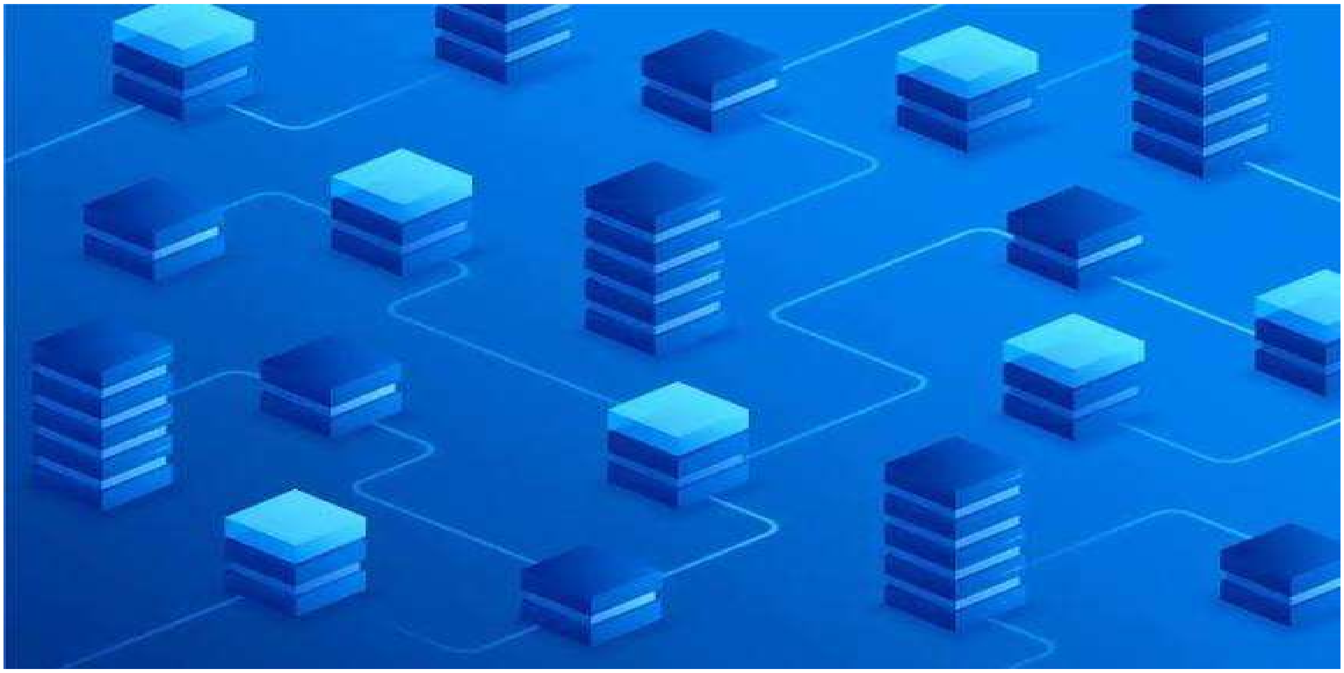
INTERNATIONAL
UNIVERSITY OF
APPLIED SCIENCES

CONCEPTION PHASE



NAME: Bokang Tjamela **DATE: 27/11/2025**

MATRICULATION: 102401207

TUTOR: Sharam Dadashnia

Specification: Database modeling for Airbnb platform.

Abstract:

This project aims to develop a comprehensive database for an online booking platform like Airbnb, following three essential phases: Conception, Development, and Finalization. Airbnb is a global platform that connects hosts offering accommodations with guests looking for temporary lodging. Guests can search for available properties based on various criteria such as location, price, and dates, and make reservations. Payment is processed through credit or debit cards. Each user, whether a host or a guest, has a profile page. Hosts are required to upload a picture and phone number, while guests provide additional information. Users can link their profiles to social networks like Facebook and Instagram. Both hosts and guests can receive discounts via coupons and eave reviews, ratings, and complaints about their experiences.

Summary: Roles and Actions –

- **Guests –** search for properties, book rentals, leave reviews and manage their profiles. Therefore the database handle these tasks by organizing and retrieving the necessary data effectively .

*** Hosts –** lists properties, manage bookings, respond to guest queries and leave reviews. Property details and availability are tracked in the database.

- **TravelAdmins –** oversee everything, platform's operations, manage user accounts, handle customer service and generate reports. Access to all data that is required for reporting and operations.

Required Data and Functions:

User data: UserID, Name, Email, Phone, ProfilePicture.

