CSCC43: Introduction to Databases

MyBnB

Final Project

By: Howard Yang, 1006722478
Tarushi Thapliyal, 1006862851

Purpose

To create an application that can be used to book short-term rentals, complete with a command-line based user interface and a simplistic backend connected to a MySQL database. This platform aims to replicate some of the features present in the well-known home sharing service known as AirBnB by serving as a marketplace for rentals that connects hosts and renters. Users need only register once to be able to access rentals listings and post their own listings. In addition, several features have been implemented to improve the user experience for renters and hosts. Renters are able to easily search, filter, and sort through listings that are available in order to book a suitable listing. Bookings can be viewed and cancelled up until and including the start date of the booking. If a stay has been completed within the past 180 days, then clients can choose to leave a review for the listing as well as the host. Hosts are able to add and remove listings, modify the details of a listing, such as the availability and the amenities provided. Hosts can view and cancel bookings as well as leave a review for the client if a stay has been completed within the past 180 days. A host toolkit has been provided to aid hosts in pricing new listings. Additionally, an administrator is able to view reports to see important statistics and information extracted from the data stored within the application regarding bookings and listings.

Conceptual Problems and Solutions

Types of Users

Determining whether to allow any user to access both host and renter functionality or to make it so that users have to sign up exclusively as renters or hosts was a decision that we had to make early on. We favoured the idea of not differentiating between users and allowing any user to be able to access both host and renter functionality as it would prevent users from having to sign up again under a different role in the cases where a host wishes to rent places and a renter wishes to host places. This required making the application's flow very simplistic in order to not confuse new users. We achieved this by having separate dashboards for renter and host operations and the ability to switch between these dashboards seamlessly. Having this flexibility forced us to add the additional constraint that the listing owner of a listing that a renter wishes to book cannot be the same as the renter attempting to make the booking.

Deleting an Account or a Listing

Renters can only successfully delete their account if they do not have any active bookings in the future. Hosts can only successfully delete their account if they do not have any listings that have active bookings in the future. Listings can only be removed if they do not have any active bookings in the future. When deleting accounts, we had to decide whether we would like to store the user data even after the account has been deleted or if we would like to delete all of the user's data. Although this comes at the cost of making reports on the data inaccurate, we decided to go with the later approach as it seemed more reasonable to us to dutifully remove all user data at the user's request. Moreover, it is not so common for users to delete their accounts, so we believe that deleting the user data entirely would not significantly impact the accuracy of reports.

Storing and Displaying Availabilities for Listings

Each listing must have a calendar of the dates when it is available for booking by renters. Additionally, the price of a listing can vary from day to day. Since there are many listings, several possible dates, and equally several possible prices, we decided to create a relation that stores for each listing, the dates that it is available and its price on each day that it is available. We added the constraint that the dates can only be added up to 2 years into the future, but this limit can be modified based on necessity. Moreover, once a range of dates has been booked, those dates cannot be booked again unless the original booking is cancelled. To make it easier for hosts to add availability and for renters to book dates, we used the Python *tkcalendar* library to implement a GUI component in the form of a calendar. This simplifies the process of adding and booking valid dates as well as modifying the prices for available dates. We then also decided to add another field to the relation for listing availability to determine if an available date for a listing or not has already been booked. The purpose of this field primarily is to make the search results for available listings more accurate by only considering the prices for those listings that have available dates in the future that have not yet been booked.

Storing Address and Location of Listing

Since we were tasked with implementing the feature of searching listings by their exact address, we decided to store the address of a listing using multiple fields in the listings relation, instead of just a single field that stores free-form text, in order to make address validation easier and have it conform to the format that is required for an exact search by address. The components of listing addresses that we decided to store are the street numbers, street names, cities, countries, and zip codes. User addresses, on the other hand, which are acquired during sign up, have no such restrictions and are stored in a single field as free-form text. For the feature that searches by the latitude and longitude, we would ideally like to use the address of the listing to determine the latitude and longitude in the backend. However, for now, the latitude and longitude values for a listing are supplied by the user upon the creation of a new listing. For the feature that searches for listings with similar zip codes, our heuristic for determining a similar zip code is a zip code that shares the same first 3 characters as the one that is supplied for the search.

