Final Project for Software Engineering CSC648/848

Spring 2019

BetterHome

Team 3

Taso Grigoriou (team lead - agrigori@mail.sfsu.edu)
Henok Kassegn (frontend lead)
Sawyer Nixon (backend lead)
Cole (Michael) Tormey (github master)
Austin Sy-Velasco (backend / DB manager)
Liwang Gao (frontend)

URL:

https://better-home-234220.appspot.com

Milestone 5

05/23/2019

Table of Contents:

1. Product Summary:

Our product named “BetterHome” will be the leading property marketplace dedicated to providing consumers with services they need to buy, sell, or rent the place they call home. BetterHome will provide many services to users of all kinds, including landlord hosts, apartment / real-estate searchers and resellers. BetterHome will give property owners the ability to host their property on the website, and provide all general and customized information needed to make their property an attractive candidate for someone’s next home. The website will also provide this same service to “house-flippers,” or resellers, and will allow them to browse all listings with filters, all while being able to host a property up on the website and maintain each aspect simultaneously. What makes BetterHome a unique product is our incredibly detailed advanced search which allows users to browse the various property listings on the website with feature rich and detailed customization of search options. This includes accessibility filters such as laundry and wheelchair access, and hospital / BART proximity that other property marketplaces don’t possess.

URL:

https://better-home-234220.appspot.com

Priority 1 Committed Functions:

For Admins:

(Admin username: **admin**
Admin password: **better-home-admin**)

1. Admins are able to update and maintain user information in the case where users forget their password or if there is inappropriate content on their profile.
2. Admins are able to delete user accounts who host properties (landlords), as well as users who search for property listings (clients).
3. Admins are able to lock accounts for both landlord and clients, as well as unlock their accounts.
4. Admins are able to find registered landlord and clients on the users page by their username, full name, email address, etc.
5. Admins are able to view a gallery collection of property listing images on the images page.
6. Admins are able to delete images on the images page.
7. Admins are able to access username and email for all users on the users page.

For Landlords:

1. Landlords are able to browse and experience all regular features of the website without having to login.
2. Soon to be registered landlords are able to register on the registration page.
3. Registered landlords are able to post their own property with all the information about their property.
4. Registered landlords are able to update the location of their hosted property.
5. Registered landlords are able to update the price of their hosted property.
6. Registered landlords are able to update the city and zipcode of their hosted property.
7. Registered landlords are able to select if their property is a house or apartment.
8. Registered landlords are able to select if their property is wheelchair accessible.
9. Registered landlords are able to login and logout at any and all times.
10. Registered landlords are able to add photos of their newly created listings, as well as adding new photos to their already hosted listing.
11. Registered landlords are able to delete photos of their already hosted property.

For Clients:

1. Soon to be clients are able to create new accounts.
2. Registered clients are able to login and logout whenever they see fit.
3. Registered clients are be able to modify all information on their profile, including email address, phone number, etc.
4. Registered clients are able to add properties they view to their list of favorite properties.
5. Registered clients are able to revisit any and all of their favorite properties.
6. All clients are able to browse the website and search for property listings without having to login.
7. All clients are able to view listings that are currently for sale.
8. All clients are able to see listings that have been recently bought or rented out.
9. A basic search and advanced search box is displayed on the home page to all clients.
10. All clients are able to search property listings by the city they want to view.
11. All clients are able to search listings by a minimum and maximum price range.
12. All clients are able to search listings by the type of property, including houses, apartments, and condos.
13. All clients are able to change the search options they last entered on the search results page.

2. Milestone Documents (M1 - M4):

Milestone 1

03/17/2019

Table of Contents:

1. Executive Summary:

Team 43 is designing a full-stack web application titled “BetterHome” that will be the leading property marketplace dedicated to providing consumers with all the knowledge and services they need to buy, sell, or rent the place they call home. BetterHome will provide many services to users of all kinds, including landlord hosts, apartment / real-estate searchers and resellers. BetterHome will give property owners the ability to host their property on the website, and provide all general and customized information needed to make their property an attractive candidate for someone’s next home. BetterHome will allow searchers to browse the various property listings on the website with feature rich and detailed customization of search options, no matter if they are looking to buy or rent an apartment or house. The website will also provide this same service to “house-flippers,” or resellers, and will allow them to browse all listings with filters, all while being able to host a property up on the website and maintain each aspect simultaneously.

2. Use Cases:

Admin:

Tom the admin receives emails from clients complaining about a landlord that is discriminating against clients. After reviewing several emails as evidence, the Admin decides to take action and delete the offending users account.

Landlord (host):

Joe owns a house in San Francisco, but he is going to move to Seattle next month with his family. So he decides to sell the house. He registers a new account and he fills out the the property informat

Apartment Searcher:

- Austin is moving to San Francisco to attend SFSU, and he is looking to rent an apartment close to campus. He is hoping to find an apartment below the price range of $1500 a month. He would like to have his own room and bathroom. Austin should be able to locate apartments within his requirements.

- John has moved to the Bay Area from out of state for a new job. He is looking for an apartment that is reasonably close to his new workplace or is accessible to public transportation such as BART. In his search for property, he should be able to limit the scope of his search based on location.

Reseller:

Cole is in the business of buying old homes and reselling after refurbishing the property. When he looks for homes that are on the market, the two most important things for him to look out for are location and price. As is the nature of his work, he wants to be able to view the cheapest homes available with ease. While having images to preview the property are also important, he won’t make a purchase before making a trip out to look at the property, so it is vital that he be able to contact the seller in order to arrange a meeting.

Handicap User:

Tom, a wheelchair user, is looking to rent an apartment. He uses a wheelchair because of his back injury from 2010 car accident. He is looking to rent an apartment that accommodate a wheelchair. He finds apartment that has a wider Entryways and hallways that will allow him to turn around in his wheelchair.

3. Data Definition (items and entities):

Admin:
User from BetterHome team that can maintain use information and ensure listings posted on our website are valid.

Client (Registered User):
User that has registered with the site, can access all listings on the site, can message the listing owners, and can post their own listings.

Guest User:
User can see the listings on the site without making an account, but cannot message listing owners nor access their contact information.

Privacy policy:
Policy to maintain and protect our user information.

Map View:
A list of all available home/apartment listings in the designated area.

Handicap:
Indicator on listings to show that they have handicap accessible pathways.

Listing:

- Photo:
Some photos to showcase the house/apartment.

- Price/Rate:
A general idea of how much each listing is.

- Address:
Area for where the listing is located.

BART Accessible:
Within a close proximity to a BART Station.

Availability:
Indicator to show that if the listing is available (could signify that an admin should take down the listing OR can just scrap this and have the owner take down the listing themself).

Reseller:

A user that is planning to purchase houses/apartments not for their own use, but to improve and then resell back on the market.

Handicap User:
A user that would like special accommodations with their house: wheelchair accessible (elevator to apartment, ramp) or within a relatively close distance to a hospitable or care unit.

Landlord:
A user that will be posting listings on the website in order to get a buyer's attention.

Rooms:
Amount of rooms available in the property listing.

Parking:
How much parking comes with the property.

Profile:
A place on our website that will show personal information of the user.

Help:
A brief explanation of how our website works.

4. <u>Functional Requirements:</u>

For Admin:

1. Admin shall be able to update/maintain user information.
2. Admin shall be able to delete Landlord and Clients.
3. Admin shall be able to lock and unlock Landlord and Clients.
4. Admin shall be able to find Landlord and Clients.
5. Admin shall be able to view gallery.
6. Admin shall be able to delete photos.
7. Admin shall be able to reset password for both Landlord and Clients.
8. Admin shall be able to access username and email of Landlord and Clients.

For Landlord:

1. Landlord shall be able to browse the website without login.
2. New Landlord shall be able to register.
3. Registered Landlord shall be able to post property descriptions.

4. Registered Landlord shall be able to post the location of the property.
5. Registered Landlord shall be able to post the price of the property.
6. Registered Landlord shall be able to post the city and zipcode of the property.
7. Registered Landlord shall be able to select if the property is a house or apartment.
8. Registered Landlord shall be able to select if the property is wheelchair accessible.
9. Registered Landlord shall be able to login.
10. Registered Landlord shall be able to add photos.
11. Registered Landlord shall be able to delete photos.
12. Registered Landlord shall be able to contact registered clients.
13. Registered Landlord shall be able to view gallery.
14. Registered Landlord shall be able to contact admin.
15. Registered Landlord shall be able to change the property status to available, pending, and sold.
16. Registered Landlord shall be able to update price, location, and pictures.

For Client:

1. New clients shall be able to register.
2. Registered clients shall be able to login.
3. Registered clients shall be able to sign out.
4. Registered clients shall be able to modify their profile.
5. Registered clients shall be able to add the listing as favorite .
6. Registered clients shall be able to revisit the favorite .
7. All clients shall be able to browse the website without login.
8. All clients shall be able to view the sale listing.
9. All clients shall be able to view the sold listing.
10. A free text search box shall be displayed to all clients.
11. The default display in the free text search box shall be “address, neighborhood, zip”.
12. All clients shall be able to search the listing by city.
13. All clients shall be able to search the listing by price range.
14. All clients shall be able to search the listing by property type.
15. All clients shall be able to filter the search result by relevance.
16. All clients shall be able to modify the last search
17. The search results shall be able to be sorted by relevance.
18. All clients shall be able to view the search results in a map view based on current location.
19. All clients shall be able to contact the landlord.
20. All clients shall be able to contact the admin.
21. Listings thats are similar to a viewed listing should be displayed to all clients
22. A Q&A page shall be provided for all clients.
23. All handicap users shall be able to search for listings by accessibility.

24. All resellers shall be able to repost the listing onto the site.

For handicap:

1. Handicap users shall be able to flag their profile, denoting a need for handicap-specific accessibilities.
2. Handicap users shall be able to specify in their profile the exact nature of their accessibility needs.
3. Handicap users shall be able to specify in their search that they require a ramp.
4. Handicap users shall be able to specify in their search that they require a ground floor unit.
5. Handicap users shall be able to search the listing by number of floors.
6.

For reseller:

1. Resellers shall be able to access the original post for the property they purchased.
2. Resellers shall be able to modify and repost the original listing with updated information.
3. Original listing shall be linked to new listing so other users may view and verify improvements and/or alterations to property.

For guest User:

1. Guest users shall be able to register.
2. Guest users shall be able to search listing by city.
3. Guest users shall be able to search listing by price range.
4. Guest users shall be able to search listing by property type.
5. Guest users shall be able to modify the last search.
6. Guest users should be able to view the search results in a map view based on current location.
7.

For apartment Searcher:

1. Apartment searcher shall be able to search apartments only available for rent
2. Apartment searcher shall be able to search the apartment by number of bedrooms
3. Apartment searcher shall be able to search the apartment by monthly payment.
4. Apartment searcher shall be able to sort by the results by relevance.
5. Apartment searcher shall be able to filter the results by relevance.
6. Apartment searcher shall be able to view the apartment features.

5. Non-Functional Requirements:

Security:

1. Login shall be required for Clients and Admins.
2. Username shall be the Client’s email.

3. Password shall be encrypted when stored.
4. Client's session shall end upon leaving the site.
5. Client's session shall only end by code design.

Performance:

1. Loading time for site shall be less than 3 seconds for any screen.

Capacity:

1. The total data storage allowed by the web site shall not exceed of 80 % of the server capacity for this site.
2. The web site shall be prepared to support scalability for adding future new features.
3. The web site shall be capable to handle at least 50 Clients simultaneously.

Reliability:

1. Downtime for maintenance shall be less than 3 hours per month.
2. Downtime for maintenance shall not affect the main functionality of the site.
3. In all cases, downtime for maintenance shall be informed to the Client through email.

Recovery:

1. In a total failure case, the whole site should be put down to revision.
2. If broken, the mean time to recovery shall not exceed one day.

Data Integrity:

1. Database tables shall be backed up every day.
2. Administrator shall be able to execute a recovery when needed.
3. Image Sizes shall be limited up to 1 megabyte.
4. Images shall be uploaded in correct format (jpg, jpeg, or pdf) to the server.
5. Links to images on the server shall be uploaded to the database.

Compatibility:

1. The site shall be compatible with the last version of Microsoft Edge browser (44.17763).
2. The site shall be compatible with the last version of Safari browser (12).
3. The site shall be compatible with the last version of Firefox browser (65.0.2).
4. The site shall be compatible with the last version of Chrome browser (7.3).
5. Third party applications shall not be able to modify any content that may affect the site compatibility.
6. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions.
7. The site should be compatible to escalate to new relational databases.

Conformance with Coding Standards:

1. Architecture and design standards shall meet all the requirements listed under the High Level Architecture section of this document.
2. Only working code that meets all the code standards shall be submitted to the project repository.
3. Any working code shall be tested and debugged before being considered working code.
4. Any internal errors or exceptions returned by the code shall be stored in a log.
5. Any error that may affect the functionality of the site shall be reported to the Client.
6. Any error shall be handled in a way that does not affect the functionality of the site.
7. The whole production cycle of this site shall be finished 2 weeks before the delivery date.
8. This site shall not be launched without all the priority one features completed and tested.

Look and Feel Standards:

1. The application and its layouts shall look professional.
2. The site shall be simple enough to handle by all the parties involved.
3. Elements on screen shall have the correct density to meet the compatibility standard of the browsers.
4. Elements on screen shall have rich and beautiful colors for Client delight.
5. The site shall be able to work correctly without mouse interaction.
6. The site shall be able to work correctively without keyboard interaction.
7. Elements in screen shall be resized automatically without Client interaction when being loaded in all the different platforms supported by the site.

Internationalization / Localization Requirements:

1. Default language shall be English.
2. The site shall support scalability to add other languages.
3. The site shall support geolocation in order to show listing locations.
4. Any copyrighted material shall be immediately be taken down upon reception of an official DMCA takedown request.

Website Policies:

1. A link to the policies of this site shall be always visible in all its pages to be accessible by all the parties.
2. Clients' data shall not be sold to third parties.
3. Clients and Landlords data that do not add any functionality to the system shall not be collected.
4. Clients that post inappropriate listings(false listings/ copyright images/ sexual images) shall have their postings taken down.

6. Competitive Analysis:

	SMCHousing	Trulia	Zillow	Apartments	BetterHome
Search For Property	+	+	+	+	+
Post Listings	+	+	+	+	+
Set Favorites	-	+	+	+	+
View accessibility (Handicap /BART)	+	-	-	-	+
Show Property Status	+	+	+	+	+

Competitors:
http://smchousingsearch.org/
https://www.trulia.com/
https://www.zillow.com/
https://www.apartments.com/

Most of our competitors have a great deal of functionality and features so our goal as a team was to highlight and emphasize the features they don't have. One of the ways we tend to accomplish this is to have a few unique search terms. All of these home searching websites provide the essential functions of being able to search/post properties to be sold or rented out. SMCHousing focuses on providing an online house search service for people in the San Mateo county. On top of their service, they provide housing tools such as a Moving Cost Calculator and Rental Checklists to help people in the moving process. Trulia provides a community aspect to the home searching market by providing information on the area around the house, such as schools to the neighborhood or crime rates around the house. All these sites provide their own uniqueness to the home searching market, but they always lack in an area that we feel is just as important. That is to see if a home provides accessibility options for new homeowners, whether it be in BART accessibility or disabled accessibility. BetterHome intends to include all these important features, as well as add more accessibility options.

7. High-level system architecture and technologies used:

1. Application shall be developed using a variation of the MEAN stack consisting of a MySQL database, Express/Node.js backend, and an Angular/HTML/CSS frontend.
2. Application shall be hosted and deployed on Google Cloud Platform.
3. Data shall be stored in the MySQL database hosted by Google Cloud Platform.
4. Application shall be developed using the latest Webstorm version 2018.3.
5. Any other tools or frameworks shall be approved by the Team Lead and either the backend lead or frontend lead as is relevant.
6. Application shall be viewable and accessible on standard browsers up to the two latest version of: Mozilla (64, 65), Safari (11, 12), Chrome (72, 73), and Edge (44, 42).
7. Privacy of all users shall be protected and all privacy policies shall be appropriately communicated to the users.
8. Application shall be simple and intuitive to use for all potential users.
9. Google Analytics shall be added for reported web traffic to the website.
10. Modern SE processes and practices must be used as specified in the class, including collaborative and continuous SW development, using the tools approved by the instructor.

8. Team:

- Taso Grigoriou – Team Lead
- Henok Kassegn – Front-End Team Lead
- Sawyer Nixon – Back-End Team Lead
- Cole (Michael) Tormey – GitHub Master / Front-End Developer
- Austin Sy-Velasco – Back-End Developer / Document master
- Liwang Gao – Front-End Developer

9. Checklist:

· Team found a time slot to meet outside of the class.

DONE

· Github master chosen.

DONE

· Team decided and agreed together on using the listed SW tools and deployment server.

DONE

· Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing.

ON TRACK

· Team lead ensured that all team members read the final M1 and agree/understand it before submission.

DONE

· Github organized as discussed in class (e.g. master branch, development branch, folder for milestone, documents, etc).

DONE

Milestone 2

04/19/2019

Table of Contents:

1. Data Definitions V2:

Admin:
User from BetterHome team that can maintain use information and ensure listings posted on our website are valid.

Client (Registered User):
User that has registered with the site, can access all listings on the site, can message the listing owners, and can post their own listings.

Guest User:
User can see the listings on the site without making an account, but cannot message listing owners nor access their contact information.

Privacy policy:
Policy to maintain and protect our user information.

Map View:
A list of all available home/apartment listings in the designated area.

Handicap:
Indicator on listings to show that they have handicap accessible pathways.

Listing:
- Photo:
Some photos to showcase the house/apartment.
- Price/Rate:
A general idea of how much each listing is.
- Address:
Area for where the listing is located.

BART Accessible:
Within a close proximity to a BART Station.

Availability:
Indicator to show that if the listing is available (could signify that an admin should take down the listing OR can just scrap this and have the owner take down the listing themself).

Reseller:
A user that is planning to purchase houses/apartments not for their own use, but to improve and then resell back on the market.

Handicap User:
A user that would like special accommodations with their house: wheelchair accessible (elevator to apartment, ramp) or within a relatively close distance to a hospitable or care unit.

Landlord:
A user that will be posting listings on the website in order to get a buyer's attention.

Rooms:
Amount of rooms available in the property listing.

Parking:
How much parking comes with the property.

Profile:
A place on our website that will show personal information of the user.

Help:
A brief explanation of how our website works.

2. Functional Requirements V2:

1st Priority:

For Admin:

1. Admin shall be able to update/maintain user information.
2. Admin shall be able to delete Landlord and Clients.
3. Admin shall be able to lock and unlock Landlord and Clients.
4. Admin shall be able to find Landlord and Clients.
5. Admin shall be able to view gallery.
6. Admin shall be able to delete photos.
7. Admin shall be able to access username and email of Landlord and Clients.

For Landlord:

1. Landlord shall be able to browse the website without login.
2. New Landlord shall be able to register.
3. Registered Landlord shall be able to post property descriptions.
4. Registered Landlord shall be able to post the location of the property.
5. Registered Landlord shall be able to post the price of the property.
6. Registered Landlord shall be able to post the city and zipcode of the property.
7. Registered Landlord shall be able to select if the property is a house or apartment.
8. Registered Landlord shall be able to select if the property is wheelchair accessible.
9. Registered Landlord shall be able to login.
10. Registered Landlord shall be able to add photos.
11. Registered Landlord shall be able to delete photos.

For Client:

1. New clients shall be able to register.
2. Registered clients shall be able to login.
3. Registered clients shall be able to sign out.
4. Registered clients shall be able to modify their profile.
5. Registered clients shall be able to add the listing as favorite .
6. Registered clients shall be able to revisit the favorite .
7. All clients shall be able to browse the website without login.
8. All clients shall be able to view the sale listing.
9. All clients shall be able to view the sold listing.
10. A free text search box shall be displayed to all clients.
11. All clients shall be able to search the listing by city.

12. All clients shall be able to search the listing by price range.
13. All clients shall be able to search the listing by property type.
14. All clients shall be able to modify the last search.
15. The search results shall be able to be sorted by relevance.

<u>2nd Priority:</u>

For Admin:

1. Admin shall be able to filter data by property type, status, or date posted.
2. Admin shall be able to review flags, and take the appropriate action.
3. Admin shall be able to run data analysis on user data to understand how the application is being used.

For Client:

1. The default display in the free text search box shall be “address, neighborhood, zip”.
2. All clients shall be able to filter the search result by relevance.
3. All clients shall be able to view the search results in a map view based on current location.
4. All clients shall be able to contact the landlord.
5. All clients shall be able to contact the admin.
6. Listings thats are similar to a viewed listing should be displayed to all clients
7. A Q&A page shall be provided for all clients.
8. All handicap users shall be able to search for listings by accessibility.
9. All resellers shall be able to repost the listing onto the site.
10. The application shall keep track of registered user history search.
11. Registered clients shall be given a list of checkbox to select and search (Advanced search).

For Landlord:

1. Registered Landlord shall be able to contact registered clients.
2. Registered Landlord shall be able to view gallery.
3. Registered Landlord shall be able to contact admin.
4. Registered Landlord shall be able to change the property status to available, pending, and sold.
5. Registered Landlord shall be able to update price, location, and pictures.
6. Registered Landlord shall be given a list of checkbox to select if their property has access to wheelchair, BART, parking, etc.

For guest User:

1. Guest users shall be able to register.
2. Guest users shall be able to search listing by city.
3. Guest users shall be able to search listing by price range.
4. Guest users shall be able to search listing by property type.
5. Guest users shall be able to modify the last search.
6. Guest users should be able to view the search results in a map view based on current location.

For Apartment Searcher:

1. Apartment searcher shall be able to search apartments only available for rent
2. Apartment searcher shall be able to search the apartment by number of bedrooms
3. Apartment searcher shall be able to search the apartment by monthly payment.
4. Apartment searcher shall be able to sort by the results by relevance.
5. Apartment searcher shall be able to filter the results by relevance.
6. Apartment searcher shall be able to view the apartment features.

3rd Priority:

For handicap:

1. Handicap users shall be able to flag their profile, denoting a need for handicap-specific accessibilities.
2. Handicap users shall be able to specify in their profile the exact nature of their accessibility needs.
3. Handicap users shall be able to specify in their search that they require a ramp.
4. Handicap users shall be able to specify in their search that they require a ground floor unit.
5. Handicap users shall be able to search the listing by number of floors.

For reseller:

1. Resellers shall be able to access the original post for the property they purchased.
2. Resellers shall be able to modify and repost the original listing with updated information.
3. Original listing shall be linked to new listing so other users may view and verify improvements and/or alterations to property.

Non-Functional Requirements:

Security:

1. Login shall be required for Clients and Admins.
2. Username shall be the Client's email.
3. Password shall be encrypted when stored.
4. Client's session shall end upon leaving the site.
5. Client's session shall only end by code design.

Performance:

1. Loading time for site shall be less than 3 seconds for any screen.

Capacity:

1. The total data storage allowed by the web site shall not exceed of 80 % of the server capacity for this site.
2. The web site shall be prepared to support scalability for adding future new features.
3. The web site shall be capable to handle at least 50 Clients simultaneously.

Reliability:

1. Downtime for maintenance shall be less than 3 hours per month.
2. Downtime for maintenance shall not affect the main functionality of the site.
3. In all cases, downtime for maintenance shall be informed to the Client through email.

Recovery:

1. In a total failure case, the whole site should be put down to revision.
2. If broken, the mean time to recovery shall not exceed one day.

Data Integrity:

1. Database tables shall be backed up every day.
2. Administrator shall be able to execute a recovery when needed.
3. Image Sizes shall be limited up to 1 megabyte.
4. Images shall be uploaded in correct format (jpg, jpeg, or pdf) to the server.
5. Links to images on the server shall be uploaded to the database.

Compatibility:

1. The site shall be compatible with the last version of Microsoft Edge browser (44.17763).
2. The site shall be compatible with the last version of Safari browser (12).
3. The site shall be compatible with the last version of Firefox browser (65.0.2).
4. The site shall be compatible with the last version of Chrome browser (7.3).
5. Third party applications shall not be able to modify any content that may affect the site compatibility.
6. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions.
7. The site should be compatible to escalate to new relational databases.

Conformance with Coding Standards:

1. Architecture and design standards shall meet all the requirements listed under the High Level Architecture section of this document.
2. Only working code that meets all the code standards shall be submitted to the project repository.
3. Any working code shall be tested and debugged before being considered working code.
4. Any internal errors or exceptions returned by the code shall be stored in a log.
5. Any error that may affect the functionality of the site shall be reported to the Client.
6. Any error shall be handled in a way that does not affect the functionality of the site.
7. The whole production cycle of this site shall be finished 2 weeks before the delivery date.
8. This site shall not be launched without all the priority one features completed and tested.

Look and Feel Standards:

1. The application and its layouts shall look professional.
2. The site shall be simple enough to handle by all the parties involved.
3. Elements on screen shall have the correct density to meet the compatibility standard of the browsers.
4. Elements on screen shall have rich and beautiful colors for Client delight.
5. The site shall be able to work correctly without mouse interaction.
6. The site shall be able to work correctively without keyboard interaction.
7. Elements in screen shall be resized automatically without Client interaction when being loaded in all the different platforms supported by the site.

