CSCC43
Final Project

Date: Aug 8, 2022

Annanya Sharma
Vanshika Virmani

Contents

Description

MyBnb is a database that supports similar operations to home-sharing services like AirBnb.

The database and the application are implemented in Java embedded SQL using MySQL.

As this project is based on SQL and JDBC, we decided to have the minimum text-based interface on the terminal to make the queries and reports interactive.

What are some conceptual problems we encountered?

- It was our first time using JDBC, so initially, we had to get used to it.
- We also spent time discussing finding the solution to run multiple filters in a single query, while making sure that we could run the filters independently too. The solution that we found was to keep track of listings in a set and keep filtering it so that in the end we are left with a filtered set. We were able to get the filtered set and display the listings accordingly.
- We also made changes to our ER diagram and relational schema while working throughout the project, which caused us to go back to the function and re-do them.

Assumptions

- We assume that the input given by the user is valid. For example, if we say enter 1 or 2, the user will choose either one. We did not handle the edge cases considering that the project was focused on SQL.
- When the renter deletes their account, the bookings they have completed still stay but all bookings in the future associated with that renter will automatically be canceled.
- All information in review related to the renter gets deleted when the renter is deleted (including reviews they wrote and reviews they received)
- The same person with the same sin cannot be the host and renter at the same time.
- The only acceptable payment method is the credit card.
- For filtering prices, it makes sure the listing price for all the available dates is within the budget.

Operations Supported

The following are the operations supported:

> As a **user**, I can sign up or log in as a host or renter.

> As a **renter**, I can view listings.
> As a **renter**, I can book listings.
> As a **renter** I can cancel my booking(s).
> As a **renter**, I can filter listings according to our needs.
> As a **renter**, I can review a listing or a host.
> As a **renter**, I can delete my account.

> As a **host**, I can put up a listing.
> As a **host**, I can have an option to see suggestions to boost my listings.
> As a **host**, I can view my listings.
> As a **host**, I can update the listings which aren’t booked.
> As a **host**, I can cancel a booking.
> As a **host**, I can remove a listing.
> As a **host**, I can review a renter.
> As a **host**, I can delete my account.

Queries Supported

As a renter, I am able to apply the following filters independently or in combinations:

- Filter by Address
    - It returns the listings which have the same apt name, city, country, and postal code. If there is none, no results are found.
- Filter by Date Range
    - It returns the listings which are available for the specified date range (i.e. start date and end date). If there is none, no results are found.
- Filter by Price
    - For this, we make sure that the renter had specified what dates they are looking for (if the date range filter is not applied) because the prices depend on the dates. Even if we were to look at a consecutive set of dates let's say, Jan 1 to Jan 10, it could be possible that from Jan 1 - Jan 5, the host puts up the listing for $70, and for the rest, $40.
    - If the user first enters a set of dates, and there are no dates available in that range, they are given an option to look for a different set of dates or turn back to the menu. (only when the date range filter is not used)
    - The algorithm that we used to filter the listings was:
        - Checking availability.
        - Getting the max price for each listing that has the dates available to book.
        - Comparing the max price with the budget of the renter.
        - At last, sort the results with the prices in ascending order.

- Filter by Location (longitude and latitude)
    - It returns the listings in the nearby area.

    - We have a helper function that takes 2 coordinates and finds the distance between them.
    - As a renter, we have an option to go with the default distance which is 10 km, or set a new distance.
    - When viewing, we could view it with the distance in ascending order.
- Filter by postal code
    - It returns the listings in the nearby postal code.
    - We used a helper function that takes 3 postal codes and finds the distance between them. For simplicity, we are just finding the hashcodes for the two and subtracting them.
    - As a renter, we have an option to go with the default distance which is 10 km, or set a new distance.
    - When viewing, we could view it with the distance in ascending order.
- Filter by Amenities
    - As a user, we can specify the must-have amenities.
    - It filters through all the listings to find the listings that have the must-have amenities specified by the user.

Note: Whenever filter by prices, location, and postal code are applied together or in pairs, we give users the option to view the listing in either ascending order by prices, location, or postal code. We can not view listings by using the sort function on two or more.

Reports Supported

As a user, I am able to run the following reports periodically to understand the data:

- Report the total number of bookings in a specific date range by city.
- Report the total number of bookings in a specific date range by postal code within a city.
- Report the total number of listings per country, per country and city, per country, city, and postal code.
- Report ranks the hosts by the total number of listings they have overall per country.
- Report ranks the hosts by the total number of listings they have overall per city.
- Report for every city and country a report should provide the hosts that have a number of listings that is more than 10% of the number of listings in that city and country.
- Report ranks the renters by the number of bookings in a specific time period.
- Report ranks the renters by the number of bookings in a specific time period per city. (renters that have made at least 2 bookings in the year).
- Report hosts and renters with the largest number of cancellations within a year.
- Report A report that presents for each listing the set of most popular noun phrases associated with the listing (noun phrases from review).

Host Toolkit

We have a very simple function for the host toolkit.

When the host is posting up a new listing, they have an option to get suggested prices and/or amenities for their new listings.

