MyBnB User Manual

MyBnb is a bnb hosting and booking application that allows users to rent and book BnBs. It comes with a wide variety of additional features such as a host-toolkit to aid hosts in choosing the best features for their BnB, or a commenting feature to comment on users and hosts. This manual outlines these features as well as discusses some limitations and improvements for this application.

Menu

Once the user has started the application they are presented with six options to navigate the site. Users must login to use the Host feature, and some of the features in Book are also only restricted to logged in users. To begin Users may simply enter a number from 1-6 on the console to pick what they want to do.

Login

If the user has an account they may simply enter their userID and password to get access to their account. If the User does not have an account simply enter any incorrect userID and password to exit this page.

Register

The register page allows users to add themselves to the User database so that they can interact with the rest of the features to the fullest extent. The sign-up page receives input for the userID and password. Note that the userID must be **unique**, then it takes input for the rest of the information such as name, address, Date of Birth, Occupation and Social Insurance Number. The program enforces that the age must be **at least 18 years** to proceed.

Host

To begin the host page can only be accessed if the user is logged in. The Host page receives information from the user step by step to generate a listing. Once all the basic listing information is added such as the location and amenities the user chooses the availability and price. Furthermore the Host toolkit provides some useful insights to Hosts as they are creating the listing. It determines a **suitable price** for the Bnb by analyzing the successful bookings in surrounding areas. Then as the user progresses to the amenities it suggests some basic amenities as well as checks surrounding areas to suggest more amenities. Lastly, as the amenities are displayed the toolkit provides users with recommendations on amenities and pricing as well as expected revenue increase based on trends of previous bookings.

The following is the algorithm that is used to determine to recommend amenities which is done by computing the Euclidean distance on the longitude and latitude coordinates and then using a threshold of 20 determines the amenities that the nearby listings have to recommend some amenities.

```
public void recommendAmenities(double longitude, double latitude) {
    String distanceQuery = "SQRT(POW(longitude - ?, 2) + POW(latitude - ?, 2))";
    String query = "SELECT DISTINCT name FROM Listing NATURAL JOIN Book NATURAL
JOIN Amenities " +
            "WHERE " + distanceQuery + " < 20;";

    /* OMITTED QUERY EXECUTION */

    try (ResultSet rs = pstmt.executeQuery()) {
        while (rs.next()) {
            String name = rs.getString("name");
            System.out.println(name);
        }
    }
}
```

Tho following is the algorithm used to recommend a price to the host as they are creating their listing. Similar to the amenities the pricing first takes the euclidean distance and ensure its in a threshold of 20. Once that is confirmed then the average of all the nearby prices are taken to get a recommendation of the pricing.

```
public void recommendPricing(double longitude, double latitude) {
    String distanceQuery = "SQRT(POW(longitude - ?, 2) + POW(latitude - ?, 2))";
    String query = "SELECT AVG(price) FROM Listing NATURAL JOIN Book " +
            "WHERE " + distanceQuery + " < 20;";

    try (PreparedStatement pstmt = connection.prepareStatement(query)) {
        pstmt.setDouble(1, longitude);
        pstmt.setDouble(2, latitude);

        try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
                double price = rs.getDouble("AVG(price)");
                price = price == 0 ? 180 : price;
                System.out.print("$" + price);
            } else {
                System.out.print("$1000");
            }
        }
        System.out.println("\n");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

The following is the algorithm that is used to determine the expected increase in revenue after adding an amenity, which simply receives the average price of the listings that have the amenity and the listings that do not have the amenity and returns the difference. If the differences happens to be negative then 0 is simply returned.

