

Requirements Specification for Airbnb Database

1. Introduction

This document outlines the requirements for designing a database for a booking system similar to Airbnb. The system enables interaction between guests and hosts, allowing them to book and manage accommodations seamlessly.

2. User Roles

- **Guest:** A user looking to book accommodations.
- **Host:** A user listing and renting out accommodations.
- **Administrator:** A system user responsible for managing the platform.

3. Actions Performed by Each Role

Guest Actions

- Create and update profile
- Search for accommodations
- Book and cancel reservations
- Pay for bookings
- Leave reviews and ratings for hosts
- Message hosts

Host Actions

- Create and update profile
- List, update, and delete accommodations
- Manage booking requests (accept/reject)
- Set rental prices and availability
- Leave reviews and ratings for guests
- Receive payments after check-in

Administrator Actions

- Verify user accounts and accommodations

- Oversee transactions and disputes
- Manage platform commission fees
- Ensure compliance with policies

4. Data and Functions Required

Data Requirements:

- User details (name, email, phone, role, payment information).
- Accommodation details (title, description, location, pricing, availability).
- Booking information (guest, accommodation, check-in/out dates, status, total price).
- Payment records (amount, method, status, associated booking).
- Reviews and ratings (reviewer, recipient, rating, comment, timestamp).
- Messages between users (sender, receiver, content, timestamp).
- Platform management data (fees, disputes, complaints, support tickets).

System Functions:

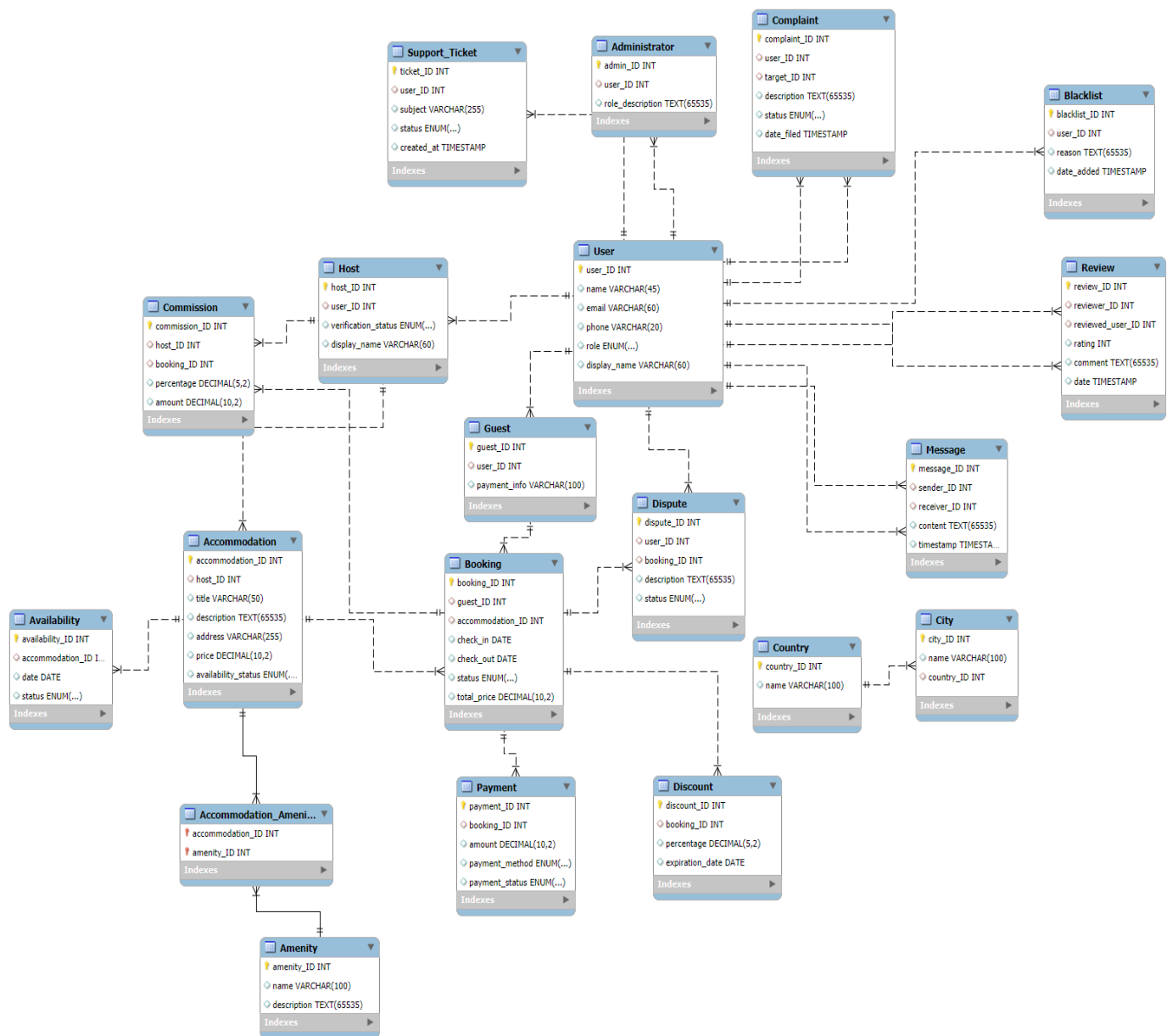
- **User Management:** Register, authenticate, and update user profiles.
- **Accommodation Management:** Hosts can list, modify, or remove accommodations.
- **Search & Filtering:** Guests can search for accommodations based on location, price, availability, and amenities.
- **Booking & Payments:** Guests can book accommodations, make payments, and cancel reservations; hosts receive payments post-check-in.
- **Messaging System:** Guests and hosts can communicate within the platform.
- **Review System:** Guests and hosts can leave ratings and feedback for each other.
- **Admin Oversight:** Manage disputes, verify listings, monitor transactions, and enforce platform policies.