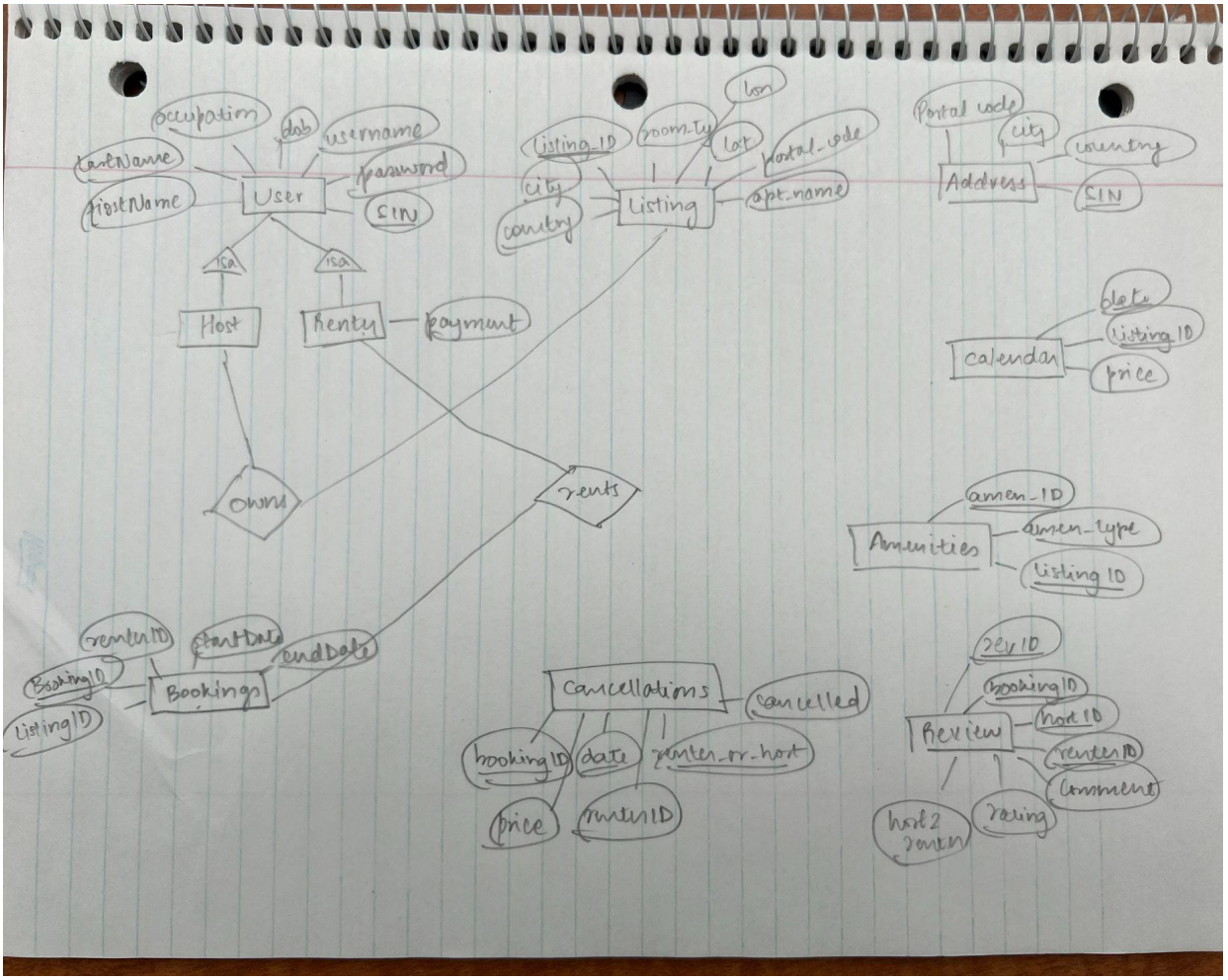
The way we implemented this was, that we first ask the host, if they would want to use this functionality and if they opt, we get the data from the host and compare it to the listings nearby.

The algorithm that we followed was to get the city and look at the listings in that city, find the market price (avg price for the listing), and suggest that to the host. Similarly, we also look at the amenities that were being offered nearby and displayed both the available amenities and the suggested amenities (which were the ones that were not included in the neighborhood, to make the listing stand out).

We were short on time, otherwise, there was another way we thought of implementing it. It was to get the type of listings and see those types around, by refining the results to the postal code. We could also use the coordination to know more about the location for eg, if it’s a warm place, we would suggest the host to add “AC” as one of the amenities and so on and so forth.

ER Diagram

The following is our ER diagram with the relational schema:



Relational Schema:

- User(SIN, occupation, lastname, firstname, username, password)
- Renter (SIN, payment)
- Host(SIN)
- Listing(lisintg_id, room_type, lon, lat, city, country, postal_code, apt_name)
- Address(Postal_code, city, country, SIN)
- Owns(listing_id, host_id)
- Rents(booking_id, renter_id)
- Calendar(date, lisitng_id, price)
- Amenities(amen_id, amen_type, listing_id)
- Review(rev_id, booking_id, host_id, renter_id, comment, rating, host_to_renter)
- Cancellations(booking_id, date, renter_or_host, renter_id, price, cancelled)

User Manual

How to use it?

- The application is pretty self-explanatory.
- Initially, a user is given an option to choose the role (i.e a renter or a host)
- Then after choosing, they are asked if they want to sign up or log in.
- If they chose to sign-up, they are asked for their information.
- If they chose to log in, they are required to enter a valid username and password.

After a user is in their account as a **renter**:
They have multiple filters, to filter through the listings, view them, book a listing by providing the listing_id displayed when they viewed the listing, cancel a listing by providing lisitng_id, rate a host or listings by providing a valid listing_id, and delete their account.

After a user is in their account as a **host**:
They have the option to create a listing, view their listings, update their listings by providing the valid lisitng_id, cancel a booking by providing the booking_id, remove a listing by providing the valid lisitng_id, review the renter by providing the valid renter_id, and delete their account.

    - Description of system limitations
    - Possibility for improvement

Description of system limitations:

- Because of time limitations, we weren't able to get to the last 2 reports.
- Works on a small scale database because of the filtering.

Possibility for improvement:

- We would want to figure out some other way to filter out the listings to make them more effective.
- We also planned to print out the distance with the listings when sorting by location and postal code.