

## Requirement Specification for Airbnb Use-case

**Users** (Guest, Host, Admin)

### Guest

A Guest is a user who searches for and books accommodations.

#### Actions

- Search for available properties based on location, date, price, etc.
- View detailed information about a specific property.
- Book a property for a specified date range.
- Post a review and rating after a stay.
- View past and upcoming bookings.
- Update personal information and payment methods.

#### Required Data and Functions

- **Guest Data:** Guest ID, name, email, phone, payment details, booking history, reviews.
- **Search Functionality:** Location, date, filters (price, amenities, property type).
- **Booking Functionality:** Booking ID, property ID, dates, total price, payment status.
- **Review Functionality:** Review ID, property ID, rating, comments, review date.

### Host

A Host is a user who lists properties and manages bookings.

#### Actions

- Add a new property to the platform with details and photos.
- Edit or remove existing property listings.
- See incoming booking requests and manage accepted bookings.
- Respond to booking requests from Guests.
- Set and update the availability calendar for properties.
- Post a review of the Guest after their stay.
- Track earnings from bookings.
- Update personal information and property details.

#### Required Data and Functions

- **Host Data:** Host ID, name, email, phone, reviews.
- **Listing Data:** Property ID, title, description, location, price per night, amenities, photos, availability calendar.
- **Booking Management:** Booking ID, guest details, dates, status (pending, approved, declined), payment status.

- **Review Functionality:** Review ID, guest ID, rating, comments, review date.
- **Earnings Data:** Booking ID, property ID, earnings amount, payout status.

## Admin

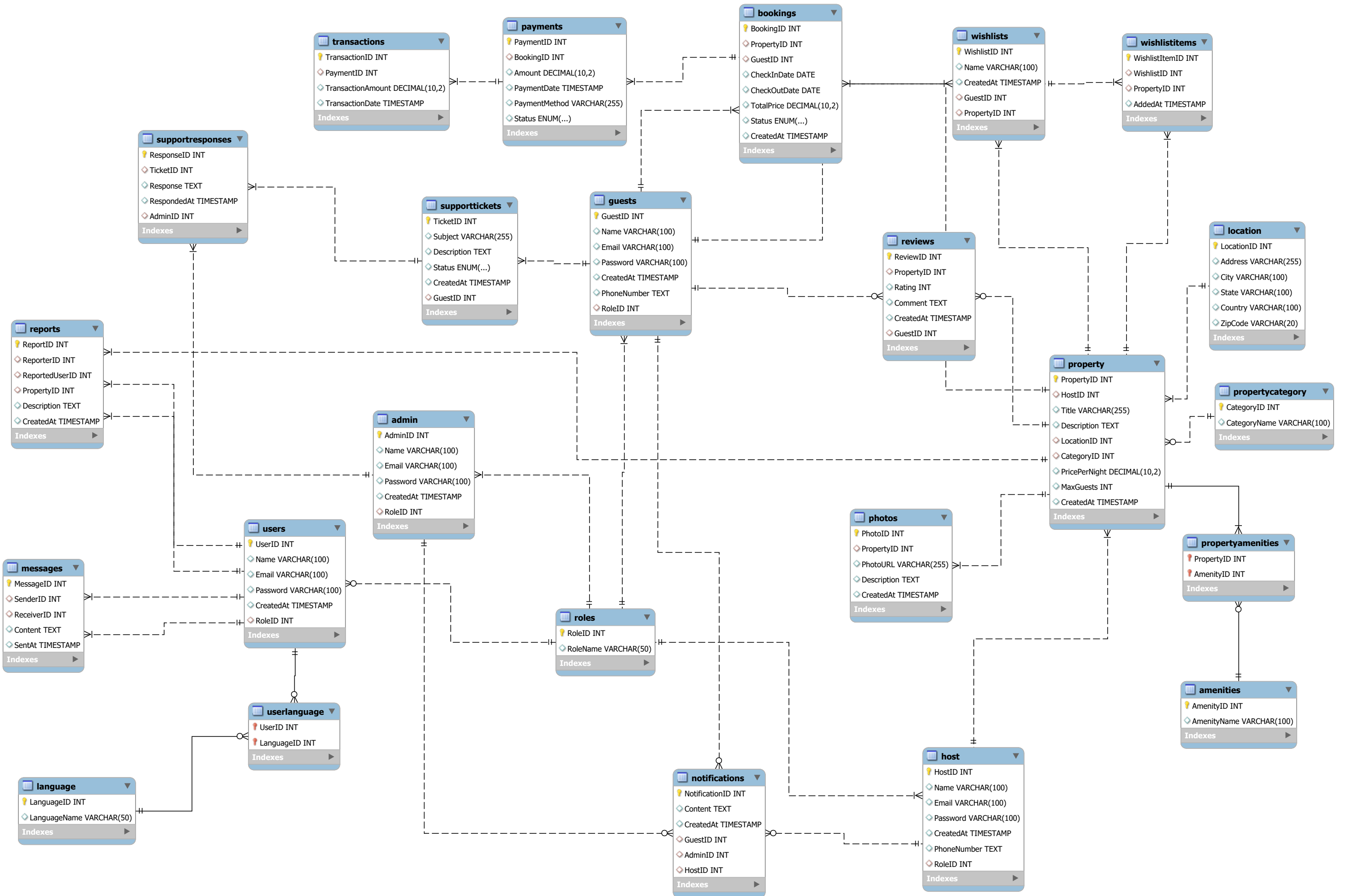
An Admin manages the overall platform, ensuring smooth operations and compliance with policies.

