

# **Airbnb in Boston: The Connection between Host and Consumer**

**Fengyi Quan, Anni Yuan, Wanqi Peng**

April 2020

# Table of Content

1. [Introduction](#)
2. [Abstract](#)
3. [Data Domain and Capability](#)
4. [Textual Description](#)
5. [SQL vs. NO SQL Storage](#)
6. [Languages, Libraries and Framework](#)
7. [UML Model and EER Model](#)
8. [User Flow](#)
9. [Technical Specifications](#)
10. [Group Role](#)
11. [Lesson Learned](#)
  - [11.1 Technical expertise](#)
  - [11.2 Group Insights](#)
  - [11.3 Alternative Design/Approaches](#)
12. [Future Work](#)
  - [12.1 Planned Uses of Database](#)
  - [12.2 Potential Areas for Added Functionality](#)
13. [README](#)

# 1. Introduction

Airbnb, short for AirBed and Breakfast, has developed rapidly in recent years. It is a website providing new short-term housing which different from traditional hotel. This new type of housing provides people with more choices and experiences. Also, the price is a big advantage over hotels, so it is liked by a large number of travelers, especially young generations. We, as the users of Airbnb, want to discover more details of it in order to get a better user experience and introduce it to more people in need.

## 2. Abstract

Our data is mainly based on Airbnb public dataset (website called Inside Airbnb). Among these, we are only interested in property that located in Boston. Python is our host language. We provided a graph user interface (GUI) based project for both hosts and customers. Each user can choose to sign up as a host or a customer. Hosts are able to use our project to manage their current listing, account, delete listing and add new listing while customers are able to view all the listing information, make reservation, manage their account. Both hosts and customers can directly access to their own login page through home page and find their respective needs.

## 3. Data Domain and Capability

Inside Airbnb (source: <http://insideairbnb.com/get-the-data.html>)

This is the dataset we are using which hosts public available data from Airbnb. We focused data is Boston region. This dataset is mainly comprised of three main csv files (all files are compiled on 13 Feb 2020):

- i. Listings: detailed listing data showing 96 attributes for each of the listing
- ii. Reviews: detailed review data given by guests with 6 attributes. They are listing\_id, id, date, reviewer\_id, reviewer\_name and comments.
- iii. Calendar: details about booking for the next year by listing

In general, our project is a simple browsing and renting system, which is aimed at the host and guests at the same time. The host can list the houses they want to rent on the while guests can make reservations according to their own needs. The two user groups have different user pages with small overlap.

## 4. Textual Description

We are choosing to represent the data about Airbnb housing in Boston, including the information of rental space, host, customer and reservation.

To represent rental space, we collected the data of the name of each house, price for rental, number of bedrooms, number of bathrooms, description written by hosts, capacity of each property and average rating for this house which calculated by all stars given by reviews. We also kept track of the latest date of which host updated its information to let customers know how long it has been since the last update. In addition, we presented some pictures that demonstrates the surroundings and the house itself.

Also, we have five components which belong to certain housing: room type, property type, amenities, address and price. Among these, room type provides various types of room that customers can choose, such as entire home/apt, private room, shared room etc. Each room type can have zero or more corresponding rental spaces, while each rental space can only have one room type. Property type represents various house types, including apartment, guest suite, condominium, house, boat, townhouse, loft, bungalow, villa, bed & breakfast and others. Each property type can have zero or more corresponding rental spaces, while each rental space can only have one property type. Amenities indicate the services that supported by each house represented by Boolean values, such as wi-fi, parking, kitchen, heating and hot water etc. We have an amenity id to represent as identifier. Similarly, each amenity can have zero or more corresponding rental spaces, while each rental space can only have one amenity. Address contains the specific location of the house including zip-code, neighborhood, city, state and country where zip-code is the primary key. Each rental space is binding to only one address. Moreover, price involves price id, daily price, weekly price, monthly price, security deposit, cleaning fee, extra people where price id is the primary key of price.

Host is the owner of the house. Each host can host many rental spaces and has own identifier; however, each house must have only one host. We collected URL, name, since, total listing and is verified of each house owner. What's more, we cared pictures and whether a host is super host (friendly and nice) to offer a more customer-friendly service.

