# KASPER – PROGRAMMER DOCUMENTATION

SOFTWARE DEVELOPMENT TEAM:

Project Manager: Tushita Patel

Dev Lead: Kristof Mercier, Dylan Prefontaine

Test Lead: Jeremy Liau

Build Manager: Christopher Mykota-Reid (ChrisMR)

Developers: Gaurav Arora, Haotian (Justin) Ma, Melody (Tian) Zhao

Test Team: Christopher May (ChrisJ), Ryan Tetland

Documentation: Arianne Butler

# Contents

# 1.0 Introduction

Kasper home app is a real estate application that facilitates communication between sellers and prospective buyers. Sellers refers both to home owners wishing to sell, and to real-estate agents selling on their behalf. Buyers refers to users seeking to purchase a home, and to real-estate agents looking on their behalf. Sellers can post information and images of their properties (called Listings), and buyers can browse through all published Listings; filtering, liking, and disliking as they please. If a user likes a Listing, it will be saved to their Favourites. If a user dislikes a Listing, it will be removed from their Browse page permanently. Buyers should be able to initiate contact with sellers by sending a message containing their email and phone number. Although applications with similar functionality exist, Kasper home app aims to fill the gap for individuals wishing to buy or sell exclusively, purposely excluding renters and leasers. By dividing the market between buyers and renters/leasers, Kasper home app hopes to provide a direct, easy-to-use link between sellers and buyers.

# 2.0 System Overview

## 2.1 Tools and Frameworks

The front-end uses Node JS, which supplies the NPM package manager. NPM provides both Cordova and Ionic. Cordova builds the application on the three mobile platforms (iOS, Android, and Windows), as well in the Browser. Ionic extends Cordova and is built on Angular2; a development platform that uses Typescript, SCSS, and HTML.

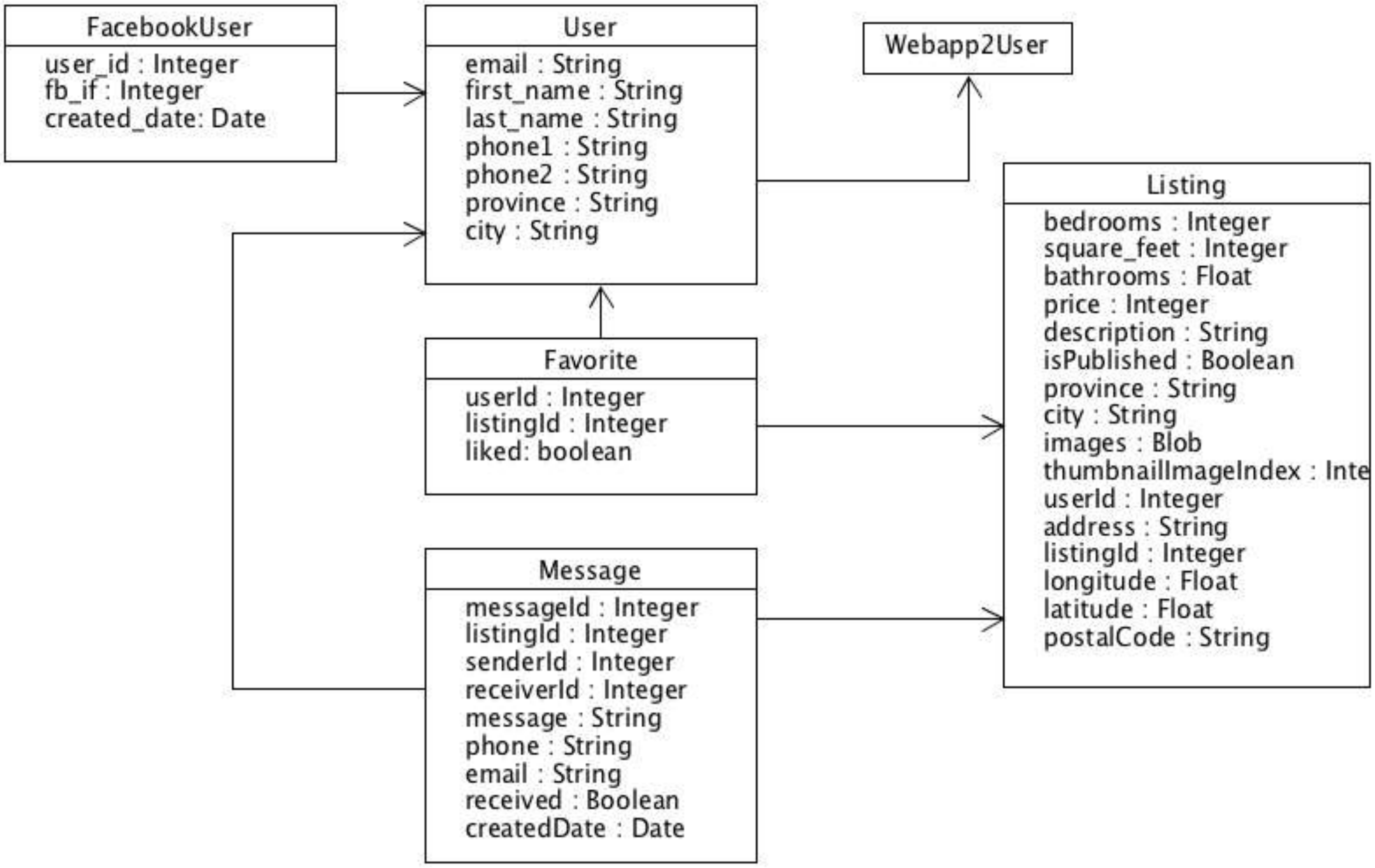
The back-end uses Google App Engine and Python 2.7 for the API calls. The interface for Google App Engine is provided by Web App. Web App is extended by Web App 2, which provides URI routing and better exception handling. Back-end testing is handled by the unittest module in Python. To create our mock test server, we are using Test Bed from the Google App Engine.