Entity Relationship Model (ERM) Document

1. Introduction

This document presents the Entity Relationship Model (ERM) for the accommodation booking platform. It defines entities, attributes, relationships, and cardinalities using Crow's Foot notation. The model adheres to the project requirements by incorporating at least 20 entities, 2-3 triple relationships, and recursive relationships.

2. ER Diagram



3. Entities and Relationships

3.1 Entities and Primary Keys

1. **User** (**user_ID**, name, email, phone, role, display_name)
2. **Guest** (**guest_ID**, user_ID, payment_info)
3. **Host** (**host_ID**, user_ID, verification_status, display_name)
4. **Accommodation** (**accommodation_ID**, host_ID, title, description, address, price, availability_status)
5. **Booking** (**booking_ID**, guest_ID, accommodation_ID, check_in, check_out, status, total_price)
6. **Payment** (**payment_ID**, booking_ID, amount, payment_method, payment_status)
7. **Review** (**review_ID**, reviewer_ID, reviewed_user_ID, rating, comment, date)
8. **Message** (**message_ID**, sender_ID, receiver_ID, content, timestamp)
9. **Administrator** (**admin_ID**, user_ID, role_description)
10. **Amenity** (**amenity_ID**, name, description)
11. **Accommodation_Amenity** (**accommodation_ID**, **amenity_ID**)
12. **Commission** (**commission_ID**, host_ID, booking_ID, percentage, amount)
13. **Availability** (**availability_ID**, accommodation_ID, date, status)
14. **Dispute** (**dispute_ID**, user_ID, booking_ID, description, status)
15. **City** (**city_ID**, name, country)
16. **Country** (**country_ID**, name)
17. **Discount** (**discount_ID**, booking_ID, percentage, expiration_date)
18. **Complaint** (**complaint_ID**, user_ID, target_ID, description, status, date_filed)
19. **Blacklist** (**blacklist_ID**, user_ID, reason, date_added)
20. **Support_Ticket** (**ticket_ID**, user_ID, subject, status, created_at)

3.2 Triple Relationships

1. **Host ↔ Commission ↔ Booking**
 - A host receives a commission from a booking made by a guest. The commission is calculated as a percentage of the booking amount

2. **Accommodation ↔ Accommodation_Amenity ↔ Amenity**

- An accommodation can have multiple amenities, and an amenity can be available in multiple accommodations. The Accommodation_Amenity table serves as a linking table to associate them.

3. **Booking ↔ Discount ↔ Payment**

- A booking may qualify for a discount, which affects the total amount paid. The discount is applied to the payment associated with the booking.

3.3 Recursive Relationships

1. **User ↔ Review ↔ User**

- A user can review another user (guest reviewing a host, host reviewing a guest).

2. **User ↔ Message ↔ User**

- Users can send messages to each other.

4. Cardinality Specifications

4.1 Key Cardinalities

- **User → (1) ↔ (0,1) Host:** A user can either be a host or not be a host (each host is a user, but not all users are hosts).
- **User → (1) ↔ (0,1) Guest:** A user can either be a guest or not be a guest (each guest is a user, but not all users are guests).
- **User → (1) ↔ (0, N) Booking:** A user can make multiple bookings as a guest.
- **User → (1) ↔ (0, N) Accommodation:** A user can list multiple accommodations as a host.
- **Accommodation → (1) ↔ (0, N) Booking:** One accommodation can have multiple bookings.
- **Booking → (1) ↔ (1, 1) Payment:** Each booking corresponds to one payment.
- **User → (M) ↔ (N) Review:** Users can review multiple other users.
- **Accommodation → (M) ↔ (N) Amenity:** An accommodation can have multiple amenities, and an amenity can belong to multiple accommodations.

Entity	Attribute	Data Type	Description
User	user_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each user.
	name	VARCHAR(25)	Full name of the user.
	email	VARCHAR(60), UNIQUE	User's email address for login and notifications.
	phone	VARCHAR(20)	User's phone number.
	role	ENUM('Guest', 'Host', 'Administrator')	Defines the user type.
	display_name	VARCHAR(60)	Public display name for user profiles.
Guest	guest_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each guest.
	user_ID	INT (FK to User)	Links to the User entity.
	payment_info	VARCHAR(100)	Stores payment method details.
Host	host_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each host.
	user_ID	INT (FK to User)	Links to the User entity.
	verification_status	ENUM('Pending', 'Verified', 'Rejected')	Host's account verification status.
	display_name	VARCHAR(60)	Public display name for host profiles.