## Actions

- View, edit, or deactivate user accounts (both Guests and Hosts).
- Approve or remove property listings based on content policies.
- Oversee payment processes, resolve disputes, and handle refunds.
- Generate reports on platform activity, bookings, earnings, user activity.
- Resolve issues between Guests and Hosts, such as disputes over bookings or reviews.
- Update platform settings, such as fees, policies, and support information.

## Required Data and Functions

- **Admin Data:** Admin ID, name, email, role permissions.
- **User Management Data:** User ID, role, status (active/inactive), contact details, history.
- **Listing Data:** Property ID, host details, listing status, content (photos, descriptions).
- **Payment Data:** Payment ID, booking ID, amounts, status, refund status.
- **Report Data:** Aggregated data on users, bookings, earnings, disputes.
- **Dispute Management:** Ticket ID, booking ID, involved parties, dispute details, resolution status.
- **Platform Settings:** Fee structures, policy text, support contact information.



## Data Dictionary

Admin	<u>AdminID</u>	Unique identifier	INT
	Name	Name of the admin	varchar
	Email	Admin's unique email	varchar
	Password	Admin's authentication information	varchar
	CreatedAt	Time of creation	Timestamp
	UserID	Foreign key referencing users	INT

Amenities	AmenityID	Unique identifier	INT
	AmenityName	Name of the Amenity	varchar

Bookings	<u>BookingID</u>	Unique identifier	INT
	PropertyID	Foreign key	INT
	GuestID	Foreign key	INT
	CheckInDate	Date of check in	Date
	CheckOutDate	Date of check out	Date
	TotalPrice	Toatal cost of booking	Decimal
	Status	Current state of entity	ENUM
	CreatedAt	Time of creation	Timestamp

Guests	<u>guestID</u>	Unique identifier	INT
	Name	Name of the guest	varchar
	Email	Guest email	varchar
	Password	Guest authentication information	varchar
	CreatedAt	Time of creation	Timestamp
	phonenumber	Guest phone number	INT
	UserID	Foreign key	INT

Host	<u>hostID</u>	Unique identifier	INT
	Name	Name of the host	varchar
	Email	Host email	varchar
	Password	Host authentication information	varchar
	CreatedAt	Time of creation	Timestamp
	phonenumber	Host phone number	INT
	UserID	Foreign key	INT

Language	<u>LanguageID</u>	Unique identifier	INT
	LanguageName	Name of the language	varchar

Location	LocationID	Unique identifier	INT
----------	------------	-------------------	-----

	Address	Property address	varchar
	City	Property city	varchar
	State	Property state	varchar
	Country	Property country	varchar
	ZipCode	Property zipcode	varchar

Messages	MessageID	Unique identifier	INT
	SenderID	Foreign key	INT
	ReceiverID	Foreign key	INT
	Content	Information stored in the message	Text
	SentAt	Time message was sent	Timestamp