At the customer perspective, customers should have unique identifier as primary key as well as phone, email, date of birth, pictures to prove their identities. Customers can browse multiple houses that they desired. Additionally, customers make reservations to book rental spaces.

For each reservation, there is information about start date, end date, number of guests and booking date. Each reservation has its own id number and total price calculated by corresponding data.

Lastly, customers can write many reviews to give feedbacks and rate properties to these housing that they have lived in. Each review contains its own identifier. A review can only comment on one property; whereas each rental space can take advice from zero or more reviews.

## **5. SQL vs. NO SQL Storage**

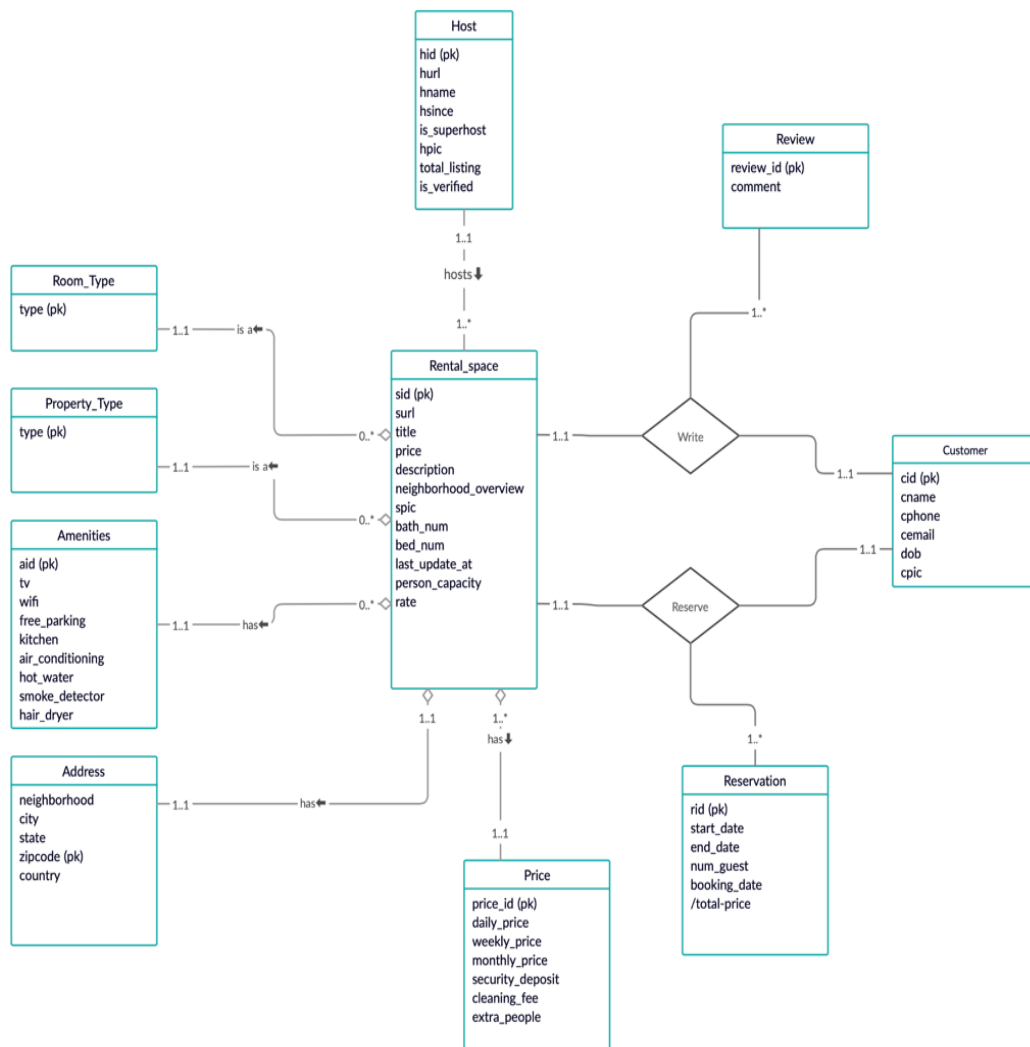
We are using SQL storage in this project.