Test frameworks include Node JS and NPM which access Jasmine and Protractor. End-to-end tests are written in Jasmine and run on Protractor. We are employing Gulp to toggle a line in the kasper-config.ts file, so that testers can switch between the real database and the test database.

The Build uses Travis CI. For further information regarding build frameworks, see the Build Report on page 17 in the following document:

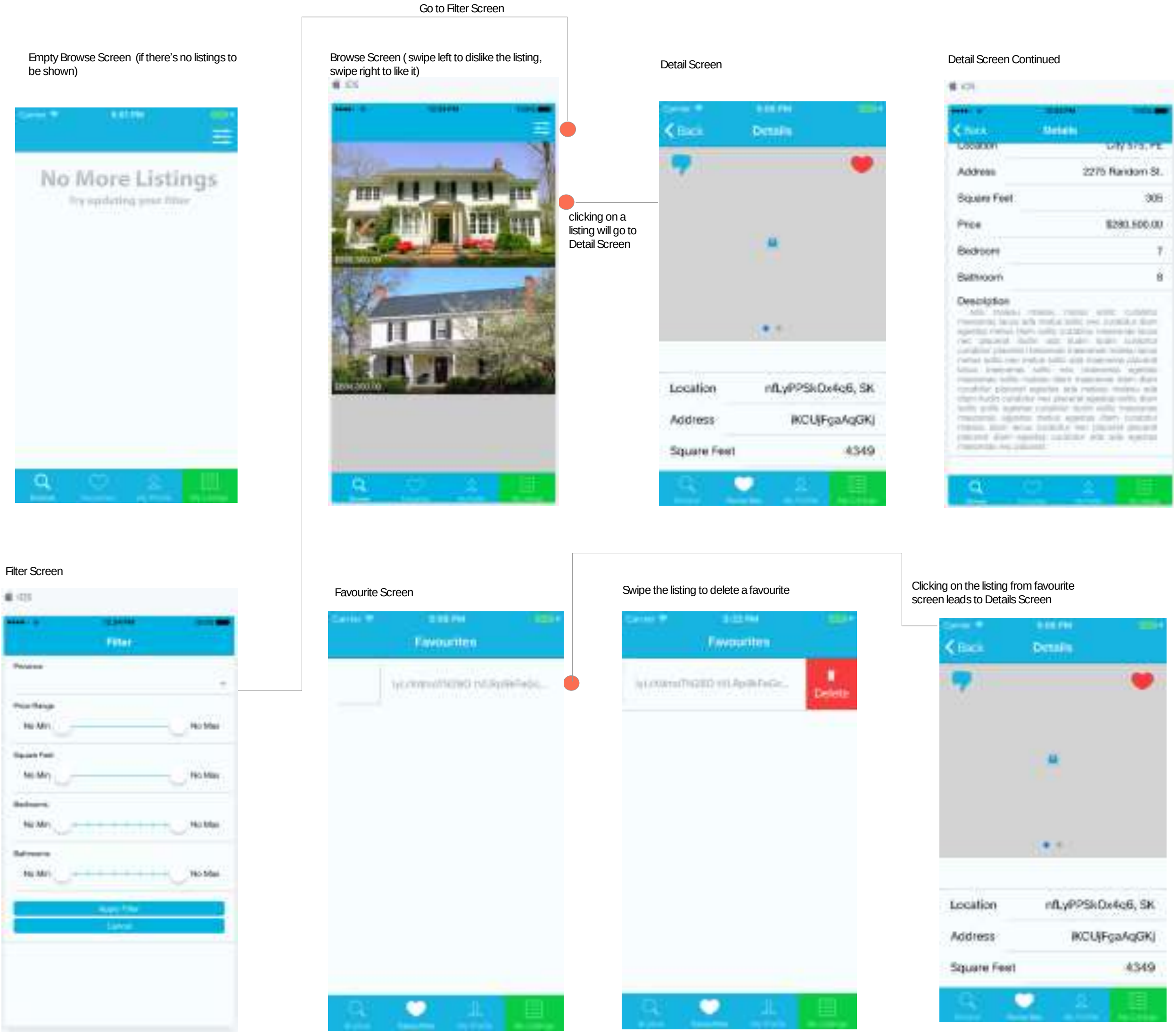
> https://github.com/CMPT371Team1/Documentation/blob/master/ID5-Documentation/ID5-ProjectDocumentation.pdf

## 2.2 Back-end (Database) Data Structures



## 2.3 UI Sequence Flow Diagrams

### iOS:



Go to Filter Screen

