

1. Motivation for the idea/description of the problem the application solves

For our project we will create an application which evaluates the relationship between AirBnBs and the weather for the Boston area. Our target user for our web application is a person who lives in Boston and is considering becoming an AirBnB host. Our application will give a user the opportunity to query through Boston listings and assist them in how this user should price their rental according to how people price their listings in their surrounding area. Moreover, the user will be able to look up the ratings and other hosts and listing statistics so that the user can be well informed about how competitive the area is. We will display the popular month for a particular listing and also the weather on the day of the booking to help users make a better choice.

2. List of features you will definitely implement in the application

- Filtering listings by host
- Sorting listings by by views
- Sorting listings by number of bookings
- Bookmark/Unbookmark listings
- Display most common month for bookings
- Average temperature & humidity on the day of bookings

3. List of features you might implement in the application, given enough time

- Users have their own profile with name, profile photo, description along with their own listings.
- Users can add new listings.
- Users can delete listings they have previously added
- Display most common day of the week for bookings

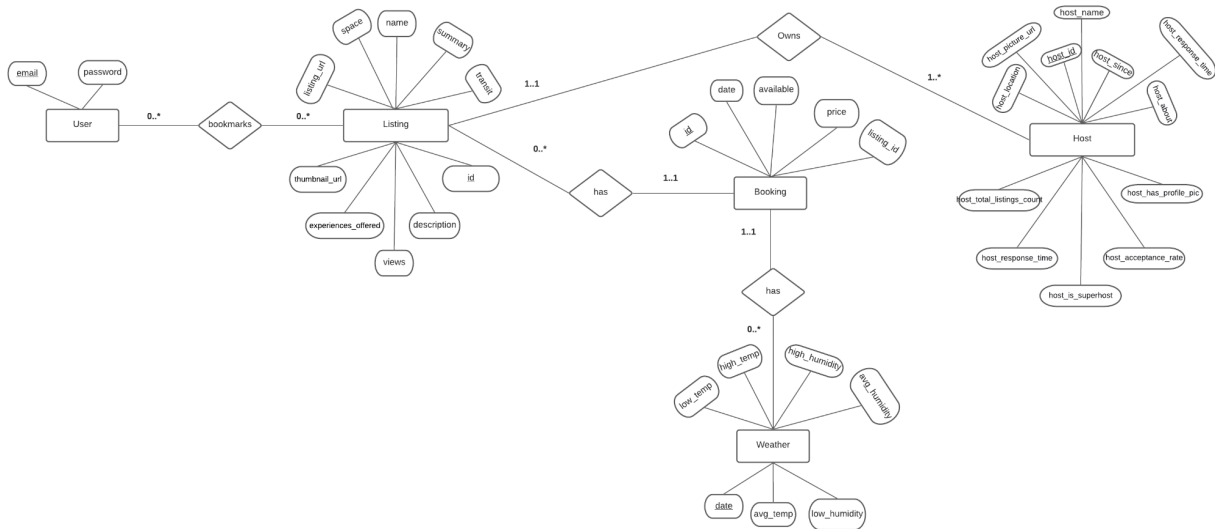
4. List of pages the application will have and a 1-2 sentence description of each page.

- **Sign up/ Login page.** User creates an account by specifying email address and password or user can use an existing account to log in.
- **View Listings Page.** Users can view all the listings from the database in a table. Each entry in the table includes: the listings name, a link to the listing, reviews per month and the host name.
- **Individual Listing Page.** User clicks on a listing in the view listings page, and it is directed to a page which contains a bunch of information about the listing including the host information, a picture of the listing, reviews, and the most popular time for bookings including weather data. User can also like/book mar
- **Bookmark.** A user can view the listings/hosts that they bookmarked. The listing/host will be displayed in the same way as the view listings page and the individual listings page.
- **Individual Hosts Page.** See all the information about a particular host including: Host_id Host_url Host_name Host_since Host_location Host_about Host_response_time

Host_response_rate Host_acceptance_rate Host_is_superhost Host_picture_url
 Host_total_listings_count Host_has_profile_pic User can view all the listings of the host.
 They will be redirected to the View Listings Page which will have listings filtered by the
 host.

- **Hosts List Page.** A user can view all the hosts in a table and click on one of them to see all their listings.

5. Relational schema as an ER diagram



6. SQL DDL for creating the database

```
CREATE TABLE User(
email varchar(255),
password varchar(255),
PRIMARY KEY (email)),
```

```
CREATE TABLE Bookmarks(
email varchar(255),
listing_id int,
PRIMARY KEY (email,listing_id)
FOREIGN KEY listing_id REFERENCES Listing(id),
FOREIGN KEY email REFERENCES User(email))
```

```
CREATE TABLE Listing(
id int ,
listing_url varchar(255),
last_scraped,
name varchar(255),
summary varchar(255),
```

```
space varchar(255),
description varchar(255),
experiences_offered,
transit varchar(255),
thumbnail_url varchar(255),
views int,
PRIMARY KEY (id))
```

```
CREATE TABLE Booking(
id int,
listing_id int,
date int,
available boolean,
price int
PRIMARY KEY (id),
FOREIGN KEY listing_id REFERENCES Listing(id),
)
```

```
CREATE TABLE Host (
host_id int,
host_name varchar(12),
host_since,
host_location varchar(12),
host_about varchar(255),
host_response_time int,
host_response_rate int,
host_acceptance_rate int,
host_is_superhost boolean,
host_picture_url varchar(255),
host_total_listings_count,
host_has_profile_pic,
PRIMARY KEY (host_id))
```

```
CREATE TABLE Owner (
host_id int,
id int,
PRIMARY KEY (host_id, id),
FOREIGN KEY host_id REFERENCES Host(host_id),
FOREIGN KEY id REFERENCES Listing(id),
)
```

```
CREATE TABLE Weather (
date date,
high_temp int,
```

```
avg_temp int,  
low_temp int,  
high_humidity int,  
avg_Humidity int,  
low_Humidity int  
PRIMARY KEY (date)  
)
```

7. Explanation of how you will clean and pre-process the data.

- **Clean missing/ misentered values:**
Depending on the dataset, either drop all the rows with missing values or replace it manually/ with defaults.
- **Detect and solve entity resolution problems:**
Entity resolution is applied to all the datasets having common entities by editing the names (if different) in a dataset to match the other. Unpaired entities is also removed at this point.
- **Replace categorical variables with numeric indicators:**
Convert the columns to numeric indicators (if required) for efficiency. This is done by mapping each value name to an integer and finding a unique index (single column index or multi column index) for each table.
- **Export the data:**
This is the final step where the data is ready to export.

8. List of technologies you will use. You must use some kind of SQL database. We recommend using MySQL or Oracle specifically because you will use MySQL in HW2, and we will provide guidance for setting up a MySQL database. Some groups in the past have had issues with MySQL, but Oracle is another option.

Backend: MySQL, Node.js, Express

Frontend: React.js, Javascript, HTML/CSS

9. Description of what each group member will be responsible for

Each group member will be responsible for implementing a particular page end-to-end.

- **Sign up/ Login page:** Gabriela
- **View Listings Page:** Gabriela, Yiyasu
- **Individual Listing Page:** Yasmine, Purva
- **Bookmark Page:** Yiyasu
- **Individual Hosts Page:** Yasmine
- **Hosts List Page:** Purva