## **6. Languages, Libraries and Framework**

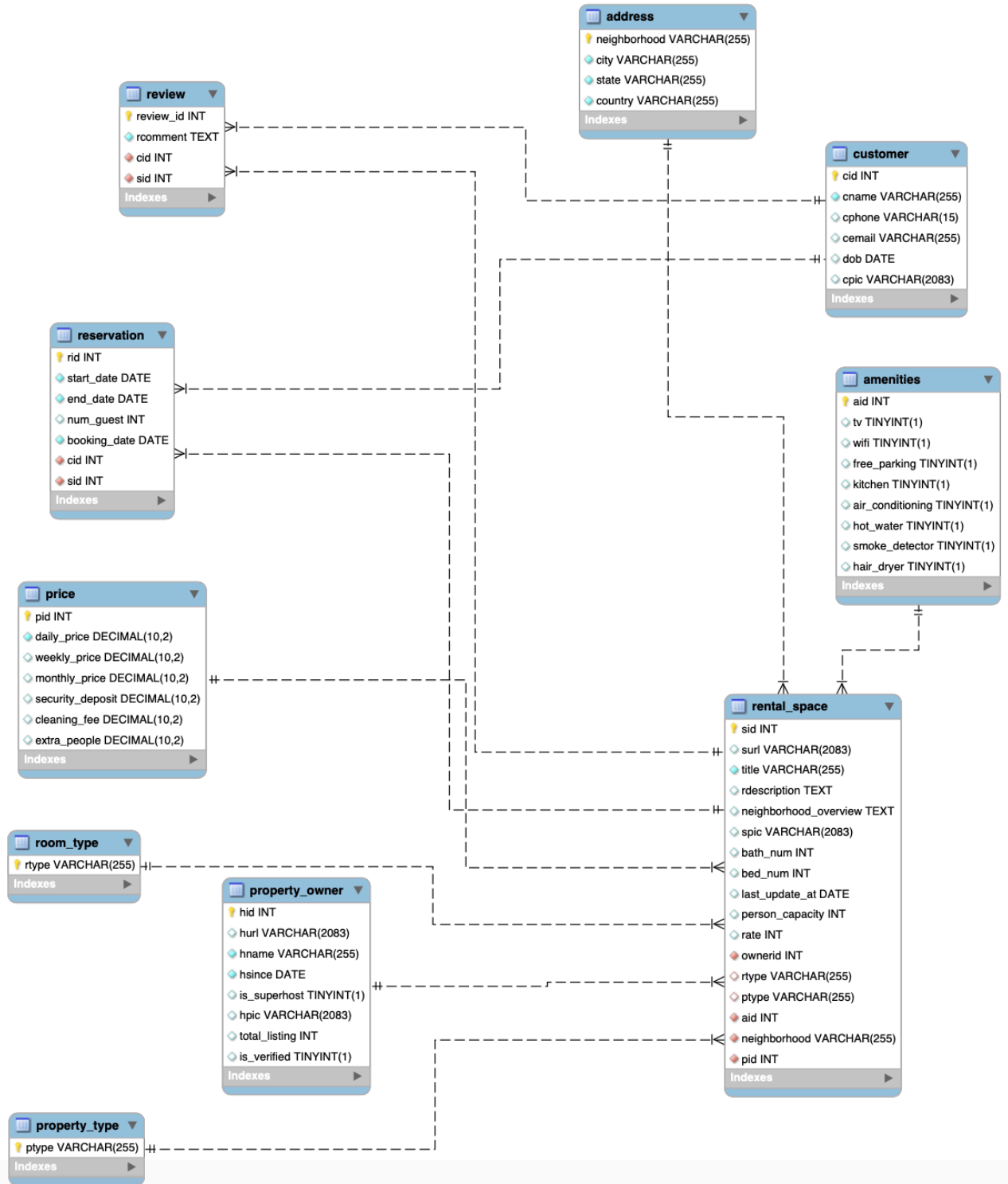
We choose to use python as host language and SQL as database storage. We import MySQL. Connector library as a bridge to connect python to SQL server, and some other common libraries, such as NumPy, matplotlib for computing and visualizations. We are using tkinter as frame to build our user interface.