Internationalization / Localization Requirements:

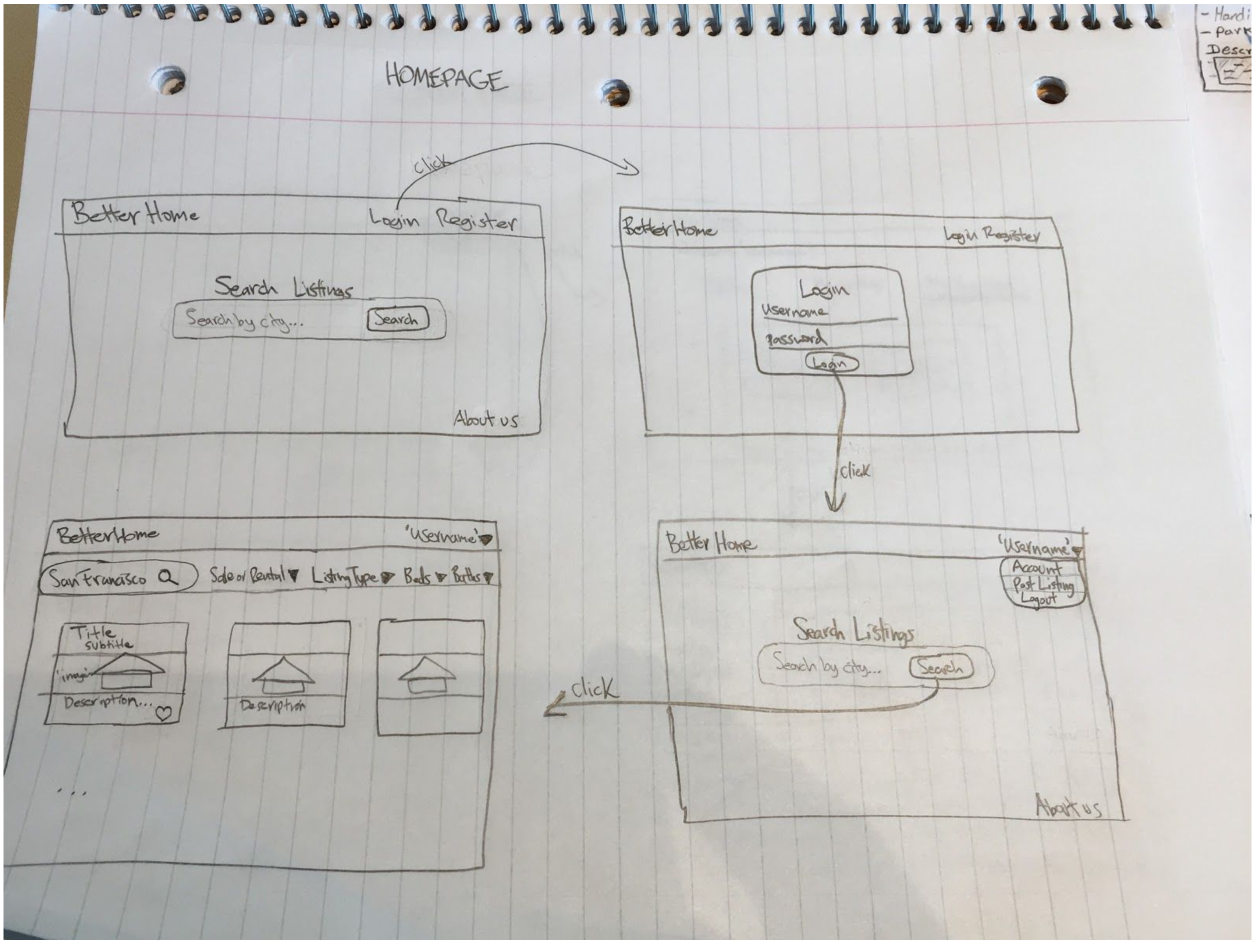
1. Default language shall be English.
2. The site shall support scalability to add other languages.
3. The site shall support geolocation in order to show listing locations.
4. Any copyrighted material shall be immediately be taken down upon reception of an official DMCA takedown request.

Website Policies:

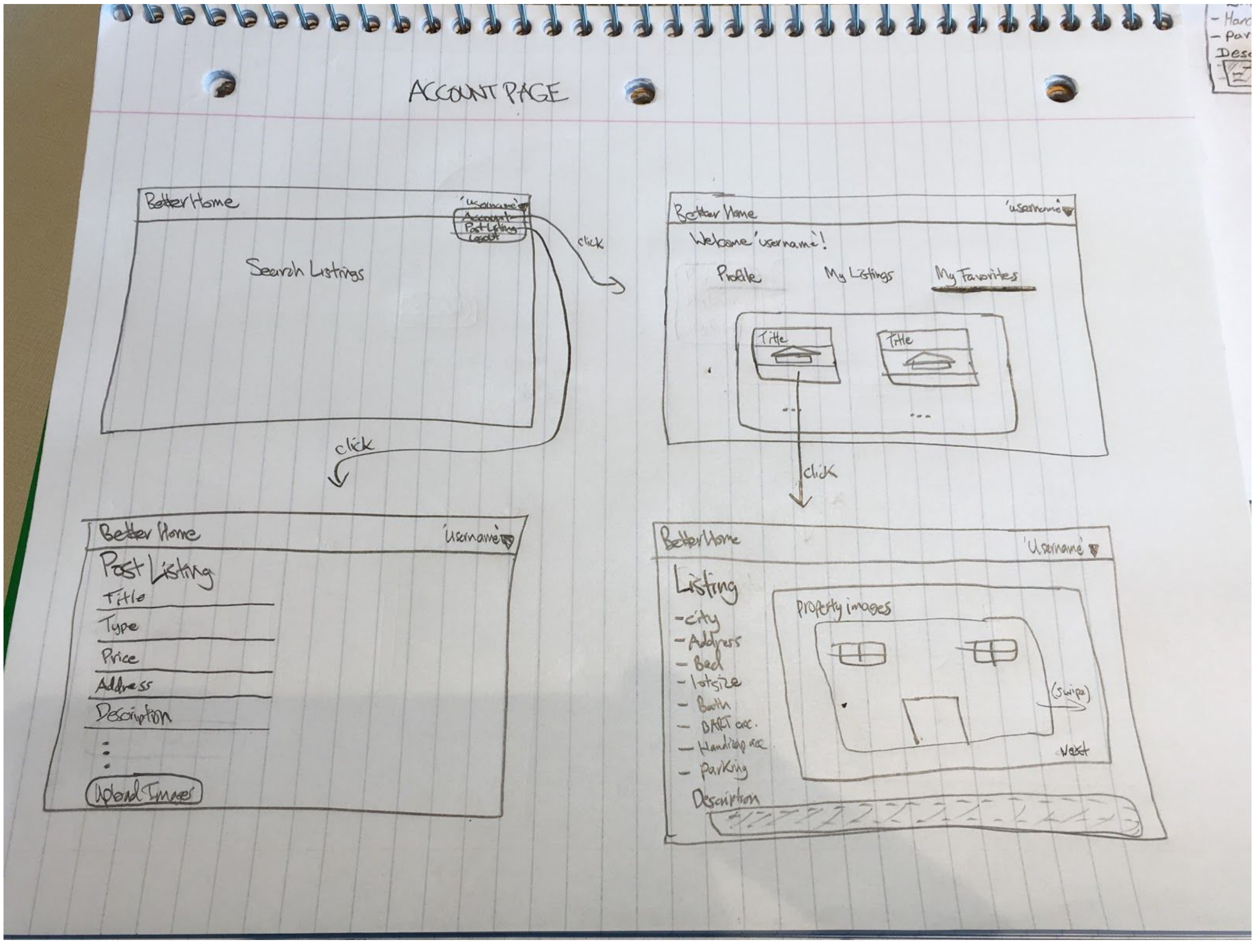
1. A link to the policies of this site shall be always visible in all its pages to be accessible by all the parties.
2. Clients' data shall not be sold to third parties.
3. Clients and Landlords data that do not add any functionality to the system shall not be collected.
4. Clients that post inappropriate listings(false listings/ copyright images/ sexual images) shall have their postings taken down.

3. UI Mockups and Storyboards

Login flow:



Account Flow:

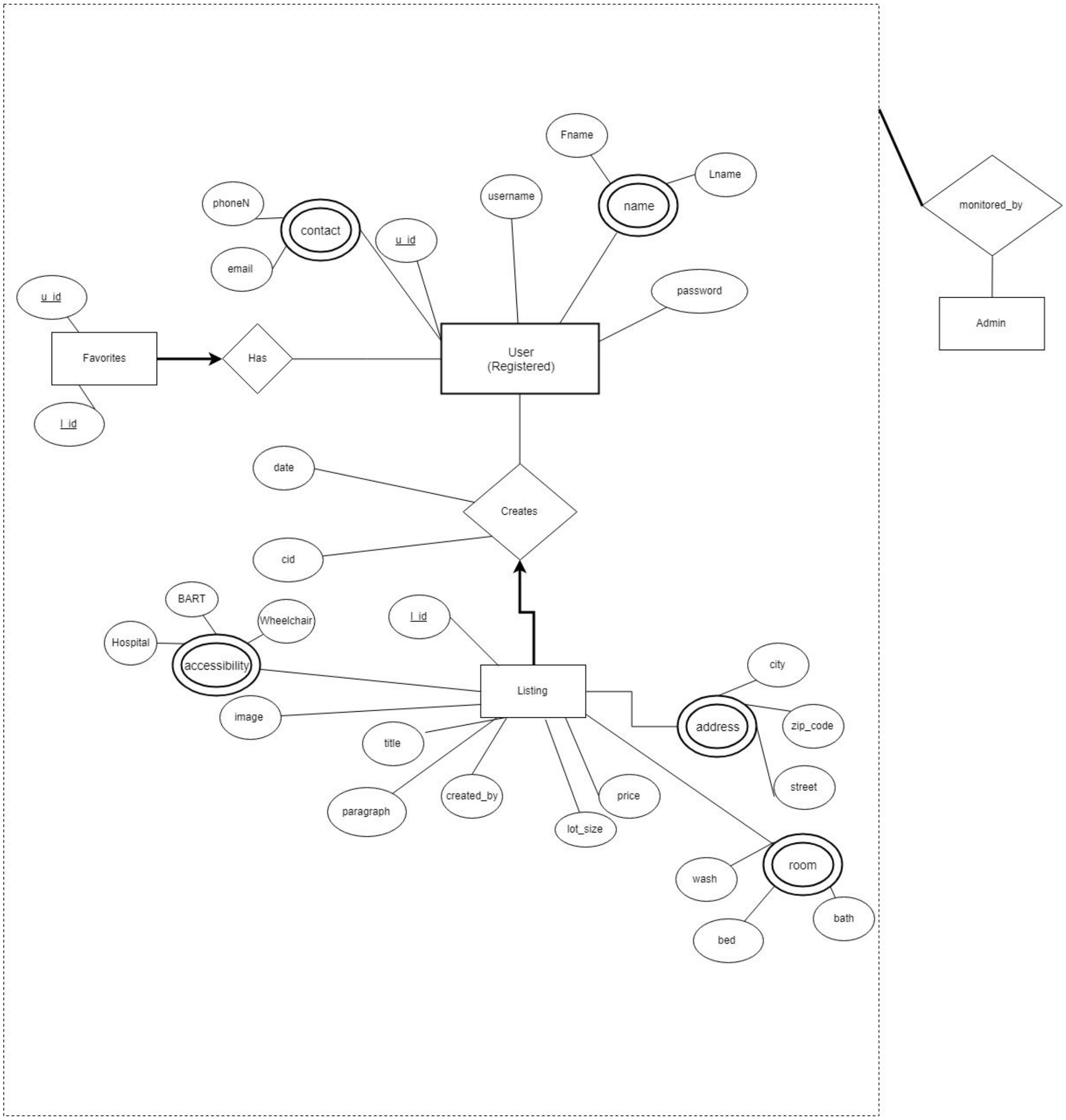


4. High-level Architecture, Database Organization:

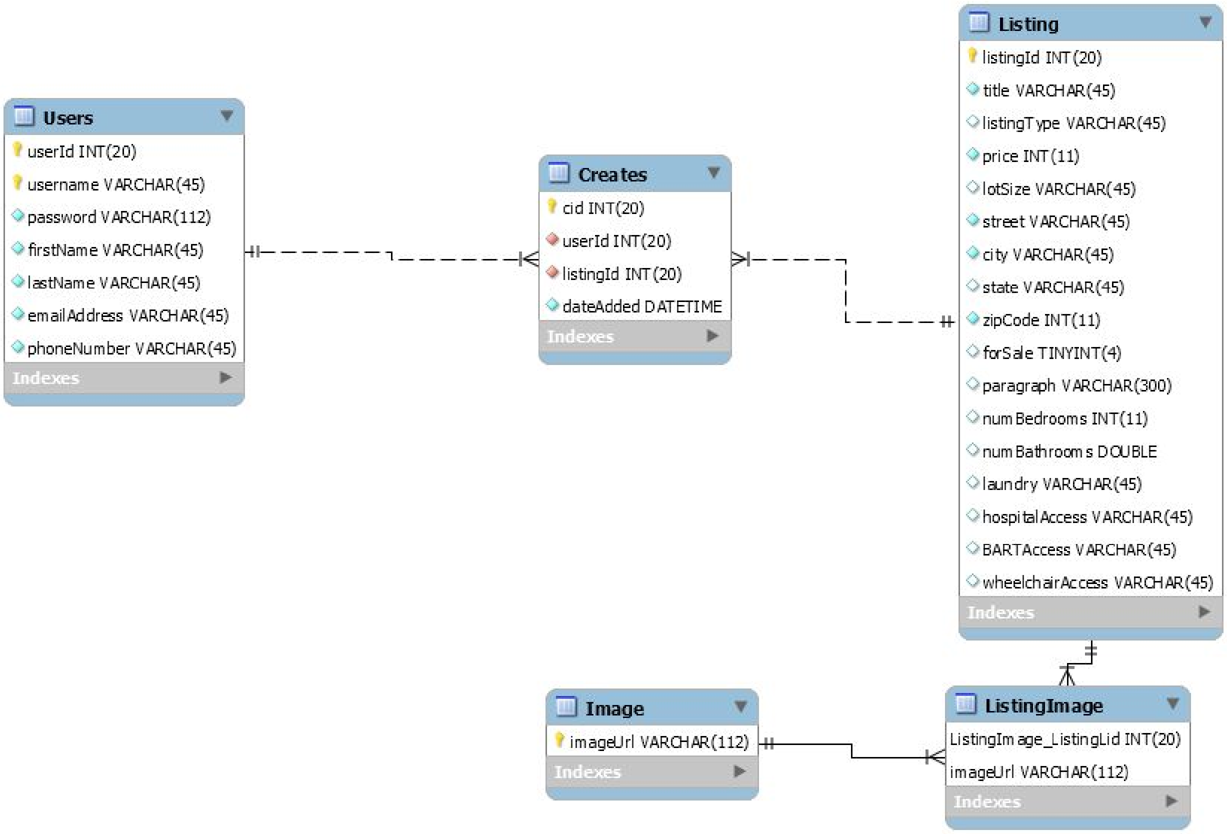
Business Rules:

1. Users can create multiple listings on the website.
2. Users can view multiple listings with a specific amount of rooms.
3. A listing must have only one description describing its characteristics.
4. A listing can be displayed on multiple listing boards.
5. Admins must monitor all listings posted by all users.

ERD Diagram:



DB Diagram:


Users
userId INT(20)
username VARCHAR(45)
password VARCHAR(112)
firstName VARCHAR(45)
lastName VARCHAR(45)
emailAddress VARCHAR(45)
phoneNumber VARCHAR(45)
Indexes
Creates
cid INT(20)
userId INT(20)
listingId INT(20)
dateAdded DATETIME
Indexes
Listing
listingId INT(20)
title VARCHAR(45)
listingType VARCHAR(45)
price INT(11)
lotSize VARCHAR(45)
street VARCHAR(45)
city VARCHAR(45)
state VARCHAR(45)
zipCode INT(11)
forSale TINYINT(4)
paragraph VARCHAR(300)
numBedrooms INT(11)
numBathrooms DOUBLE
laundry VARCHAR(45)
hospitalAccess VARCHAR(45)
BARTAccess VARCHAR(45)
wheelchairAccess VARCHAR(45)
Indexes
Image
imageUrl VARCHAR(112)
Indexes
ListingImage
ListingImage_ListingId INT(20)
imageUrl VARCHAR(112)
Indexes

DBMS:

We chose MySQL as our Database Management System since it provides great support alongside MySQLWorkbench and the workflow is very intuitive and efficient.

Media Storage:

Images (.jpeg, .png, .tiff, .bmp) will be the only form files that the user can input into the database. They will be stored as URLs on the “Image” table and be a primary key in “ListingImage” that will connect it to the “Description” table.

Search/Filter Architecture and Implementation:

Our search functionality requires a city name in the free text search box from the frontend. We utilize the SQL LIKE clause in this format ‘%{city}%’ which finds all Listing.city values that have {city} in any position, provided by the request body.

The listing type, number of bedrooms and bathrooms are also (optionally) provided in the search query in order to filter out Listings which don’t match the query using the WHERE clause.

All together the search query looks something like this (JavaScript/SQL):

```
let sqlQuery = `SELECT * FROM Listing WHERE city LIKE ‘%${req.body.cityName}%’ AND
listingType = '${req.body.listingType}' AND numBedrooms = ${req.body.numBedrooms} AND
numBathrooms = ${req.body.numBathrooms}`;
```

API’s:

We have several endpoints provided on our backend service that allow the angular services to make HTTP requests in the form of CRUD operations (create, retrieve, update, delete). They all begin with `/api` so as to not point to the static files in our public directory. Some examples are:

- **POST `/api/search`**
    - Pass in a search query into the request body and retrieve an array of Listing objects that make the given query
- **GET `/api/listing/:id`**
    - Pass in a Listing id to the GET request header to retrieve a single Listing from the database
- **POST `/api/login`**
    - Pass in a username and password request body, the backend queries the DB for the User with the provided username. Then, compare the encrypted password with password from the request body and send a success/failure response back

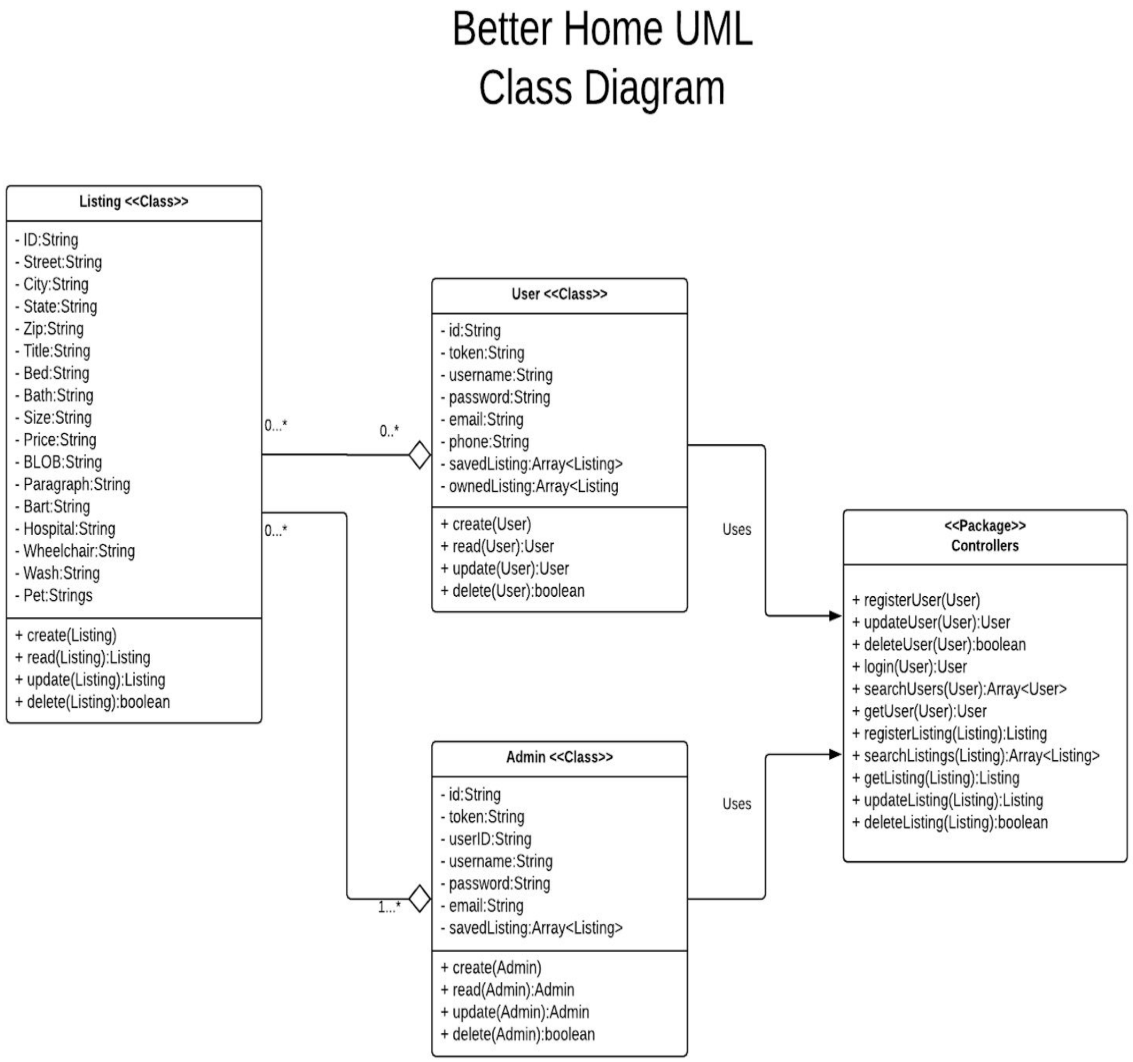
Non-Trivial Algorithm:

No non-trivial algorithm is required nor has been divized as of yet for our project.

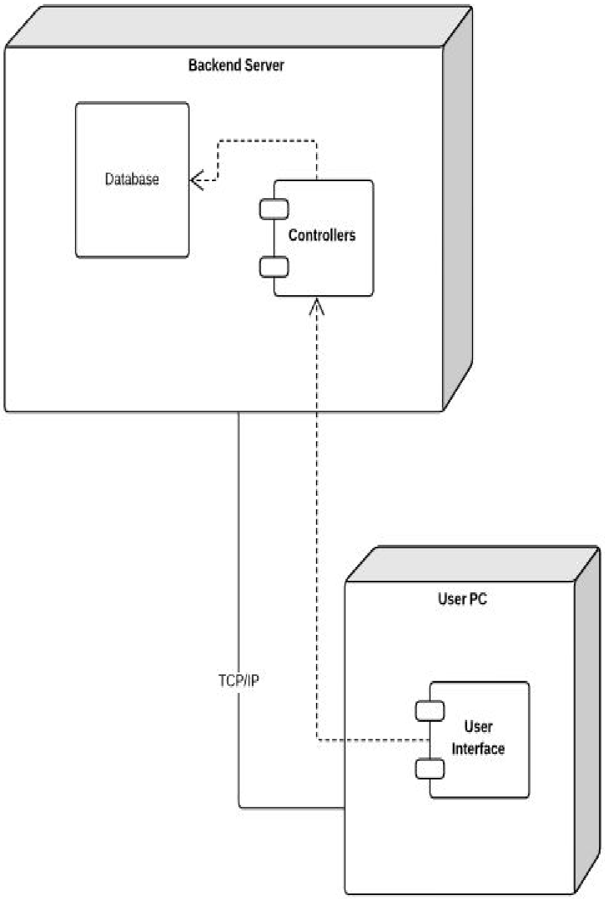
Software Tools and Frameworks:

We have not added any additional SW tools or frameworks for our project.

5. High-level UML Diagrams:



Deployment Diagram



6. Identify actual key risks for your project at this time:

- Skills risks (do you have the right skills):
  - As a whole, the frontend and backend teams are developing the necessary skills in order to take the database and UI mockup design and learn the frameworks in order to implement the requirements in code

- Schedule risks (can you make it given what you committed and the resources):
  - Due dates for the milestones so far have felt tight, however with proper time management and getting started early as we have, we should be in good shape

- Technical risks (any technical unknowns to solve):
  - With the architecture and structure provided in the codebase by the team lead, we don't foresee any legitimate technical issues during development

- Teamwork risks (any issues related to teamwork):
  - The team has well defined roles (frontend / backend) and is operating well on their own, however there could be more time spent to meet up outside of lecture in order to sync up

- Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright):
  - There are risks of copyrighted images being uploaded to our server and how we will address this, however this is the only major hurdle.

7. Project Management:

We have decided to use Trello for agile development and project management. We have split up the work by categorizing them into user stories inside the following sections:

- Defined
- In Progress
- Completed
- Testing
- Accepted

You may view and track the progress of our user stories by visiting our Trello board by clicking the link below.

https://trello.com/b/8fSVt9YJ/csc-648-betterhome-team43-board

Milestone 3

04/26/2019

Table of Contents:

1. Data Definitions V2:

Admin:
User from BetterHome team that can maintain use information and ensure listings posted on our website are valid.

