Guest Accommodation and Fee Management System

Neha Kantheti (cs22b1081)
Varshitha Masaram (cs22b1071)
Naga Sripada Penumetsa (cs22b1018)
Sharvani Choppadhandi (cs22b2011)
Rishika Bontha (cs22b2027)

March 19, 2024

Contents

1 Software Requirements System(SRS)

1.1 Introduction

The Guest Accommodation and Fee Management System is designed to streamline the process of booking rooms for guests and managing their fees within a college environment. This system provides a user-friendly interface for both students and administrators to efficiently manage accommodation bookings and payments.

1.2 Purpose

The purpose of this system is to automate and simplify the process of guest accommodation booking and fee management within a college campus. By providing an intuitive interface, it aims to enhance the overall experience for both guests and administrators.

1.3 Scope

The system encompasses functionalities such as user authentication, room booking, guest details management, payment processing, room availability tracking,and receipt generation. It caters to both students and administrators, providing role-based access control.

1.4 Functional Requirements

1.4.1 User Authentication

- Students can log in using their email/password or roll number.
- Administrators can log in using their credentials.

1.4.2 Room Booking

- Students can provide their details and the guest details.
- The system allocates rooms based on availability (AC/non-AC).

1.4.3 Payment Processing

- Various payment modes are available.
- Students can complete payment for the booked room.

1.4.4 Real-time availability updates

- The system should provide real-time updates on room availability, reflecting bookings and cancellations instantly.
- Users should be notified promptly if their selected room becomes unavailable during the booking process.

1.4.5 Receipt Generation

- Upon successful fee payment, the system should generate receipts or invoices for users, containing transaction details
- Receipts should be accessible and downloadable for users for record-keeping purposes.

1.4.6 Administrative Access

- Administrators should have access to administrative tools for managing user accounts, room inventory, and payment records.
- They should be able to view reports on booking trends, occupancy rates, and revenue generation.
- Administrators can update, insert, delete, and view data related to rooms, bookings, and fees.

1.5 Non-Functional Requirementss

1.5.1 Security

- User passwords should be securely hashed and stored in the database.
- Role-based access control ensures data integrity.

1.5.2 Performance

- The system should be able to handle multiple concurrent user sessions without significant degradation in performance.
- Response times for critical operations such as room booking and fee payment should be kept minimal.

1.5.3 Usability

- The user interface should be intuitive and easy to navigate, catering to users with varying levels of technical proficiency.
- Error messages should be informative and user-friendly, guiding users through any issues encountered.

1.5.4 Reliability

- The system should be reliable and available 24/7, minimizing downtime and service interruptions.
- Backup and recovery mechanisms should be in place to safeguard against data loss and ensure business continuity.

1.5.5 Scalability

- The system should be scalable to accommodate future growth in user base and transaction volume.
- It should be able to handle an increasing number of room bookings and fee transactions without compromising performance.

1.5.6 Compliance

- The system should comply with relevant regulations and standards regarding data privacy and payment processing (e.g., GDPR, PCI DSS).
- Compliance with accessibility standards (e.g., WCAG) should be ensured to make the system accessible to users.

1.6 Use Cases

1.6.1 User Authentication

Description:This use case describes the process of authenticating users into the system.
Actors:

- Student
- Administrator

Flow of Events:

- The system presents the login interface.
- The user enters their credentials (email/password or roll number).
- The system verifies the credentials.
- If the credentials are valid, the user is logged into the system.
- If the credentials are invalid, an error message is displayed, and the user is prompted to retry.

1.6.2 Administrator Access

Description:This use case outlines the actions an administrator can perform within the system. Actors:

- Administrator

Flow of Events:

- The administrator logs into the system.
- The system validates the administrator's credentials.
- Once authenticated, the administrator gains access to administrative functionalities.
- The administrator can view, update, insert, or delete data related to rooms, bookings, and fees as required.

1.6.3 Student Access

Description: This use case outlines the actions a student can perform within the system. Actors:

- Student

Flow of Events:

- Student logs into the system.
- System validates the student's credentials.
- Once authenticated, student gains access to administrative functionalities.
- Student functionalities are restricted only to entering details as per requirement i.e, student do not have any kind of insert/delete permissions.