Searching with Filter and Sort

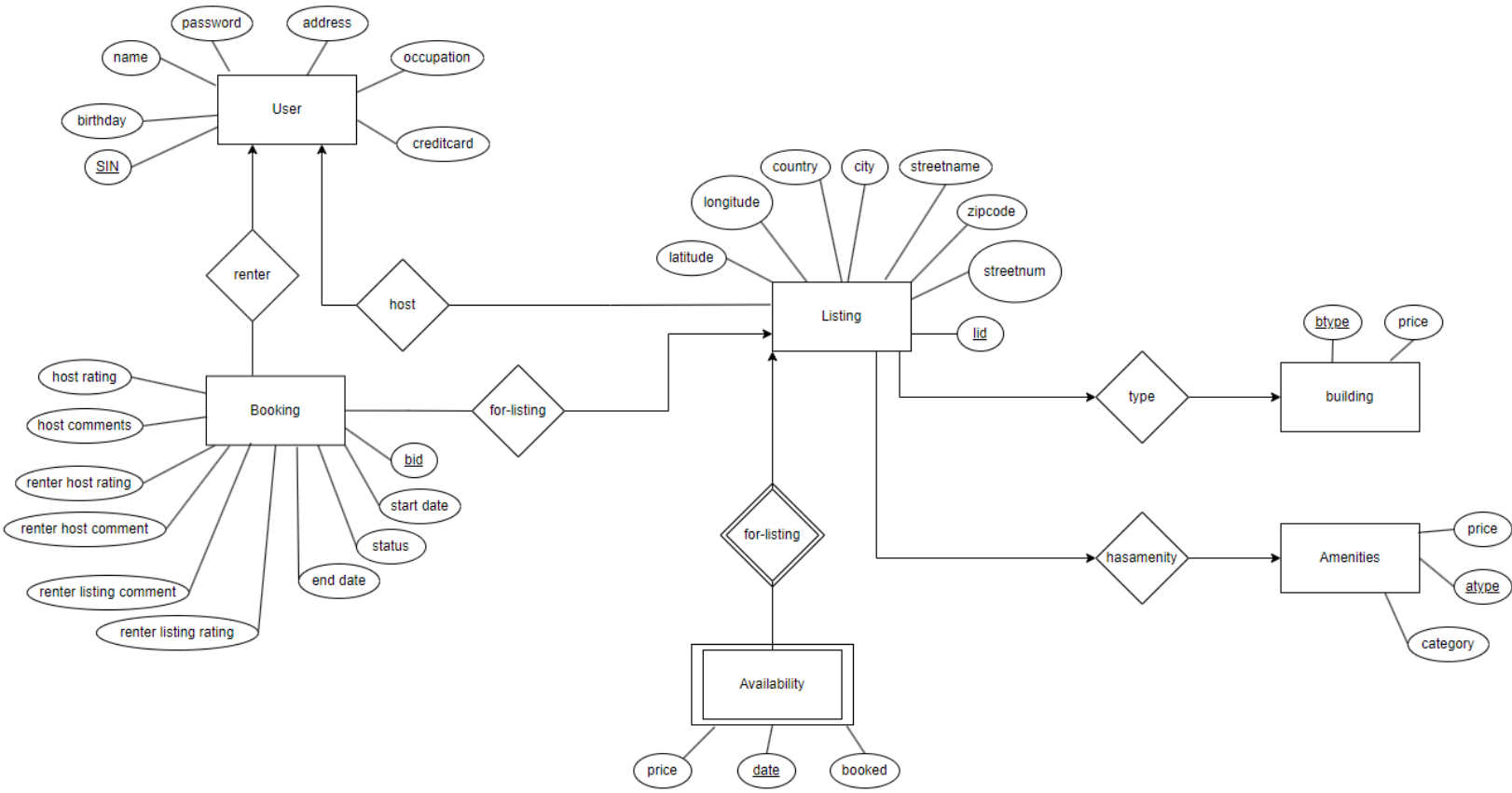
Since we were tasked with implementing tools to make the search for booking a suitable listing easier by adding the ability to filter and sort search results, we took the approach of first displaying all search results for possible searches such as searching for all available listings, searching by location, by similar zip codes, and by exact addresses, and then, allowing the user to filter and sort through the displayed listings. To add the filter functionality, each time a user decides to filter through displayed results, the application asks whether the user would like to apply each filter in the form of a series of binary questions. If the user agrees to have the filter applied, then it will be applied immediately after the relevant information for the filter is acquired from the user. If the user does not agree, then the filter is removed from the search results. In the backend, we apply the selected filters by modifying the search query incrementally as the user