Empty Browse Screen (if there's no listings to be shown)

Browse Screen ( swipe left to dislike the listing, swipe right to like it)

Detail Screen

Detail Screen Continued

clicking on a listing will go to Detail Screen

Filter Screen

Favourite Screen

Swipe the listing to delete a favourite

Clicking on the listing from favourite screen leads to Details Screen

Location nfLyPPSkDx4c6, SK

Address IKCUjFgaAqGKj

Square Feet 4349

Address 2275 Random St.

Square Feet 308

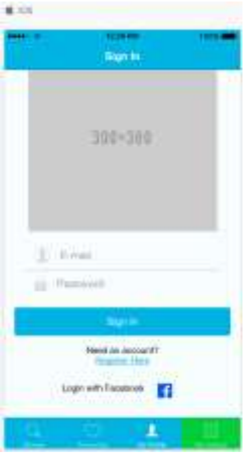
Price $280,500.00

Bedroom 7

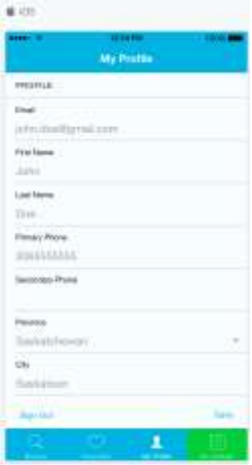
Bathroom 8

## iOS continued:

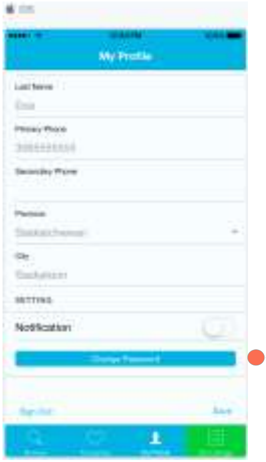
Profile Screen (not signed in)



