Conception Phase

The aim of this study is to design and implement a database for the Airbnb use case. The Airbnb platform consists of different roles, such as guests and hosts, each performing various operations. In this project, a relational database model has been created to store and process data specific to Airbnb.

First, during the conceptual design phase, the core components of the platform were analyzed, and an Entity-Relationship Model (ERM) was developed, including fundamental entities such as users, properties, bookings, payments, and communication mechanisms. This model consists of tables and relationships that support the platform's essential functions.

This system is designed in a modular and easily scalable manner. However, as data volume increases, performance issues may arise. Due to these factors, the system has both positive and negative aspects**.**

Following this phase, the implementation process will begin. Initially, the designed database model will be defined as a .sql file in MySQL, and the necessary tables will be created. Then, sample data entries will be made to test the system's functionality. During the testing process, improvements will be made by evaluating table relationships, data integrity, and performance criteria**.**

Methodology and Tools

This project is developed based on a relational database management system (RDBMS). Crow's Foot Notation is used to visualize the relationships between tables and to express the data model more clearly. This notation makes the database structure easier to understand by clearly showing the cardinality and types of relationships between entities. MySQL has been chosen due to its strong community support, robust data management tools, and reliability. Building on this foundation, the technical modeling phase will further detail how the platform operates and explain the flow of data. The following section provides a detailed overview of user roles, relationships, and system functionality**.**

User Roles and System Functionality

The system consists of different types of users, each performing specific functions. The primary roles include Guests and Hosts. These roles define the level of interaction a user has with the platform and the actions they can perform. Guests are users who browse properties, make reservations, and complete payments for their stays. They can leave reviews and ratings based on their experiences and communicate with hosts through the platform's messaging system.

Hosts, on the other hand, are responsible for creating and managing property listings. They set availability, determine pricing, and handle booking requests. Once a booking is confirmed, hosts manage guest interactions, ensuring a smooth chck-in process and responding to any inquiries or issues that may arise. Hosts also receive payments based on platform policies and can respond to reviews left by guests to maintain their reputation.

To support these roles and actions, the system relies on a well-structured Entity-Relationship Model (ERM) that includes various tables and relationships. The user table stores basic user information such as names, emails, and passwords. Property listings are managed within the property table, which includes details like descriptions, amenities, and pricing. Location of properties can be accessed from this table. The booking table tracks reservations, linking guests to specific properties, while the payment table stores transaction records. Communication between users is facilitated through the conversation and message tables, ensuring seamless interaction. Reviews and ratings are handled via the guest_review table, allowing users to provide feedback on their experiences.

Key Functionalities

A few key functions are crucial for the platform to run smoothly. Authentication mechanism provides secure access to user accounts. For this, users must have uniquely identifiable e-mail addresses. For reservation management, the availability table manages room and property bookings, ensuring accurate availability updates. Payment processing features facilitate a smooth financial flow by securely managing transactions between guests and hosts. Cancellation policy's role is ensuring transparency, defining refund conditions, and preventing disputes. Additionally, the messaging system enables effective communication, while review and feedback mechanisms help maintain the reliability and quality of listings on the platform. By implementing these roles, actions and data structures, the system ensures a smooth and efficient experience for all users.