answers each question. At the end of the series of questions, the search results are displayed again with all selected filters applied. In addition, the search results can always be sorted by price in any order as the user pleases. This is also done by modifying the search query to add or edit the sorting parameters and then displaying the results again in sorted order.

Assumptions

1. A host can have many listings, and a listing belongs to one host only.
2. A booking for a listing belongs to one renter and covers a range of consecutive dates.
3. There can be more than one listing at an address.
4. Renters can only have 1 credit card, and the only payment information required is the credit card number.
5. A review by a renter or a host only consists of a comment and a rating.
6. A review for a stay can be made and overwritten as long as the stay has been completed within the past 180 days.
7. Listing addresses consist of a street number, street name, city, country, and zip code.
8. User account addresses are freeform text without any specific format, while listings must follow the above format.
9. Zip codes are exactly 6 characters long and in the format of A#A#A#.
10. You can only leave a review on a past booking.
11. Reviews are tied to bookings, so a user is able to leave multiple bookings for a single listing if they have stayed more than once.
12. Different listings may have the same address.
13. The renter's review and rating for the listing is separate from the one for the host

ER Diagram



Relational Schema

User(sin, birthday, name, password, address, occupation)
Listing(lid, longitude, latitude, country, city, streetname, streetnum, zipcode, btype, host)
Amenity(atype, category, price)
Building(btype, price)
HasAmenity(lid, atype)
Availability(date, for-listing, price, booked)
Booking(
    bid,
    start_date,
    end_date,
    status,
    for-listing,
    renter,
    renter_host_comment,
    renter_host_rating,
    renter_listing_comment,
    renter_listing_rating,
    host_comment,
    host_rating)

Booking(for-listing), Availability(for-listing), and HasAmenity(lid) are subsets of Listing(lid)
Booking(renter) and Listing(host) are subsets of User(sin)
HasAmenity(atype) is a subset of Amenity(atype)
Listing(btype) is a subset of Building(btype)

User Manual

MyBnB is a command-line based application that can be launched by compiling and running main.py. The user is then greeted by the home page with menu options to create an account, login, or exit the application. Upon signing up or logging in, the user is directed to the client dashboard page of the application with several more menu options to choose from. Menu options can be chosen by entering the number corresponding to each option, and the user is always prompted for these and any additional information required at the relevant instances. Where possible, user input is first validated before executing the desired option.

Reviews are free-form text entries and cannot have any newline characters. Zip codes cannot contain any spaces. A user's SIN, password, and credit card number, when inputted, should not contain any spaces as well. Prices, latitude, and longitude can be entered as decimal numbers. Price ranges for searches are inclusive.

Sign Up and Log In

Creating an account prompts the user for some personal information including their SIN, which is used as their login ID for future logins, and the password. Payment information

is not collected at this time. Successful account creation or log in directs the user to the dashboard page of the application, where the user has more options to choose from.

Search Listings

Users can search for available listings to book and also have the option to filter and sort through the search listings by average price. There are 3 specific ways to narrow down available listing results such as searching by location, searching for similar zip codes, and searching by exact address. The filters allow users to filter listings given a range of dates, range of prices, and a list of amenities.

Book Listings

Once users have found a suitable listing to book, they can select that listing from the displayed results by entering its listing ID. They will then be able to see more information about the amenities that the listing offers and will have the option to book the listing. Booking the listing requires a selection of a valid range of dates for the booking by using the GUI calendar component as well as the payment information if it is the user's first time making a booking.

Manage Bookings

Once a booking has been made, the renter and the host of the listing both have the option to cancel the booking if the stay has not yet taken place. For stays that have been completed within the past 180 days, renters have the option to leave a review for the host as well as the listing and overwrite it any number of times. Hosts also have the option to leave a review for the renter.