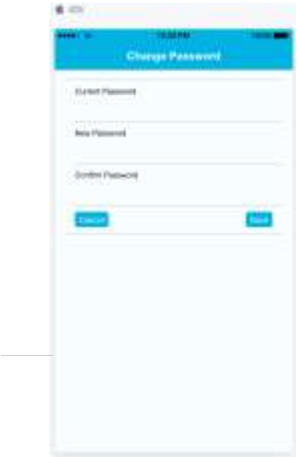
Profile Screen (signed-in)



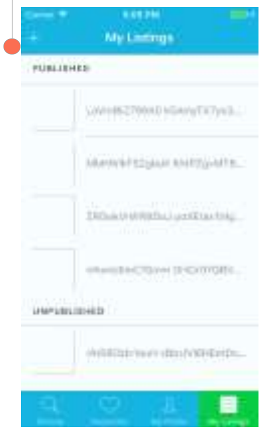
Profile Screen Continued (signed-in)



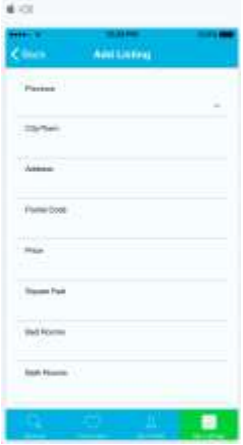
Change Password Screen



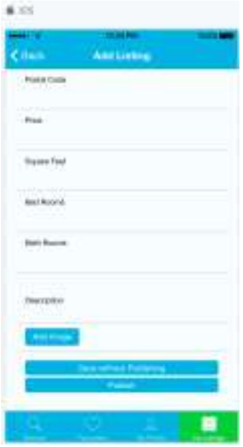
My Listings Screen



Add Listings Screen



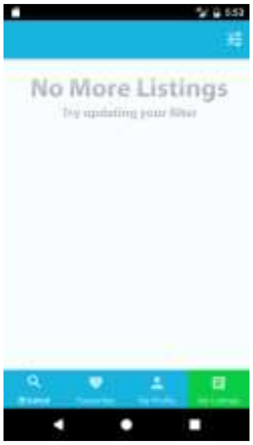
Add Listings Screen ( Continued)



# Android:

Go to Filter Screen

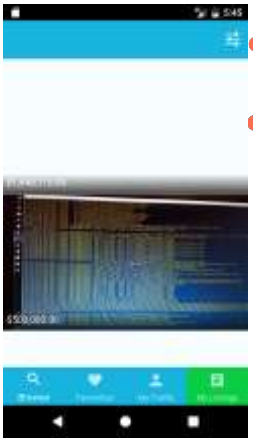
Empty Browse Screen (if there's no listings to be shown)



No More Listings

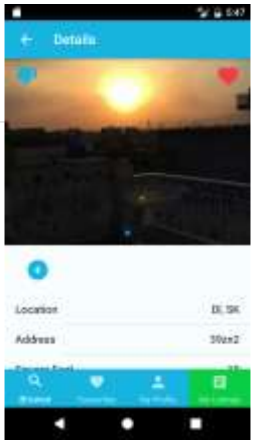
Try updating your filter

Browse Screen ( swipe left to dislike the listing, swipe right to like it)