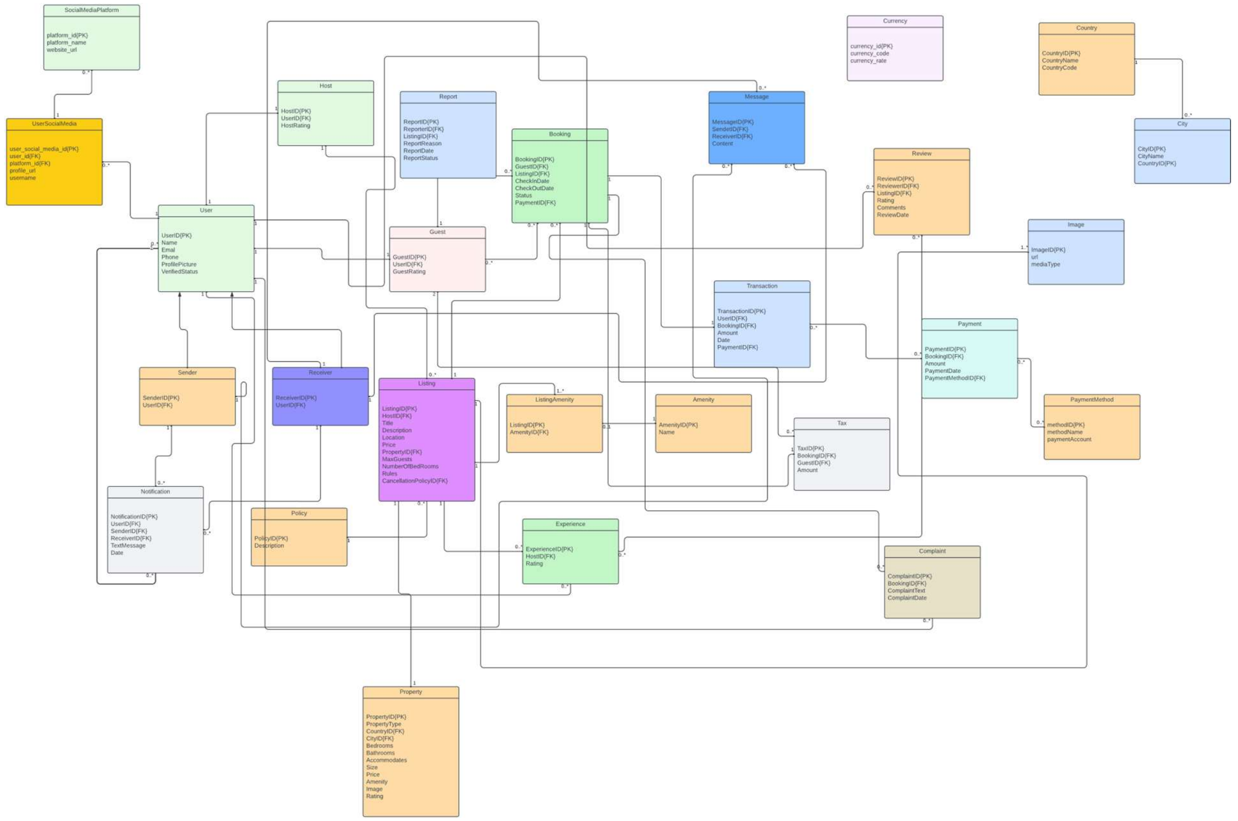
Listing data: ListingID, HostID, Title, Description.

Booking data: BookingiD, GuestID, ListingID, BookingDates, Status, Payment**.**

Review data: ReviewID, ReviewerID, ListingID, Rating, Comments.

Entity Relationship Model (ERM):

- The revised ERM reflects on improved payment process, with the attributes like TransactionType, PaymentMethod, and RefundProcessedDate for the transaction management. The Review entity allows both guests and hosts to leave reviews, with attributes like ReviewType and ReviewID. The model minimizes redundancy and ensures data integrity across the platform's 21 entities, covering profiles, reservation, payments and reviews.



- **Triple Relationships:** exist between tables like Guest, Booking and Transaction to connect users, reservations and payments. A similar structure connects Guests, Bookings and Reviews.
- **Recursive Relationships:** the host table allows tracking referrals
- **Cardinality:** Martin notation, also known as Crow's Foot notation, is used to define relationships and cardinality between entities.

Normalization:

- During the design process, normalization was prioritized to minimize redundancy and enhance data integrity. For example, the Policy entity collects policy information, while full address details (street, state, country) are stored in appropriate entities. Foreign keys link tables as Location and VacationRented to maintain consistency without repeating data.

Tools and Documentation:

- MySQL from Microsoft was used for ERM creation, testing, and managing the design process. Each of the 21 entities is populated with 20 entities. The SQL code documentation contains information about the database structure and covers the creation, relationships and sample data entities. The design's functionality was confirmed through tests cases that verify proper handling of bookings and payments.