## 7. UML Model and EER Model

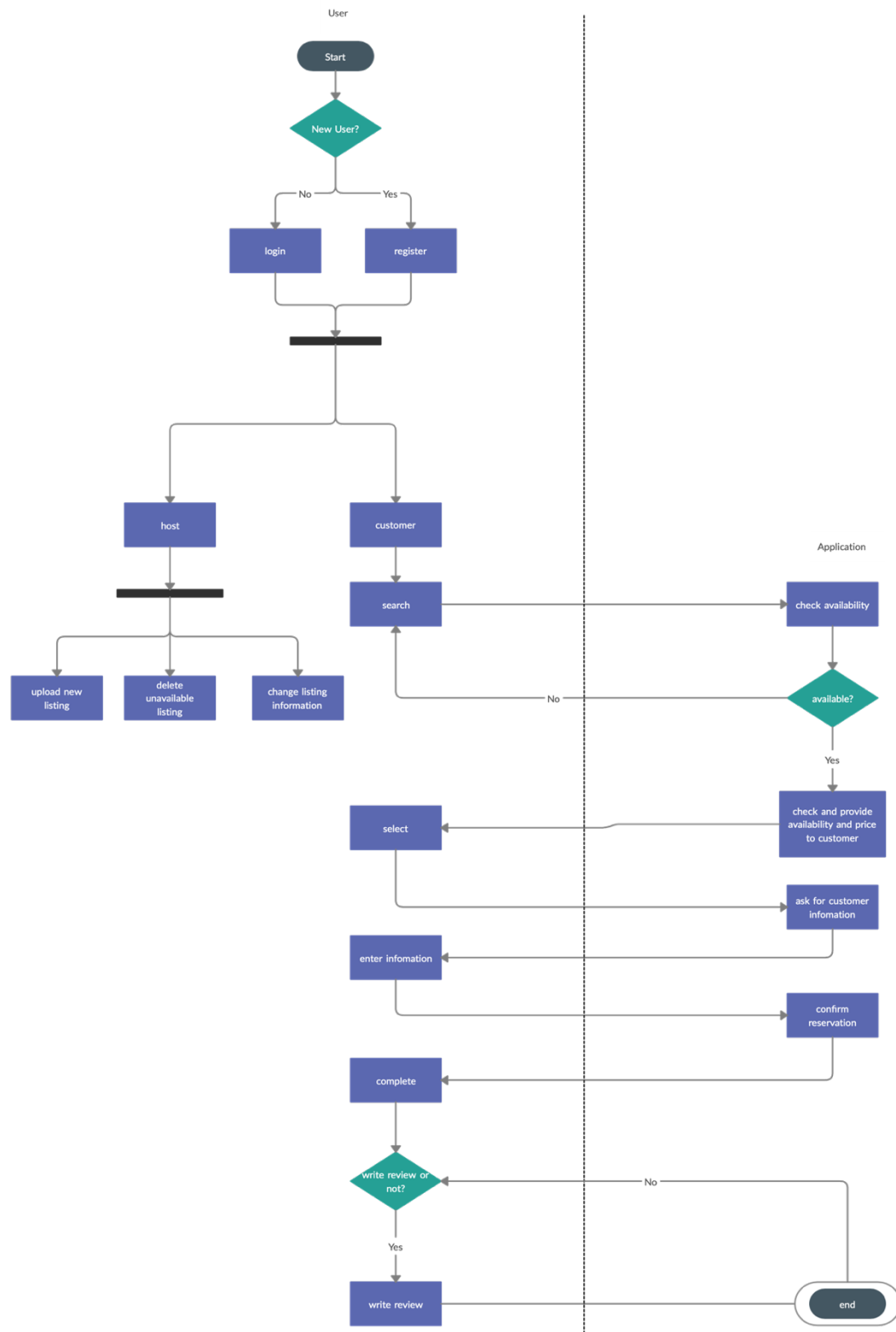
### UML Model



## EER Model



## 8. User Flow





## 9. Technical Specifications

As mentioned above, this is a simple renting system just like Airbnb. User groups are hosts and guests, so the main function is mainly divided into hosts and guests. There are four parts on the initial entry page: host login, customer login, About and Exit. The two major user groups, hosts and guests, have different entry pages. For the each enter system, we demonstrated two text boxes which lead users to enter their username and passwords (no passwords needed right now). They could also choose to sign up for an account if they don't have one. After logging in, they two user interfaces would have a huge difference since they demand a difference usage. Start from host page, there are five buttons below where users can easily click the certain button on the page that we designed to increase and delete the listings. For instance, 'Add' and 'Delete' allows hosts to modify their properties to be rented. While for the customer system, a bunch of housing information appear after we are logging into the account. They can select properties that they are interested in and look up the more detail information by hitting 'more info' which jumping to another page. There are also three buttons 'Availability check', 'Reserve' and 'Back' below info page which guide customers to check availability and make reservations. If the guest selects hit reserve, another page will pop up and ask the customer for specific check-in information, such as date and number of people. After filling out all the information and click 'reserve', the whole booking process is complete. All codes are working properly. Some bonus was done as a group of three:

- i. Interesting Visualizations  
Our graphic user interface is user-friendly. Although the GUI is simple, it is very clear and logical, conforming to the user's operating habits.
- ii. Deal with some complicated translations from user to database operation  
At the beginning of importing data, we need to process all the data, such as the data format and escape symbol. When implementing, we need to pack information entered by customers, and then added to the database in sequence.
- iii. Deal with complicated schema
- iv. Difficult process for data extraction for the database

## **10. Group Role**

There are three members in our group: Fengyi Quan, Anni Yuan and Wanqi Peng. We roughly divided our work according to these four deliverables: project presentation, client code, database design and this final report. We discussed database together and came up with the final edition of the design. Fengyi is responsible for 40% of the work in this project. He covered almost every part in front end and built up the database. Anni and Wanqi each account for 30% of the remaining work. Anni implemented code in user graphic interface and database as well as the project presentation. Wanqi documented the project progress report and final project report.

## **11. Lesson Learned**

### **11.1 Technical expertise**

In general, we have a deeper understanding of the field of data science. First, after choosing the topic of a project, we take the good use UML and EER diagrams to organize the fragmented ideas more intuitively. Secondly, by applying various SQL statements, we can sort out a huge data in an order. We have a deeper experience in the cooperation between SQL and host languages. In this project, we used the Python. Because it has one of the most mature libraries and can achieve the function we want within the least amount of code. Especially when we paired it with SQL, making us realize the complementary effect of various programming languages.

### **11.2 Group Insights**

This assignment is completed in the form of group cooperation, so we have gained the ability of cooperation and communication while improving the technical expertise. Our group is a group of three people, so everyone has got their own tasks from the beginning. This division clearly push everyone to be responsible while saving a lot of ineffective communication.

Teamwork is a very important skill in the future work and life, this project frames a preliminary understanding for us in this regard.

### **11.3 Alternative Design/Approaches**

Before finalizing the idea, we thought about making a project to value the house based on the information and pictures of houses. But soon we found that finding photos of corresponding house was very difficult. In addition, we also thought about making a website. Eventually, we found that the focus of this project was on data processing and analysis rather than web design. Therefore, we gave up the above ideas and made a simple rental system that oriented to both hosts and guests.

## **12. Future Work**

### **12.1 Planned Uses of Database**

We consider not only focusing on the Boston area in the future, but also facing the entire United States. So, we need the database of housing, hosts and guests for the whole country.

### **12.2 Potential Areas for Added Functionality**

Due to the time constrains and other certain reasons we are not able to make this project the most perfect, here are some parts that we think should be changed and improved:

i. Password

This program does not support password functionality for all users. We want to improve this functionality to make this login in system more reasonable.

ii. Picture View

Customer cannot see the actual picture through this program right now. Adding picture would make it more user-friendly and realistic.

iii. Customize Searching

This program does not support any customized searching based on preferences which would cause inconvenience.

iv. Scoring System

Some attributes need to be evaluated, like `is_superhost`, `is_verified`, etc. For now, users can change these values which is not a reasonable design.

#### v. Prevent SQL Injection

We should escape the user input first (not implement yet) and prevent from other security threat.

## 13. README

### # Airbnb Simulator

This project simulates the basic operation of how the Airbnb works. It allows different customers (host and visitor) to search, modify and add new information to the database via our UI. We want to perform all the rental information around Boston area to give users a better access to make reservation and posting house.

### ## Description

This program provides the basic functions for both hosts and visitors. Both hosts and visitors start within the same entry point. Two view is very different with small overlap. Host cannot make any reservation and visitors cannot make any changes on the listings. It requires customer to choose the identity to login in and provide their id number. For hosts, it can add new listing, modify own listings. For visitors, they can browse all listings to choose their desired house and time to make reservation. For the sake of both hosts and visitors, all the required field starts from a star (\*) symbol.