1.6.4 Room Booking

Description:This use case outlines the process of booking a room for a guest.
Actors:

- Student

Flow of Events:

- The student selects the option to book a room.
- The system prompts the student to enter their details and the guest details.
- The student provides the necessary information.
- The system checks room availability based on the preferences (AC/non-AC).
- If a room is available, it is allocated to the guest, and a confirmation is displayed.
- If no room is available, the student is informed, and they may choose to modify their preferences or try again later.

1.6.5 Payment Processing

Description:This use case describes the process of processing payments for room bookings. Actors:

- Student

Flow of Events:

- After booking a room, the student selects the option to make a payment.
- The system presents various payment modes (e.g., credit card, debit card, cash).
- The student selects their preferred payment mode.
- The system processes the payment.
- Upon successful payment, a receipt is generated and provided to the student.
- If the payment fails, an error message is displayed, and the student can retry or choose an alternative payment method.

1.7 Constraints

- The system must be compatible with modern web browsers.
- It should comply with college policies and regulations regarding data handling and privacy.

1.8 Glossary

GDPR: General Data Protection Regulation PCI DSS: Payment Card Industry Data Security Standard WCAG: Web Content Accessibility Guidelines

1.9 Revision History

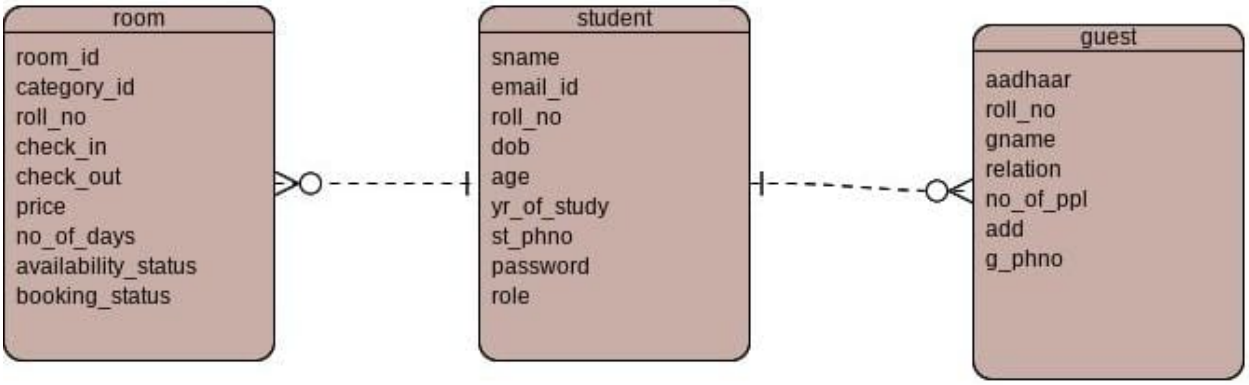
- Version 1.0: Initial release (Date: [date])
- Version 1.1: Updated functional requirements based on feedback (Date: [date])