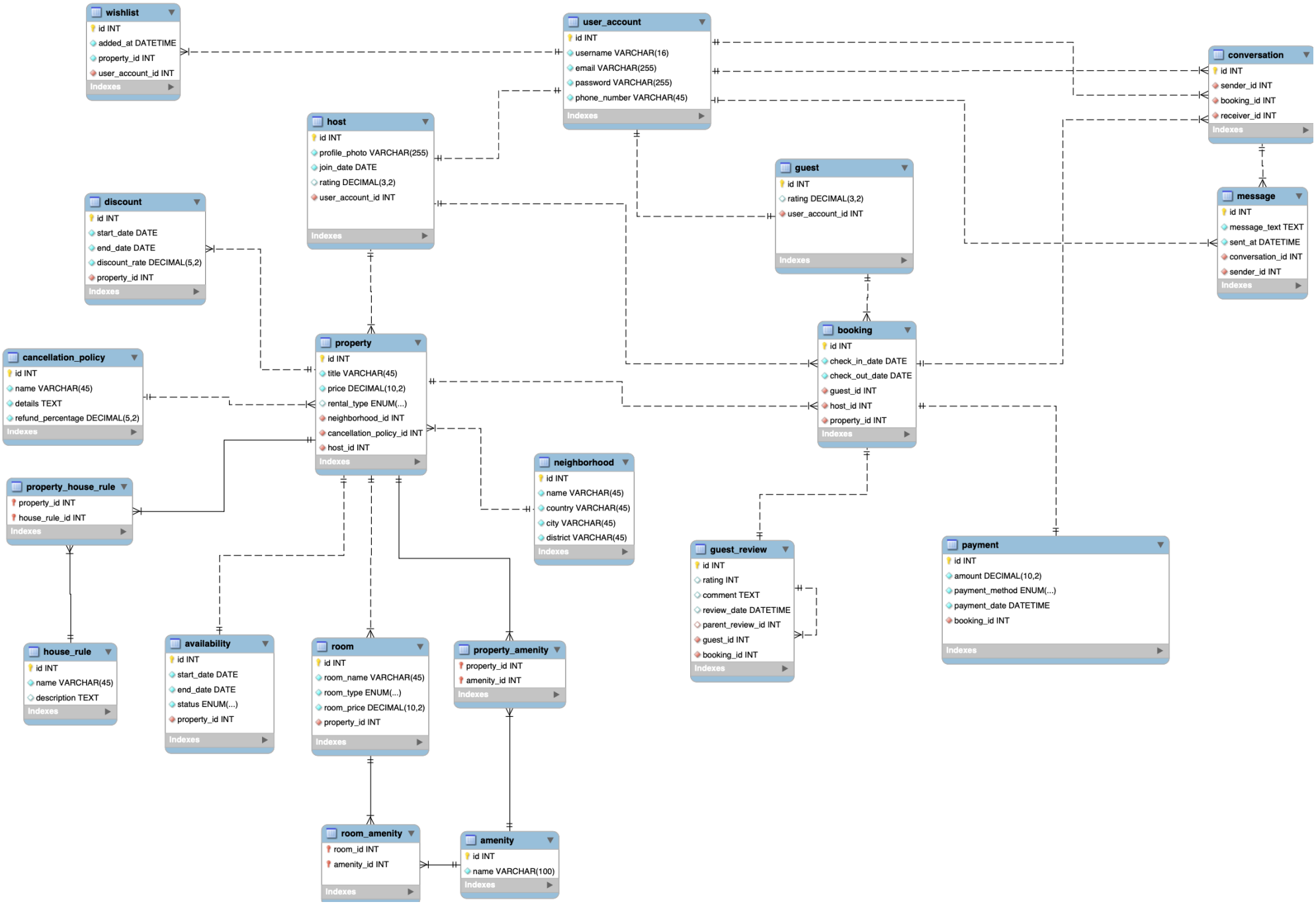
wishlist
id INT
added_at DATETIME
property_id INT
user_account_id INT
Indexes
user_account
id INT
username VARCHAR(16)
email VARCHAR(255)
password VARCHAR(255)
phone_number VARCHAR(45)
Indexes
conversation
id INT
sender_id INT
booking_id INT
receiver_id INT
Indexes
host
id INT
profile_photo VARCHAR(255)
join_date DATE
rating DECIMAL(3,2)
user_account_id INT
Indexes
guest
id INT
rating DECIMAL(3,2)
user_account_id INT
Indexes
message
id INT
message_text TEXT
sent_at DATETIME
conversation_id INT
sender_id INT
Indexes
discount
id INT
start_date DATE
end_date DATE
discount_rate DECIMAL(5,2)
property_id INT
Indexes
booking
id INT
check_in_date DATE
check_out_date DATE
guest_id INT
host_id INT
property_id INT
Indexes
cancellation_policy
id INT
name VARCHAR(45)
details TEXT
refund_percentage DECIMAL(5,2)
Indexes
property
id INT
title VARCHAR(45)
price DECIMAL(10,2)
rental_type ENUM(...)
neighborhood_id INT
cancellation_policy_id INT
host_id INT
Indexes
neighborhood
id INT
name VARCHAR(45)
country VARCHAR(45)
city VARCHAR(45)
district VARCHAR(45)
Indexes
property_house_rule
property_id INT
house_rule_id INT
Indexes
guest_review
id INT
rating INT
comment TEXT
review_date DATETIME
parent_review_id INT
guest_id INT
booking_id INT
Indexes
payment
id INT
amount DECIMAL(10,2)
payment_method ENUM(...)
payment_date DATETIME
booking_id INT
Indexes
house_rule
id INT
name VARCHAR(45)
description TEXT
Indexes
availability
id INT
start_date DATE
end_date DATE
status ENUM(...)
property_id INT
Indexes
room
id INT
room_name VARCHAR(45)
room_type ENUM(...)
room_price DECIMAL(10,2)
property_id INT
Indexes
property_amenity
property_id INT
amenity_id INT
Indexes
room_amenity
room_id INT
amenity_id INT
Indexes
amenity
id INT
name VARCHAR(100)
Indexes