Delete Account

Users can choose to delete their account in the dashboard. However, deleting an account will only be successful if the user does not have any active bookings for the future or any listings with any active bookings for the future.

Create Listing

Users can create listings by providing the address, building type, and location specified by latitude and longitude. Users are also prompted to enter the amenities that the listing offers by entering the number corresponding to each amenity displayed. To submit the list, users may enter a value of 0. Users receive helpful suggestions for amenities that they should add as well as an estimate generated by the host toolkit in the application for the price per day of the listing.

Manage Listing

Users can view and manage their listings after the creation of a listing by adding or removing amenities, adding or removing date availability, and modifying the prices of the listing on any date for which it is available. The user can select the listing to manage by

entering its listing ID. For managing availability and modifying prices, we use the GUI calendar component once again to validate the dates selected by the user. The host toolkit price per day estimate is displayed wherever it may be of use to the user, including whenever prices are changed, dates are added, and amenities are added or removed. Users are able to see the future and past bookings for each listing in order to cancel a booking or leave a review respectively. A listing can be removed as well but only if it does not have any active bookings.

View Reports

- The administrator can view reports by signing into the application. The administrator login is special with a SIN of 1 and a password of 1. Some of the reports include ranking renters based on the number of bookings, ranking hosts based on the number of listings, ranking renters and hosts on the number cancellations, determining the commercial host, getting the amount of bookings and listings based on various parameters, and generating a word cloud.

Host Toolkit

During the creation of a new listing, users can choose to enter the amenities that their listing offers. The host toolkit aids with this by providing an estimate for the increase in a listing's price resulting from the addition of an amenity. Users choose amenities by entering the number corresponding to each amenity. The list of amenities can be saved by entering a value of 0. When the list is submitted, the amenities added are compared against the recommendations for the safety and essential amenities that a listing should have. The user receives suggestions to add any of these important amenities that have not already been added to the list along with the price per day estimate generated by the host toolkit based on the amenities already added to the list. If the user wishes to continue adding amenities based on these recommendations, then they have an option to do so as well. The final estimate is displayed after the user has finished adding amenities.

Every time an availability for a listing is added, the host toolkit price per day estimate is displayed to help the user enter the price per day for their newly added availabilities. Similarly, when the price of an available date is modified, an amenity is added, or an amenity is removed, the host toolkit estimate is displayed and updated if required as well. The host toolkit estimate is determined by considering the price for each amenity and the price for the building type of the listing as we believe that these two factors have a significant impact on the price of a listing. Different building types have different prices as some building types inherently require more maintenance by the host than the others. DIfferent categories of amenities, such as essential, safety, and standout amenities, have different price ranges as well. It is recommended that the host provides most of the safety and essential amenities at the very least. Each amenity in particular has its own price as well, since different amenities have different maintenance costs.

System Limitations & Suggestions for Improvement

- Renters cannot view their cancelled bookings. To reduce confusion when a booking is cancelled they should be able to view cancelled bookings.
- Hosts and renters cannot view a summary of the reviews on a listing, in the future an option to view all the reviews and average rating would improve user experience.
- We do not use actual geographic data to validate certain addresses and countries. In the future we could use an API to retrieve actual real world data.
- Latitude and longitude data on listing creation is manually inputted. Longitude and latitude the future could be determined by the address they inputted by querying an API.
- No way for a host to see how much money a listing has made or is making per month other than manually counting the bookings. In the future we could add more robust host management options.
- No way for a host to aggregate the reviews of a renter that has made a future booking. In the future, this would allow them to adequately decide if they should cancel the future booking.
- When we delete a user, we delete all of the user's data, possibly leading to inaccurate reports in the future. We could store as much data as is possible while keeping the user anonymous at the time the user is deleted to make the reports more accurate.
- No way to restore a listing that was deleted. This could be added in as an option.
- The host toolkit does not take into account the location of the listing. In order to better estimate prices, it could consider the average price of accommodation in an area as well as the average cost of maintaining an amenity.
- Users can only add amenities one by one at this time. An option to add all amenities of a category at once or to add all amenities recommended by the host toolkit could be added.

Github Repository containing the code, DDL statements, and sample data:
https://github.com/ectodrop/mybnb-c43