Client (Registered User):
User that has registered with the site, can access all listings on the site, can message the listing owners, and can post their own listings.

Guest User:
User can see the listings on the site without making an account, but cannot message listing owners nor access their contact information.

Privacy policy:
Policy to maintain and protect our user information.

Map View:
A list of all available home/apartment listings in the designated area.

Handicap:
Indicator on listings to show that they have handicap accessible pathways.

Listing:
- Photo:
Some photos to showcase the house/apartment.
- Price/Rate:
A general idea of how much each listing is.
- Address:
Area for where the listing is located.

BART Accessible:
Within a close proximity to a BART Station.

Availability:
Indicator to show that if the listing is available (could signify that an admin should take down the listing OR can just scrap this and have the owner take down the listing themself).

Reseller:
A user that is planning to purchase houses/apartments not for their own use, but to improve and then resell back on the market.

Handicap User:
A user that would like special accommodations with their house: wheelchair accessible (elevator to apartment, ramp) or within a relatively close distance to a hospitable or care unit.

Landlord:
A user that will be posting listings on the website in order to get a buyer's attention.

Rooms:
Amount of rooms available in the property listing.

Parking:
How much parking comes with the property.

Profile:
A place on our website that will show personal information of the user.

Help:
A brief explanation of how our website works.

2. Functional Requirements V2:

1st Priority:

For Admin:

8. Admin shall be able to update/maintain user information.
9. Admin shall be able to delete Landlord and Clients.
10. Admin shall be able to lock and unlock Landlord and Clients.
11. Admin shall be able to find Landlord and Clients.
12. Admin shall be able to view gallery.
13. Admin shall be able to delete photos.
14. Admin shall be able to access username and email of Landlord and Clients.

For Landlord:

12. Landlord shall be able to browse the website without login.
13. New Landlord shall be able to register.
14. Registered Landlord shall be able to post property descriptions.
15. Registered Landlord shall be able to post the location of the property.
16. Registered Landlord shall be able to post the price of the property.
17. Registered Landlord shall be able to post the city and zipcode of the property.
18. Registered Landlord shall be able to select if the property is a house or apartment.
19. Registered Landlord shall be able to select if the property is wheelchair accessible.
20. Registered Landlord shall be able to login.
21. Registered Landlord shall be able to add photos.
22. Registered Landlord shall be able to delete photos.

For Client:

16. New clients shall be able to register.
17. Registered clients shall be able to login.
18. Registered clients shall be able to sign out.
19. Registered clients shall be able to modify their profile.
20. Registered clients shall be able to add the listing as favorite .
21. Registered clients shall be able to revisit the favorite .
22. All clients shall be able to browse the website without login.
23. All clients shall be able to view the sale listing.
24. All clients shall be able to view the sold listing.
25. A free text search box shall be displayed to all clients.
26. All clients shall be able to search the listing by city.
27. All clients shall be able to search the listing by price range.

28. All clients shall be able to search the listing by property type.
29. All clients shall be able to modify the last search.
30. The search results shall be able to be sorted by relevance.

<u>2nd Priority:</u>

For Admin:

4. Admin shall be able to filter data by property type, status, or date posted.
5. Admin shall be able to review flags, and take the appropriate action.
6. Admin shall be able to run data analysis on user data to understand how the application is being used.

For Client:

12. The default display in the free text search box shall be “address, neighborhood, zip”.
13. All clients shall be able to filter the search result by relevance.
14. All clients shall be able to view the search results in a map view based on current location.
15. All clients shall be able to contact the landlord.
16. All clients shall be able to contact the admin.
17. Listings thats are similar to a viewed listing should be displayed to all clients
18. A Q&A page shall be provided for all clients.
19. All handicap users shall be able to search for listings by accessibility.
20. All resellers shall be able to repost the listing onto the site.
21. The application shall keep track of registered user history search.
22. Registered clients shall be given a list of checkbox to select and search (Advanced search).

For Landlord:

7. Registered Landlord shall be able to contact registered clients.
8. Registered Landlord shall be able to view gallery.
9. Registered Landlord shall be able to contact admin.
10. Registered Landlord shall be able to change the property status to available, pending, and sold.
11. Registered Landlord shall be able to update price, location, and pictures.
12. Registered Landlord shall be given a list of checkbox to select if their property has access to wheelchair, BART, parking, etc.

For guest User:

7. Guest users shall be able to register.
8. Guest users shall be able to search listing by city.
9. Guest users shall be able to search listing by price range.
10. Guest users shall be able to search listing by property type.
11. Guest users shall be able to modify the last search.
12. Guest users should be able to view the search results in a map view based on current location.

For Apartment Searcher:

7. Apartment searcher shall be able to search apartments only available for rent
8. Apartment searcher shall be able to search the apartment by number of bedrooms
9. Apartment searcher shall be able to search the apartment by monthly payment.
10. Apartment searcher shall be able to sort by the results by relevance.
11. Apartment searcher shall be able to filter the results by relevance.
12. Apartment searcher shall be able to view the apartment features.

3rd Priority:

For handicap:

6. Handicap users shall be able to flag their profile, denoting a need for handicap-specific accessibilities.
7. Handicap users shall be able to specify in their profile the exact nature of their accessibility needs.
8. Handicap users shall be able to specify in their search that they require a ramp.
9. Handicap users shall be able to specify in their search that they require a ground floor unit.
10. Handicap users shall be able to search the listing by number of floors.

For reseller:

4. Resellers shall be able to access the original post for the property they purchased.
5. Resellers shall be able to modify and repost the original listing with updated information.
6. Original listing shall be linked to new listing so other users may view and verify improvements and/or alterations to property.

Non-Functional Requirements:

Security:

6. Login shall be required for Clients and Admins.
7. Username shall be the Client's email.
8. Password shall be encrypted when stored.
9. Client's session shall end upon leaving the site.
10. Client's session shall only end by code design.

Performance:

2. Loading time for site shall be less than 3 seconds for any screen.

Capacity:

4. The total data storage allowed by the web site shall not exceed of 80 % of the server capacity for this site.
5. The web site shall be prepared to support scalability for adding future new features.
6. The web site shall be capable to handle at least 50 Clients simultaneously.

Reliability:

4. Downtime for maintenance shall be less than 3 hours per month.
5. Downtime for maintenance shall not affect the main functionality of the site.
6. In all cases, downtime for maintenance shall be informed to the Client through email.

Recovery:

3. In a total failure case, the whole site should be put down to revision.
4. If broken, the mean time to recovery shall not exceed one day.

Data Integrity:

6. Database tables shall be backed up every day.
7. Administrator shall be able to execute a recovery when needed.
8. Image Sizes shall be limited up to 1 megabyte.
9. Images shall be uploaded in correct format (jpg, jpeg, or pdf) to the server.
10. Links to images on the server shall be uploaded to the database.

Compatibility:

8. The site shall be compatible with the last version of Microsoft Edge browser (44.17763).
9. The site shall be compatible with the last version of Safari browser (12).

10. The site shall be compatible with the last version of Firefox browser (65.0.2).
11. The site shall be compatible with the last version of Chrome browser (7.3).
12. Third party applications shall not be able to modify any content that may affect the site compatibility.
13. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions.
14. The site should be compatible to escalate to new relational databases.

Conformance with Coding Standards:

9. Architecture and design standards shall meet all the requirements listed under the High Level Architecture section of this document.
10. Only working code that meets all the code standards shall be submitted to the project repository.
11. Any working code shall be tested and debugged before being considered working code.
12. Any internal errors or exceptions returned by the code shall be stored in a log.
13. Any error that may affect the functionality of the site shall be reported to the Client.
14. Any error shall be handled in a way that does not affect the functionality of the site.
15. The whole production cycle of this site shall be finished 2 weeks before the delivery date.
16. This site shall not be launched without all the priority one features completed and tested.

Look and Feel Standards:

8. The application and its layouts shall look professional.
9. The site shall be simple enough to handle by all the parties involved.
10. Elements on screen shall have the correct density to meet the compatibility standard of the browsers.
11. Elements on screen shall have rich and beautiful colors for Client delight.
12. The site shall be able to work correctly without mouse interaction.
13. The site shall be able to work correctively without keyboard interaction.
14. Elements in screen shall be resized automatically without Client interaction when being loaded in all the different platforms supported by the site.

Internationalization / Localization Requirements:

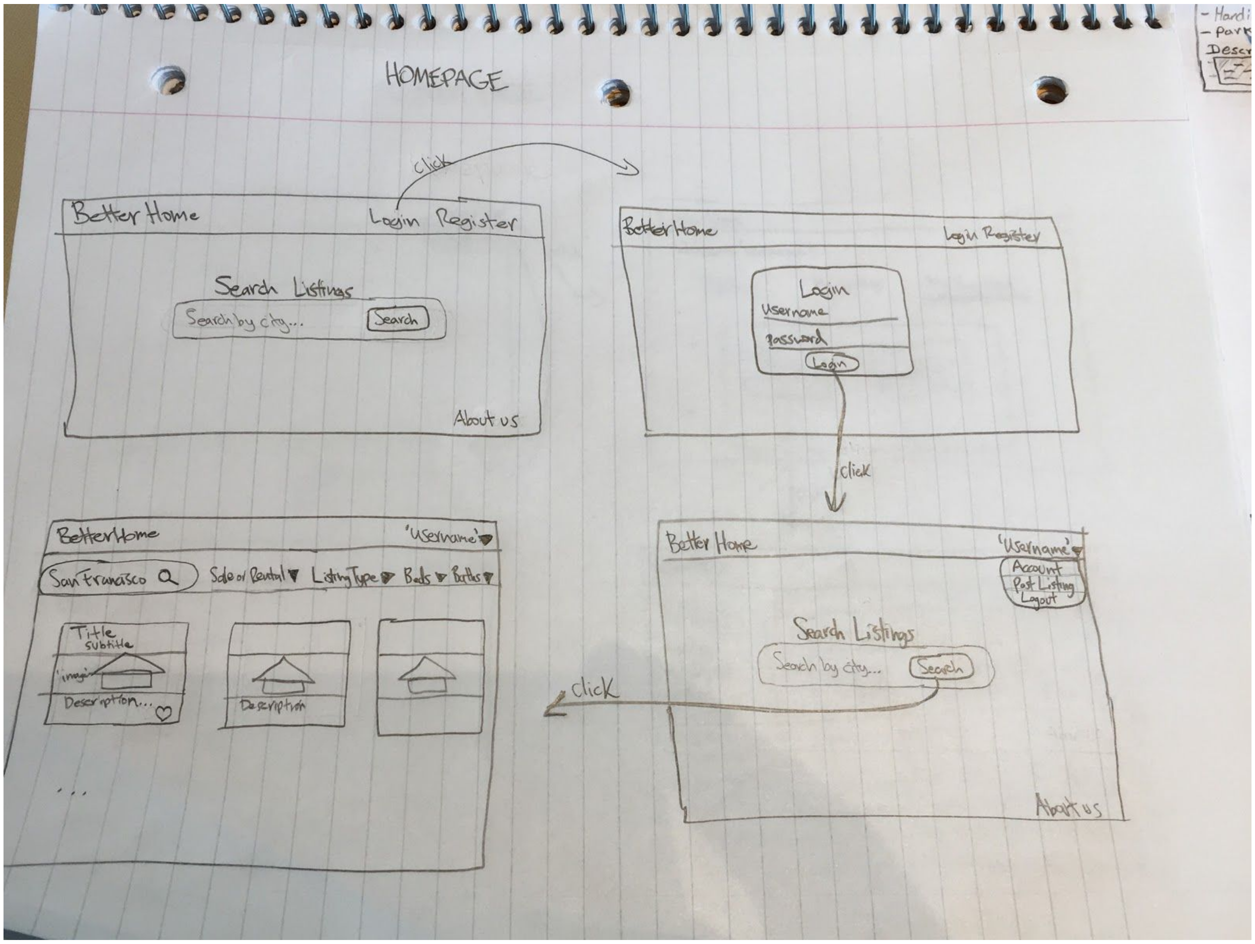
5. Default language shall be English.
6. The site shall support scalability to add other languages.
7. The site shall support geolocation in order to show listing locations.
8. Any copyrighted material shall be immediately be taken down upon reception of an official DMCA takedown request.

Website Policies:

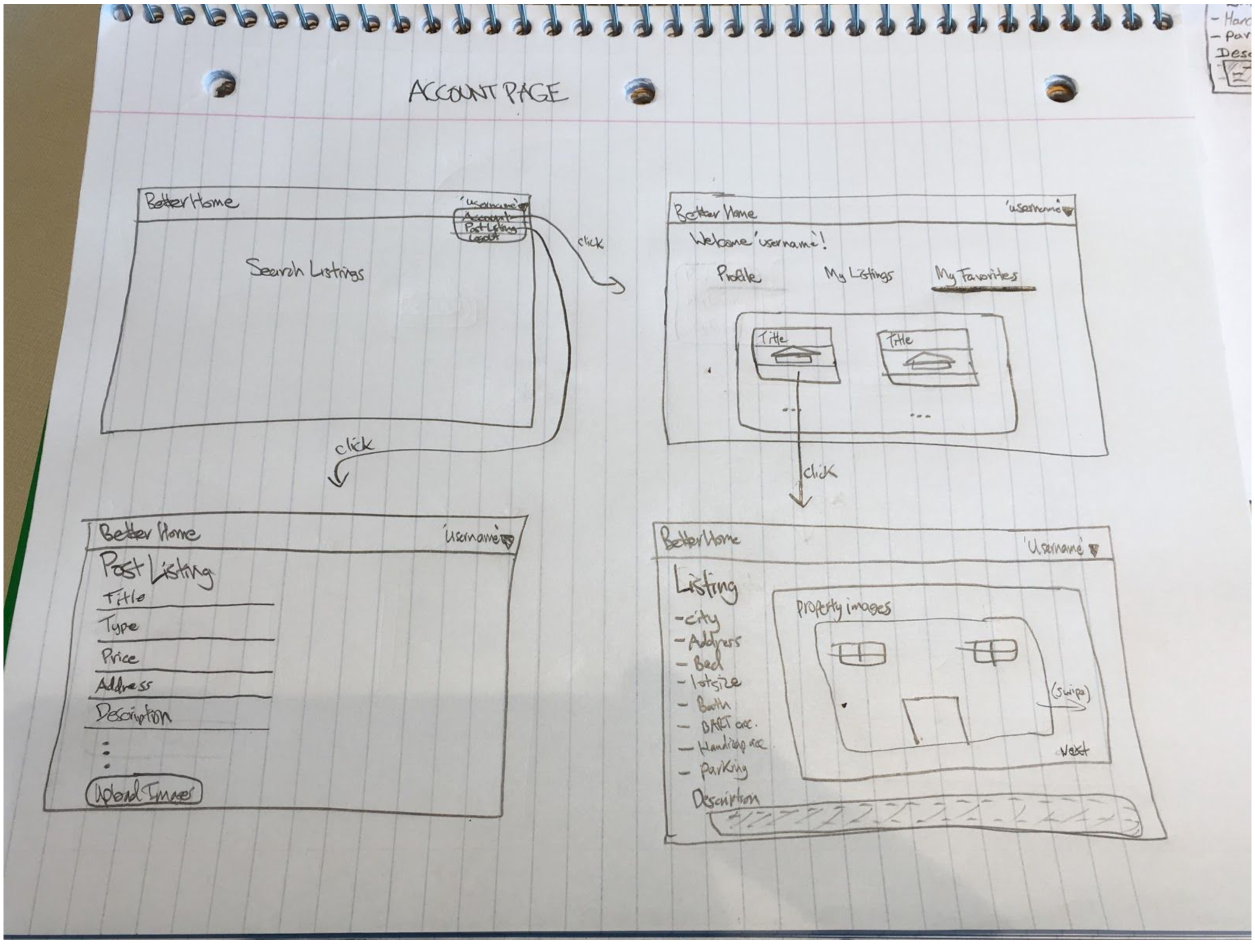
5. A link to the policies of this site shall be always visible in all its pages to be accessible by all the parties.
6. Clients’ data shall not be sold to third parties.
7. Clients and Landlords data that do not add any functionality to the system shall not be collected.
8. Clients that post inappropriate listings(false listings/ copyright images/ sexual images) shall have their postings taken down.

3. <u>UI Mockups and Storyboards</u>

Login flow:



Account Flow:

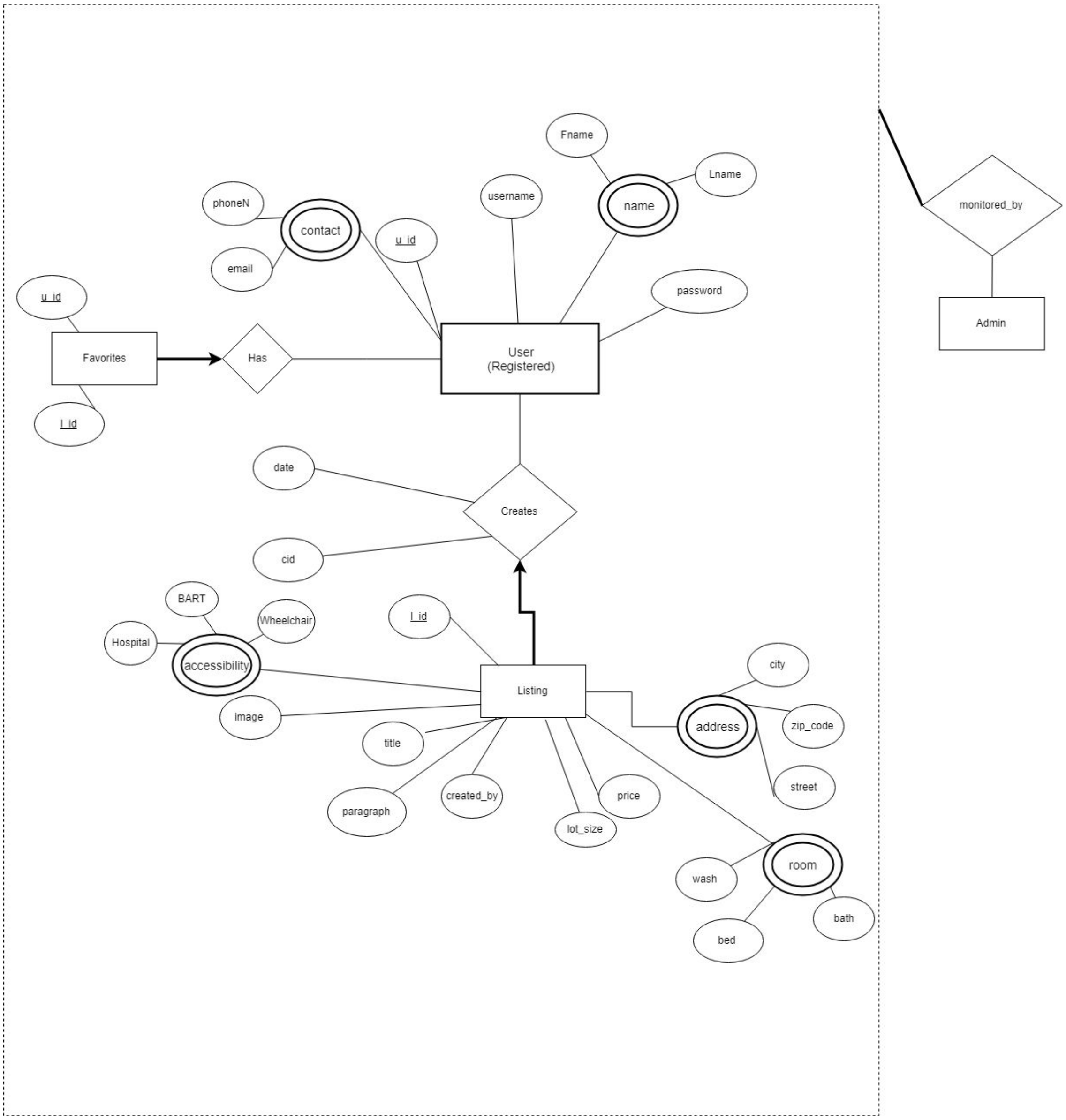


4. High-level Architecture, Database Organization:

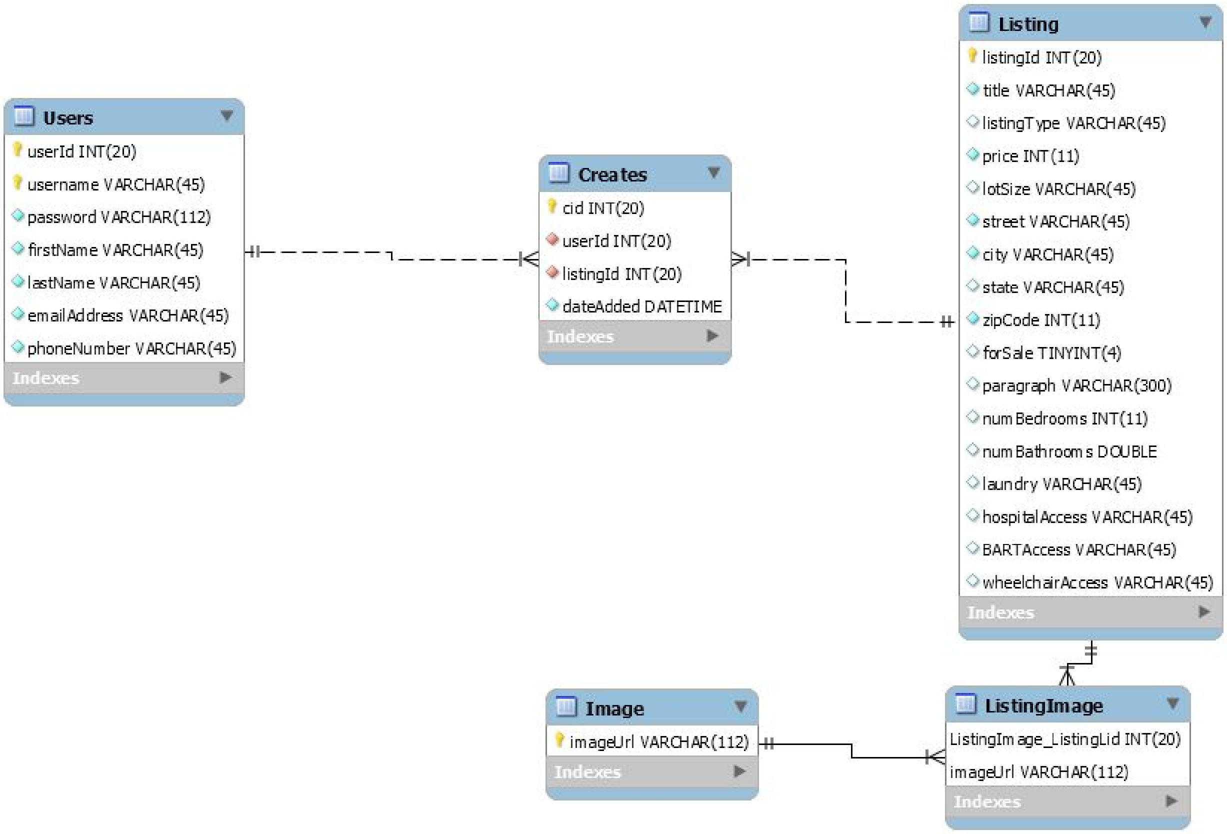
Business Rules:

6. Users can create multiple listings on the website.
7. Users can view multiple listings with a specific amount of rooms.
8. A listing must have only one description describing its characteristics.
9. A listing can be displayed on multiple listing boards.
10. Admins must monitor all listings posted by all users.

ERD Diagram:



DB Diagram:


Users
userId INT(20)
username VARCHAR(45)
password VARCHAR(112)
firstName VARCHAR(45)
lastName VARCHAR(45)
emailAddress VARCHAR(45)
phoneNumber VARCHAR(45)
Indexes
Creates
cid INT(20)
userId INT(20)
listingId INT(20)
dateAdded DATETIME
Indexes
Listing
listingId INT(20)
title VARCHAR(45)
listingType VARCHAR(45)
price INT(11)
lotSize VARCHAR(45)
street VARCHAR(45)
city VARCHAR(45)
state VARCHAR(45)
zipCode INT(11)
forSale TINYINT(4)
paragraph VARCHAR(300)
numBedrooms INT(11)
numBathrooms DOUBLE
laundry VARCHAR(45)
hospitalAccess VARCHAR(45)
BARTAccess VARCHAR(45)
wheelchairAccess VARCHAR(45)
Indexes
Image
imageUrl VARCHAR(112)
Indexes
ListingImage
ListingImage_ListingLid INT(20)
imageUrl VARCHAR(112)
Indexes

DBMS:

We chose MySQL as our Database Management System since it provides great support alongside MySQLWorkbench and the workflow is very intuitive and efficient.

Media Storage:

Images (.jpeg, .png, .tiff, .bmp) will be the only form files that the user can input into the database. They will be stored as URLs on the "Image" table and be a primary key in "ListingImage" that will connect it to the "Description" table.

Search/Filter Architecture and Implementation:

Our search functionality requires a city name in the free text search box from the frontend. We utilize the SQL LIKE clause in this format '%{city}%' which finds all Listing.city values that have {city} in any position, provided by the request body.

