CIT-594 HW6 Group Project

Tianxiang Dong, Lijun Mao, Hanyu Yang

Introduction:

Function:

For this group project, our team is mainly developing a real-estate search application based on java. It is similar to some websites for selling or buying houses, but our program emphasizes on those in Philadelphia region and show the selling or buying results through java Gui with google map and some detail information inserted to it.

Feature:

Our program can be divided into two sections, Sell and Buy. When running the program, the main search page will pop up as a java GUI. It is a nicely designed user-interface with several images. Every time the user clicks a house type from commercial, residential and industrial, the corresponding image to the house type will show up in the background. Users can choose to buy or sell a house by clicking sub-page in the main search window. Picture will change when user updated their search option.

If user choose to buy a house, after typing in and submitting an address or zip code, the program will take the information from user input and search the google for the target location. Then the search engine will select all the houses with user specified type and a location that is near the searching address. Noting that, user could search any type of address, since we will use the user input to get the location from google map, so it is google map that predicts the exact location of the input. After searching the database, we will get an array list of houses, that will be stored in the model, and we sort the first thirty houses as the nearest ones to the target location user has input. The result will be shown on the second page, which gives an overview of the returned houses, the left part of the page is a google map with listed houses marked with blue dot. The table only shows thirty houses, and when user click one of them, the selected house will show a flag on the map to give user an exact location of the house . If user want to further search houses with more conditions. We provide several fields including sale price, build year, and living area for user to filter their dreamed house. Clicking the search button will immediately update the google map and result table with the filtered houses that satisfying user's selection. Additionally, user could sort the result houses in the table with customized ranks. For example, user could select sorting by build year, and in descending order, then click sort. The table will show a list of houses based on descending build year from large amount of houses user filtered. As before, only thirty of them will be shown on the table and map. If user feels interested in a certain house after their filtering and sorting, then double clicking a house is a good option. The most detailed information would be shown on a third page, with google map of marked location of this house, the street view of the house and most detailed information related to this house. We are trying to provide a perfect user experience by providing filtering, sorting, and viewing details functionality. We also work hard to make it look as professional as possible so that we could ensure the users find satisfying houses in Philadelphia.

If user wants to sell a house, the sell subpage should be selected. Then typing in an address of your house and then second page will pop up. In this second page, a more detailed information is required from user. Lack of details of the house will cause failure to select similar houses from database. When user complete this mini-form, the program will search the database based on the address of the house, then store the returned information in a model. Compute the most similar houses with the given information in the second page, and a third page will pop up with these similar houses. Meanwhile, an estimated price will be calculated which could viewed as a suggested price for the house with the given information. The basic idea for calculating the price is selecting the most similar thirty houses, then sorting them based on price. From the remaining houses, truncate ten highest and ten lowest price houses. Take the mean of sale price for the median ten houses. The estimated price could give user a professional guide on selling a house from our database information.

For both buying and selling parts, we take long time optimize our GUI design and functionality, and we think it is a fair game for our current model when deciding estimated price or filtering and sorting similar houses. Our program provides an all-around searching service for Philadelphia real-estate.

User Manual:

1. Set up

   There is two options to run our apps:

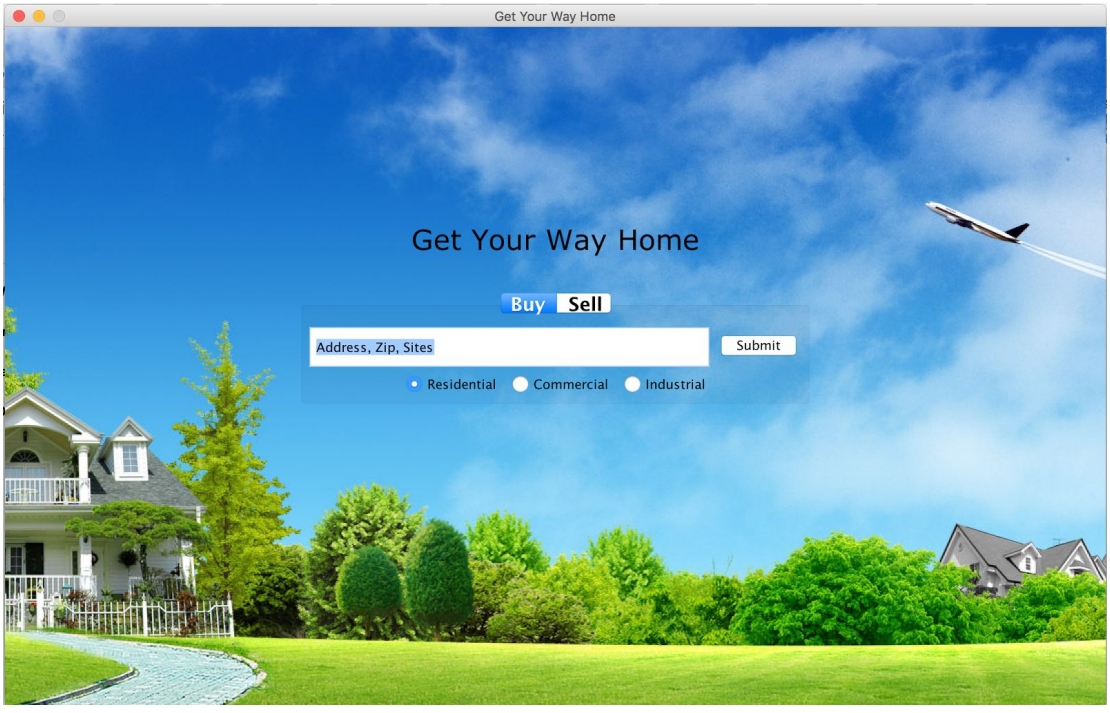
   A. Import our project package into eclipse, and import all libraries from link in Appendix #1 into build path of project, and compile the whole project by **using the main method in BuyHomePage class as entry point**.

   B. Download Appendix #2, which includes **zip file with a jar file and other jpg file inside, just run jar file.** To ensure the correctness for displaying all the picture and google map in this program, the jar file should always be put along with other jpg files, or the picture would be lost while running.

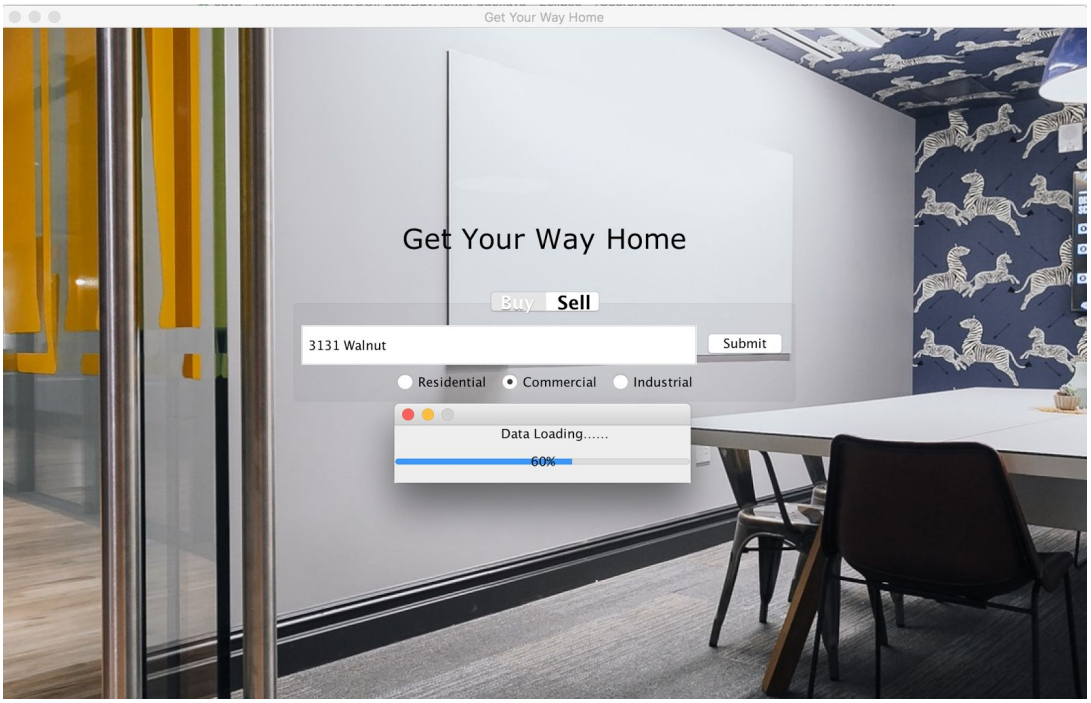
2. Service for Buying

   The first service we provide for user is buying:

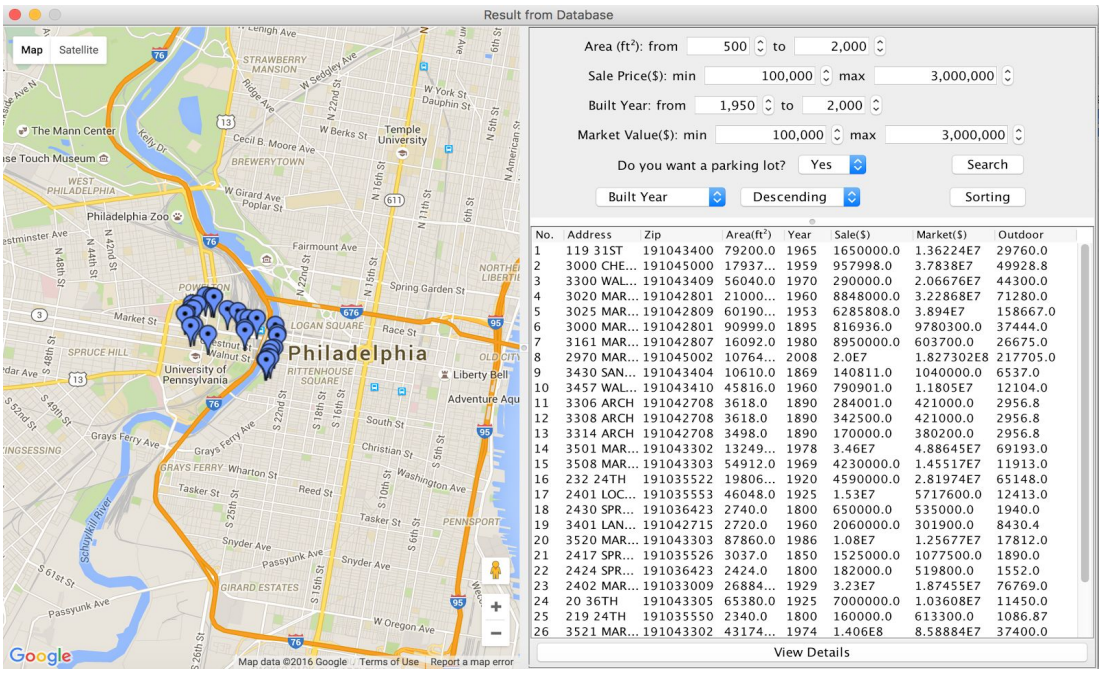
   1. The home page is like following. In the "Buy" tab, you should either enter your detailed address, or zip code. Our program would automatically judge what you have entered. Meanwhile, the type of the real estate you are looking for should be specifically chosen, which includes residential, commercial and industrial. The pictures may be changing when you are choosing different types, to inform your related choice.



