CSCC43 project

Muhammad Hamza
Student ID: 1007996411

July 2023

1 Objective

The objective of this project is to create a BnB hosting and renting system that allows users to seamlessly interact with a database.

2 Problems and Solutions

A conceptual problem occurred with the report that had to determine the popular noun phrases from the comments. It was tough to determine what a noun was as there were many factors that can determine if a word is a noun or not. As a solution I used the most common words before nouns and checked for those in the comments and simply counted the noun that came after. Once that was done I displayed the ten most frequent nouns.

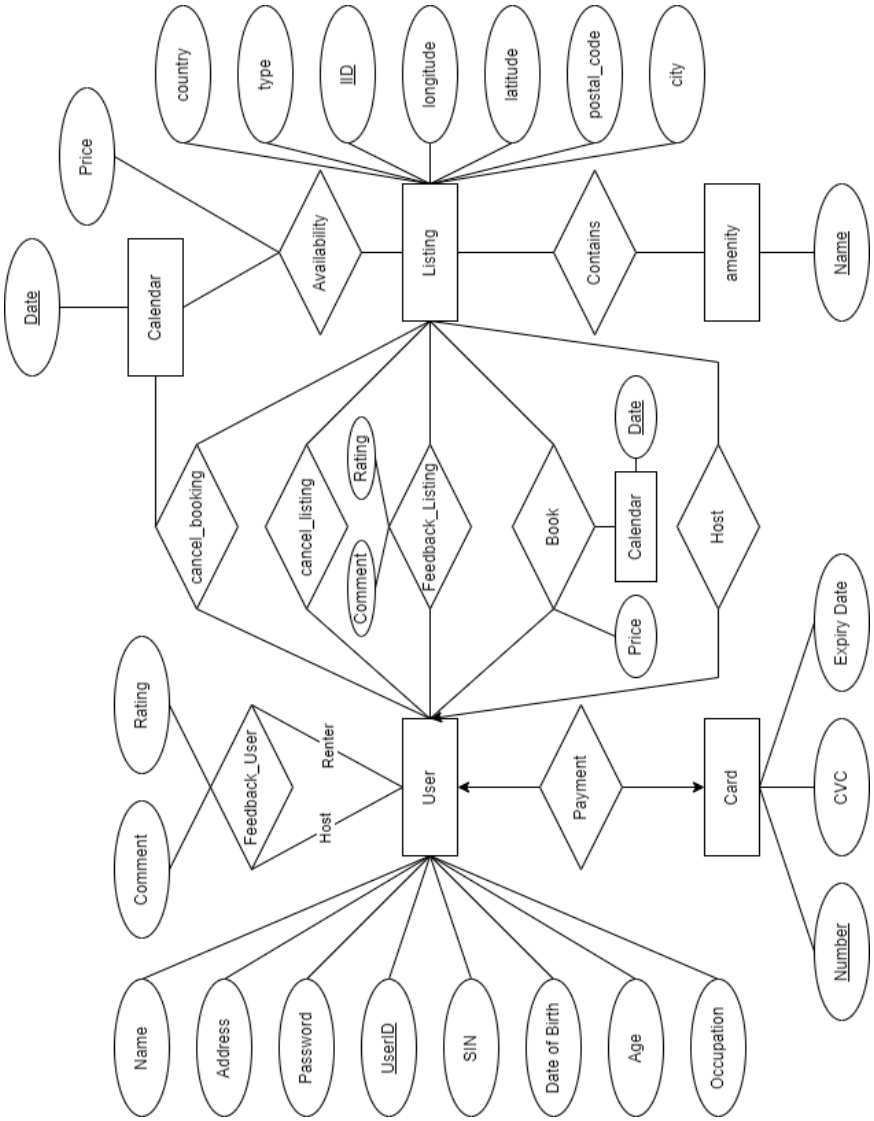
Another problem was with receiving user input and writing it as a query since it was very inefficient to do a bunch of string concatenations. As a solution I found the Prepared statements simply let you use a ? as a placeholder for variables and insert them into queries efficiently so I adapted this in some places of the code.