The listing type, number of bedrooms and bathrooms are also (optionally) provided in the search query in order to filter out Listings which don't match the query using the WHERE clause.

All together the search query looks something like this (JavaScript/SQL):

```
let sqlQuery = `SELECT * FROM Listing WHERE city LIKE '%${req.body.cityName}%' AND
listingType = '${req.body.listingType}' AND numBedrooms = ${req.body.numBedrooms} AND
numBathrooms = ${req.body.numBathrooms}`;
```

API's:

We have several endpoints provided on our backend service that allow the angular services to make HTTP requests in the form of CRUD operations (create, retrieve, update, delete). They all begin with `/api` so as to not point to the static files in our public directory. Some examples are:

- **POST `/api/search`**
    - Pass in a search query into the request body and retrieve an array of Listing objects that make the given query
- **GET `/api/listing/:id`**
    - Pass in a Listing id to the GET request header to retrieve a single Listing from the database
- **POST `/api/login`**
    - Pass in a username and password request body, the backend queries the DB for the User with the provided username. Then, compare the encrypted password with password from the request body and send a success/failure response back

Non-Trivial Algorithm:

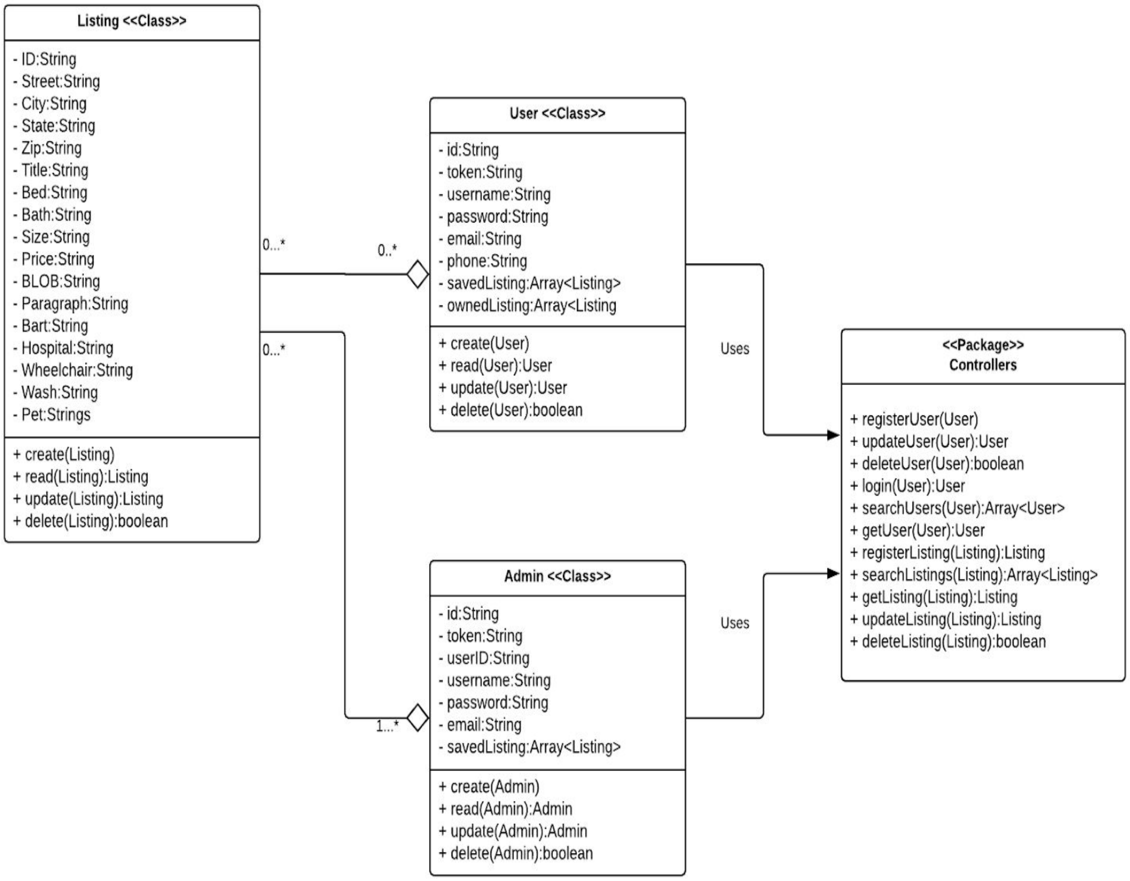
No non-trivial algorithm is required nor has been divized as of yet for our project.

Software Tools and Frameworks:

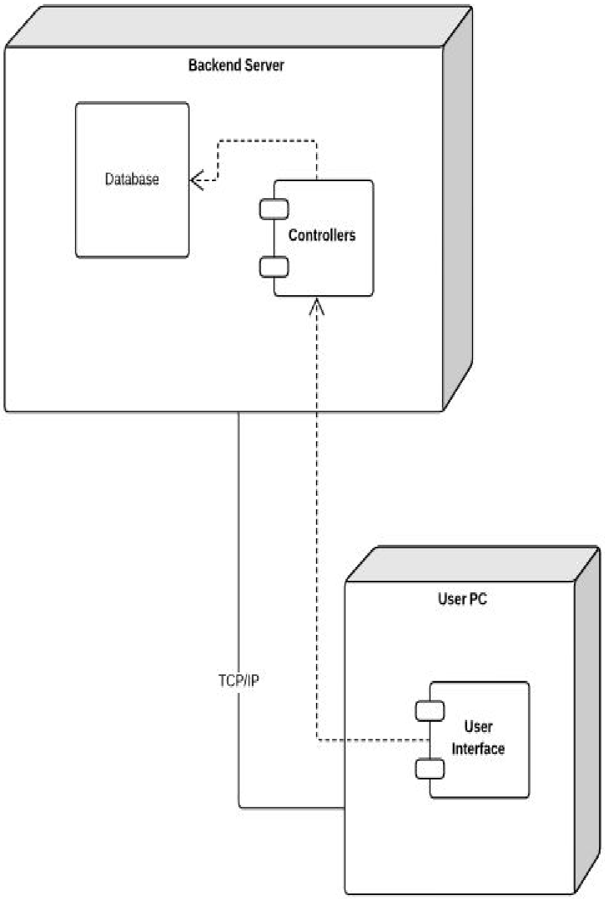
We have not added any additional SW tools or frameworks for our project.

5. High-level UML Diagrams:

Better Home UML
Class Diagram



Deployment Diagram



6. Identify actual key risks for your project at this time:

- Skills risks (do you have the right skills):
  - As a whole, the frontend and backend teams are developing the necessary skills in order to take the database and UI mockup design and learn the frameworks in order to implement the requirements in code

- Schedule risks (can you make it given what you committed and the resources):
  - Due dates for the milestones so far have felt tight, however with proper time management and getting started early as we have, we should be in good shape

- Technical risks (any technical unknowns to solve):
  - With the architecture and structure provided in the codebase by the team lead, we don't foresee any legitimate technical issues during development

- Teamwork risks (any issues related to teamwork):
  - The team has well defined roles (frontend / backend) and is operating well on their own, however there could be more time spent to meet up outside of lecture in order to sync up

- Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright):
  - There are risks of copyrighted images being uploaded to our server and how we will address this, however this is the only major hurdle.

7. **Project Management:**

We have decided to use Trello for agile development and project management. We have split up the work by categorizing them into user stories inside the following sections:

- Defined
- In Progress
- Completed
- Testing
- Accepted

You may view and track the progress of our user stories by visiting our Trello board by clicking the link below.

https://trello.com/b/8fSVt9YJ/csc-648-betterhome-team43-board

8. M3 Feedback:

Search results page:

- different search options should come from the DB
- add number of results/pages to top of page
- make it obvious that you can click on one of the property cards
- add some sort of highlight/underline/drop shadow

Post listing page:

- amount of information necessary might be too much
- use fewer options to create property, and user can go in and add extra options after the fact
- cannot see the submit button, needs to be relocated

Home page:

- Page feels too empty, needs more links on top or maybe a logo
- (Maybe add a list of recently added properties?)

Admin page:

- Need to create the page since it’s in priority 1 requirements!

Milestone 4

05/12/2019

Table of Contents:

1. Product Summary:

Our product named “BetterHome” will be the leading property marketplace dedicated to providing consumers with services they need to buy, sell, or rent the place they call home. BetterHome will provide many services to users of all kinds, including landlord hosts, apartment / real-estate searchers and resellers. BetterHome will give property owners the ability to host their property on the website, and provide all general and customized information needed to make their property an attractive candidate for someone’s next home. The website will also provide this same service to “house-flippers,” or resellers, and will allow them to browse all listings with filters, all while being able to host a property up on the website and maintain each aspect simultaneously. What makes BetterHome a unique product is our incredibly detailed advanced search which allows users to browse the various property listings on the website with feature rich and detailed customization of search options. This includes accessibility filters such as laundry and wheelchair access, and hospital / BART proximity that other property marketplaces don’t possess.

URL:

https://better-home-234220.appspot.com

Priority 1 Committed Functions:

For Admins:

(Admin username: **admin**
Admin password: **better-home-admin**)

1. Admins are able to update and maintain user information in the case where users forget their password or if there is inappropriate content on their profile.
2. Admins are able to delete user accounts who host properties (landlords), as well as users who search for property listings (clients).
3. Admins are able to lock accounts for both landlord and clients, as well as unlock their accounts.
4. Admins are able to find registered landlord and clients on the users page by their username, full name, email address, etc.
5. Admins are able to view a gallery collection of property listing images on the images page.
6. Admins are able to delete images on the images page.
7. Admins are able to access username and email for all users on the users page.

For Landlords:

1. Landlords are able to browse and experience all regular features of the website without having to login.
2. Soon to be registered landlords are able to register on the registration page.
3. Registered landlords are able to post their own property with all the information about their property.
4. Registered landlords are able to update the location of their hosted property.
5. Registered landlords are able to update the price of their hosted property.
6. Registered landlords are able to update the city and zipcode of their hosted property.
7. Registered landlords are able to select if their property is a house or apartment.
8. Registered landlords are able to select if their property is wheelchair accessible.
9. Registered landlords are able to login and logout at any and all times.
10. Registered landlords are able to add photos of their newly created listings, as well as adding new photos to their already hosted listing.
11. Registered landlords are able to delete photos of their already hosted property.

For Clients:

1. Soon to be clients are able to create new accounts.
2. Registered clients are able to login and logout whenever they see fit.
3. Registered clients are be able to modify all information on their profile, including email address, phone number, etc.
4. Registered clients are able to add properties they view to their list of favorite properties.
5. Registered clients are able to revisit any and all of their favorite properties.
6. All clients are able to browse the website and search for property listings without having to login.
7. All clients are able to view listings that are currently for sale.
8. All clients are able to see listings that have been recently bought or rented out.
9. A basic search and advanced search box is displayed on the home page to all clients.
10. All clients are able to search property listings by the city they want to view.
11. All clients are able to search listings by a minimum and maximum price range.
12. All clients are able to search listings by the type of property, including houses, apartments, and condos.
13. All clients are able to change the search options they last entered on the search results page.

2. Usability test plan:

Software usability testing is a key methodology that ensures applications are intuitive and easy to use for the target audience. Usability testing has direct benefits for companies as usability improvements often are fundamental to the success of a product. A standard usability test includes the following five steps: obtain suitable participants, design test scripts, conduct usability sessions, interpret test outcomes, and produce recommendations.The main objective of Usability Testing is to identifies usability errors in our BetterHome website in its early development cycle and to save our website from failure. For our BetterHome website, we will test the usability of the Advanced Search functionality. The objective of the Advanced Search usability testing include establishing a baseline of user performance in using the Advanced Search and identifying potential design concerns to be addressed in order to improve the efficiency, productivity, and end-user satisfaction in the Advanced Search functionality.

Test objectives:

- To verify that BetterHome website is intuitive to the average user that would be likely to search for Apartments.
- To check how the navigation back and forward in a search impacted the result.
- Establish baseline user performance and user-satisfaction levels of the user interface and search functionality for future usability evaluations.
- To check the total number of results that displayed on the search result page

Problem statement:

Is the Advanced Search function of BetterHome website easy to use?

Potential sources of error may include:

- Advanced Search result errors – failure to display the correct search result and if a keyword is typed incorrectly, then the relevant result message should be displayed

- Presentation errors – failure to locate and properly act upon desired information in the search result screens.

Test Description and System setup:

The participants' responsibilities will be to attempt to complete a set of representative task scenarios presented to them in as efficient and timely manner as possible, and to provide feedback regarding the usability and acceptability of the search functionality. The participants will be directed to provide honest opinions regarding the usability of the website search usability, and to participate in post-session subjective questionnaires and debriefing.

Participants will take part in the usability test at San Francisco State University, Business building Room number 217. A personal laptop with the BetterHome Website will be used in a typical school class environment to test the Advanced Search functionality. The participant's interaction with the Website will be monitored by the facilitator seated in the classroom. Note takers and data logger(s) will monitor the sessions.

The facilitator will brief the participants on the Web site and instruct the participant that they are evaluating the website. Participants will sign an informed consent that acknowledges: the participation is voluntary, that participation can cease at any time. The facilitator will ask the participant if they have any questions. After each task, the participant will complete the post-task questionnaire and elaborate on the task session with the facilitator. After all task scenarios are attempted, the participant will complete the post-test satisfaction questionnaire.

Hardware and software setup:

- Hardware Setup – Website on Google Cloud Platform running on Linux Machine
- Software Setup – BetterHome homepage (logged out) available on Chrome 72 browser of Ubuntu 16.04.5. The MySQL database currently has 7 property listings.

URL: **https://better-home-234220.appspot.com**

Legal issues: This test is voluntary, no identifying information will be kept, only experience with website. Any personal information entered upon registration will be deleted at the completion of the project.

Report: Will contain information based on how easy the users found it to navigate and operate BetterHome

Usability Task Table:

Task	**% Completed**	**Error**	**Comments**
Search Apartments	100	None	None
Search Apartment by Listing Type	100	None	None
Search apartment by City	100	None	None
Search apartment by Bedrooms	100	None	None
Advanced Search box appear on mobile or Tablet	100	None	None
Advanced Search option exclude for not available Apartments	100	None	None

Task:

Find apartment for rent in San francisco that has two bedroom using the Advanced Search.

Task	Search apartment in San Francisco
Machine State	Home page of BetterHome https://better-home-234220.appspot.com
Success Criteria	Three Search results displayed

Questionnaire:

	Agree **Disagree**
The Advanced Search feature was easy to be found	○ ○ ○ ○ ○ ○ ○ ○ ○ ○
You find the Advanced Search interface intuitive.	○ ○ ○ ○ ○ ○ ○ ○ ○ ○
The Advanced Search process was quick and simple.	○ ○ ○ ○ ○ ○ ○ ○ ○ ○
The Advanced Search option didn't ask for many information.	○ ○ ○ ○ ○ ○ ○ ○ ○ ○
It was clear how to select the search options.	○ ○ ○ ○ ○ ○ ○ ○ ○ ○

Scenarios for testing Advanced Search functionalities:

1. Advanced Search results displayed should be relevant to search keyword
2. % sign in search keyword should not redirect to 404 ERROR
3. Application should not crash if user inserted % in search field
4. When user start typing word in text box it should suggest words that matches typed keyword
5. There should be pre-defined search criteria for auto complete e.g. after typing first 3 letter it should suggest matching keyword
6. When user clicks on any link from result and navigates back, then result should be maintained
7. After clicking Search field - search history should be displayed (latest search keyword)
8. Search results should be cleared on clicking clear search button
9. History displayed in search field should be relevant to logged in user only
10. Total number of search records/results should be displayed on page
11. Search keyword should suggest similar kind of properties
12. For Advanced Search - limited search filters should be provided
13. Validate search rules defined for "Exact Match" with the search key word
14. User should be able to search when he/she enters the keyword and hits 'Enter' button on keyboard

3. QA test plan:

Test objectives:

- Purpose: Quality Assurance testing is performed to validate that system functionalities are per the requirements. Testing is performed to validate that all functionalities work according the specifications mentioned in previous phases.

- Problem statement: To verify that BetterHome's 'Advanced Search' feature returns proper listings according to the parameters specified.

Test plan:

The user will follow the test cases below to search for listings that satisfy certain characteristics of the property. They will complete this search without logging in or registering for an account.

- Hardware and software setup:

  Hardware Setup – Website on Google Cloud Platform running on Linux Machine
  Software Setup – BetterHome homepage (logged out) available on Chrome 72 browser of Ubuntu 16.04.5. The MySQL database currently has 7 property listings.

Features to be tested:

Listings can be returned by entering at least the city in which to search, and can be further specified to include available amenities in the options located in the Advanced Search bar. The search functionality is to be tested through the 3 test cases below:

1. Search for a listing in San Francisco with access to Bart using the search bar
2. Search for a listing in San Francisco with wheelchair access and onsite laundry access using the search bar
3. Search for a listing in Daly City using the search bar

Test Cases:

Test #	**Description**	**Test Input**	**Expected Output**	**Pass/Fail**
1	Search for a listing in San Francisco with access to Bart using the search bar	Select ‘Advanced Search’. Select Accessibility - ‘BART’ Enter ‘San Francisco’ into search by city	Property Listings page appears and displays the one (1) property that matches these parameters.	PASS
2	Search for a listing in San Francisco with wheelchair access and onsite laundry access using the search bar	Select ‘Advanced Search’. Select Accessibility - ‘laundry’ Enter ‘San Francisco’ into search by city	Property Listings page appears and displays the three (3) properties that match these parameters.	PASS
3	Search for a listing in Oakland using the search bar	Enter ‘Daly City’ into search by city	A popup is displayed that reads: “Unable to retrieve any listing based on your search and filter options. Please try again”	PASS

4. Code Review:

Code Style of HTML:

1. Use lowercase for element names
2. Use lowercase letters in attribute names
3. Use double quote ("") around attribute values
4. Close all elements
5. Do not add blank lines without a reason
6. Provide alternative contents for multimedia
7. Remove spaces around equal signs
8. Two spaces of indentation instead of using tab key
9. When line-wrapping, each continuation line should be indented at least 4 additional spaces from the original line.
10. Use a new line for every element

Code Style of CSS:

1. Use shorthand properties where possible
2. Do not use units after 0 values unless they are required
3. Alphabetize declarations
4. Use a semicolon after every declaration
5. Undent all block content
6. Separate rules by new lines and put a blank line between rules.

Code Style of Typescript:

1. Use camelCase for variable and function names
2. Use PascalCase for class names
3. Use camelCase of class members and methods
4. Use PascalCase for type name.
5. Use camelCase for type members
6. Use single quotes (') unless escaping
7. Use 2 spaces for indentation. Not tabs
8. Space before type
9. Use semicolons
10. Annotate arrays as foos: Foo[] instead of foos:Array<Foo>.
11. Name files with camelCase
12. Put else on a separate line from the closing curly brace
13. Put the opening curly braces on the same line.
14. Do suffix a service class name with Service.
15. Name test specification files with a suffix of .spec

Code Style of Javascript:

1. Declare variable with let or const (no vars)
2. Always use semicolons
3. For name: functionNamesLikeThis, variableNamesLikeThis,ClassNamesLikeThis, EnumNamesLikeThis, methodNamesLikeThis,CONSTANT_VALUES_LIKE_THIS, foo.namespaceNamesLikeThis.bar, andfilenameslikethis.js.
4. Use Array and Object literals instead of Array and Object constructors.
5. Follow C++ formatting rules (https://google.github.io/styleguide/cppguide.html#Formatting)
6. Use parentheses only where required
7. Strings should use ‘ over “ but either is acceptable
8. Encouraged, use JSDoc annotations @private and @protected

Code Review from Team 6 (Reviewer - Huy Nguyen):

Tuesday, May 7, 2019, 4:58 PM -0700 from Liwang Gao <lgao2@mail.sfsu.edu>:

Hi Huy,

The below is a piece of the front-end code for the search function which is written in typescript, please take a look and give us some feedback/code review. Thank you!

```
import {Component, OnDestroy, OnInit} from '@angular/core';
import {Router} from "@angular/router";
import {ListingSearch, SearchListingsService} from "../core/services/search.listings.service";
import {Listing} from "../core/services/listings.service";
import {MatDialog} from "@angular/material";
import {RegisterDialog} from "../register/register.dialog";
import {FormControl} from "@angular/forms";

@Component({
  selector: 'app-advanced-search',
  templateUrl: './advanced-search.component.html',
  styleUrls: ['./advanced-search.component.css']
})
export class AdvancedSearchComponent implements OnInit, OnDestroy {

  listingSearch: ListingSearch;
  listings: Listing[];

  accessibilities = new FormControl();
  accessibilityList: string[] = ['Laundry', 'Hospital', 'Wheelchair', 'BART'];

  isLoaded = true;
```

```
constructor(
  private router: Router,
  private searchService: SearchListingsService,
  public dialog: MatDialog
) {}

ngOnInit() {
  if (localStorage.getItem('listingSearch')) {
    this.listingSearch = JSON.parse(localStorage.getItem('listingSearch'));
  }
  else {
    this.listingSearch = {
      city: ''
    }
  }
}

ngOnDestroy() {
  if (this.listingSearch) {
    localStorage.setItem('listingSearch', JSON.stringify(this.listingSearch));
  }
}

onAccessibilityChange() {
  this.listingSearch.accessibilities = this.accessibilities.value;
}

onSearchClick() {
  if (!this.listingSearch.city.length) {
    this.openDialog('Please enter some text for the city field');
  }
  else {
    this.isLoaded = false;
    this.searchService.getSearchListings(this.listingSearch)
      .subscribe(listings => {
          this.isLoaded = true;
          this.searchService.saveSearchListings(listings);
          this.router.navigate(['/properties']);
        },
        err => {
          this.isLoaded = true;
          this.openDialog('Unable to retrieve any listing based on your search. Please try again');
        });
    }
  }

  numberOnly(event): boolean {
    const charCode = (event.which) ? event.which : event.keyCode;
    return !(charCode > 31 && (charCode < 48 || charCode > 57));
  }

  openDialog(message: string, subscribe: boolean = false) {
    const dialog = this.dialog.open(RegisterDialog, {
      width: '250px',
      data: {
        message: message
      }
    });
    if (subscribe) {
      dialog.afterClosed().subscribe(result => {
        this.router.navigate(['/properties']);
      });
    }
  }
}
```

Huy Nguyen <huy9997@gmail.com> Today, 2:55 PM Liwang Gao
Hi Liwang, I did the basic code review below. It was good for the most part.
I think their is nothing wrong with your code as far as I can see. I only only work on making this more moduler.
That means reach function such as openDialog should be its own file. I think that would make it cleaner.
Then you connect them all to a index.js
overall good code though
Example:
Folder Search:
file openDialog
file numberOnly
file onSearchClick
file onAccessibilityChange
file ngOnDestroy
file ngOnInit
file index.js

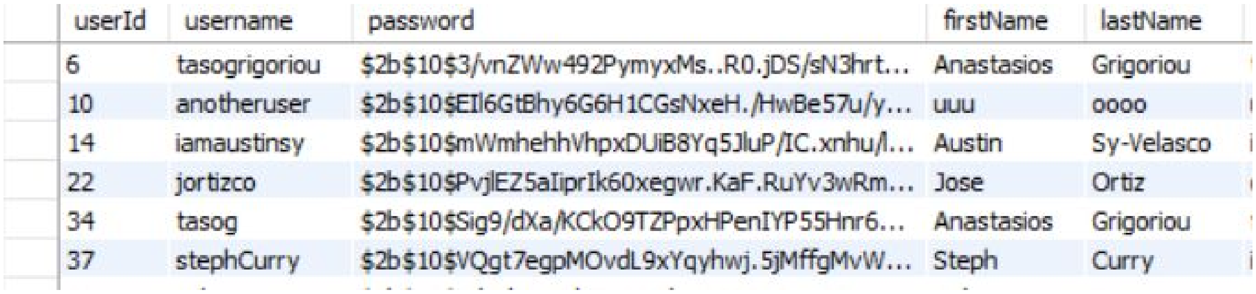
5. Self-check on best practices for security:

Major Assets to Protect:

1. Passwords

Confirmation of Password encryption:

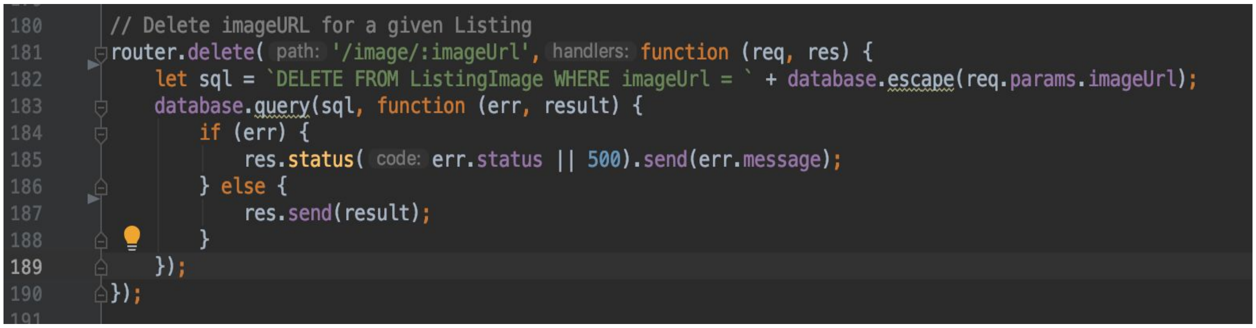
Passwords are encrypted using the bcrypt library upon registration of the user. This password is validated using the bcrypt compare function upon logging in. Here is an example of the encoded passwords in the database.