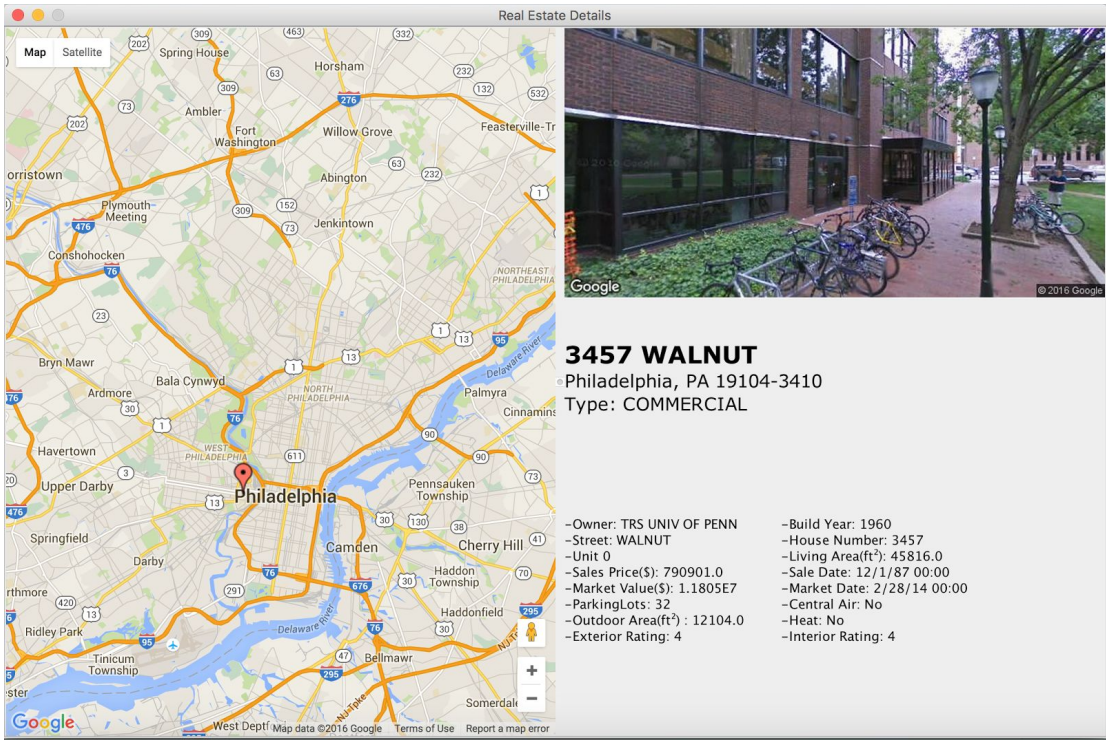
   2. When you submit, the Progress Bar would be displaying. Since we store all our data in our cloud MongoDB, the network environment the user is in will decide the time it spends. For average, we just spend about 20-30 seconds finishing this process in the AirPennNet network environment. If the process taking too long, it's better for you to change your network. Sorry for the inconvenience.



3. Then our system will displaying all the feasible options according to your requirement. The left part is the google map API we use for displaying all the locations of real estate we have listed. Also, there is a filter & sorting part, you can specify the filtering conditions or choose your sorting style. In this part, we only displaying the most meet-requirement real estate in our database. Therefore, the list of real estate would be added new options, or be removed some listed options after sorting and of course, filtering. The ResultListing Page is as follows:



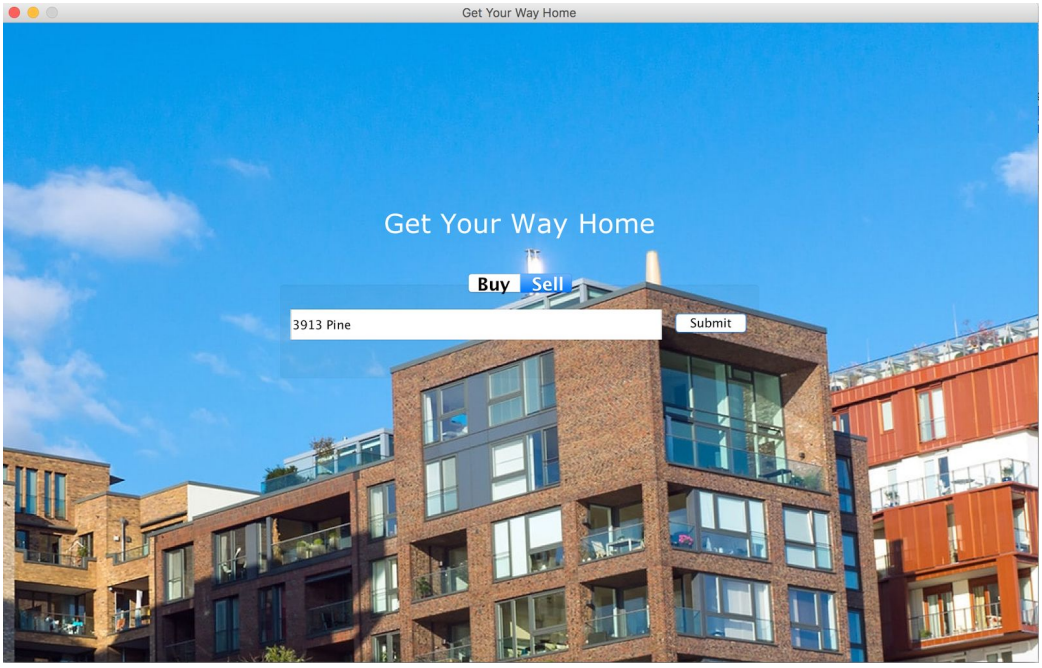
4. The Detail Page of one real estate will be displaying when you either double-click on any row of listing, or use the “View Details” button at the bottom of window after you have choose one option. The Detail Page is as follows. The left part is its location on Google Map, the picture is the streetscape. And also all the detailed information stored in our database, for users’ reference.