### ## To Start

- 1) To see the host perspective, click `Host Login`, then enter `8229` for host account and leave password empty to `Login` for demo. You can also create your own hid or account (must be unique).
- 2) You will see 6 different listing in this page. These are all the host `8229` owned. You can click `Add` to add new listing. It will ask you provide some information in order to create your new listing. You can choose any listing to delete or modify as well. You can click `My Profile` to see the detail of the personal information and make some changes by clicking `Update` if you wish. For information details, please see \*Rental Space (Host View) \*.
- 3) To see the customer(visitor) perspective, back to main page and click `Customer Login`, then enter `1` as cid and leave password empty to `Login`. You can also create your own sid by clicking `Sign Up`.
- 4) Once your login, you can see all the posted listing. You can choose any one you want to see more detailed information by choose a listing and click `More Info`. It will lead you to a new window that show all the information about your choice.

- 5) Then you can click `Availability Check` to check the availability of the house;  
`Reserve` to make actual reservation  
to this house, this would you to provide some basic information.
- 6) When you back to customer page, you can also click `My Profile` to view your  
personal information and make some  
changes by clicking `Update`; `My Reservation` to see reservation you make and  
modify it.
- 7) For data visualization, back to main page (the start page). Click `Data Visualization`  
and choose what information  
you want to see. It will pop up a web page to show the plot. There are four different type  
of plots. For more  
information about the plots, see section `Data Visualization`.
- 8) About page is just for present the information of group members.
- 9) `Exit` button is to exit the program.

## ## Host Perspective

As a host, it needs to provide their hid (host id) in order to login and **\*\*no password required\*\***. Once he login, he can only see the listings he owned.

### ### Host Sign Up Page

Host need to provide basic information in order to make a new account. The information  
needs to provided lists below:

- \* hid: host id. It allows host to customize his own id. But it needs to be unique to  
distinguish other's uid. Each user has one unique uid. **\*\*It cannot be empty.\*\***
- \* hurl: host url. It can be empty, and program does not check if it is a valid url.
- \* hname: host name. It can be empty.
- \* hsince: time when create this account assigned by system. It cannot be modified.
- \* is\_superhost: a Boolean value represents whether it is a super host. It allows users to  
change these values. But **\*\*it cannot be empty\*\***. 0 represents it is not a super host,  
otherwise, 1 represents it is a super host.
- \* hpic: host picture. It can be empty, and program does not check if it is a valid url.
- \* is\_verified: a Boolean value represents whether the host has verified this account. It  
allows users to change these values. But **\*\*it cannot be empty\*\***. 0 represents this  
account has not been verified, otherwise, 1 represents it has been verified.

### ### Rental Space (Host View)

He can see all his listings as well as make changes to these listings. Also, he can click  
`My Profile` to see his personal information and click `Update` to modify his information.  
For listings, he can choose one listing and click  
`Modify` to change the information for the particular house or click `Delete` to delete this  
listing. In addition, he can make a new listing by clicking `Add` Button. For the fields that  
shows in modifying and adding page. **\*sid, city, state, country\*** are not allowed to be  
modified. Reason states below:

- \* sid: unique identifier of a house (rental\_space).

\* city, state, country: we only allowed to post listings around Boston area, so, these fields are not allowed to be changed.

Fields that allowed to be **\*\*empty\*\*** string:

- \* url: url of this rental space. Program does not check if it is a valid url.
- \* title: the title of this house that host want to show first.
- \* rdescription: description of this house.
- \* neighborhood\_overview
- \* spic: picture of this house (url). Program does not check if it is a valid url.
- \* neighborhood

Fields that **\*\*cannot be empty\*\***:

- \* sid: id
- \* bath\_num: number of bathrooms that this house has. It can only be integer.
- \* bed\_num: number of bedroom that this house has. It can only be integer.
- \* rate: rate of this room ranges from 0 to 100.
- \* rtype: room type (can only be 'Entire home/apt', 'Hotel room', 'Private room', 'Shared room')
- \* ptype: property type (can only be 'Apartment', 'Barn', 'Bed and breakfast', 'Boat', 'Boutique hotel', 'Bungalow', 'Castle', 'Condominium', 'Cottage', 'Guest suite', 'Guesthouse', 'Hotel', 'House', 'Houseboat', 'Loft', 'Other', 'Serviced apartment', 'Townhouse', 'Villa')
- \* city, state, country: as reason states above.
- \* kinds of price
- \* amenities: boolean value to represents if this house provides this service. 0 is no and 1 is yes.