Table name	Attribute	Data type	Description
user	id	INT (PK)	Unique id for each user
	user_name	VARCHAR	User’s full name
	email	VARCHAR	User’s email address
	password	VARCHAR	User’s password
	phone_number	VARCHAR	User’s phone number
host	id	INT (PK)	Unique ID for each host
	profile_photo	VARCHAR	URL of host's profile photo
	join_date	DATE	Date when the host joined
	rating	DECIMAL	Host's rating
	user_id	INT (FK)	User ID of the host
guest	id	INT (PK)	Unique guest ID
	rating	DECIMAL	Guest’s rating
	user_id	INT (FK)	User id of the guests
wishlist	id	INT (PK)	Unique wishlist entry id
	added_at	DATETIME	Timestamp when added to wishlist
	user_id	INT (FK)	User who added the property
	property_id	INT (FK)	Property added to wishlist
property	id	INT (PK)	Unique identifier for each property
	title	VARCHAR	Title of the property listing
	price	DECIMAL	Nightly rental price
	rental_type	ENUM(‘Entire house’, ‘Shared’)	Rental type of the property
	neighborhood_id	INT (FK)	Neighborhood location
	cancellation_policy_id	INT (FK)	Cancellation policy applied
	host_id	INT (FK)	Host who owns the property
availability	id	INT (PK)	Unique identifier for each availability entry
	start_date	DATE	Start date of availability
	end_date	DATE	End date of availability
	status	ENUM (‘available’, ‘booked’)	Availability status (available, booked)
	property_id	INT (FK)	Property associated with the availability
room	id	INT (PK)	Unique identifier for each room
	room_name	VARCHAR	Name of the room
	room_type	ENUM ('private', 'shared')	Type of the room (private, shared)
	room_price	DECIMAL	Price of each room
	property_id	INT (FK)	Property that this room belongs to
room_amenity	room_id	INT (PK-FK)	The room this amenity is associated with
	amenity_id	INT (PK-FK)	The amenity that belongs to this room
property_amenity	property_id	INT (PK-FK)	Property associated with the amenity
	amenity_id	INT (PK-FK)	The amenity that belongs to this property
amenity	id	INT (PK)	Unique identifier for each amenity
	name	VARCHAR	Name of the amenity (WiFi, Pool, etc.)
neighborhood	id	INT (PK)	Unique identifier for each neighborhood
	name	VARCHAR	Neighborhood name
	country	VARCHAR	Country the neighborhood is located in

	city	VARCHAR	City the neighborhood is in
	district	VARCHAR	District name
discount	id	INT (PK)	Unique identifier for each discount
	start_date	DATE	Start date of the discount
	end_date	DATE	End date of the discount
	discount_rate	DECIMAL	Percentage discount offered
	property_id	INT (FK)	Property eligible for the discount
cancellation_policy	id	INT (PK)	Unique identifier for each cancellation policy
	name	VARCHAR	Policy name
	details	TEXT	Policy details
	refund_percentage	DECIMAL	Percentage of refund available
property_house_rule	property_id	INT (PK-FK)	The property associated with this house rule
	house_rule_id	INT (PK-FK)	The house rule applied to this property
house_rule	id	INT (PK)	Unique identifier for each house rule
	name	VARCHAR	Name of the house rule
	description	TEXT	Detailed description of the rule
booking	id	INT (PK)	Unique identifier for each booking
	check_in_date	DATE	Start date of the booking
	check_out_date	DATE	End date of the booking
	guest_id	INT (FK)	Guest who made the booking
	host_id	INT (FK)	Host associated with the booking
	property_id	INT (FK)	Property booked
guest_review	id	INT (PK)	Unique identifier for each review
	rating	INT	Star rating given in the review
	comment	TEXT	Review comments
	review_date	DATETIME	Date the review was written
	parent_review_id	INT (FK)	References another review if it's a reply
	guest_id	INT (FK)	Guest being reviewed
	booking_id	INT (FK)	Booking associated with the review
payment	id	INT (PK)	Unique identifier for each payment
	amount	DECIMAL	Total amount of payment
	payment_method	ENUM(‘paypal’, ‘credit_card’)	Payment method used ('paypal', 'credit_card')
	payment_date	DATETIME	Timestamp of the payment
	booking_id	INT (FK)	Booking associated with the payment
conversation	id	INT (PK)	Unique identifier for each conversation
	receiver_id	INT (FK)	User receiving the message
	sender_id	INT (FK)	User sending the message
	booking_id	INT (FK)	Booking associated with the conversation
message	id	INT (PK)	Unique identifier for each message
	message_text	TEXT	The actual text content of the message
	sent_at	DATETIME	The time the message was sent
	conversation_id	INT (FK)	The conversation this message belongs to
	sender_id	INT (FK)	The user who sent the message