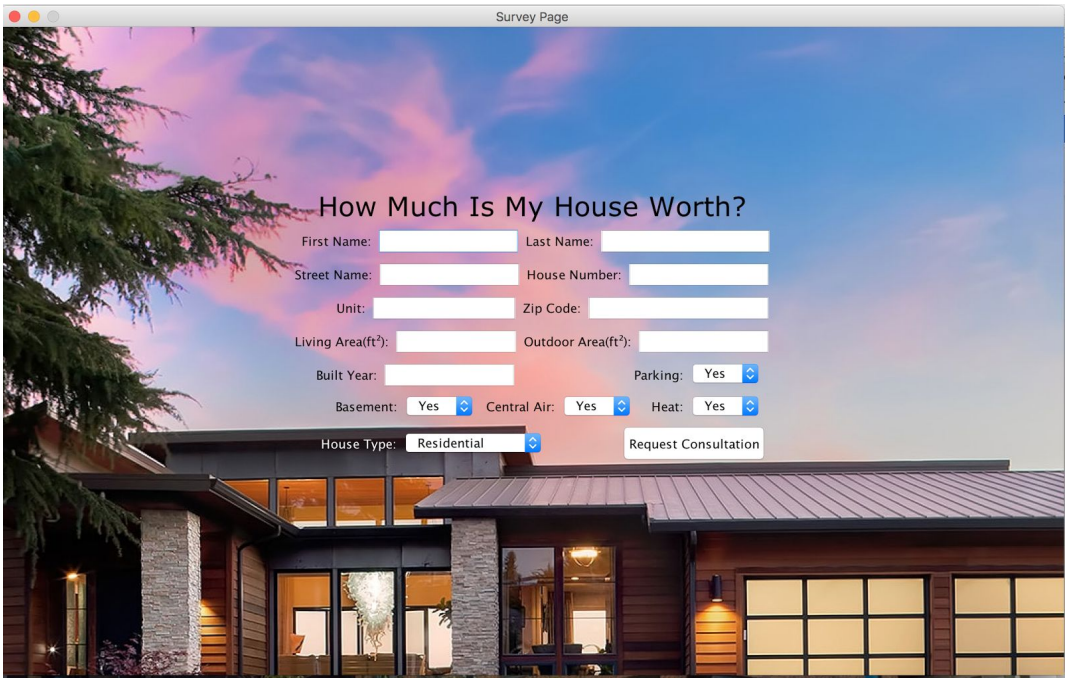
3. Service for Selling

The second service we provide for user is Selling

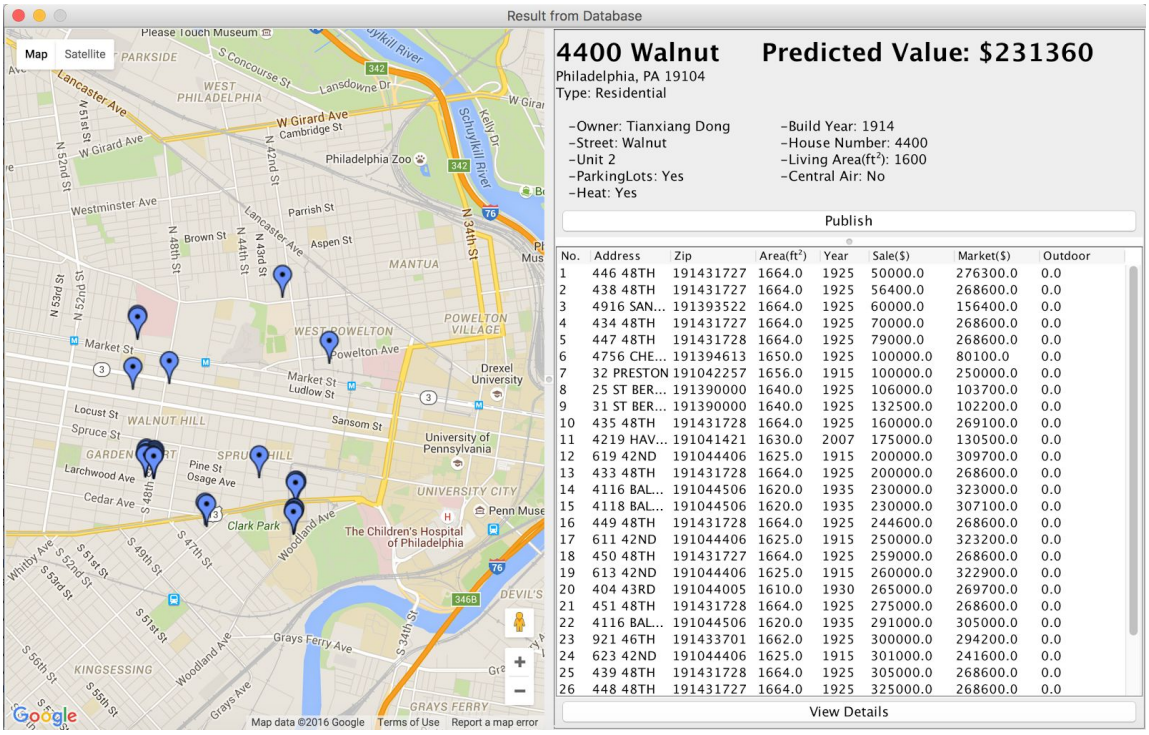
A. As in the same home page, the selling tab requires you to enter your address or zipcode. However at this time, you don’t have to choose your house type at this step. Just click on Submit for further information.



B. In Selling part, the survey page will be displaying when you click on submit in the home page. It requires you to give more detailed information about your real estate. All the information is required and some blanks like zip code, house number and living area, etc. can only accept digits. Once you have click on “Request Consultation”, the program will search in database for suggestions.



C. After searching in database, the system will find similar real estate stored in database and give the evaluated price for your house. Your details and predicted value is displaying, as well as other related real estate and their location on Google Map. As the same thing mentioned before, the speed of searching process depends on your network environment. You can also view details of the listed houses. Apart from that, if you consider the predicted value is appropriate, you can just publish your information, which actually will be collected into our database for other users to review.



Technical Documentation:

MVC:

1. Model
   For model part, it contains two part, Model class and DataBaseEngine class, both of them are acting as model. One is for processing, calculating, sorting and filtering, the other is for database query, data retrieval, instance creation and pre-sorting.
2. View (GUI Pages)
   View part has been separated into several sub components, because we have to generate multiple pages, so we design index page (BuyHomePage), ResultListingPage, DetailPage, SellPage and EsitimateListingPage. Inside these pages, we embed our view part into them, including tables and browserView in EsitimateListingPage and ResultListingPage, in static street view in detail pages. Each part of view is independently controlled by controller, which belongs to same GUI sub component class. For instance, ResultListingPage combine its own browserview, tables and its corresponding controller inside.
3. Controller
   For controller part, we design a multi-level controller. The main search page acts as the first level controller, it collects basic selling or buying information from user, which is address or zip code of the house. The program will connect database retrieving houses related to the searching information and return these houses. Then this controller initialize model, GUI pages based on the results and the second level controller pops up. In the second controller, it provides google map information, filtering and sorting option and table displaying of the houses. This controller call the methods from model to filter or sort the houses when certain action is performed. Meanwhile, the map and table display will be updated responding to the update. So the view will be changed as a result of this controller receiving certain input commands from users. In conclusion, the controller part is two-level, first level gets house information from user, connects database and initialize second controller. The second controller provides option of filtering, sorting and displaying. It communicates with model methods and update views of house table and google map markers.

