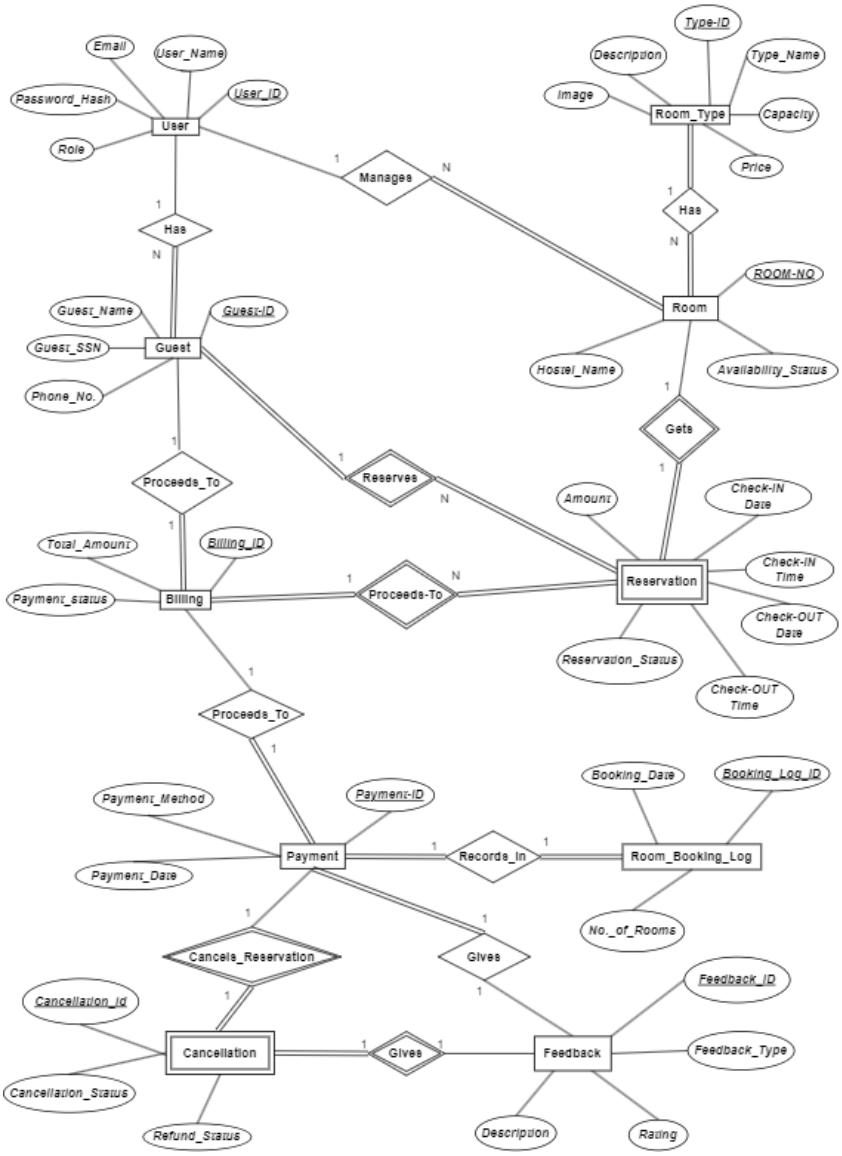
- If deleting rooms, the system prompts the administrator to confirm the deletion and updates the room inventory accordingly.

Postconditions:

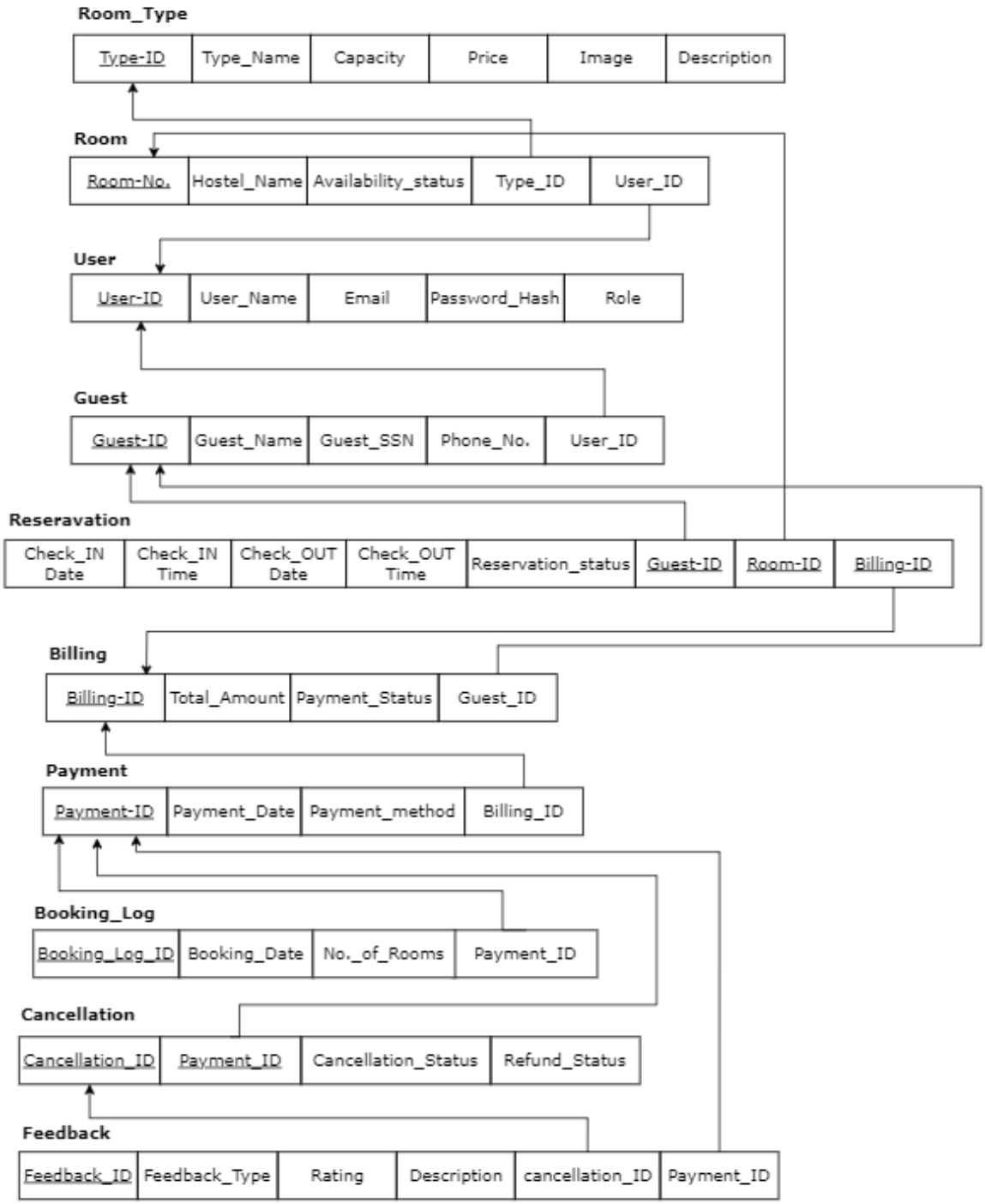
- The room inventory is updated with the changes made by the administrator.

II . Data Design

Entity Relationship Diagram



2.Schema


Room_Type
Type-ID
Type_Name
Capacity
Price
Image
Description
Room
Room-No.
Hostel_Name
Availability_status
Type_ID
User_ID
User
User-ID
User_Name
Email
Password_Hash
Role
Guest
Guest-ID
Guest_Name
Guest_SSN
Phone_No.
User_ID
Reseravation
Check_IN Date
Check_IN Time
Check_OUT Date
Check_OUT Time
Reservation_status
Guest-ID
Room-ID
Billing-ID
Billing
Billing-ID
Total_Amount
Payment_Status
Guest_ID
Payment
Payment-ID
Payment_Date
Payment_method
Billing_ID
Booking_Log
Booking_Log_ID
Booking_Date
No._of_Rooms
Payment_ID
Cancellation
Cancellation_ID
Payment_ID
Cancellation_Status
Refund_Status
Feedback
Feedback_ID
Feedback_Type
Rating
Description
cancellation_ID
Payment_ID

3.Entities and Attributes

This section of the document explains the entities used in the project, their attributes and how they will work together. Basically, this is intented to make the design more easy and understandable to everyone.

3.1 Entities:

a. User

b. Room

c. Room Type

d. Guest

e. Reservation

f. Billing

g. Payment

h. Room Booking Log

i. Cancellation

j. Feedback

a.User

Participates in:
Manages Room: (1:N) - One user(Role: Admin) can manage many rooms, but many rooms are managed by one user . User has partial participation, and Room has total participation.
Has Guest: (1:N) - One user can have many guests, but a guest can only be associated with one user. User has partial participation, and Guest has total participation.

Attributes:

User		
Attribute_Name	**Data_Type**	**Attribute_Type**
User_ID	VarChar	Primary_key
User_Name	VarChar	Non_key
Email	VarChar	Non_key
Password_Hash	VarChar	Non_key
Role	VarChar	Non_key

b.Room

Participates in:
Managed by User: (N:1) - Many rooms can be managed by one user , but one room must managed be by one user . Room has total participation, and User has partial participation.
Gets Reservation: (1:N) - One room can contain one reservation, but a reservation can only be for one room. Room has partial participation, and Reservation has total participation.

Attributes:

Room		
Attribute_Name	**Data_Type**	**Attribute_Type**
Room_No	Int	Primary_key
Hostel_Name	VarChar	Non_key
Availability_Status	VarChar	Non_key
Type_ID	VarChar	Foreign_key
User_ID	VarChar	Foreign_key

c.Room_Type

Participates in:
Classifies Room: (1:N) - One room type can classify many rooms, but a room can only belong to one room type. Room Type has total participation, and Room has total participation.