1.10 Conclusion

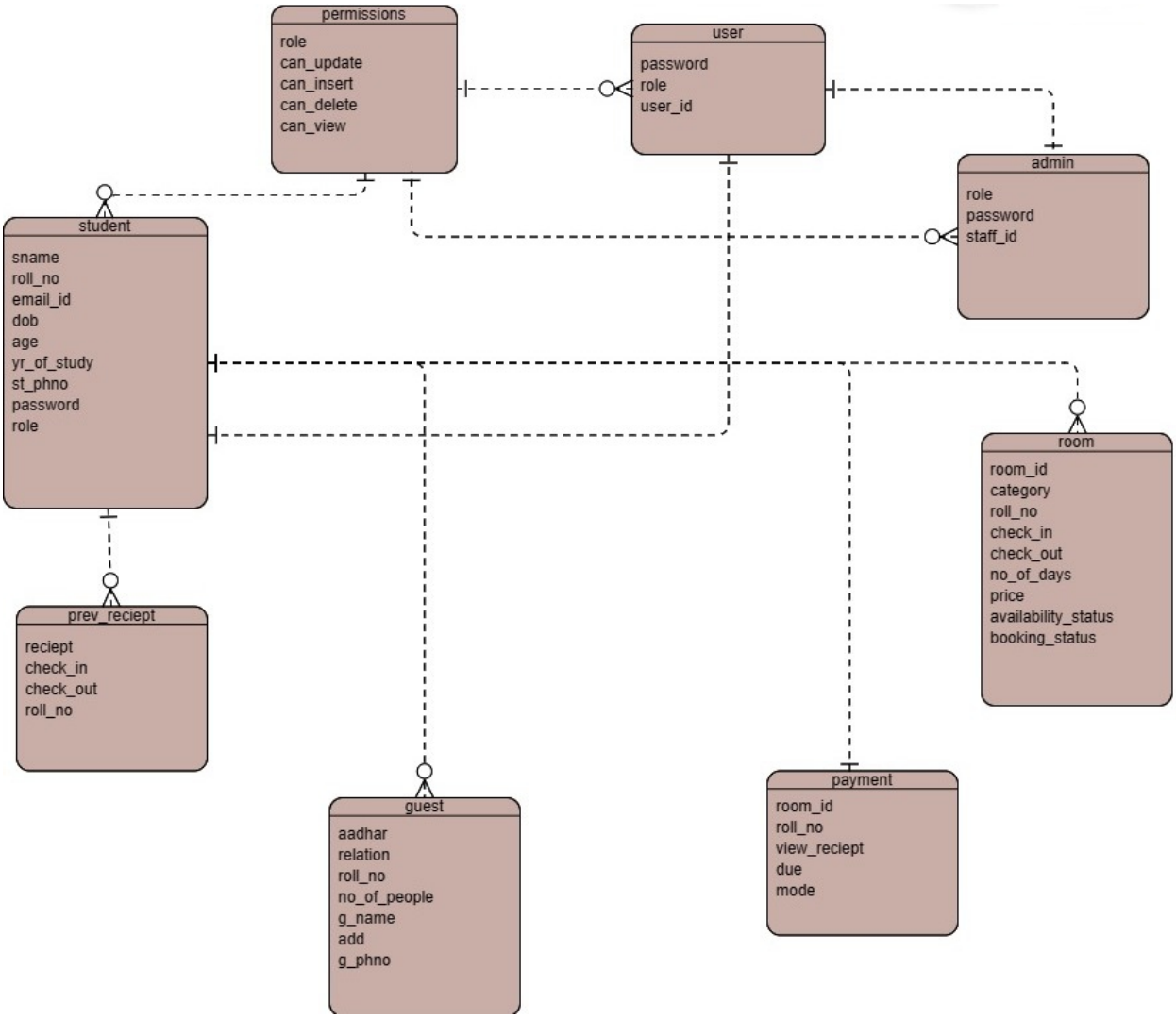
The Guest Accommodation and Fee Management System aims to streamline the process of booking and managing guest accommodation within a college campus.By providing an intuitive interface and robust functionalities, it enhances user experience and ensures efficient management of accommodation bookings and fees.