Database:

1. Data Set Source
   For data set choosing, we use Philadelphia property assessment data from https://www.opendataphilly.org/dataset/opa-property-assessments and http://metadata.phila.gov/#home/datasetdetails/5543865f20583086178c4ee5/ , the detail information about fields and attributes in this data set can be referenced in http://metadata.phila.gov/#home/datasetdetails/5543865f20583086178c4ee5/representationdetails/55d62f07ee9c74144746ccfd/ . For convenience, we have picked 37 attributes for data insertion, later put about 20 attributes in program, remove the entries with unreasonable market value and sales value. After that, we get 220690 entries, and transfer that to JSON format.
2. DataBase Setup
   In order to speed up development on our project, we first set up our database locally, to save time to retrieve data back and forth from remote server. MongoDB NoSQL database is used to store source data. For data query and data processing part, we use

Mongo JAVA driver open source library, to build connection with local database server or remote server, retrieve data, parse data and generate HouseType objects for future processing. For final version, we set up a AWS EC2 instance, which keeps running mango database server, so that anyone using our product can get data from anywhere, anytime. But running time is highly dependent on latency (RTT) of network users are using. Typically, it takes 20 SECs to load data.

Design Pattern:

1. Factory Method
   In the program, the **House is an interface** and the HouseType imlementing House is an abstract class which uses the factory method. Three sub-classes extends the HouseType with different attributes and get methods for the specific type. The HouseType provides an interface for all houses with common attributes and methods. It is the controller that decides which type of house to initialize according to user selection. This kind of design provides flexibility of initializing houses.
2. Singleton
   For LocationLookupMap class, we use singleton pattern, since searching a location should only use one map, and should be globally unique and accessible. For this instance, it has several options to run its search for certain location user put in. First, it will lookup its local memory map, which contains about 24 unique mappings including famous places usually used in Philly. If got local Map miss, it will change to remote database running on server, to see if there is a zip code match, if still got no luck, it will use getCoodinateByGoogle method which uses Geocoding API to get accurate coordinate from google API.
3. Server-worker
   In many class, we have buttons for interaction with users, like the submit button in BuyHomePage, the view details button in ResultLisingPage. All this interaction with database may cost much time due to the network environment. Therefore, SwingWorker is necessary. In BuyHomePage, ResultListingPage and SellPage, we all use swing worker for buttons. So that the requirement requested will be processed in the background. In this case, the user can even send new requests while waiting for the responds. All this requests will start new threads so we don't need to worry about the reaction.

Open Source API:

1. Google Map
   For this project, we use Google Maps Embed API, Google Street View API and Google Geocoding API. Embed API is used with JXBrowser API to be shown in a embedded JXBrowser container in GUI, and we also put marks on the map to show the final results of search. Google Street View API is used for final detail demo of property, it will automatically generate corresponding street view on certain property and shown in detail page. And Geocoding API is also used to get coordinates from any natural language input from user, which is passed to database Engine to get property data from certain range of distance.
2. Mongo JAVA driver

MongoDB API is used to build connection to database and interact with database. For buying activity, we only query database for certain info, for selling activity, user can insert data into our dataset, then retrieve data about similar properties info from database.

3. JXBrowser API
   JXbrowser Library is used to visualize the result list data from previous search. In order to use embed google API, we have to combine JXBrowser and Google Map, to put browser component into JAVA GUI. And change marks and center point of map by changing local html file, so that JXBrowser can visualize all information in a map.

Appendix

1. Library Download Link
   https://drive.google.com/open?id=0B-nzYBpunHxCOHA4NUpmUTJqNzg
   The above Link including all library needed to build our project.
2. JAR File Link