Attributes:

Room_Type		
Attribute_Name	**Data_Type**	**Attribute_Type**
Type_ID	VarChar	Primary_key
Type_Name	VarChar	Non_key
Capacity	Int	Non_key
Price	Decimal	Non_key
Image	Blob	Non_key
Description	Text	Non_key

d.Guest

Participates in:
Has User: (N:1) - Many guests can be associated with one user, but a guest can only be associated with one user. Guest has total participation, and User has partial participation.
Reserves Reservation: (1:N) - One guest can reserve many reservations, but a reservation can only be reserved by one guest. Guest has total participation, and Reservation has total participation.
Proceeds To Billing: (1:1) - One Guest proceeds to one billing record, and one billing record is for one Guest. Guest has partial participation and billing has total participation.

Attributes:

Guest		
Attribute_Name	**Data_Type**	**Attribute_Type**
Guest_ID	VarChar	Primary_key
Guest_Name	VarChar	Non_key
Guest_SSN	VarChar	Non_key
Phone_No	VarChar	Non_key
User_ID	VarChar	Foreign_key

e.Reservation

Participates in:
Contains Room: (N:1) - Many reservations can be for one room, but a reservation can only be for one room. Reservation has total participation, and

Room has partial participation.
Reserved by Guest: (N:1) - Many reservations can be reserved by one guest, but a reservation can only be reserved by one guest. Reservation has total participation, and Guest has total participation.
Proceeds To Billing: (N:1) - Many reservation proceeds to one billing record, and one billing record is for one reservation. Reservation and Billing have total participation.

Attributes:

Reservation (Weak Entity)		
Attribute_Name	**Data_Type**	**Attribute_Type**
Guest_ID	VarChar	Foreign_key
Room_ID	Int	Foreign_key
Billing_ID	VarChar	Foreign_key
Check_IN_date	Date	Non_key
Check_IN_time	Time	Non_key
Check_OUT_date	Date	Non_key
Check_OUT_time	Time	Non_key
Reservation_status	VarChar	Non_key

f.Billing

Participates in:
Proceeds from Guest: (1:1) - One Guest proceeds to one billing record, and one billing record is for one guest. Guest has partial participation and billing has total participation.
Proceeds To from Reservation: (1:N) - One Billing proceeds from many reservations record, and one billing record can be for many reservation. Reservation and Billing have total participation.
Proceeds to Payment: (1:1) - One billing record can has only one payment , Billing has partial participation and payment has total participation.

Attributes:

Billing		
Attribute_Name	**Data_Type**	**Attribute_Type**
Billing_ID	VarChar	Primary_key
Total_Amount	Decimal	Non_key
Payment_Status	VarChar	Non_key
Guest_ID	VarChar	Foreign_key

g.Payment

Participates in:
Records in Room booking log: (1:1) - For one payment only one room booking log is recorded.Both Payment and room booking log have total participation.
Cancels reservation Cancellation: (1:1) - For one payment only one cancellation is possible,Payment has partial participation and Cancellation has total participation.
Gives feedback: (1:1) - For one payment only one feedback is possible,Payment has total participation and feedback has partial participation.

Attributes:

Payment		
Attribute_Name	**Data_Type**	**Attribute_Type**
Payment_ID	VarChar	Primary_key
Payment_Date	Date	Non_key
Payment_Method	VarChar	Non_key
Billing_ID	VarChar	Foreign_key

h.Room_Booking_Log

Participates in:
Records in payment: (1:1) - For one room booking log there will be only one payment, Both room booking log and payment have total participation.

Attributes:

Room_Booking_Log		
Attribute_Name	**Data_Type**	**Attribute_Type**
Booking_log_ID	VarChar	Primary_key
Booking_Date	Date	Non_key
No._of_Rooms	int	Non_key
Payment_ID	VarChar	Foreign_key

i.Cancellation

Participates in:
Gives feedback: (1:1) - For one cancellation there will be one feedback, Cancellation has total participation and feedback has partial participation.
Cancels reservation Payment: (1:1) - For one cancellation there will be only one payment, Payment has partial participation and Cancellation has total participation.

Attributes:

Cancellation (Weak Entity)		
Attribute_Name	**Data_Type**	**Attribute_Type**
Cancellation_ID	VarChar	Key
Payment_ID	VarChar	Foreign_key
Cancellation_Status	VarChar	Non_key
Refund_Status	VarChar	Non_key

j.Feedback

Participates in:
Gives Cancellation: (1:1) - For one feedback there is one Cancellation,Cancellation has total participation and feedback has partial participation.
Gives Payment: (1:1) - For one feedback there is only one payment ,Feedback has partial participation and Payment has total participation.

Attributes:

Feedback		
Attribute_Name	**Data_Type**	**Attribute_Type**
Feedback_ID	VarChar	Primary_Key
Feedback_type	VarChar	Non_key
Rating	Int	Non_key
Description	Text	Non_key
Cancellation_ID	VarChar	Foreign_key
Payment_ID	VarChar	Foreign_key

_____ ***** _____