

Team GetRichQuick

Site name: DollaDollaBill

Developers:

Ziyan (Project Manager). Harry, Yikai

We are designing a Craigslist style website for anyone capable of using the internet. The issue with Craigslist is that the site is too compact and we are focusing on a simpler and pleasing UI design. Furthermore, in addition to “add to watchlist” option for listings, we also have a commenting or review feature that asks the seller for additional information. For positing, we would incorporate Google Maps API (see Figure 1) so the seller would enter the location of where to meet up to obtain the service or product.

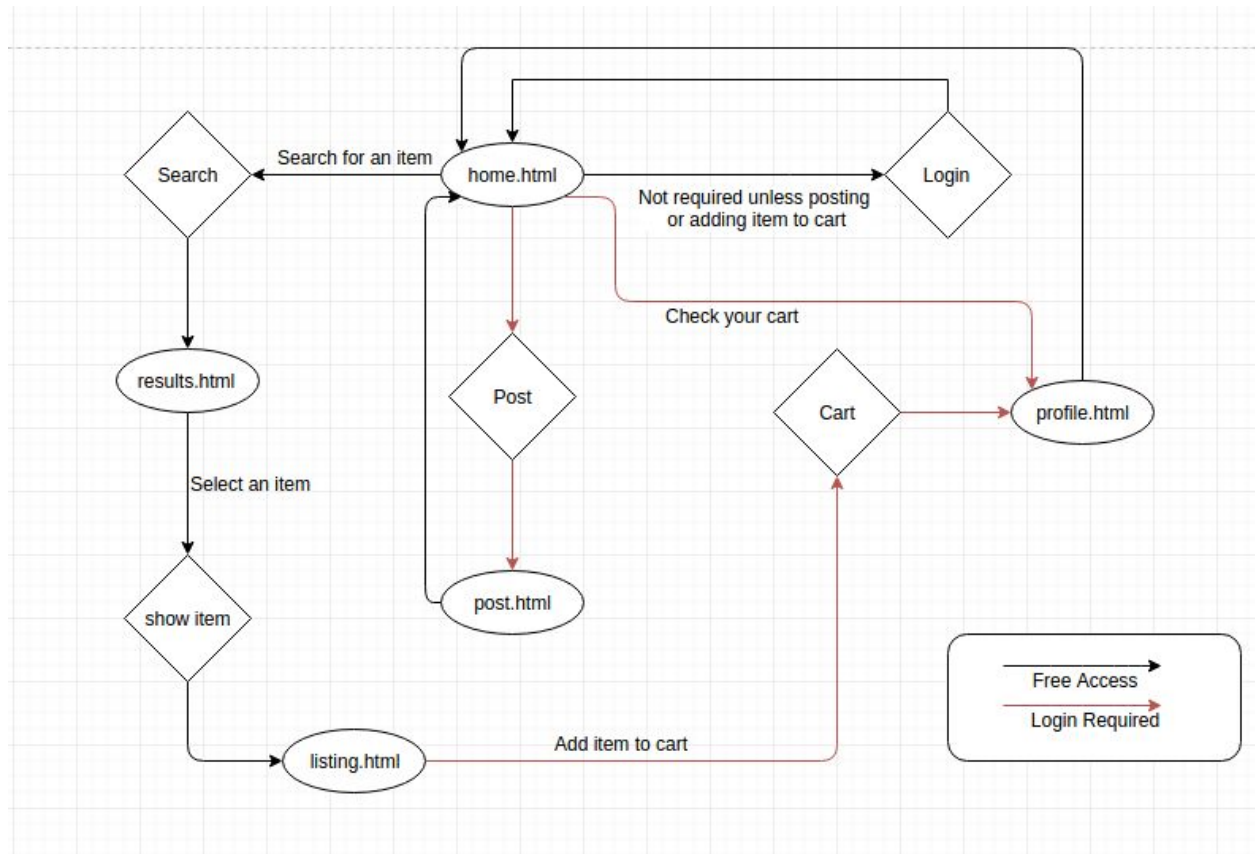
Figure 1: Allows seller to enter address using Google Maps API