2 Data Models

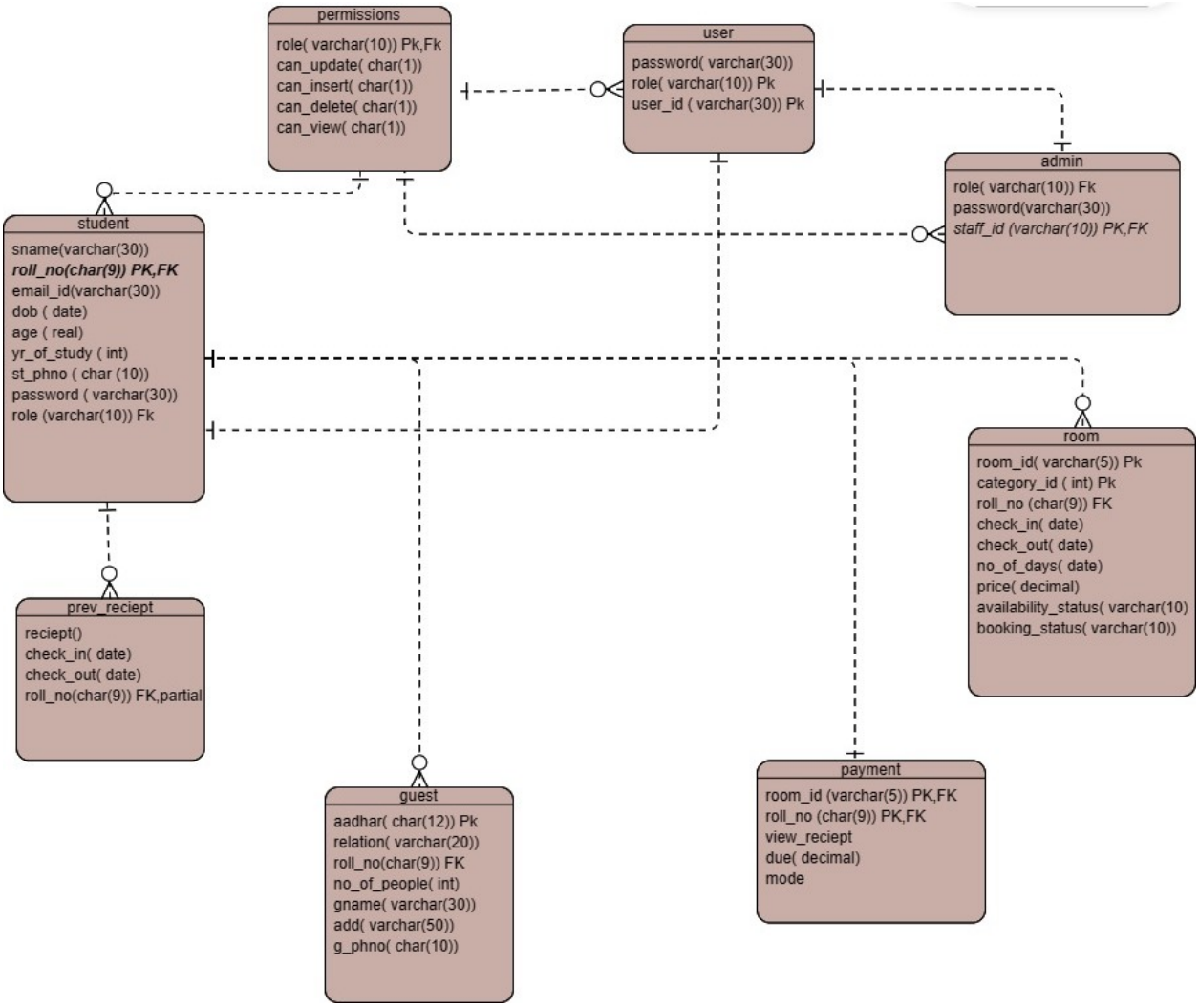
2.1 Conceptual



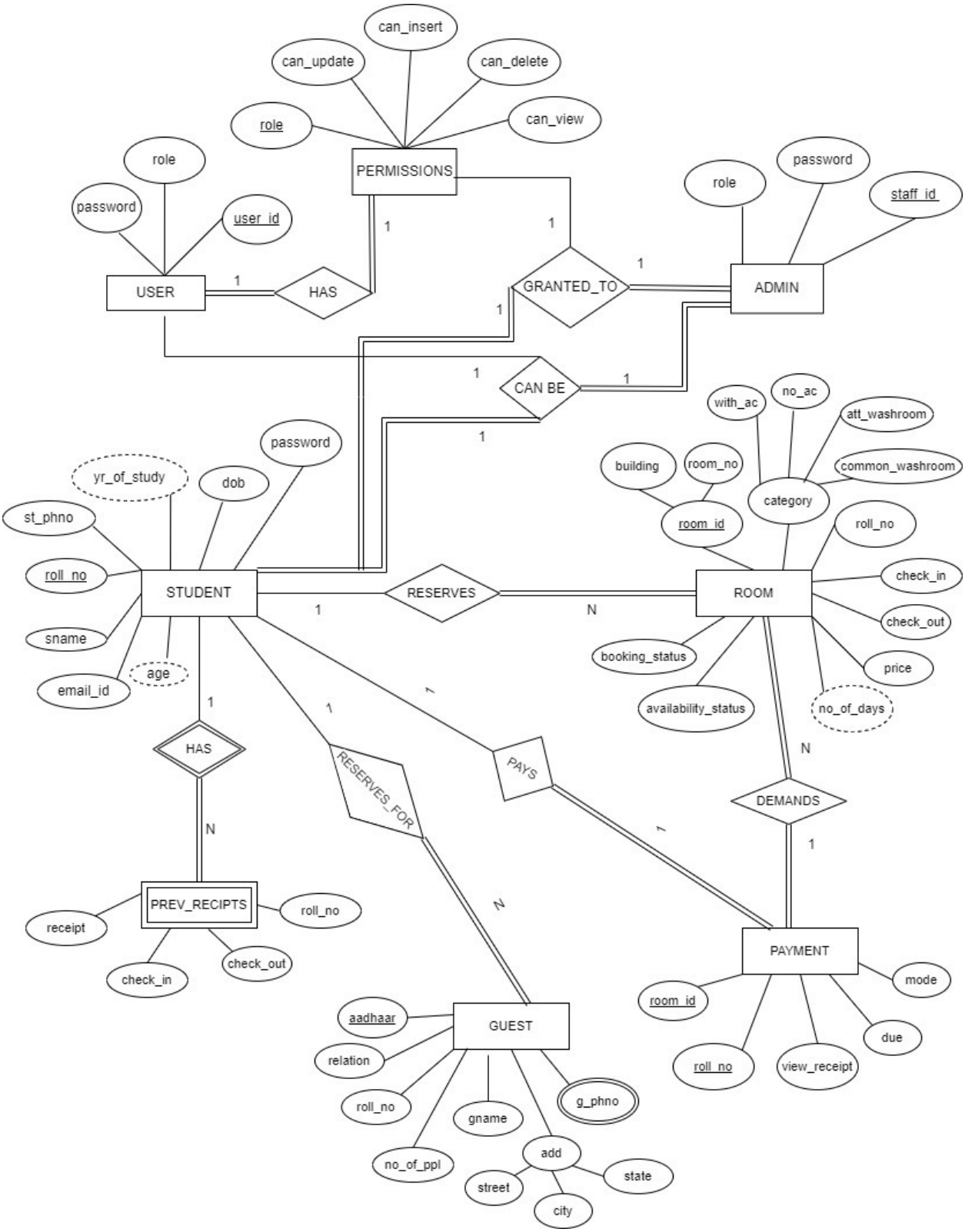
2.2 Logical


permissions
role
can_update
can_insert
can_delete
can_view
user
password
role
user_id
admin
role
password
staff_id
student
sname
roll_no
email_id
dob
age
yr_of_study
st_phno
password
role
room
room_id
category
roll_no
check_in
check_out
no_of_days
price
availability_status
booking_status
prev_reciept
reciept
check_in
check_out
roll_no
guest
aadhar
relation
roll_no
no_of_people
g_name
add
g_phno
payment
room_id
roll_no
view_reciept
due
mode

2.3 Physical


permissions
role(varchar(10)) Pk,Fk
can_update(char(1))
can_insert(char(1))
can_delete(char(1))
can_view(char(1))
user
password(varchar(30))
role(varchar(10)) Pk
user_id (varchar(30)) Pk
admin
role(varchar(10)) Fk
password(varchar(30))
staff_id (varchar(10)) PK,FK
student
sname(varchar(30))
roll_no(char(9)) PK,FK
email_id(varchar(30))
dob (date)
age (real)
yr_of_study (int)
st_phno (char (10))
password (varchar(30))
role (varchar(10)) Fk
room
room_id(varchar(5)) Pk
category_id (int) Pk
roll_no (char(9)) FK