Data Dictionary:

1. User:

UserID	INTEGER	Unique identifier for the user
Name	STRING	Name of the user
Email	STRING	Email address of the user
Phone	STRING	Phone number of the user
ProfilePicture	STRING	URL of the profile picture
VerifiedStatus	BOOLEAN	Verification status of the user

2. Guest:

GuestID	INTEGER	Unique identifier for the host
UserID	INTEGER	Foreign key to user
GuestRating	FLOAT	Rating of the guest

3. Host:

HostID	INTEGER	Unique identifier for the host
UserID	INTEGER	Foreign key to user
HostRating	FLOAT	Rating of the guest

4. Listing:

ListingID	INTEGER	Unique identifier for the listing
HostID	INTEGER	Foreign key to host
Title	STRING	Title of the listing
Description	STRING	Description of the listing
Location	STRING	Location of the listing
Price	FLOAT	Price per night
PropertyID	INTEGER	Type of the property (Foreign key)
MaxGuests	INTEGER	Maximum number of guests allowed
NumberOfBedRooms	INTEGER	Number of bedrooms
Rules	STRING	Rules for the listing

CancellationPolicyID	INTEGER	Foreign key to policy.

5. Booking:

BookingID	INTEGER	Unique identifier for the booking
GuestID	INTEGER	Foreign key to guest
ListingID	INTEGER	Foreign key to listing
CheckInDate	DATE	Check-in date
CheckOutDate	DATE	Check-out date
Status	STRING	Status of the booking
PaymentID	INTEGER	Foreign key to payment

6. Payment:

PaymentID	INTEGER	Unique identifier for the payment
BookingID	INTEGER	Foreign key to booking
Amount	Float	Amount paid
PaymentDate	DATE	Date of payment
PaymentMethodID	INTEGER	Foreign key to payment method

7. PaymentMethod:

MethodID	INTEGER	Unique identifier for the method
MethodName	STRING	Name of the payment
PaymentAccount	STRING	Account used for payment

8. Transaction:

TransactionID	INTEGER	Unique identifier for the transaction
UserID	INTEGER	Foreign key to user
BookingID	INTEGER	Foreign key to booking
Amount	FLOAT	Transaction amount
Date	DATE	Transaction date
PaymentID	INTEGER	Foreign key references to payment

9. Property:

PropertyID	INTEGER	Unique identifier for the property
PropertyType	VARCHAR	Type of property
CountryID	INTEGER	Foreign key references to country
CityID	INETEGER	Foreign key references to city
Bedrooms	INTEGER	Number of bedrooms
Bathrooms	INTEGER	Number of bathrooms
Accommodates	INTEGER	
Size	FLOAT	Size of the property
Amenitites	VARCHAR	Amenities
ImageURL	VARCHAR	URL to the image
Rating	FLOAT	Rating of the property

10. Tax:

TaxID	INTEGER	Unique identifier
BookingID	INTEGER	Foreign key to booking
Amount	FLOAT	Tax amount

11. Review:

ReviewID	INTEGER	Unique identifier for the review
ReviewerID	INTEGER	Foreign key to the user
ListingID	INTEGER	Foreign key to listing
Rating	FLOAT	Rating given
Comments	STRING	Review comment
ReviewDate	DATE	Date of the review

12. Report:

ReportID	INTEGER	Unique identifier for the report
ReporterID	INTEGER	Foreign key to the reporter
ListingID	INTEGER	Foreign key to listing
ReportReason	STRING	Reason for reporting
ReportDate	DATE	Date for reporting

ReportStatus	STRING	Status of the report

13. Message:

MessageID	INTEGER	Unique identifier for the message
SenderID	INTEGER	Foreign key to the user
ReceiverID	INTEGER	Foreign key to user
Content	STRING	Content of the message

14. Notification:

NotificationID	INTEGER	Unique identifier for the notification
UserID	INTEGER	Foreign key to user
SenderID	INTEGER	Foreign key to the sender
ReceiverID	INTEGER	Foreign key to the receiver
TextMessage	STRING	Notification text
Date	DATE	Date of the notification

15. Policy:

PolicyID	INTEGER	Unique identifier for the policy
Description	STRING	Description of the policy

16. Amenity

AmenityID	INTEGER	Description
Name	STRING	Name of the amenity

ListingID	INTEGER	Foreign key to the listing
AmenityID	INTEGER	Foreign key to the amenity

17. Experience:

ExperienceID	INTEGER	Unique identifier for the experience
HostID	INTEGER	Foreign key to the host
Rating	FLOAT	Rating of the experience

18. Image:

ImageID	INTEGER	Unique identifier for the image
URL	STRING	URL to the image
MediaType	STRING	Type of the media

19. Complaint:

ComplaintID	INTEGER	Unique identifier for the complaint
BookingID	INTEGER	Foreign key to booking
ComplaintText	STRING	Text of the complaint
ComplaintDate	DATE	Date of complaint

20. Social media platform:

Platform_id	INTEGER	Unique identifier for the platform
Plaform_name	STRING	Name of the platform
Website_url	STRING	URL to the platform

21. User social account:

user_social_media_id	INTEGER	Unique identifier for the account
User_id	INTEGER	Foreign key references the user
Platform_id	INTEGER	Foreign key references the platform
Profile_url	STRING	URL to user account
Username	STRING	Username of the user

END!