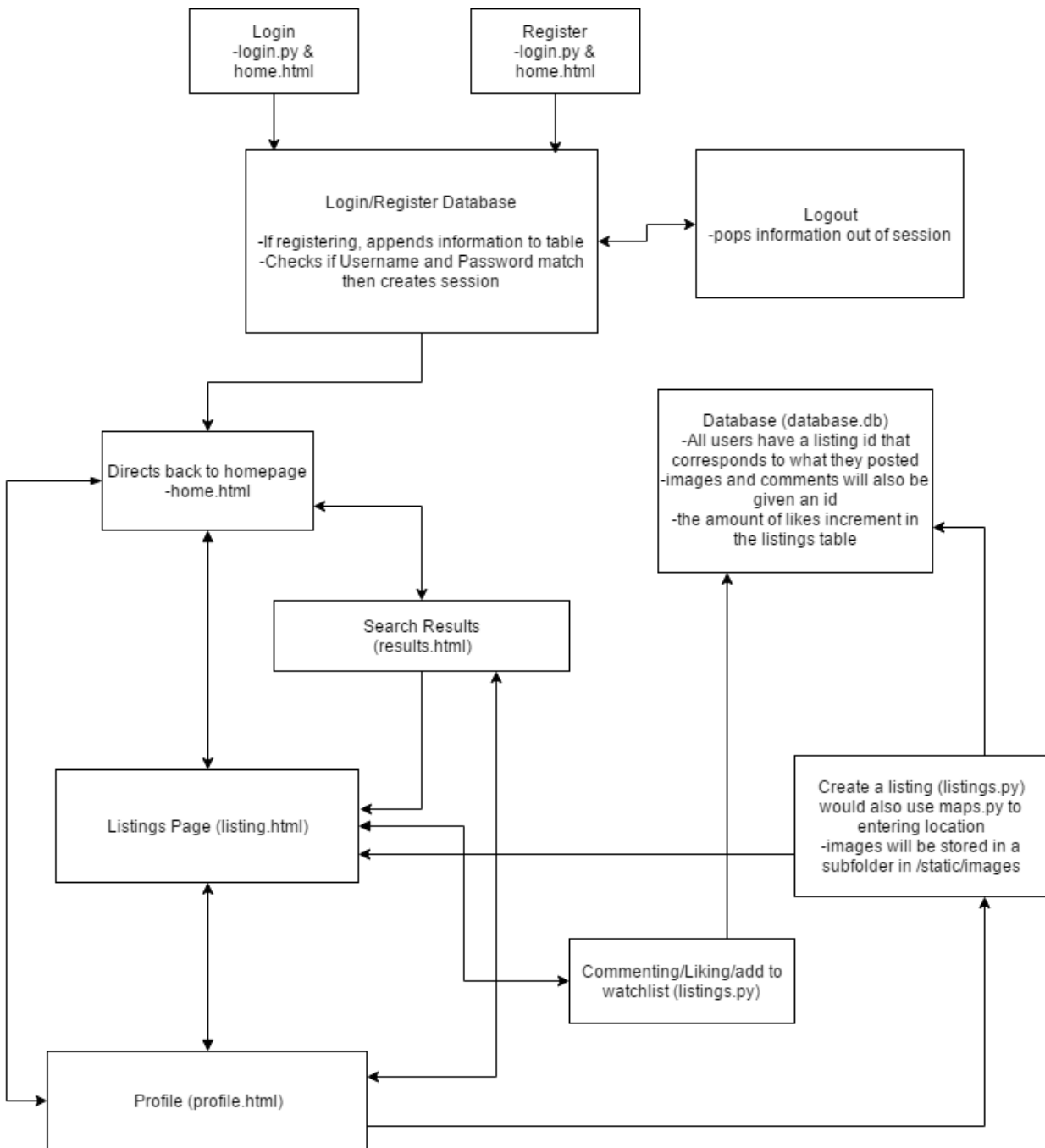
Figure 2: Listing schema

Title of the listing	
Location (where to find service/product)	
Description	Images
Username	
Date	
Number of comments and likes (when listing is clicked, it will redirect to the listing page where user can see all comments.)	

Site map:



Component Map:



Database Schema

database.db

users (stores login info)	
user TEXT	password TEXT
Stores username	Stores password

watchlist (stores watchlisted listings)	
user TEXT	id INTEGER
Stores username	Stores ID of listing

listings (stores listings)					
id INTEGER	listing TEXT	user TEXT	location TEXT	timestamp TEXT	Likes INTEGER
ID of listing	name of listing	creator of listing	location	time	Sum of the likes

images (stores images for each listing, if any)	
id INTEGER	image INTEGER
Stores listing ID	Stores image name

comments (stores comments for each listing)			
id INTEGER	user TEXT	comment TEXT	timestamp TEXT
ID of listing	commenter	location	time

File distribution:

/DollaDollaBill

app.py

/data

database.db - stores tables (see database schema for details)

/utils

login.py - for login and registering.

listings.py - for handling all functions related to listings

maps.py - for using the google maps api to select location of transactions

/templates

home.html - login/register, display most recent listings, search for listings

results.html - displays search results

post.html - form for user to either create or edit a current listing

listing.html - listing page, will contain all data related to listing

profile.html - listing page, will contain user's listings and watchlisted listings and reviews

/static

/images

contains images for listings.(format: <listingid><imagenum>.jpg)

ex: if listing #305 has 3 images, the images stored in this folder will be 3050.jpg, 3051.jpg and 3052.jpg

css files - we will be using bootstrap.

Timeline:

#We will split the work accordingly for each deadline. The details will be recorded in the Devlog

1/8: get the homepage/login working, finish profile.html & related functions

1/15: finish post.html & related functions, listing page & related functions

1/22: finish result.html & related functions, final touches (add map if we have time)

(Weekly deadlines instead of daily deadlines because of the different distributions of final exam dates & other complexities)