userId	username	password	firstName	lastName
6	tasogrigoriou	$2b$10$3/vnZWw492PymyxMs..R0.jDS/sN3hrt...	Anastasios	Grigoriou
10	anotheruser	$2b$10$EIl6GtBhy6G6H1CGsNxeH./HwBe57u/y...	uuu	oooo
14	iamaustinsy	$2b$10$mWmhehhVhpxDUiB8Yq5JluP/IC.xnhu/l...	Austin	Sy-Velasco
22	jortizco	$2b$10$PvjlEZ5aIiprIk60xegwr.KaF.RuYv3wRm...	Jose	Ortiz
34	tasog	$2b$10$Sig9/dXa/KCkO9TZPpxHPenIYP55Hnr6...	Anastasios	Grigoriou
37	stephCurry	$2b$10$VQgt7egpMOvdL9xYqyhwj.5jMffgMvW...	Steph	Curry

Confirm Input Data Validation:

In order to validate all inputs we use escaping on all data that comes from the front end. By tokenizing all input we prevent malicious sql injection through our api. Below is a snippet as an example from our delete image function.



```
// Delete imageURL for a given Listing
router.delete( path: '/image/:imageUrl', handlers: function (req, res) {
    let sql = `DELETE FROM ListingImage WHERE imageUrl = ` + database.escape(req.params.imageUrl);
    database.query(sql, function (err, result) {
        if (err) {
            res.status( code: err.status || 500).send(err.message);
        } else {
            res.send(result);
        }
    });
});
```

6. Self-Check: Adherence to original Non-Functional Specs:

Security:

1. Login shall be required for Clients and Admins. **DONE**
2. Username shall be the Client's email. **DONE**
3. Password shall be encrypted when stored. **DONE**
4. Client's session shall end upon leaving the site. **ISSUE - WE CACHE USER DATA**
5. Client's session shall only end by code design. **ISSUE - WE CACHE USER DATA**

Performance:

1. Loading time for site shall be less than 3 seconds for any screen. **DONE**

Capacity:

1. The total data storage allowed by the web site shall not exceed of 80 % of the server capacity for this site. **DONE**
2. The web site shall be prepared to support scalability for adding future new features. **DONE**
3. The web site shall be capable to handle at least 50 Clients simultaneously. **DONE**

Reliability:

1. Downtime for maintenance shall be less than 3 hours per month. **DONE**
2. Downtime for maintenance shall not affect the main functionality of the site. **DONE**
3. In all cases, downtime for maintenance shall be informed to the Client through email. **DONE**

Recovery:

1. In a total failure case, the whole site should be put down to revision. **DONE**
2. If broken, the mean time to recovery shall not exceed one day. **DONE**

Data Integrity:

1. Database tables shall be backed up every day. **DONE**
2. Administrator shall be able to execute a recovery when needed. **DONE**
3. Image Sizes shall be limited up to 1 megabyte. **DONE**
4. Images shall be uploaded in correct format (jpg, jpeg, or pdf) to the server. **DONE**
5. Links to images on the server shall be uploaded to the database. **DONE**

Compatibility:

1. The site shall be compatible with the last version of Microsoft Edge browser (44.17763). **DONE**
2. The site shall be compatible with the last version of Safari browser (12). **DONE**
3. The site shall be compatible with the last version of Firefox browser (65.0.2). **DONE**
4. The site shall be compatible with the last version of Chrome browser (7.3). **DONE**

5. Third party applications shall not be able to modify any content that may affect the site Compatibility. **DONE**
6. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions. **DONE**
7. The site should be compatible to escalate to new relational databases. **DONE**

Conformance with Coding Standards:
1. Architecture and design standards shall meet all the requirements listed under the High Level Architecture section of M1 document. **DONE**
2. Only working code that meets all the code standards shall be submitted to the project Repository. **DONE**
3. Any working code shall be tested and debugged before being considered working code. **DONE**
4. Any internal errors or exceptions returned by the code shall be stored in a log. **DONE**
5. Any error that may affect the functionality of the site shall be reported to the Client. **DONE**
6. Any error shall be handled in a way that does not affect the functionality of the site. **DONE**
7. The whole production cycle of this site shall be finished 2 weeks before the delivery date. **ON TRACK**
8. This site shall not be launched without all the priority one features completed and tested. **ON TRACK**

Look and Feel Standards:
1. The application and its layouts shall look professional. **DONE**
2. The site shall be simple enough to handle by all the parties involved. **DONE**
3. Elements on screen shall have the correct density to meet the compatibility standard of the browsers. **DONE**
4. Elements on screen shall have rich and beautiful colors for Client delight. **DONE**
5. The site shall be able to work correctly without mouse interaction. **ON TRACK**
6. The site shall be able to work correctively without keyboard interaction. **ON TRACK**
7. Elements in screen shall be resized automatically without Client interaction when being loaded in all the different platforms supported by the site. **DONE**

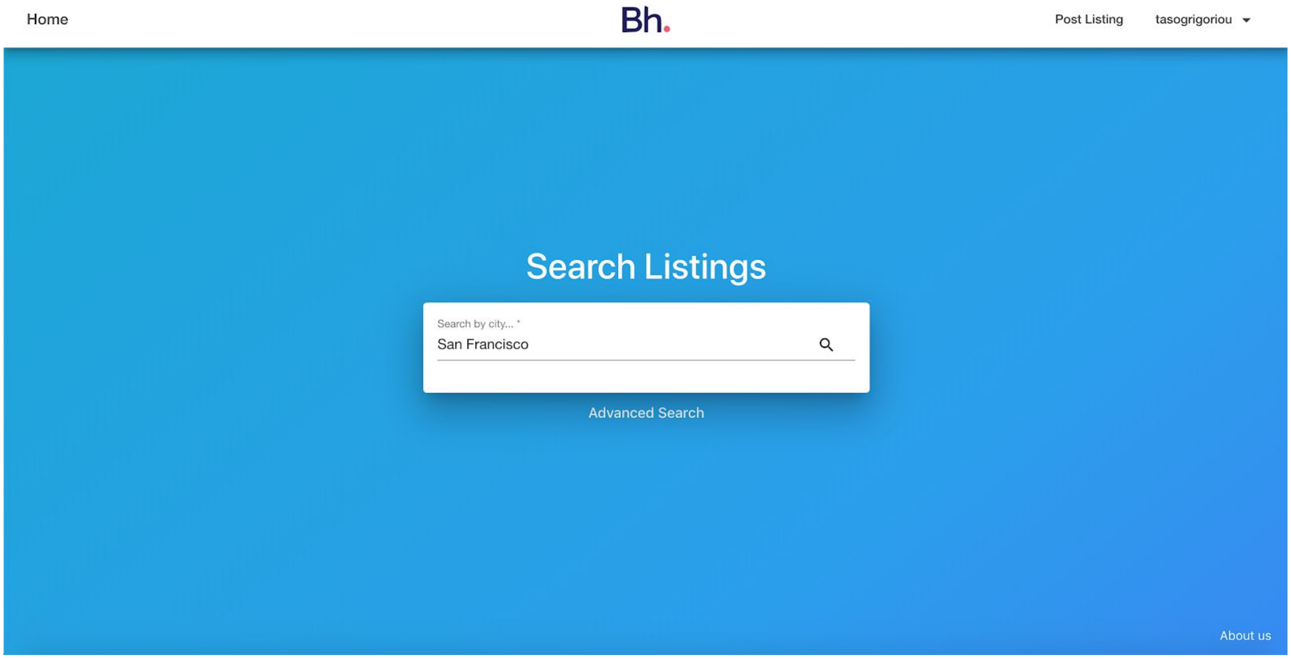
Internationalization / Localization Requirements:
1. Default language shall be English. **DONE**
2. The site shall support scalability to add other languages. **ON TRACK**
4. Any copyrighted material shall be immediately be taken down upon reception of an official DMCA takedown request. **DONE**

Website Policies:
1. A link to the policies of this site shall be always visible in all its pages to be accessible by

all the parties. **ON TRACK**
2. Clients’ data shall not be sold to third parties. **DONE**
3. Clients and Landlords data that do not add any functionality to the system shall not be Collected. **DONE**
4. Clients that post inappropriate listings(false listings/ copyright images/ sexual images) shall have their postings taken down. **DONE**

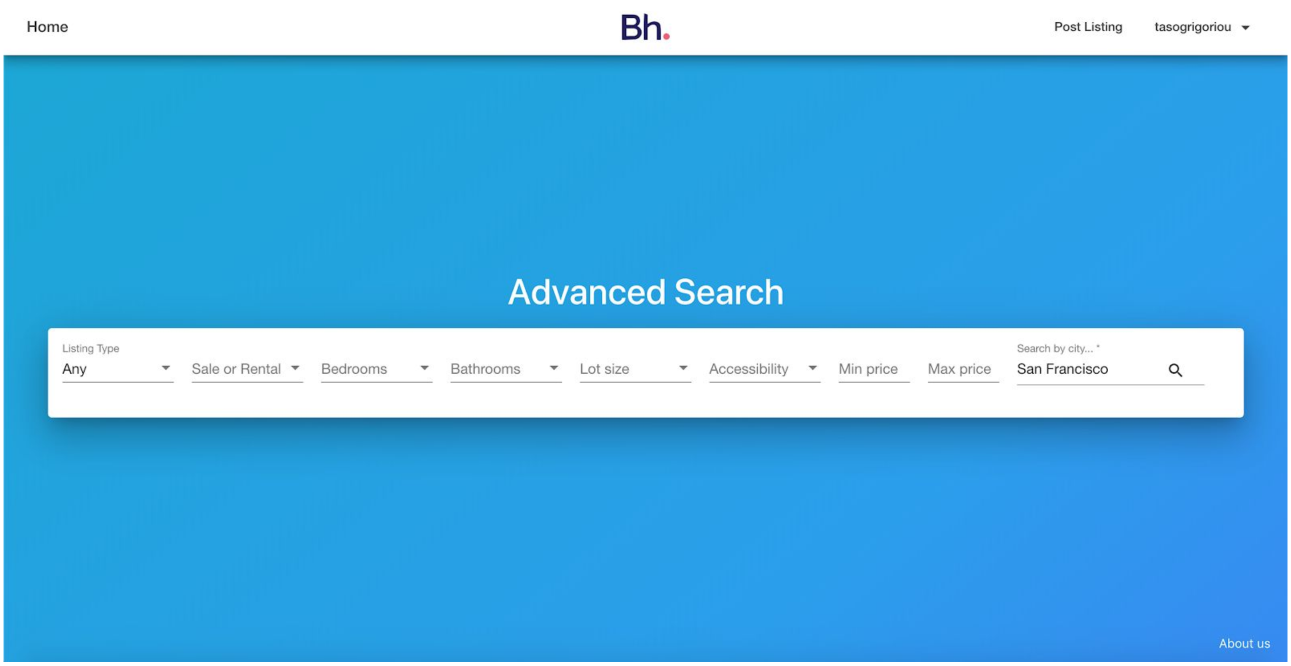
3. Screenshots of BetterHome:

Homepage and Search:



Search by city... *
San Francisco

Advanced Search:



Advanced Search

Listing Type: Any

Sale or Rental

Bedrooms

Bathrooms

Lot size

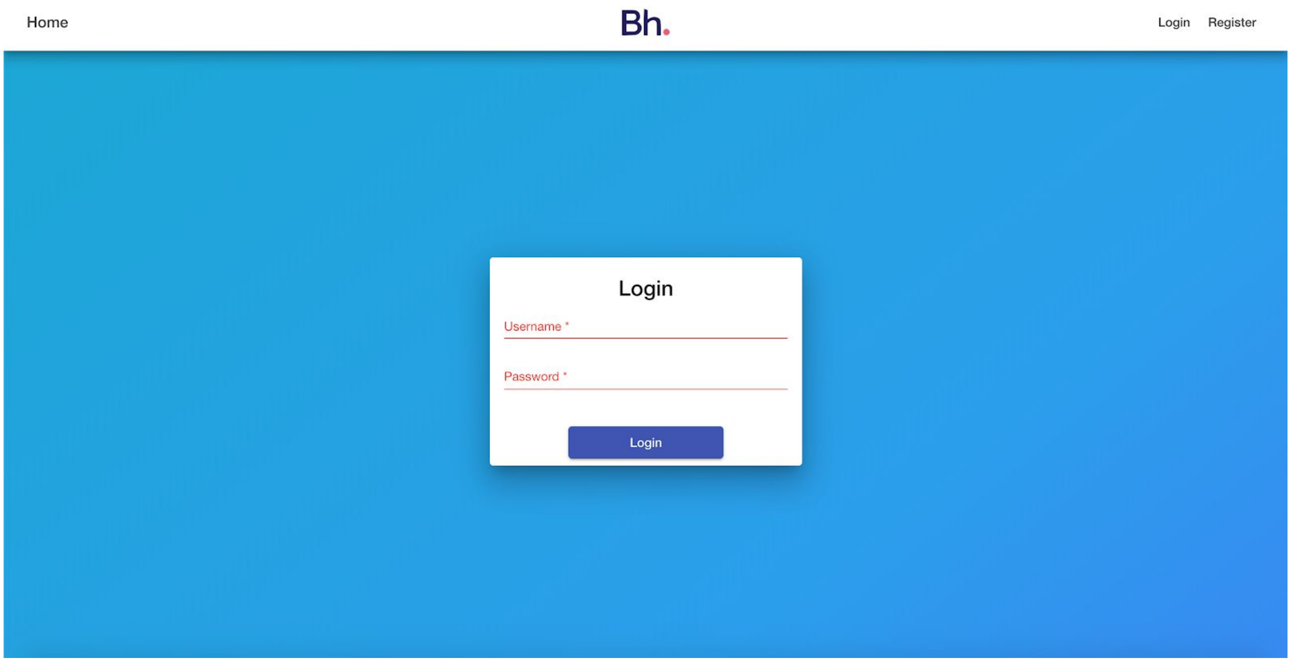
Accessibility

Min price

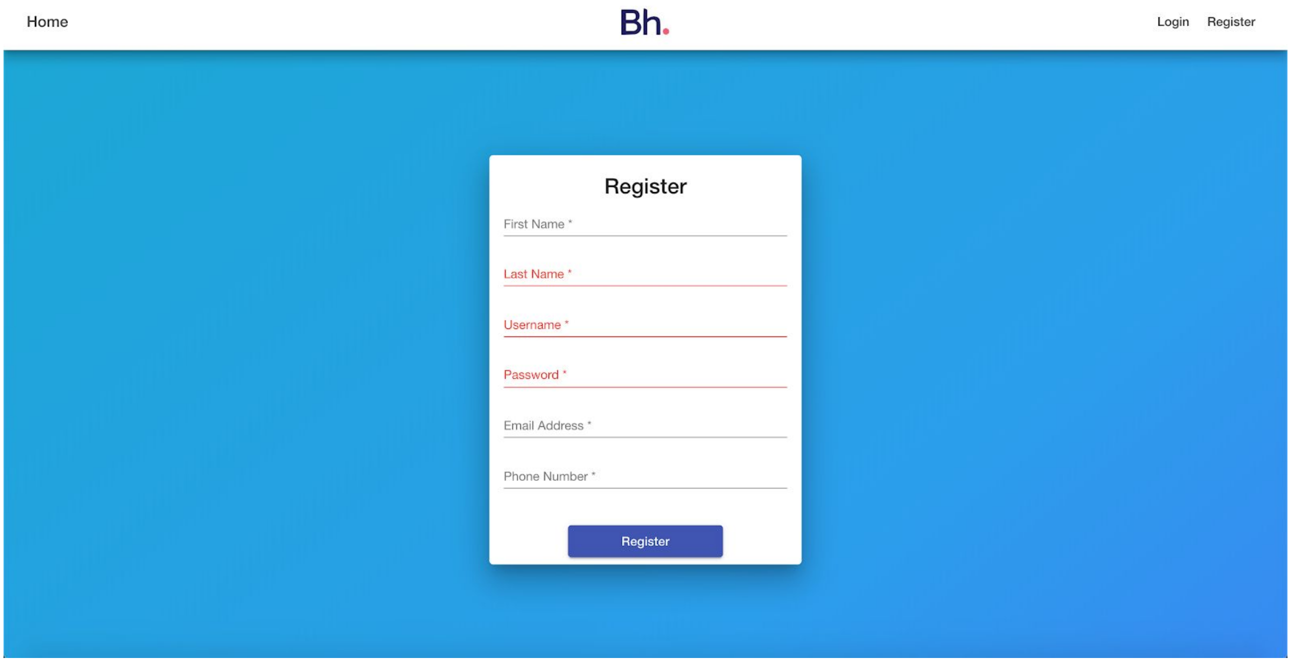
Max price

Search by city... *: San Francisco

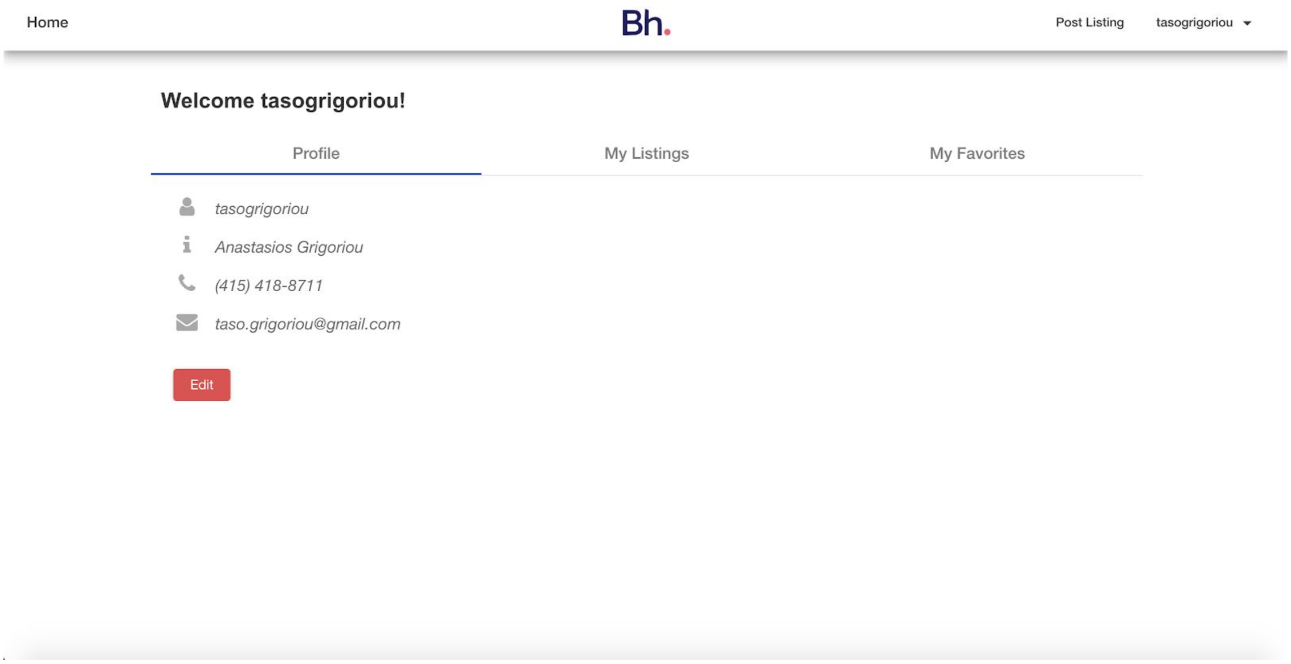
Login page:



Register page:



Account/Profile page:



Welcome tasogrigoriou!

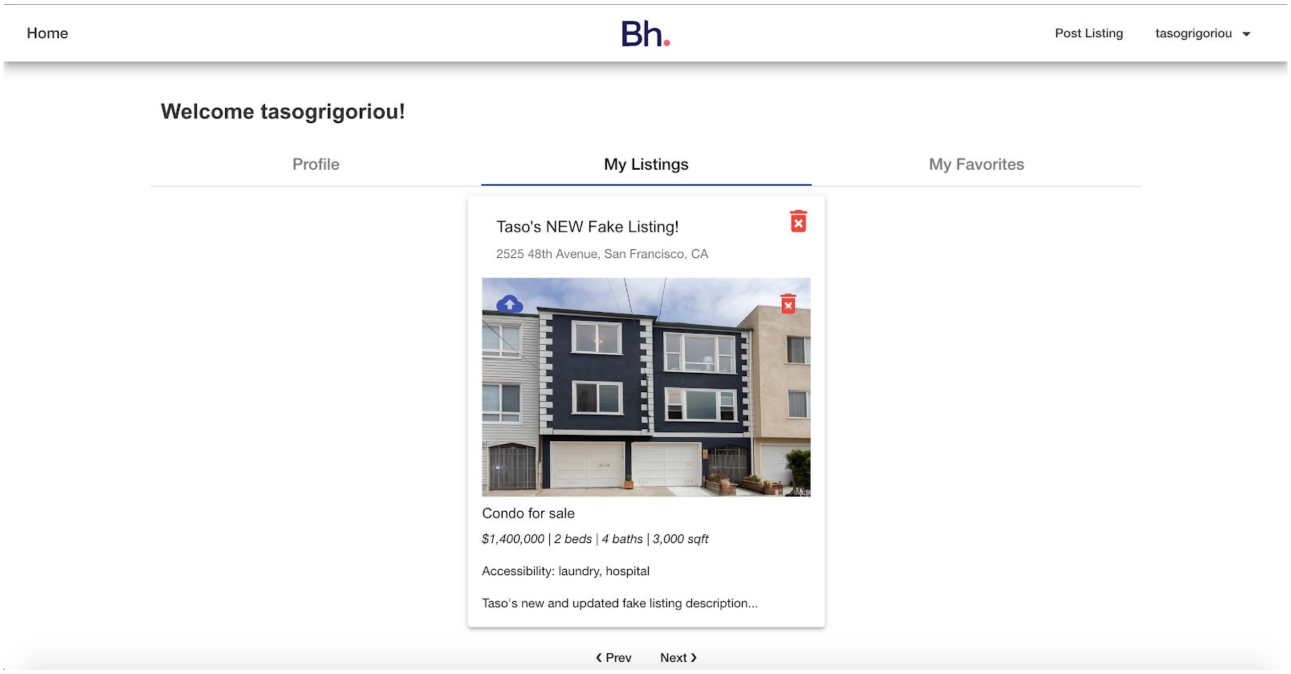
tasogrigoriou

Anastasios Grigoriou

(415) 418-8711

taso.grigoriou@gmail.com

My Listings page:



Welcome tasogrigoriou!

Taso's NEW Fake Listing!

2525 48th Avenue, San Francisco, CA

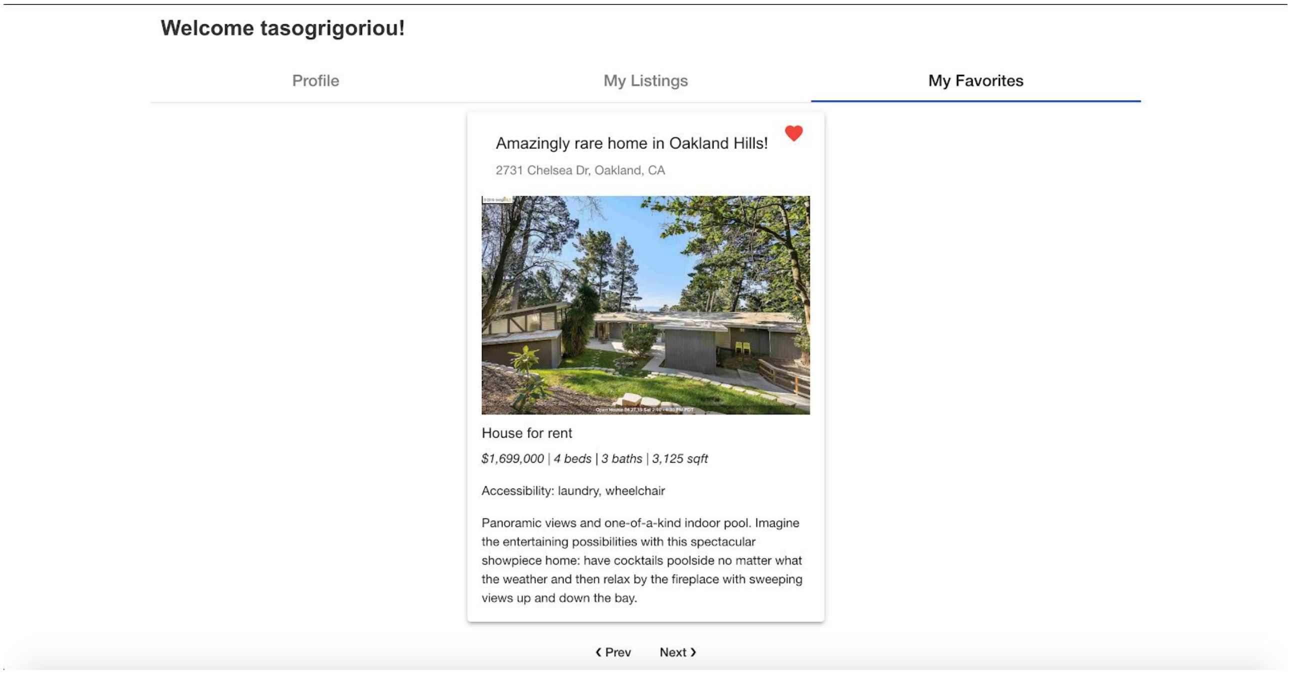
Condo for sale

$1,400,000 | 2 beds | 4 baths | 3,000 sqft

Accessibility: laundry, hospital

Taso's new and updated fake listing description...