Entity	Attribute	Data Type	Description
Accommodation	accommodation_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each accommodation.
	host_ID	INT (FK to Host)	Links to the Host entity.
	title	VARCHAR(50)	Name/title of the accommodation.
	description	TEXT(65535)	Detailed description of the accommodation.
	address	VARCHAR(255)	Location of the accommodation.
	price	DECIMAL(10,2)	Rental price per night.
	availability_status	ENUM('Available', 'Booked', 'Unavailable')	Current status of the accommodation.
Booking	booking_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each booking.
	guest_ID	INT (FK to Guest)	Links to the Guest entity.
	accommodation_ID	INT (FK to Accommodation)	Links to the Accommodation entity.
	check_in	DATE	Check-in date for the booking.
	check_out	DATE	Check-out date for the booking.

Entity	Attribute	Data Type	Description
	status	ENUM('Pending', 'Confirmed', 'Cancelled', 'Completed')	Current booking status.
	total_price	DECIMAL(10,2)	Total cost of the booking.
Payment	payment_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each payment.
	booking_ID	INT (FK to Booking)	Links to the Booking entity.
	amount	DECIMAL(10,2)	Amount paid.
	payment_method	ENUM('Credit Card', 'PayPal', 'Bank Transfer')	Payment method used.
	payment_status	ENUM('Pending', 'Completed', 'Failed')	Payment status.
Review	review_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each review.
	reviewer_ID	INT (FK to User)	User who wrote the review.
	reviewed_user_ID	INT (FK to User)	User who is being reviewed (Guest or Host).
	rating	INT (CHECK BETWEEN 1-5)	Rating score (1 to 5).
	comment	TEXT(65535)	Review text content.
	date	TIMESTAMP	Date and time of review submission.

Entity	Attribute	Data Type	Description
Message	message_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each message.
	sender_ID	INT (FK to User)	User who sent the message.
	receiver_ID	INT (FK to User)	User who received the message.
	content	TEXT(65535)	Message text.
	timestamp	TIMESTAMP	Date and time the message was sent.
Administrator	admin_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each admin.
	user_ID	INT (FK to User)	Links to the User entity.
	role_description	TEXT(65535)	Description of admin responsibilities.
Amenity	amenity_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each amenity.
	name	VARCHAR(100)	Name of the amenity (e.g., Wi-Fi, Pool).
	description	TEXT(65535)	Description of the amenity.
Accommodation_Amenity	accommodation_ID	INT (FK to Accommodation)	Links to an accommodation.
	amenity_ID	INT (FK to Amenity)	Links to an amenity.

Entity	Attribute	Data Type	Description
Commission	commission_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each commission record.
	host_ID	INT (FK to Host)	Links to the Host entity.
	booking_ID	INT (FK to Booking)	Links to the Booking entity.
	percentage	DECIMAL(5,2)	Commission percentage charged.
	amount	DECIMAL(10,2)	Commission amount calculated.
Availability	availability_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for availability records.
	accommodation_ID	INT (FK to Accommodation)	Links to the Accommodation entity.
	date	DATE	Specific date for availability.
	status	ENUM('Available', 'Booked', 'Unavailable')	Status for that date.
Dispute	dispute_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for each dispute.
	user_ID	INT (FK to User)	User who filed the dispute.
	booking_ID	INT (FK to Booking)	Related booking.

Entity	Attribute	Data Type	Description
	description	TEXT(65535)	Details of the dispute.
	status	ENUM('Open', 'Resolved', 'Rejected')	Current dispute status.
City	city_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for a city.
	name	VARCHAR(100)	Name of the city.
	country_ID	INT (FK to Country)	Country where the city is located.
Country	country_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for a country.
	name	VARCHAR(100)	Name of the country.
Discount	discount_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for a discount.
	booking_ID	INT (FK to Booking)	Links to the Booking entity.
	percentage	DECIMAL(5,2)	Discount percentage.
	expiration_date	DATE	Expiry date of the discount.
Complaint	complaint_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for a complaint.
	user_ID	INT (FK to User)	User who filed the complaint.
	target_ID	INT (FK to User)	User against whom the complaint is made.

Entity	Attribute	Data Type	Description
	description	TEXT(65535)	Complaint details.
	status	ENUM('Open', 'Resolved', 'Rejected')	Status of the complaint.
	date_filed	TIMESTAMP	Date and time complaint was submitted.
Blacklist	blacklist_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for a blacklist record.
	user_ID	INT (FK to User)	User who is blacklisted.
	reason	TEXT(65535)	Reason for blacklisting.
	date_added	TIMESTAMP	Date and time of blacklisting.
Support_Ticket	ticket_ID	INT (PK, AUTO_INCREMENT)	Unique identifier for a support ticket.
	user_ID	INT (FK to User)	User who submitted the ticket.
	subject	VARCHAR(255)	Issue summary.
	status	ENUM('Open', 'Resolved', 'Closed')	Current ticket status.
	created_at	TIMESTAMP	Time of ticket creation.