Notifications	<u>NotificationID</u>	Unique identifier	INT
	UserID	Foreign key	INT
	Content	Information in the notification	Text
	CreatedAt	Time of creation	Timestamp

Payments	<u>PaymentID</u>	Unique identifier	INT
	BookingID	Foreign key	INT
	Amount	Monetary value	Decimal
	PaymentDate	Date of payment	Timestamp
	PaymentMethod	Method of payment	ENUM
	Status	Current state of entity	ENUM

Photos	<u>PhotoID</u>	Unique identifier	INT
	PropertyID	Foreign key	INT
	PhotoURL	Link pointing to photo	varchar
	Description	Detailed information of entity	Text
	CreatedAt	Time of creation	Timestamp

Property	<u>PropertyID</u>	Unique identifier	INT
	HostID	Foreign key	INT
	Title	Descriptive name of the property	varchar
	Description	Detailed information of entity	Text
	LocationID	Foreign key	INT
	CategoryID	Foreign key	INT
	PricePerNight	Price of property per night	Decimal
	MaxGuests	Number of guest allowed	INT
	CreatedAt	Time of creation	Timestamp

Property amenities	PropertyID	Foreign key	INT
	AmenityID	Foreign key	INT

Property Category	CategoryID	Unique identifier	INT
	CategoryName	Name of the category	varchar

Reports	<u>ReportID</u>	Unique identifier	INT
	ReporterID	Foreign key	INT
	ReportedUserID	Foreign key	INT
	PropertyID	Foreign key	INT
	Description	Detailed information of entity	Text
	CreatedAt	Time of creation	Timestamp

Reviews	<u>ReviewID</u>	Unique identifier	INT
	PropertyID	Foreign key	INT
	UserID	Foreign key	INT
	Rating	Metric evaluation	INT
	Comment	Document any relevant information	Text
	CreatedAt	Time of creation	Timestamp

Roles	<u>RoleID</u>	Unique identifier	INT
	RoleName	Name of the role(admin, guest,...)	varchar

Support Responses	<u>ResponseID</u>	Unique identifier	INT
	TicketID	Foreign key	INT
	UserID	Foreign key	INT
	Response	User's response	Text
	RespondedAt	Time the response was recorded	Timestamp

Support Tickets	<u>TicketID</u>	Unique identifier	INT
	UserID	Foreign key	INT
	Subject	Reason for support	varchar
	Description	Detailed information of entity	Text
	Status	Current state of entity	ENUM
	CreatedAt	Time of creation	Timestamp

Transactions	<u>TransactionID</u>	Unique identifier	INT
	PaymentID	Foreign key	INT
	TransactionAmount	Monetary value of a transaction	Decimal
	TransactionDate	Date of transaction	Timestamp

User Language	UserID	Foreign key	INT
	LanguageID	Foreign key	INT

User Rating	<u>RatingID</u>	Unique identifier	INT
	UserID	Foreign key	INT
	RatedByUserID	Foreign key	INT
	Rating	Metric evaluation	INT
	Comment	Document any relevant information	Text
	CreatedAt	Time of creation	Timestamp

Users	UserID	Unique identifier	INT
	Name	User's name	varchar
	Email	User's email	varchar
	Password	User's authentication information	varchar
	CreatedAt	Time of creation	Timestamp
	RoleID	Foreign key	INT

Wishlist Items	<u>WishlistItemID</u>	Unique identifier	INT
	WishlistID	Foreign key	INT
	PropertyID	Foreign key	INT
	AddedAt	Time added	Timestamp

Wishlists	WishlistID	Unique identifier	INT
	UserID	Foreign key	INT
	Name	Wishlist name	varchar
	CreatedAt	Time of creation	Timestamp

## **Summary**

In this conceptual phase, the user needs and operational requirement were addressed by identifying likely problems, such as booking management, transaction processing, customer support, safety using the platform, property management and suitable accommodation search.

Following the identification of these potential issues, developing a system that handles booking request, send out reminders and notification to users, implementing a payment gateway that allows various payment options and currency, ensuring a robust customer support system, integrating review systems and verification processes is paramount, and has been implemented in this phase.

In conclusion, by addressing the likely problems and defining solutions, this conceptual design sets the foundation and aims to deliver a secure and user friendly experience for all participants.