```
public double getEstimatedRevenueIncrease(String amenity) {
    String haveAmenity = "SELECT AVG(price) as average FROM " +
            "Amenities NATURAL JOIN Book NATURAL JOIN Availability " +
            "WHERE Amenities.name = ?";

    String notHaveAmenity = "SELECT AVG(price) as average FROM " +
            "Book NATURAL JOIN Availability " +
            "WHERE lID NOT IN (SELECT lID FROM Amenities WHERE name = ?)";

    double haveAvg = 0;
    double notHaveAvg = 0;

    try {
        /* QUERY EXECUTION CODE OMITTED */

        if (rs.next()) {
            haveAvg = rs.getDouble("average");
        }

        /* QUERY EXECUTION CODE OMITTED */

        if (rs.next()) {
            notHaveAvg = rs.getDouble("average");
        }

    } catch (SQLException e) {
        e.printStackTrace();
    }
    return Math.max(0, haveAvg - notHaveAvg);
}
```

Book

The book page lets users book and cancel bookings, however also provides a convenient search feature the User can search by the following

- Exact address (longitude, latitude, rest of address is functionally determined by these 2 points)
- Proximity to a longitude and latitude
- Proximity to a postal Code
- Date Range
- Temporal filters for country, city, price range, date range, etc. which can be stacked on top of each other

These searches produce an output of the listing information as shown below, displaying the lID at the start which will be used to book a listing.

```
Listing ID: 41
--------------------------------------------
Type: Apartment
Longitude: 43.001
Latitude: 43.001
Postal Code: 12345
City: Sample City 1
Country: Sample Country
Here are the availabilities for lID: 41
Date: 2023-08-10, Price: 150.0

Amenities included:  Bedding Parking Soap Toilet paper Towels Wi-Fi

Listing ID: 42
--------------------------------------------
Type: Full House
Longitude: 43.002
Latitude: 43.002
Postal Code: 12346
City: Sample City 2
Country: Sample Country
Here are the availabilities for lID: 42
Date: 2023-08-12, Price: 200.0

Amenities included:  Bedding Heating and air conditioning Kitchen facilities
Parking Soap Toilet paper Towels TV and entertainment Washer and dryer Wi-Fi

Listing ID: 43
--------------------------------------------
Type: Room
Longitude: 43.003
Latitude: 43.003
Postal Code: 12347
City: Sample City 3
Country: Sample Country
Here are the availabilities for lID: 43
Date: 2023-08-14, Price: 100.0

Amenities included:  Bedding Kitchen facilities Parking Soap Toilet paper Towels
```

Once the User has noted the lID they cant select option 6 to book and enter the lID along with the dates they want to book. This will prompt them to complete a payment with the card that is saved in their profile. If this card is not present then the user must enter their card information to complete the booking.

Furthermore, a user can cancel a booking which is done by simply entering the lID that you wish to cancel. The application restricts users to only cancelling their own bookings.

Reports

Finally a a variety of reports can be generated using the data that is collected from myBnB. The following are the reports that can be run.

- Query total bookings in a date range by city
- Query total bookings by zip code within a city
- Query total listings per country/city/postal_code
- Rank hosts by total listings overall per country/city
- Query hosts with more than 10% of the listings in a city and country
- Rank renters by the number of bookings in a date range
- Rank renters by bookings in a date range per city
- Hosts and renters with the largest number of cancellations
- Get popular nouns from listing

Once any of these options are selected the user will be prompted for information required to execute the report, for instance a start and end date may be required or a country/city may be required which is received as input in the console.

The first three reports give an idea of popular spots where BnBs are booked and hosted which gives insight into where the company should advertise their application to improve website traffic. It helps analyze features that may make some location more booked than others which can be adapted into the host-toolkit for a better experience.

Then there are reports that rank the user and the hosts. The query to get hosts with more than 10% of listings in an area can help flag commercial hosts so that the overall experience can be made better. Furthermore, the hosts and renter with the most cancellations can also be flagged so that hosts and renters are prepared with they book a listing or when their listing is booked.

The last query gives popular nouns which can be used to determine some features about a listing. Perhaps the user gave an extra amenity such a swimming-pool that they forgot to list which can then be determined by this report.