Documentation

**Front-End**

index.html

Uses vue framework and bootstrap.css module

Uses rentsplanetvue.js, gapirentsplanet.js, housing\_calls.js, rentsplanet.css, gmap.css

gmap.css: simple google maps css

rentsplanet.css: css for vue objects in rentsplanetvue.js

rentsplanetvue.js: utilizes vue objects to handle displaying the data from the server

var searchinfo; Controls map, autocomplete, and scraping data.

var reviews; Handles array of reviews visibility and actions.

var createform; Handles the creation form visibility and actions.

gapirentsplanet.js: initializes the google map and the autocomplete search bar

initMap()

Set all necessary fields/attributes for map and places.

housing\_calls.js: client-side API calls to server

createXML()

This returns the XMLHttpRequest object but does it differently depending on browser

get\_housing\_data(address)

When passed in the address as a string will return the review we have for it as a json object

get\_rid\_of (address\_json)

After getting the json object from get\_houseing\_data if passed into this method that json will be deleted from the server

get\_info(address)

When pass the address as a string the server would return data about the house (size/ number of bathrooms) not currently working

submit\_review(address\_review\_jason)

When the user rights a review for a house it is turned into a json and passed into the server and saved using this method

**Back-End**

RentBL.java

* Business Logic class for interacting with database java object.

RentDAO.java

* Data-Access object for communicating with MySQL database

RentEntryBean.java

* Java object created using JSON data

RentEntry.java

* Maps java objects to database entries

RestService.java

* REST service calls for accepting AJAX Calls
* addEntry(RentEntryBean reb)
  + On */create* AJAX call
  + Creates and adds new entry to database
* deleteEntry(RentEntryBean reb)
  + On */delete* AJAX call
  + Looks up and removes entry with ***reb*** attributes
* displayAll()
  + On */display* AJAX call
  + Returns a JSON list of all entries in database
  + Equivalent to *SELECT \* FROM Houses;*
* updateEntry(RentEntryBean reb)
  + On */update* AJAX call
  + Updates review in database from ***reb*** object attributes
* getByAddress(String address)
  + On */entry* AJAX call
  + Returns JSON object pulling data from database using address
  + Equivalent to *SELECT \* FROM Houses WHERE address = <address>*

**Database**

Database is a MySQL Relational Database.

Users:

Root - all queries use the root account for ease of access, not for production use.

Schemas:

Rentsplanet - contains Housing Review data

Tables:

Houses - Contains Reviews for addresses

Attributes:

Address is a VARCHAR and is the primary key for the table.

Rent is a VARCHAR

Landlord is a TEXT

Review is a TEXT