## ## Customer Perspective

Customer need to provide his own cid (customer id) in order to login and **\*\*no password required\*\***. Once he login, he can only all the listings that stored in the database.

## ### Customer Sign Up Page

Host need to provide basic information in order to make a new account. The information needs to provided lists below:

- \* cid: host id. It allows host to customize his own id. But it needs to be unique to distinguish other's uid. Each user has one unique uid. **\*\*It cannot be empty.\*\***
- \* canme: name of the customer. It can be empty.
- \* cphone: phone number. It can be empty.
- \* cemail: email address. It can be empty.
- \* dob: date of birth. **\*\*It cannot be empty\*\*** and must strictly followed the format **\*\*'YYYY-MM-DD'\*\***.
- \* cpic: host picture. It can be empty, and program does not check if it is a valid url.

### ### Rental Space (Customer/Visitor View)

He can see all the listing that stored in the database. The main page only shows the information about sid, title, spic, bath\_num, bed\_num, person\_capacity of a house. For more information, click `More Info`. It will show all the information about selected house. If no house has been selected, nothing happened. Users can also view their personal information by clicking `My Profile` and edit it. Clicking `My Reservation` can show all the reservation that this particular user made. Through `My Reservation`, user have ability to modify this own reservation. But booking\_date, cid, sid cannot be changed.

### #### Single House Information Page:

Instead of seeing all information about selected house including rental space, price, amenities situation and host information. All information is read only. Visitor can also check availability of this house by clicking `Availability Check` and make reservation by clicking `Reserve`.

### #### Availability Check

When visitor click `Availability Check` button, it will go to a new window showing the reservation history including start date and end data. He can see the other reservation time and choose a valid time to make reservation. If nothing shows in the window, it means there is no reservation made for this house. Thus, all time are available to visitor.

### #### Reserve

Once visitor choose a valid time, he can click `Reserve` to actually make reservation. It requires visitor to provide information about start date, end date, number of guests. All three fields are required and have to be valid. Program will check validity. It will compare all other reservation time to make sure no time period overlap and number of guests is in range of person capacity to make sure number of guests not exceed.

### #### Data Visualization

We have four different data representation to visualize data to provide a clear way to both hosts and customers. For details, please see our presentation about what the plot looks like. We provide four basic plot which can be presented by clicking different buttons. It will guide users to a web-demonstrated plot.

- \* Neighborhood distribution: shows the number of rental spaces in the different locations(neighborhood).

- \* Price Distribution: shows the price distribution by the different locations(neighborhood).

- \* Property Type Pie Char: shows the percentage of different property type distribution.

- \* 3D House Brief Overview: shows the distribution of each house by daily price, person\_capacity, neighborhood and property type.

#### #### Exit

The exit point of this program (same as clicking close at the upper left corner).

At other page (not main page), clicking close at the upper left corner will get back to one level of the window.

#### ## About Page

Provided the author and other information.

#### ## Data Source

- \* <http://insideairbnb.com/get-the-data.html>

- \* customer, reservation, review: make up by us because it is private information, we have no access to

#### ## Load Data

To load our data, we have some methods that modify and insert the data. To re-load our data, please comment out the

method ``read\_load\_data()`` to load the data. All the data are loaded into database through csv file and these files should be please in the same path under the program.

For demo, you do not have to re-load our data. It is automatically stored in the database that in the dump sql file.

#### ## Built With

- \* tkinter - Python Interface Library

- \* pandas - Data Analysis Library

- \* plotly - plotting Library

- \* pymysql - Python MySQL Client

#### ## Future Work (working on)

- \* Password - this program does not support password functionality for all users.

- \* Picture View - customer cannot see the actual picture through this program.

- \* Customize Searching - this program does not support any customized searching based on preferences.

- \* Scoring System - some attributes need to be evaluated, like is\_superhost, is\_verified, etc. For now, users can change these values.

- \* Prevent SQL Injection - should escape the user input first (not implement yet) and prevent from other security threat.