Lastly determining the distance between 2 coordinates was also something I was unsure of. I looked through the SQL documentation as a result and found that I can use math operations a aggregations and it allowed my to use SQRT and POW to determine the distance between 2 points.

3 Assumptions

Users can only book 1 listing at a given date since bookings are for the whole day. Thus users can't book for themselves and someone else on a given day
Once a booking is cancelled it may not be booked again, likewise once a listing is cancelled it may not be activated again the host must make a new one.
I assume that all prices are in Canadian dollars regardless of the region.
Users are only limited to one card when paying for simplicity. Thus the assumption is in the schema one user can have one card, and no users can share cards.

4 ER diagram


country
type
IID
longitude
latitude
postal_code
city
Price
Availability
Listing
Contains
amenity
Name
Date
Calendar
cancel_booking
cancel_listing
Rating
Comment
Feedback_Listing
Book
Date
Calendar
Price
Host
Rating
Comment
Feedback_User
Renter
Host
User
Payment
Card
Expiry Date
CVC
Number
Name
Address
Password
UserID
SIN
Date of Birth
Age
Occupation

5 Relational Schema

Listing(lID, type, longitude, latitude, postal_code, city, country)
User(UserID, Password, SIN, Name, address, Date of Birth, Age, Occupation)
Payment(UserID, Number)
Card(Number, CVC, Expiry Date)
Contains(Name, lID) Renamed to Amenities in the Database for convenience
Availability(Date, lID Price)
Book(Date, lID, UserID, price)
Host(lID, UserID)
cancel_listing(lID, UserID)
cancel_booking(Date, lID, UserID)
Feedback_User(Host.userID, renter.userID, Comment, Rating)
Feedback_Listing(lID, userID, Comment, Rating)

The calendar schema was collapsed into the relational schemas it was used in.

Currently the schema that are not in normal form are the User and the Listing schema. This is because the SIN number functionally determines the name, address and Date of Birth, Age and Occupation. The SIN Number is not a superkey and the right hand side is not part of a key so this is not in normal form to put it in normal form a person schema can be created thus the split can be as follows.

Person(SIN, Name, address, Date of Birth, Age, Occupation)
User(UserID, Password, SIN)

For the Listing schema the longitude and latitude determine the the location thus a Location scheam can be created.

Location(longitude, latitude, postal_code, city, country)
Listing(lID, type, longitude, latitude)

The rest of the schema are in 3NF normal form as the only FD is the key functionally determining the rest of the attributes, so all FDs have a superkey on the LHS.

6 DDL Statements and Source Code

See github at https://github.com/MD-Hamza/MyBnB

7 Limitations

A limitation of this application is that due to it being a text-based application and not a GUI its a bit hard to navigate through pages. If a user wanted to go to the menu halfway through creating a listing the program wont allow that until the listing is complete. A lack of a GUI also makes it hard to keep track of information for example if you added 5 availabilities then it may be hard to see all of them at once as you may need to scroll up to the previous outputs of the console which is inconvenient as compared to having it all in one calendar display.

A limitation is the lack of data available which causes some recommendations to have error. For instance if an amenity is not found nearby it gives to expected revenue increase as $0.

The system has no way of verifying if the entered address or payment information is correct, this would have to be done by some external database which was out of scope for this project.

Lastly, the noun query requires you to search for nouns and currently there's a very primitive technique to look for nouns which is simply appending the word after any articles. This causes many words that are not nouns to be included and some nouns to be missed.

8 Possibility of Improvement

This text-based interface can be adapted into a GUI. The GUI would allow navigation between pages and would contain features such as calendars, tables, and images to enhance the user experience. Moreover, Given more data the recommendations can provide better results. Additionally, a AI model can be used to determine what nouns are present and can be used to give better recommendations for the amenities and price.