My Favorites:



Welcome tasogrigoriou!

Amazingly rare home in Oakland Hills! ♥

2731 Chelsea Dr, Oakland, CA

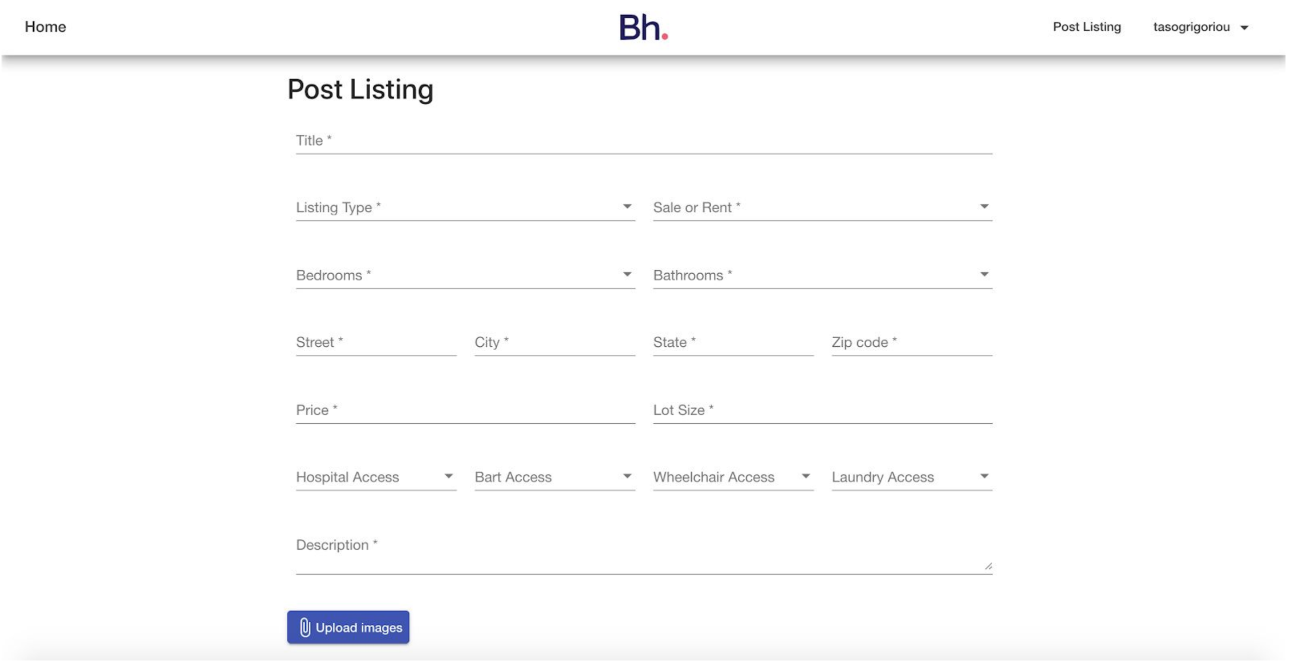
House for rent

$1,699,000 | *4 beds* | *3 baths* | *3,125 sqft*

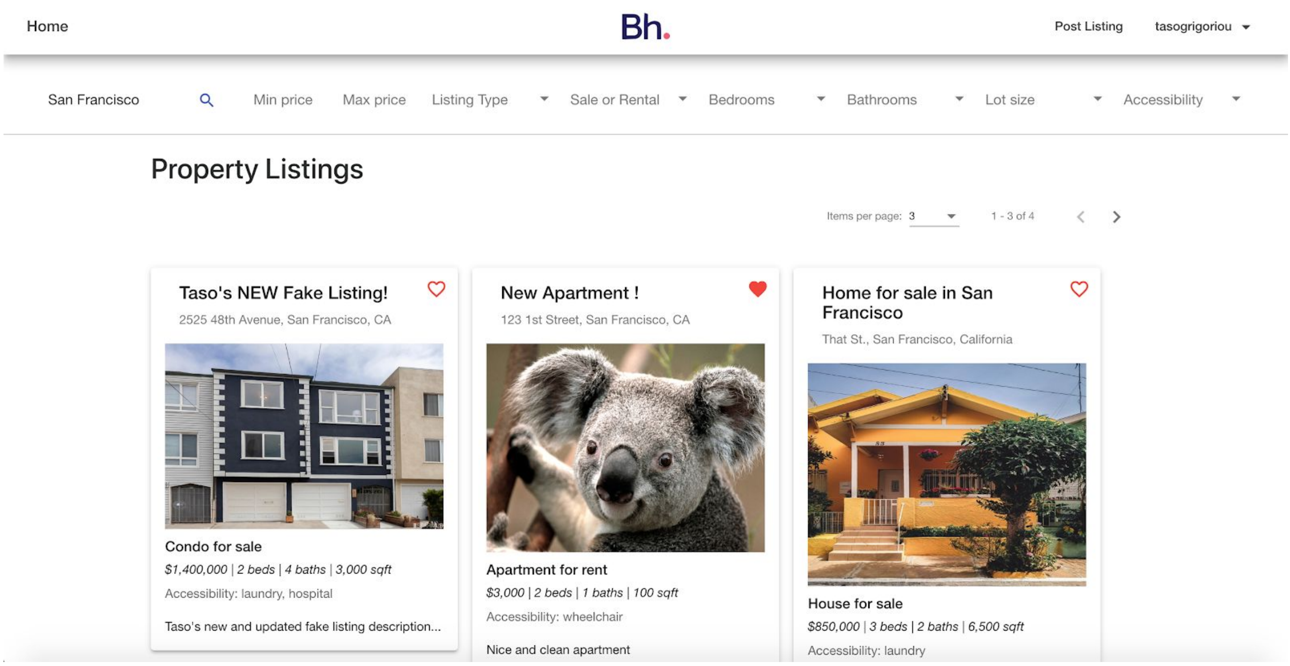
Accessibility: laundry, wheelchair

Panoramic views and one-of-a-kind indoor pool. Imagine the entertaining possibilities with this spectacular showpiece home: have cocktails poolside no matter what the weather and then relax by the fireplace with sweeping views up and down the bay.

Post Listing:



Search result page (San Francisco):



Property Listings

Items per page: 3 1 - 3 of 4

Taso's NEW Fake Listing!

2525 48th Avenue, San Francisco, CA

Condo for sale

$1,400,000 | *2 beds* | *4 baths* | *3,000 sqft*

Accessibility: laundry, hospital

Taso's new and updated fake listing description...

New Apartment !

123 1st Street, San Francisco, CA

Apartment for rent

$3,000 | *2 beds* | *1 baths* | *100 sqft*

Accessibility: wheelchair

Nice and clean apartment

Home for sale in San Francisco

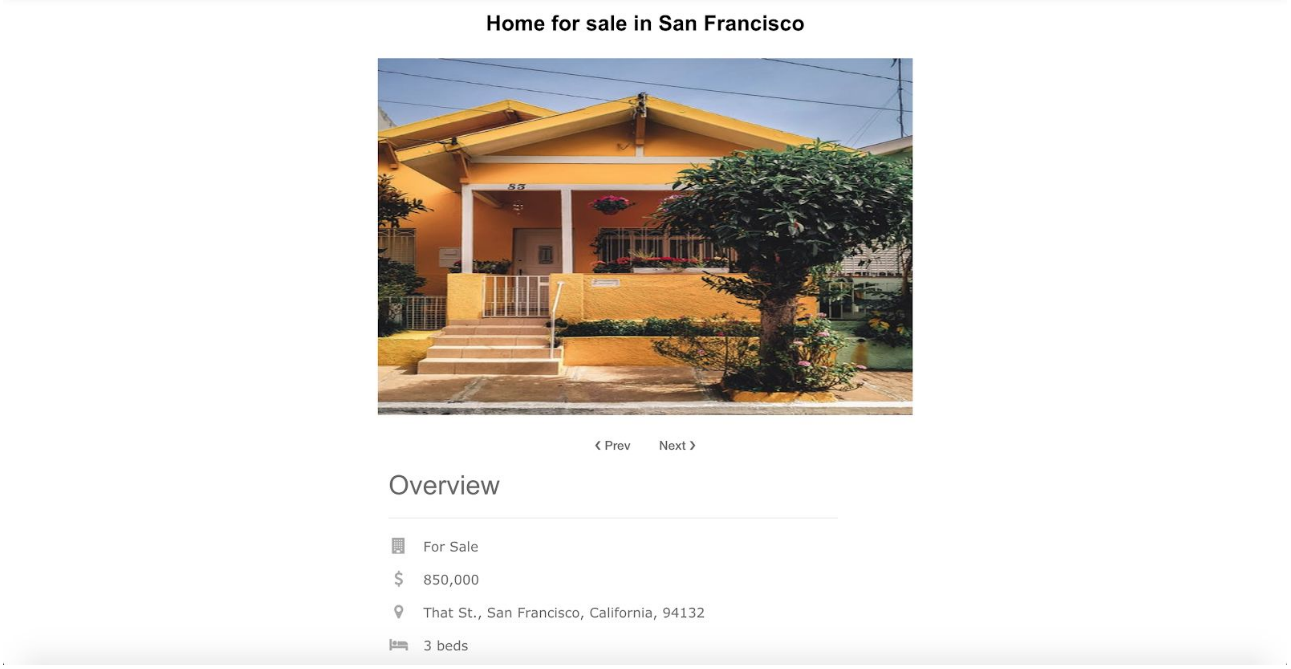
That St., San Francisco, California

House for sale

$850,000 | *3 beds* | *2 baths* | *6,500 sqft*

Accessibility: laundry

Listing Detail page:



Home for sale in San Francisco

Overview

For Sale

$ 850,000

That St., San Francisco, California, 94132

3 beds

4. Screenshots of key DB Tables:

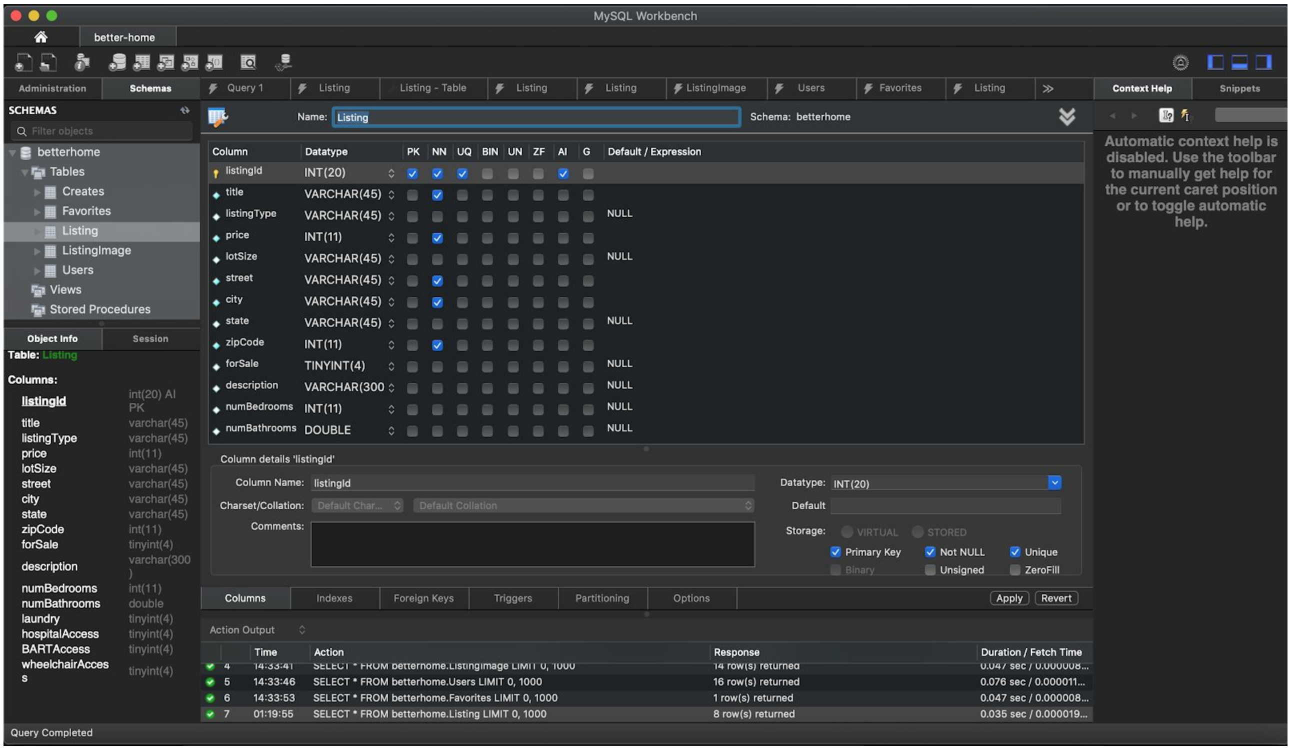


Table: Listing

Columns:

listingId	int(20) AI PK
title	varchar(45)
listingType	varchar(45)
price	int(11)
lotSize	varchar(45)
street	varchar(45)
city	varchar(45)
state	varchar(45)
zipCode	int(11)
forSale	tinyint(4)
description	varchar(300)
numBedrooms	int(11)
numBathrooms	double
laundry	tinyint(4)
hospitalAccess	tinyint(4)
BARTAccess	tinyint(4)
wheelchairAccess	tinyint(4)

Name: Listing Schema: betterhome

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
listingId	INT(20)	☑	☑	☑	□	□	□	☑	□	
title	VARCHAR(45)	□	☑	□	□	□	□	□	□	
listingType	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
price	INT(11)	□	☑	□	□	□	□	□	□	
lotSize	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
street	VARCHAR(45)	□	☑	□	□	□	□	□	□	
city	VARCHAR(45)	□	☑	□	□	□	□	□	□	
state	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
zipCode	INT(11)	□	☑	□	□	□	□	□	□	
forSale	TINYINT(4)	□	□	□	□	□	□	□	□	NULL
description	VARCHAR(300	□	□	□	□	□	□	□	□	NULL
numBedrooms	INT(11)	□	□	□	□	□	□	□	□	NULL
numBathrooms	DOUBLE	□	□	□	□	□	□	□	□	NULL

Column details 'listingId'

Column Name: listingId
Charset/Collation: Default Char... Default Collation
Comments:
Datatype: INT(20)
Default:
Storage: ○ VIRTUAL ○ STORED
☑ Primary Key ☑ Not NULL ☑ Unique
□ Binary □ Unsigned □ ZeroFill

Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options — Apply | Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
4	14:33:41	SELECT * FROM betterhome.ListingImage LIMIT 0, 1000	14 row(s) returned	0.047 sec / 0.000008...
5	14:33:46	SELECT * FROM betterhome.Users LIMIT 0, 1000	16 row(s) returned	0.076 sec / 0.000011...
6	14:33:53	SELECT * FROM betterhome.Favorites LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000008...
7	01:19:55	SELECT * FROM betterhome.Listing LIMIT 0, 1000	8 row(s) returned	0.035 sec / 0.000019...

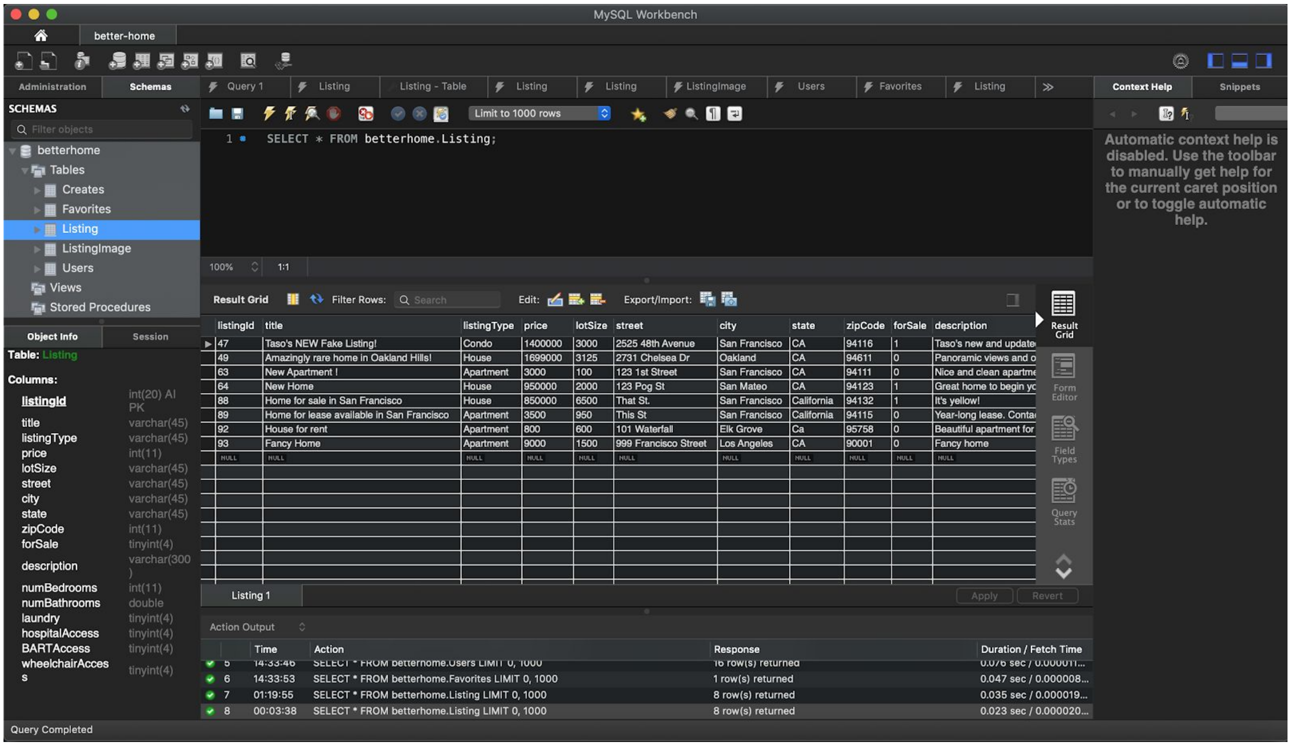


Table: Listing

Columns:

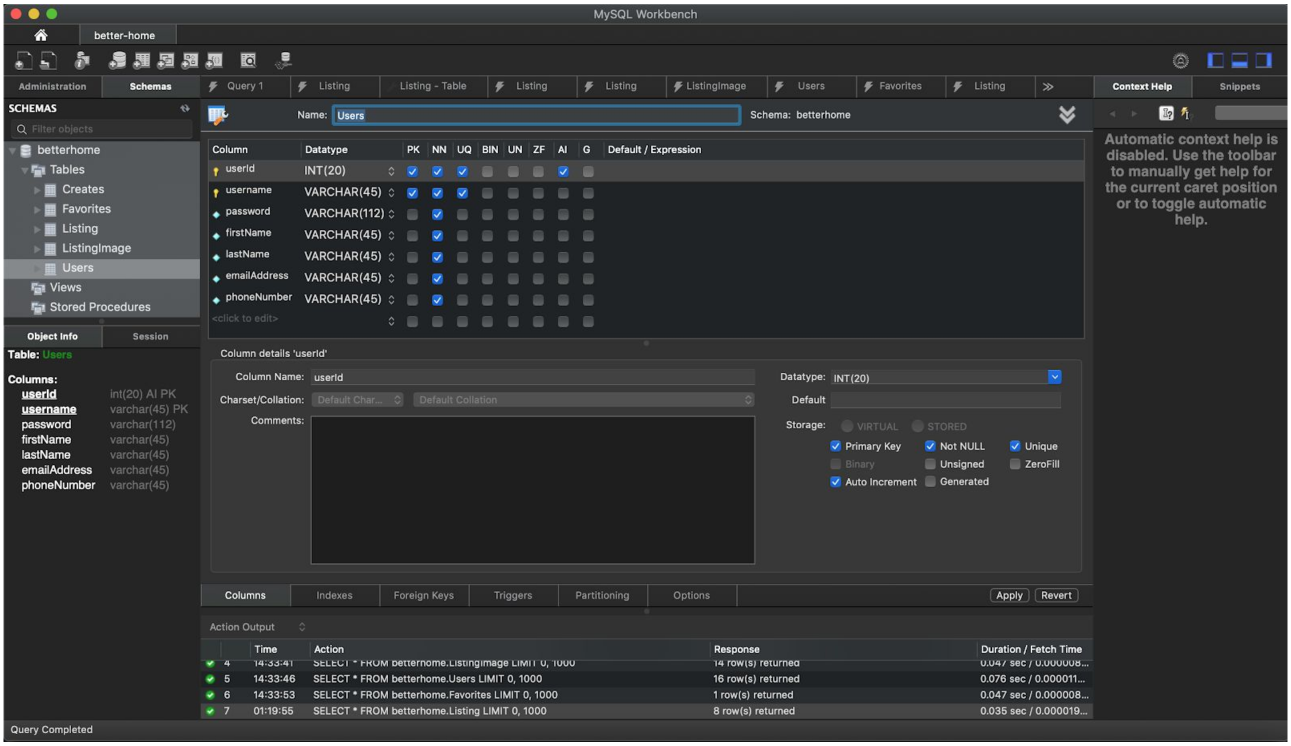
listingId	int(20) AI PK
title	varchar(45)
listingType	varchar(45)
price	int(11)
lotSize	varchar(45)
street	varchar(45)
city	varchar(45)
state	varchar(45)
zipCode	int(11)
forSale	tinyint(4)
description	varchar(300)
numBedrooms	int(11)
numBathrooms	double
laundry	tinyint(4)
hospitalAccess	tinyint(4)
BARTAccess	tinyint(4)
wheelchairAccess	tinyint(4)

```
SELECT * FROM betterhome.Listing;
```

listingId	title	listingType	price	lotSize	street	city	state	zipCode	forSale	description
47	Taso's NEW Fake Listing!	Condo	1400000	3000	2525 48th Avenue	San Francisco	CA	94116	1	Taso's new and update
49	Amazingly rare home in Oakland Hills!	House	1699000	3125	2731 Chelsea Dr	Oakland	CA	94611	0	Panoramic views and o
63	New Apartment !	Apartment	3000	100	123 1st Street	San Francisco	CA	94111	0	Nice and clean apartme
64	New Home	House	950000	2000	123 Pog St	San Mateo	CA	94123	1	Great home to begin yo
88	Home for sale in San Francisco	House	850000	6500	That St.	San Francisco	California	94132	1	It's yellow!
89	Home for lease available in San Francisco	Apartment	3500	950	This St	San Francisco	California	94115	0	Year-long lease. Conta
92	House for rent	Apartment	800	600	101 Waterfall	Elk Grove	Ca	95758	0	Beautiful apartment for
93	Fancy Home	Apartment	9000	1500	999 Francisco Street	Los Angeles	CA	90001	0	Fancy home
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Action Output

	Time	Action	Response	Duration / Fetch Time
5	14:33:46	SELECT * FROM betterhome.Users LIMIT 0, 1000	16 row(s) returned	0.076 sec / 0.000011...
6	14:33:53	SELECT * FROM betterhome.Favorites LIMIT 0, 1000	1 row(s) returned	0.047 sec / 0.000008...
7	01:19:55	SELECT * FROM betterhome.Listing LIMIT 0, 1000	8 row(s) returned	0.035 sec / 0.000019...
8	00:03:38	SELECT * FROM betterhome.Listing LIMIT 0, 1000	8 row(s) returned	0.023 sec / 0.000020...



Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Name: Users   Schema: betterhome

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
userId	INT(20)	☑	☑	☑	□	□	□	☑	□	
username	VARCHAR(45)	☑	☑	☑	□	□	□	□	□	
password	VARCHAR(112)	□	☑	□	□	□	□	□	□	
firstName	VARCHAR(45)	□	☑	□	□	□	□	□	□	
lastName	VARCHAR(45)	□	☑	□	□	□	□	□	□	
emailAddress	VARCHAR(45)	□	☑	□	□	□	□	□	□	
phoneNumber	VARCHAR(45)	□	☑	□	□	□	□	□	□	
<click to edit>		□	□	□	□	□	□	□	□	

Column details 'userId'

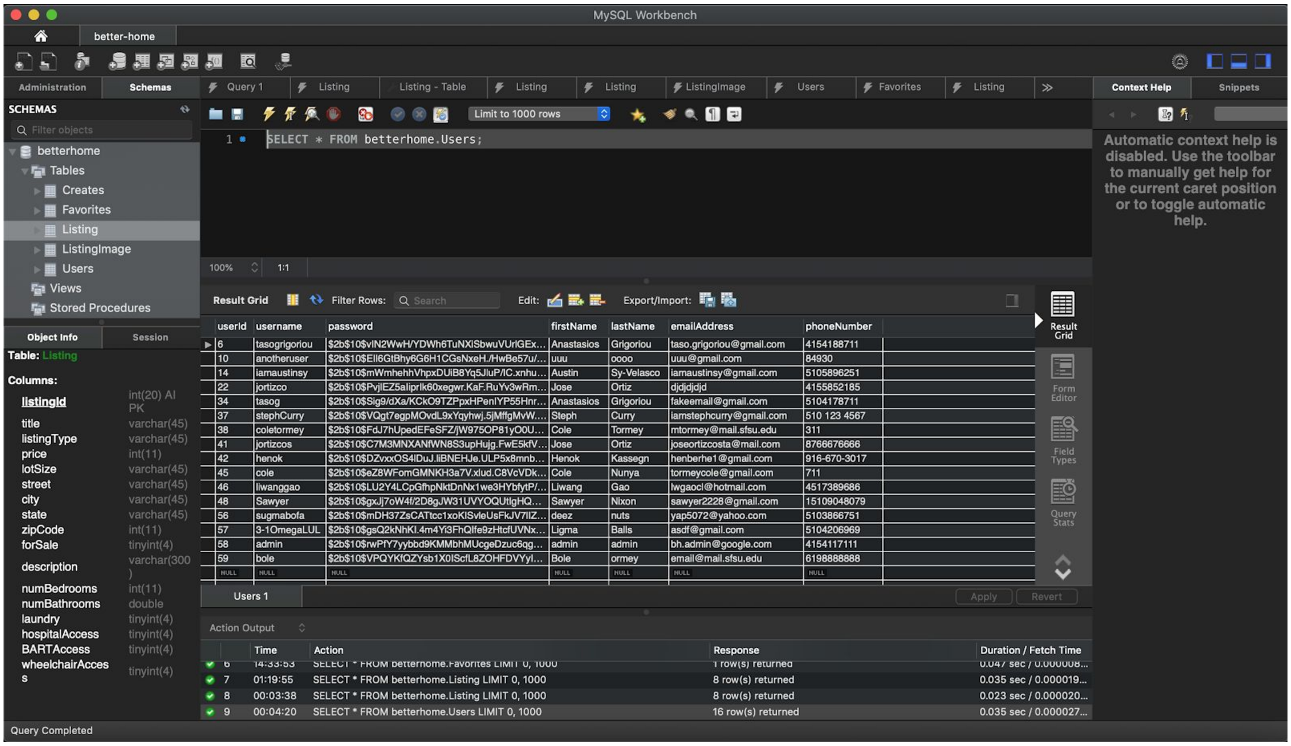
Column Name: userId
Charset/Collation: Default Char... Default Collation
Comments:
Datatype: INT(20)
Default:
Storage: ○ VIRTUAL ○ STORED
☑ Primary Key ☑ Not NULL ☑ Unique
□ Binary □ Unsigned □ ZeroFill
☑ Auto Increment □ Generated

Table: Users

Columns:

userId	int(20) AI PK
username	varchar(45) PK
password	varchar(112)
firstName	varchar(45)
lastName	varchar(45)
emailAddress	varchar(45)
phoneNumber	varchar(45)

```
    Time      Action                                                Response             Duration / Fetch Time
4   14:33:41  SELECT * FROM betterhome.ListingImage LIMIT 0, 1000   14 row(s) returned   0.047 sec / 0.000008...
5   14:33:46  SELECT * FROM betterhome.Users LIMIT 0, 1000          16 row(s) returned   0.076 sec / 0.000011...
6   14:33:53  SELECT * FROM betterhome.Favorites LIMIT 0, 1000      1 row(s) returned    0.047 sec / 0.000008...
7   01:19:55  SELECT * FROM betterhome.Listing LIMIT 0, 1000        8 row(s) returned    0.035 sec / 0.000019...
```



Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
SELECT * FROM betterhome.Users;
```

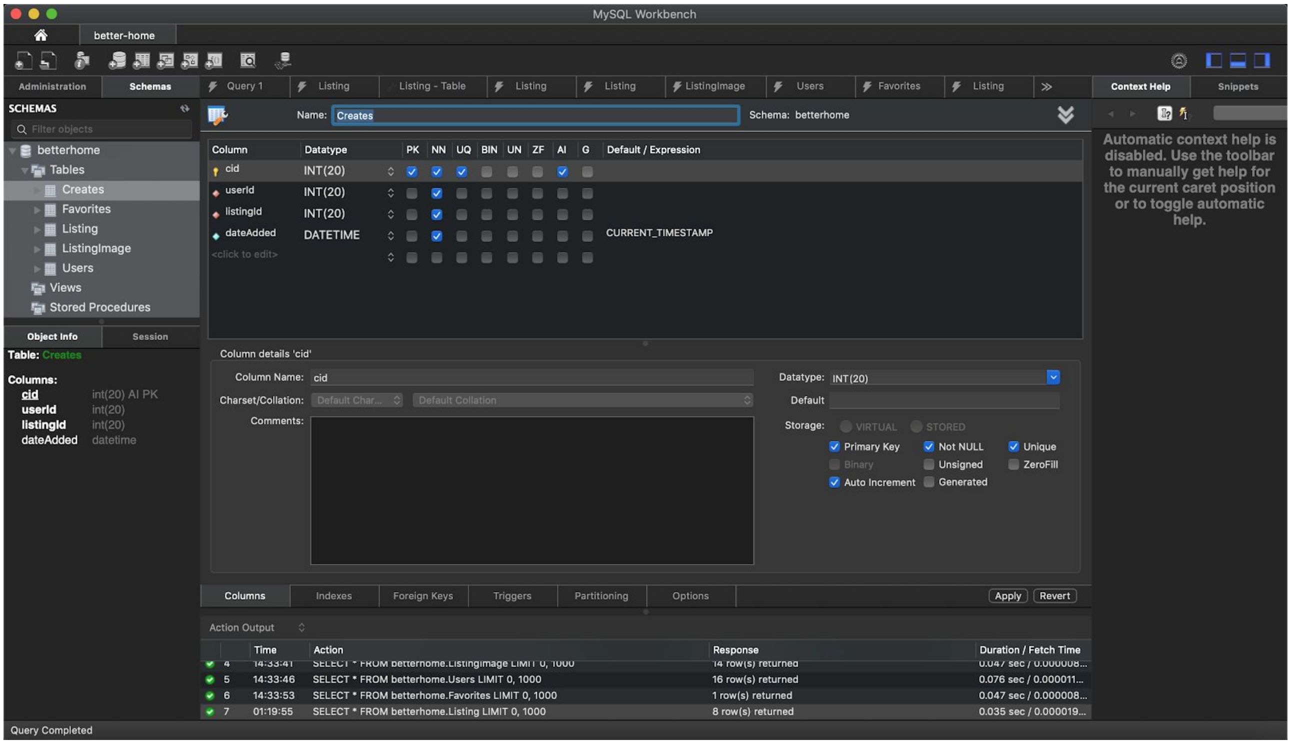
userId	username	password	firstName	lastName	emailAddress	phoneNumber
6	tasogrigoriou	$2b$10$vIN2WwH/YDWh6TuNXiSbwuVUrlGEx...	Anastasios	Grigoriou	taso.grigoriou@gmail.com	4154188711
10	anotheruser	$2b$10$EIl6GtBhy6G6H1CGsNxeH./HwBe57u/...	uuu	oooo	uuu@gmail.com	84930
14	iamaustinsy	$2b$10$mWmhehhVhpxDUiB8Yq5JluP/IC.xnhu...	Austin	Sy-Velasco	iamaustinsy@gmail.com	5105896251
22	jortizco	$2b$10$PvjlEZ5aliprlk60xegwr.KaF.RuYv3wRm...	Jose	Ortiz	djdjdjdjd	4155852185
34	tasog	$2b$10$Sig9/dXa/KCkO9TZPpxHPenIYP55Hnr...	Anastasios	Grigoriou	fakeemail@gmail.com	5104178711
37	stephCurry	$2b$10$VQgt7egpMOvdL9xYqyhwj.5jMffgMvW....	Steph	Curry	iamstephcurry@gmail.com	510 123 4567
38	coletormey	$2b$10$FdJ7hUpedEFeSFZ/jW975OP81yO0U...	Cole	Tormey	mtormey@mail.sfsu.edu	311
41	jortizcos	$2b$10$C7M3MNXANfWN8S3upHujg.FwE5kfV...	Jose	Ortiz	joseortizcosta@mail.com	8766676666
42	henok	$2b$10$DZvxxOS4lDuJ.liBNEHJe.ULP5x8mnb...	Henok	Kassegn	henberhe1@gmail.com	916-670-3017
45	cole	$2b$10$eZ8WFomGMNKH3a7V.xlud.C8VcVDk...	Cole	Nunya	tormeycole@gmail.com	711
46	liwanggao	$2b$10$LU2Y4LCpGfhpNktDnNx1we3HYbfytP/...	Liwang	Gao	lwgaocl@hotmail.com	4517389686
48	Sawyer	$2b$10$gxJj7oW4f/2D8gJW31UVYOQUtlgHQ...	Sawyer	Nixon	sawyer2228@gmail.com	15109048079
56	sugmabofa	$2b$10$mDH37ZsCATtcc1xoKISvleUsFkJV7IlZ...	deez	nuts	yap5072@yahoo.com	5103866751
57	3-1OmegaLUL	$2b$10$gsQ2kNhKI.4m4Yi3FhQlfe9zHtcfUVNx...	Ligma	Balls	asdf@gmail.com	5104206969
58	admin	$2b$10$rwPfY7yybbd9KMMbhMUcgeDzuc6qg...	admin	admin	bh.admin@google.com	4154117111
59	bole	$2b$10$VPQYKfQZYsb1X0IScfL8ZOHFDVYyl...	Bole	ormey	email@mail.sfsu.edu	6198888888
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: Listing

Columns:

listingId	int(20) AI PK
title	varchar(45)
listingType	varchar(45)
price	int(11)
lotSize	varchar(45)
street	varchar(45)
city	varchar(45)
state	varchar(45)
zipCode	int(11)
forSale	tinyint(4)
description	varchar(300)
numBedrooms	int(11)
numBathrooms	double
laundry	tinyint(4)
hospitalAccess	tinyint(4)
BARTAccess	tinyint(4)
wheelchairAccess	tinyint(4)

```
    Time      Action                                                Response             Duration / Fetch Time
6   14:33:53  SELECT * FROM betterhome.Favorites LIMIT 0, 1000      1 row(s) returned    0.047 sec / 0.000008...
7   01:19:55  SELECT * FROM betterhome.Listing LIMIT 0, 1000        8 row(s) returned    0.035 sec / 0.000019...
8   00:03:38  SELECT * FROM betterhome.Listing LIMIT 0, 1000        8 row(s) returned    0.023 sec / 0.000020...
9   00:04:20  SELECT * FROM betterhome.Users LIMIT 0, 1000          16 row(s) returned   0.035 sec / 0.000027...
```



Name: Creates Schema: betterhome

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
cid	INT(20)	☑	☑	☑	□	□	□	☑	□	
userId	INT(20)	□	☑	□	□	□	□	□	□	
listingId	INT(20)	□	☑	□	□	□	□	□	□	
dateAdded	DATETIME	□	☑	□	□	□	□	□	□	CURRENT_TIMESTAMP
<click to edit>		□	□	□	□	□	□	□	□	

Column details 'cid'

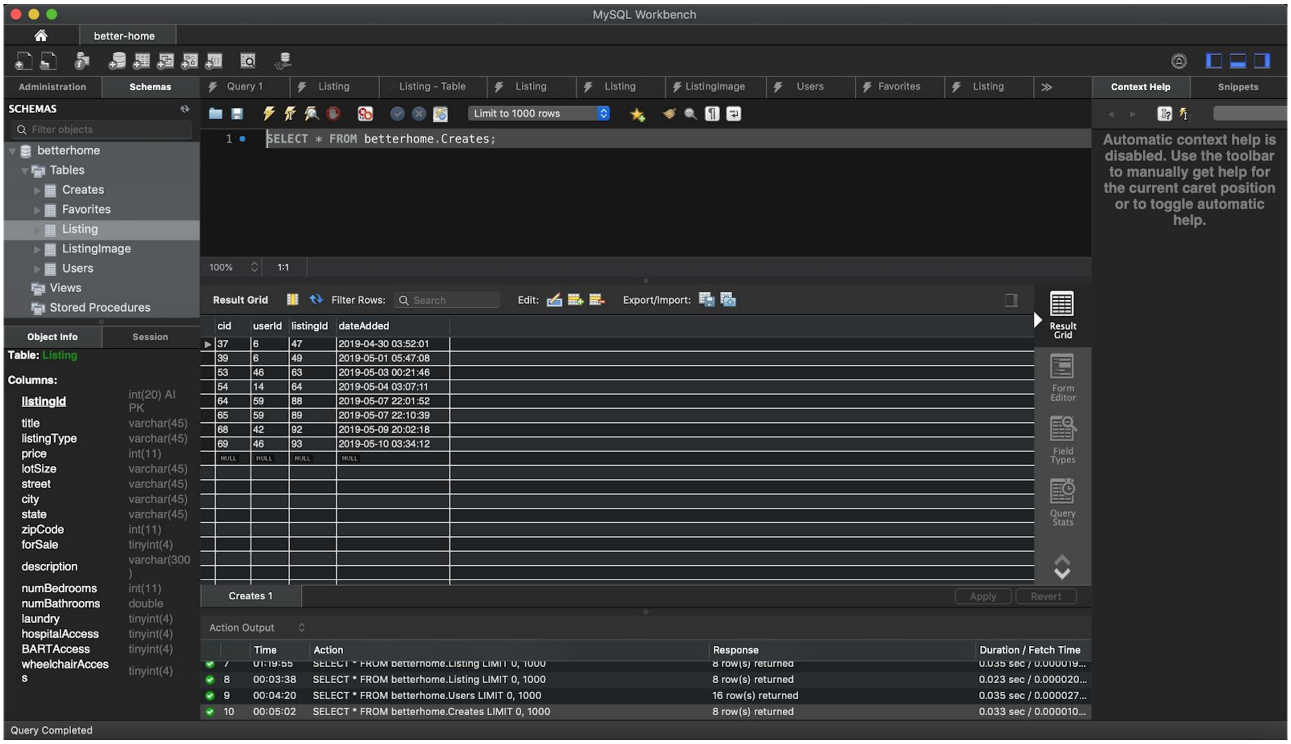
Column Name: cid
Datatype: INT(20)
Charset/Collation: Default Char... Default Collation
Default:
Comments:
Storage: ○ VIRTUAL ○ STORED
☑ Primary Key ☑ Not NULL ☑ Unique
□ Binary □ Unsigned □ ZeroFill
☑ Auto Increment □ Generated

Table: Creates

Columns:

cid	int(20) AI PK
userId	int(20)
listingId	int(20)
dateAdded	datetime

```
     Time      Action                                                  Response              Duration / Fetch Time
✓ 4  14:33:41  SELECT * FROM betterhome.ListingImage LIMIT 0, 1000     14 row(s) returned    0.047 sec / 0.000008...
✓ 5  14:33:46  SELECT * FROM betterhome.Users LIMIT 0, 1000            16 row(s) returned    0.076 sec / 0.000011...
✓ 6  14:33:53  SELECT * FROM betterhome.Favorites LIMIT 0, 1000        1 row(s) returned     0.047 sec / 0.000008...
✓ 7  01:19:55  SELECT * FROM betterhome.Listing LIMIT 0, 1000          8 row(s) returned     0.035 sec / 0.000019...
```



```
SELECT * FROM betterhome.Creates;
```

cid	userId	listingId	dateAdded
37	6	47	2019-04-30 03:52:01
39	6	49	2019-05-01 05:47:08
53	46	63	2019-05-03 00:21:46
54	14	64	2019-05-04 03:07:11
64	59	88	2019-05-07 22:01:52
65	59	89	2019-05-07 22:10:39
68	42	92	2019-05-09 20:02:18
69	46	93	2019-05-10 03:34:12
NULL	NULL	NULL	NULL

Table: Listing

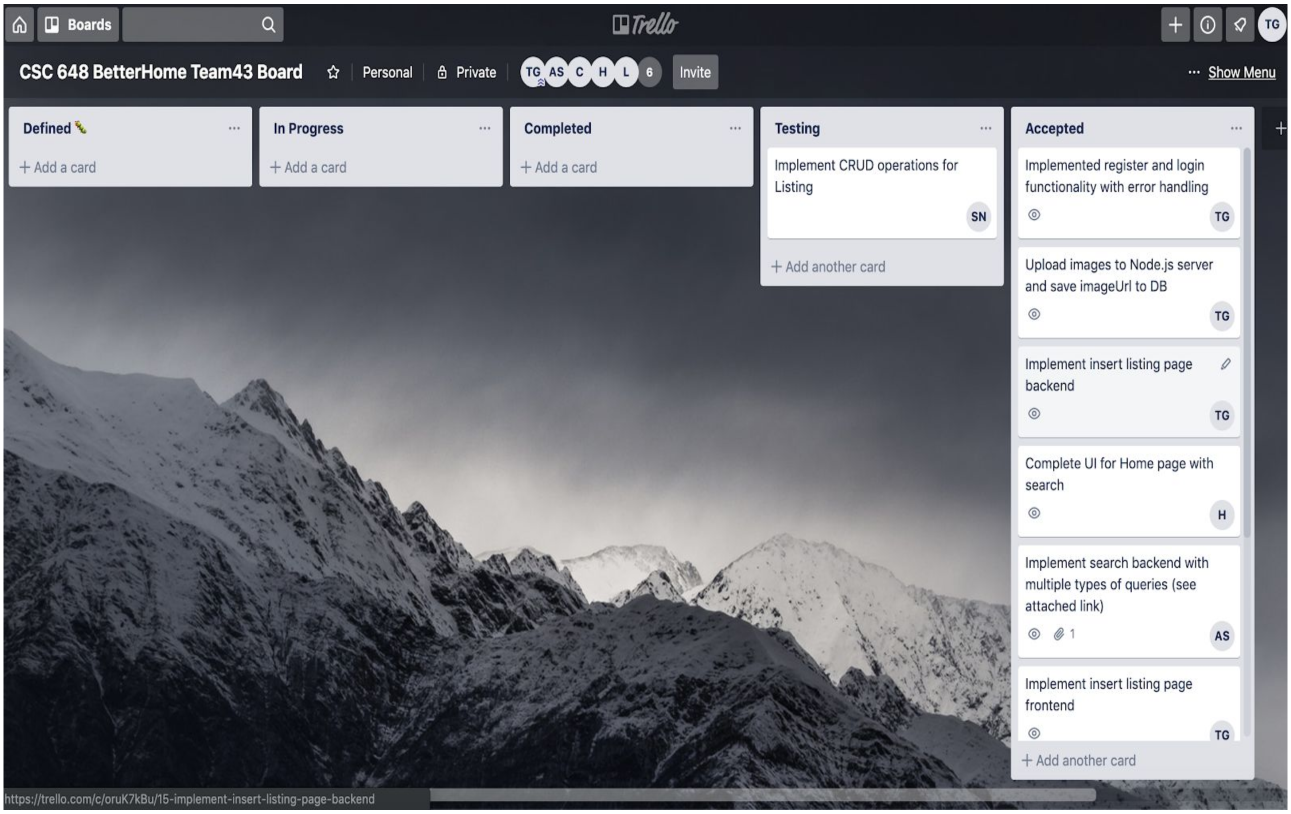
Columns:

listingId	int(20) AI PK
title	varchar(45)
listingType	varchar(45)
price	int(11)
lotSize	varchar(45)
street	varchar(45)
city	varchar(45)
state	varchar(45)
zipCode	int(11)
forSale	tinyint(4)
description	varchar(300)
numBedrooms	int(11)
numBathrooms	double
laundry	tinyint(4)
hospitalAccess	tinyint(4)
BARTAccess	tinyint(4)
wheelchairAccess	tinyint(4)