check_in(date)
check_out(date)
no_of_days(date)
price(decimal)
availability_status(varchar(10)
booking_status(varchar(10))
prev_reciept
reciept()
check_in(date)
check_out(date)
roll_no(char(9)) FK,partial
guest
aadhar(char(12)) Pk
relation(varchar(20))
roll_no(char(9)) FK
no_of_people(int)
gname(varchar(30))
add(varchar(50))
g_phno(char(10))
payment
room_id (varchar(5)) PK,FK
roll_no (char(9)) PK,FK
view_reciept
due(decimal)
mode

3 ER Model


can_insert
can_update
can_delete
role
can_view
PERMISSIONS
role
password
user_id
USER
HAS
1
1
1
1
GRANTED_TO
1
role
password
staff_id
ADMIN
1
CAN BE
1
1
password
yr_of_study
dob
st_phno
roll_no
STUDENT
sname
email_id
age
RESERVES
1
N
with_ac
no_ac
att_washroom
building
room_no
common_washroom
category
room_id
roll_no
ROOM
check_in
check_out
price
booking_status
availability_status
no_of_days
1
HAS
N
RESERVES_FOR
N
1
PAYS
1
N
DEMANDS
1
PREV_RECIPTS
receipt
roll_no
check_in
check_out
PAYMENT
mode
room_id
due
roll_no
view_receipt
aadhaar
relation
roll_no
GUEST
gname
g_phno
no_of_ppl
add
street
state
city

4 Relational Database Model