```
      Time      Action                                                  Response              Duration / Fetch Time
✓ 7   01:19:55  SELECT * FROM betterhome.Listing LIMIT 0, 1000          8 row(s) returned     0.035 sec / 0.000019...
✓ 8   00:03:38  SELECT * FROM betterhome.Listing LIMIT 0, 1000          8 row(s) returned     0.023 sec / 0.000020...
✓ 9   00:04:20  SELECT * FROM betterhome.Users LIMIT 0, 1000            16 row(s) returned    0.035 sec / 0.000027...
✓ 10  00:05:02  SELECT * FROM betterhome.Creates LIMIT 0, 1000          8 row(s) returned     0.033 sec / 0.000010...
```

5. Google Analytics plot for BetterHome:

6. Screenshots of task management system:



Testing

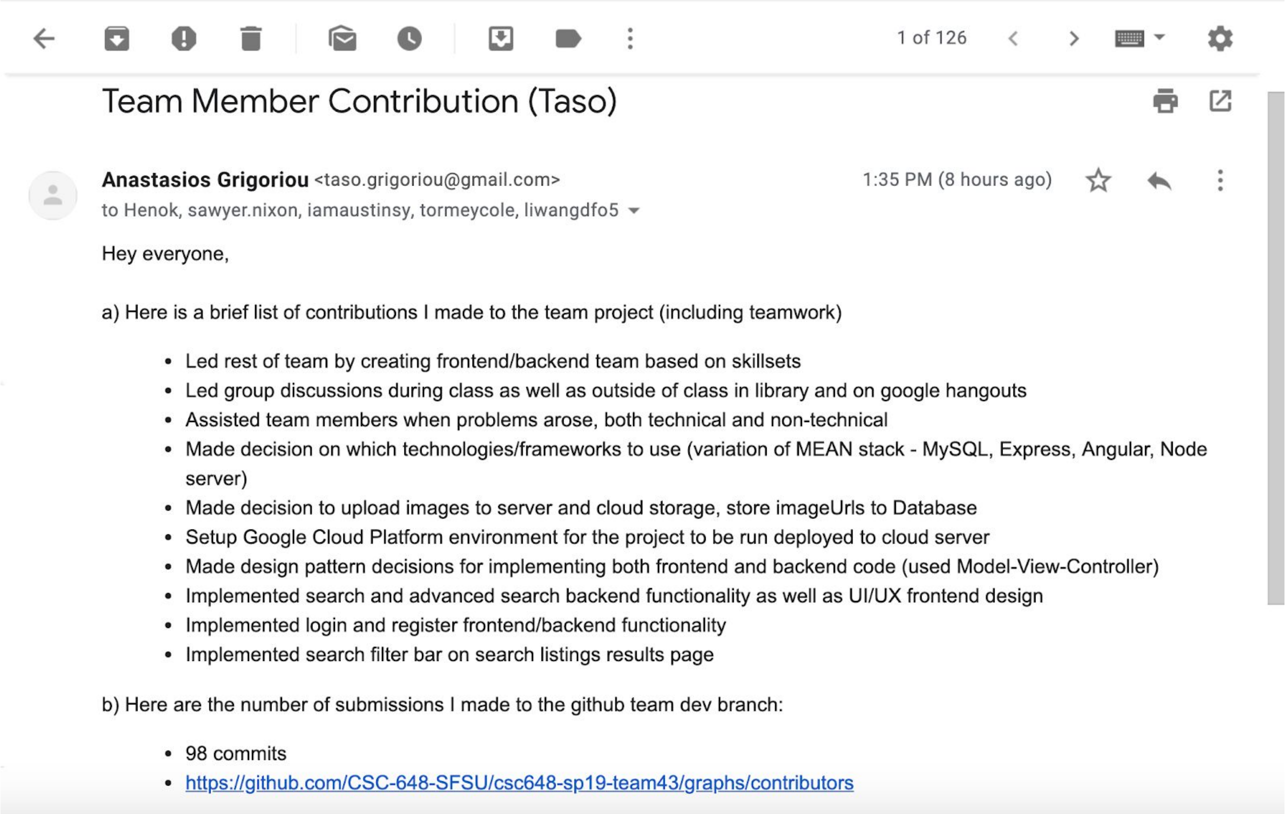
- Implement CRUD operations for Listing — SN

Accepted

- Implemented register and login functionality with error handling — TG
- Upload images to Node.js server and save imageUrl to DB — TG
- Implement insert listing page backend — TG
- Complete UI for Home page with search — H
- Implement search backend with multiple types of queries (see attached link) — 1 — AS
- Implement insert listing page frontend — TG

7. Team members contribution:

Taso Grigoriou (Team lead):



Team Member Contribution (Taso)

Anastasios Grigoriou <taso.grigoriou@gmail.com> 1:35 PM (8 hours ago)
to Henok, sawyer.nixon, iamaustinsy, tormeycole, liwangdfo5

Hey everyone,

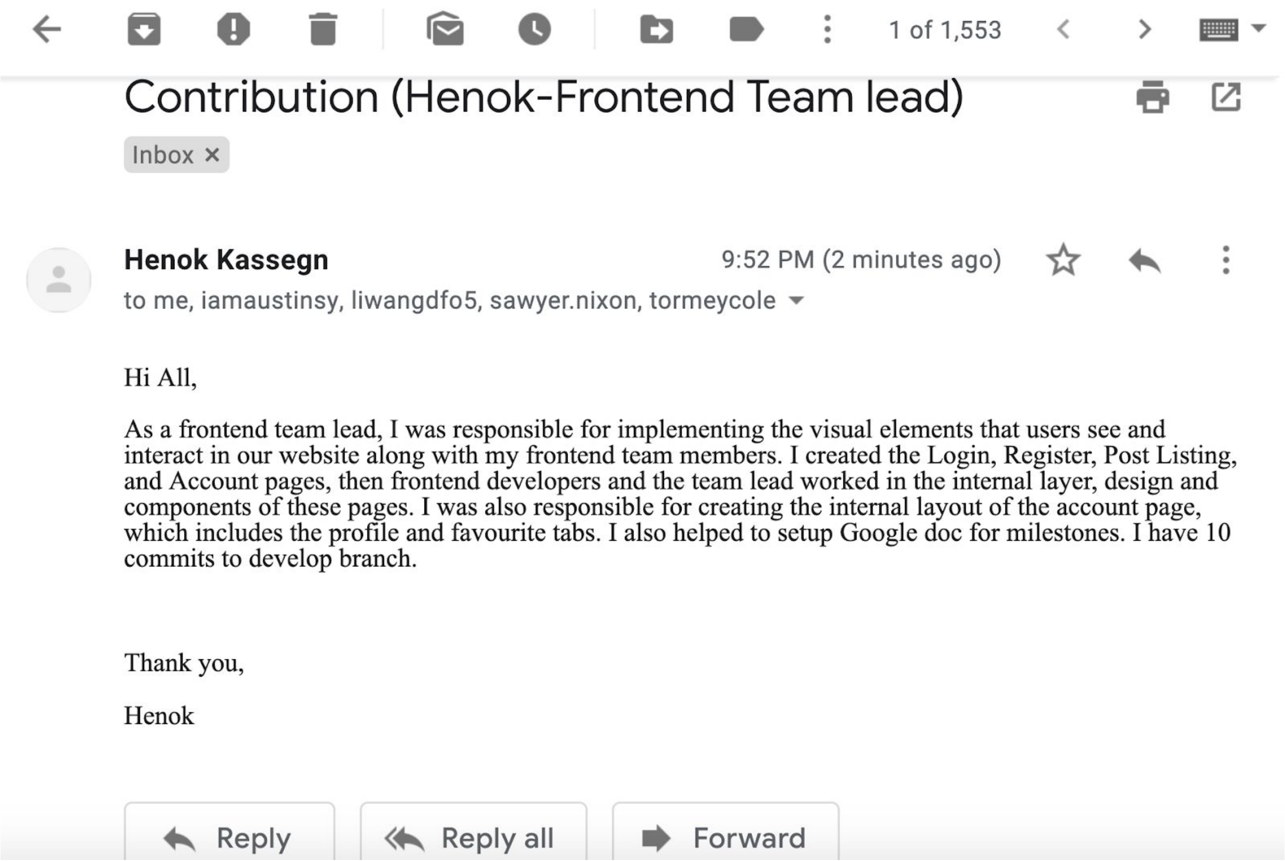
a) Here is a brief list of contributions I made to the team project (including teamwork)

- Led rest of team by creating frontend/backend team based on skillsets
- Led group discussions during class as well as outside of class in library and on google hangouts
- Assisted team members when problems arose, both technical and non-technical
- Made decision on which technologies/frameworks to use (variation of MEAN stack - MySQL, Express, Angular, Node server)
- Made decision to upload images to server and cloud storage, store imageUrls to Database
- Setup Google Cloud Platform environment for the project to be run deployed to cloud server
- Made design pattern decisions for implementing both frontend and backend code (used Model-View-Controller)
- Implemented search and advanced search backend functionality as well as UI/UX frontend design
- Implemented login and register frontend/backend functionality
- Implemented search filter bar on search listings results page

b) Here are the number of submissions I made to the github team dev branch:

- 98 commits
- https://github.com/CSC-648-SFSU/csc648-sp19-team43/graphs/contributors

Henok Kassegn (Frontend lead):



Contribution (Henok-Frontend Team lead)

Henok Kassegn 9:52 PM (2 minutes ago)

to me, iamaustinsy, liwangdfo5, sawyer.nixon, tormeycole

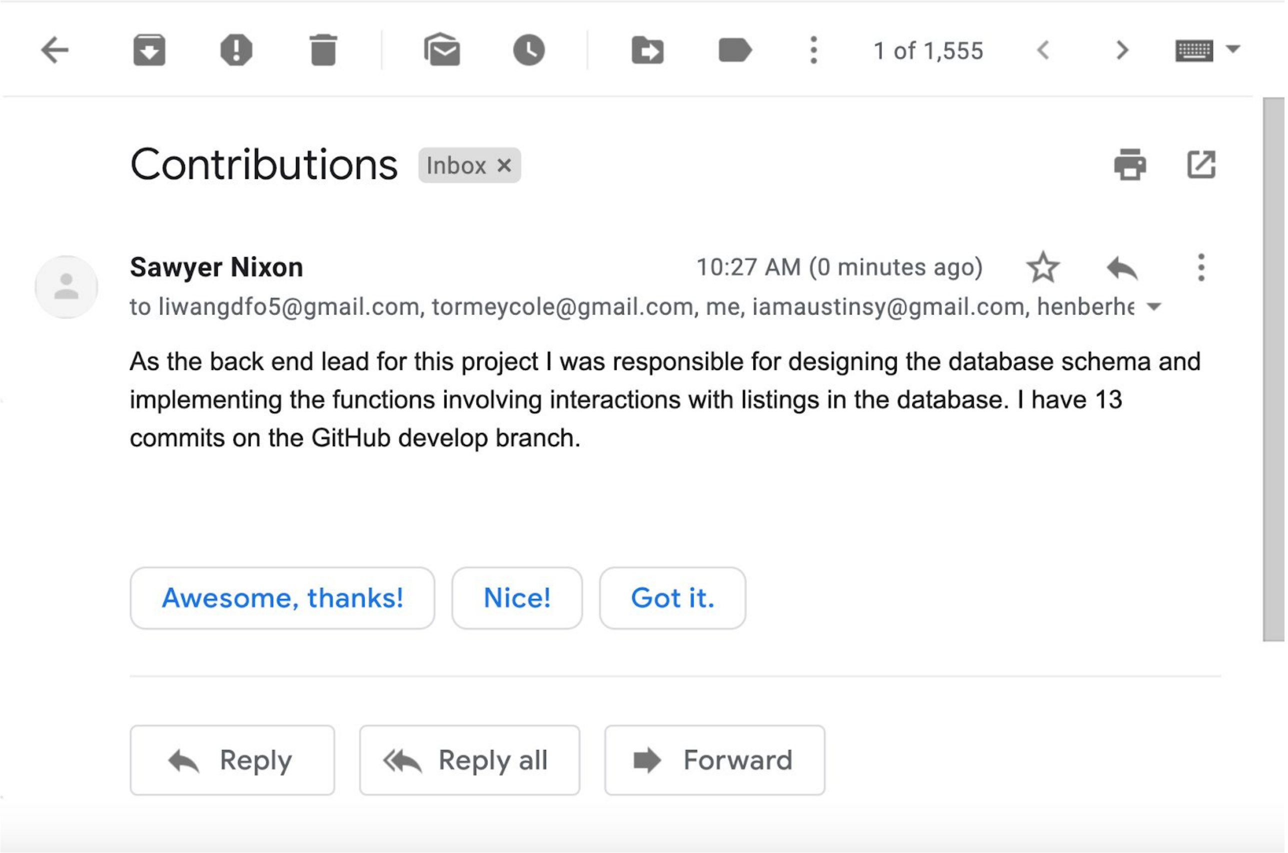
Hi All,

As a frontend team lead, I was responsible for implementing the visual elements that users see and interact in our website along with my frontend team members. I created the Login, Register, Post Listing, and Account pages, then frontend developers and the team lead worked in the internal layer, design and components of these pages. I was also responsible for creating the internal layout of the account page, which includes the profile and favourite tabs. I also helped to setup Google doc for milestones. I have 10 commits to develop branch.

Thank you,

Henok

Sawyer Nixon (Backend lead):



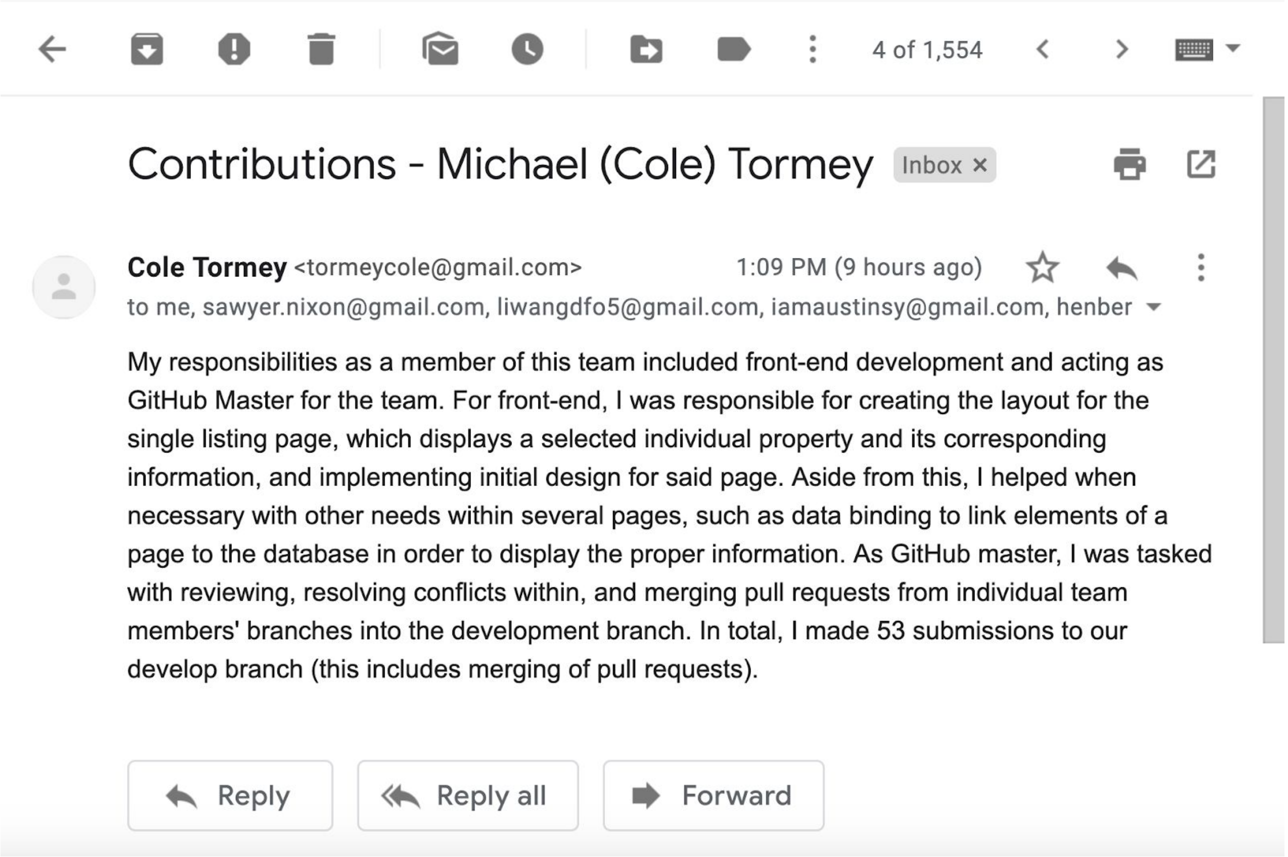
Contributions

Sawyer Nixon 10:27 AM (0 minutes ago)

to liwangdfo5@gmail.com, tormeycole@gmail.com, me, iamaustinsy@gmail.com, henberhe

As the back end lead for this project I was responsible for designing the database schema and implementing the functions involving interactions with listings in the database. I have 13 commits on the GitHub develop branch.

Cole (Michael) Tormey (github master):

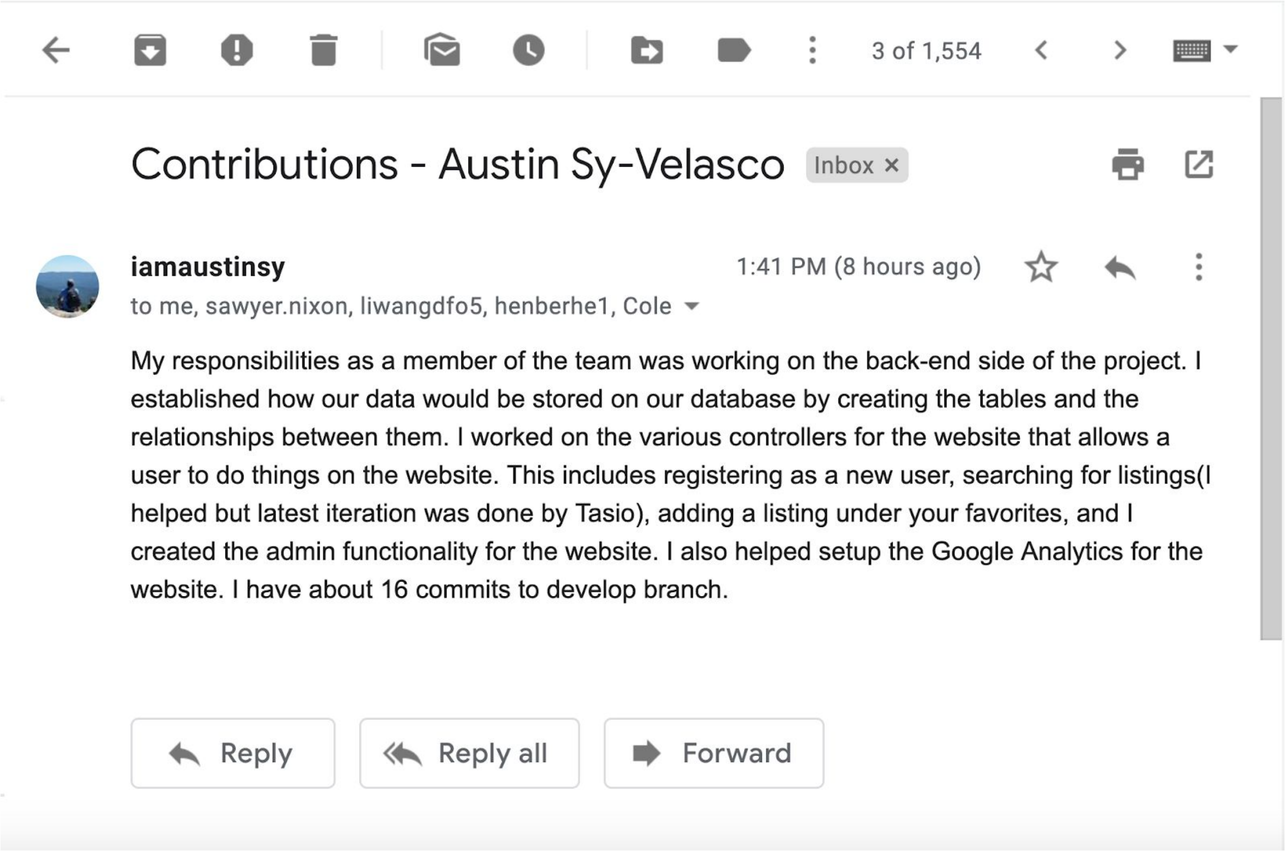


Contributions - Michael (Cole) Tormey

Cole Tormey <tormeycole@gmail.com> 1:09 PM (9 hours ago)
to me, sawyer.nixon@gmail.com, liwangdfo5@gmail.com, iamaustinsy@gmail.com, henber

My responsibilities as a member of this team included front-end development and acting as GitHub Master for the team. For front-end, I was responsible for creating the layout for the single listing page, which displays a selected individual property and its corresponding information, and implementing initial design for said page. Aside from this, I helped when necessary with other needs within several pages, such as data binding to link elements of a page to the database in order to display the proper information. As GitHub master, I was tasked with reviewing, resolving conflicts within, and merging pull requests from individual team members' branches into the development branch. In total, I made 53 submissions to our develop branch (this includes merging of pull requests).

Austin Sy-Velasco (Backend / DB manager):



Contributions - Austin Sy-Velasco

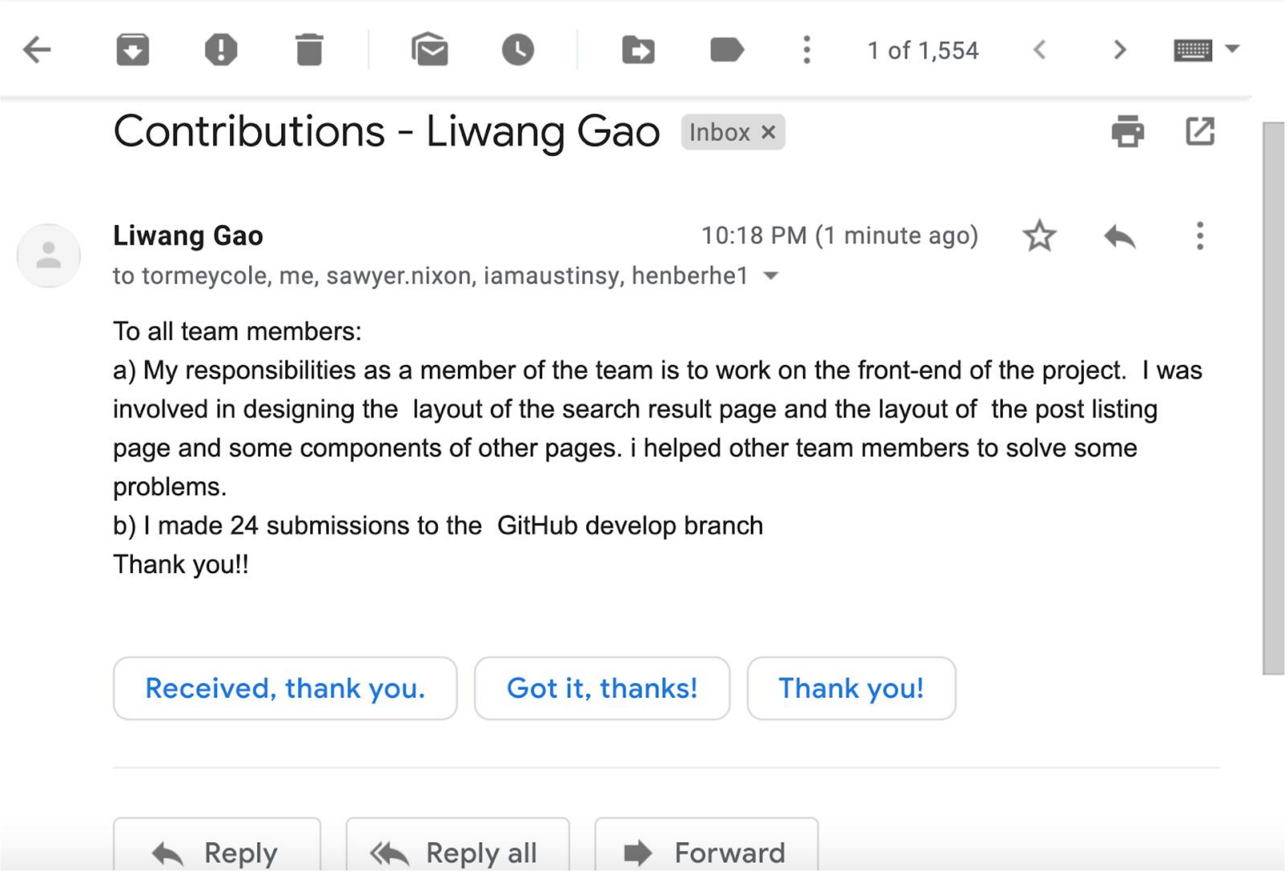
iamaustinsy

1:41 PM (8 hours ago)

to me, sawyer.nixon, liwangdfo5, henberhe1, Cole

My responsibilities as a member of the team was working on the back-end side of the project. I established how our data would be stored on our database by creating the tables and the relationships between them. I worked on the various controllers for the website that allows a user to do things on the website. This includes registering as a new user, searching for listings(I helped but latest iteration was done by Tasio), adding a listing under your favorites, and I created the admin functionality for the website. I also helped setup the Google Analytics for the website. I have about 16 commits to develop branch.

Liwang Gao (Frontend):



Contributions - Liwang Gao

Liwang Gao

10:18 PM (1 minute ago)

to tormeycole, me, sawyer.nixon, iamaustinsy, henberhe1

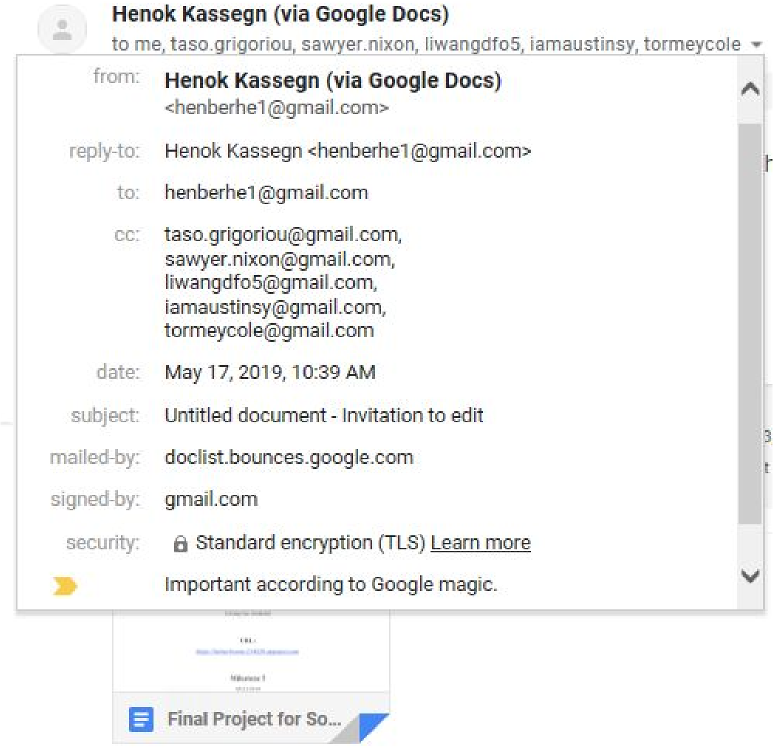
To all team members:

a) My responsibilities as a member of the team is to work on the front-end of the project. I was involved in designing the layout of the search result page and the layout of the post listing page and some components of other pages. i helped other team members to solve some problems.

b) I made 24 submissions to the GitHub develop branch

Thank you!!

Milestone 5 Google doc invitation to all team members



Henok Kassegn (via Google Docs)
to me, taso.grigoriou, sawyer.nixon, liwangdfo5, iamaustinsy, tormeycole

from:	Henok Kassegn (via Google Docs) <henberhe1@gmail.com>
reply-to:	Henok Kassegn <henberhe1@gmail.com>
to:	henberhe1@gmail.com
cc:	taso.grigoriou@gmail.com, sawyer.nixon@gmail.com, liwangdfo5@gmail.com, iamaustinsy@gmail.com, tormeycole@gmail.com
date:	May 17, 2019, 10:39 AM
subject:	Untitled document - Invitation to edit
mailed-by:	doclist.bounces.google.com
signed-by:	gmail.com
security:	🔒 Standard encryption (TLS) Learn more
➤	Important according to Google magic.

Final Project for So...

8. Post analysis:

Throughout this entire semester working on our product all the way from Milestone 0 to now Milestone 5, we have overcome a lot of obstacles and have a great product to show for. It was an incredible experience working my teammates, and I'm sure they can attest to that. We came from different backgrounds, and had very different skill sets, yet we came together and collaborated very effectively after a rough start. Our rough start mostly consisted of us not knowing each other well enough, and not knowing the technologies and frameworks needed in order to build a good product, all the while trying to create good quality milestone documents. Once we got to know each other better, and had a more open communication path via various messaging platforms (Text messages, google hangouts, in-person meetups, etc), we were able to come together to build an awesome product. A real technical challenge we faced in the beginning was that none of my teammates had any real web development experience. The overhead of learning and understanding the web and how to develop applications on that platform was something that needed to be taken account for, and we spent a lot of time as a team trying to bootstrap this learning process. I was proud of both my frontend and backend team for putting in the effort to learn these new technologies, and even moreso, was happy that they were willing to ask for help in crucial times. One thing I've learned is that no one knows everything, and asking for help early on can avoid a ton of time wasted. Even myself as team lead needed to ask my teammates questions at times, and I was never afraid to do so, which says a lot about our teamwork and how comfortable we felt with each other. Some things I could have handled differently was that I could have guided my team more early on so that they didn't feel overwhelmed learning the new technologies while creating the milestone documents. At times I felt like I could have been more clear on the requirements presented for us, and created a better plan in order to implement the required tasks. However, as a whole, I am very satisfied with my teammates and our final product, and am really looking forward to showcasing our product with my teammates.