clicking on a listing will go to Detail Screen

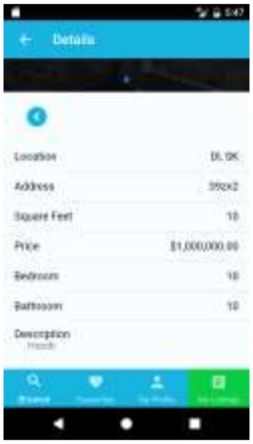
Detail Screen



Details

| | |
|---|---|
| Location | DI, SK |
| Address | 392x2 |

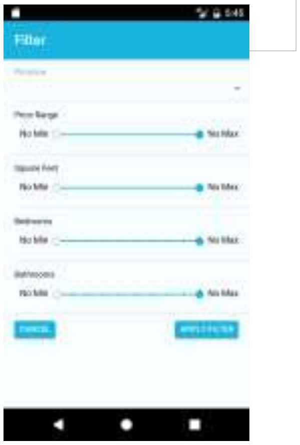
Detail Screen Continued



Details

| | |
|---|---|
| Location | DI, SK |
| Address | 392x2 |
| Square Feet | 10 |
| Price | $1,000,000.00 |
| Bedroom | 10 |
| Bathroom | 10 |
| Description | |

Filter Screen



Filter

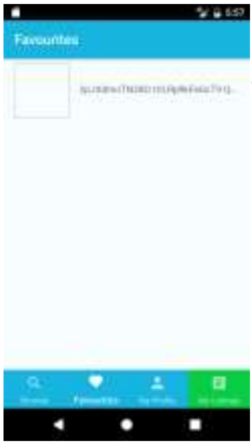
Price Range: No Min – No Max

Square Feet: No Min – No Max

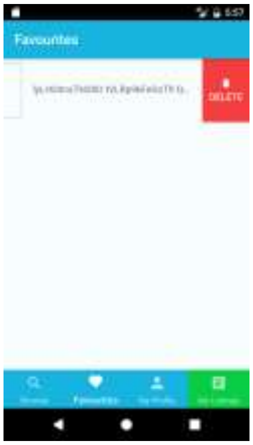
Bedrooms: No Min – No Max

Bathrooms: No Min – No Max

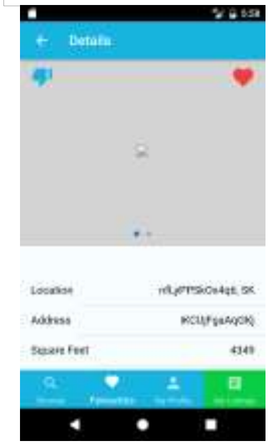
Favourite Screen



Swipe the listing to delete a favourite



Clicking on the listing from favourite screen leads to Details Screen

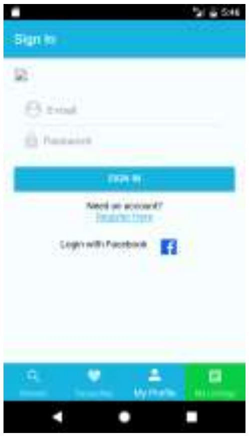


Details

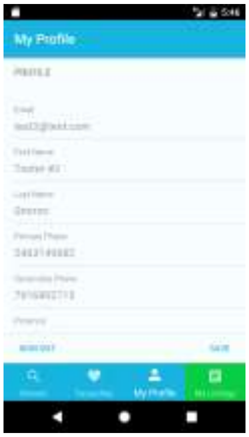
| | |
|---|---|
| Location | nfLj#FPSkOs4q6, SK |
| Address | lKCUjFgaAqOKj |
| Square Feet | 4349 |

## Android continued:

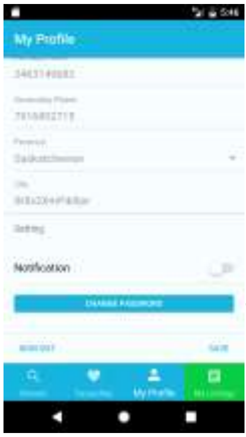
Profile Screen (not signed in)



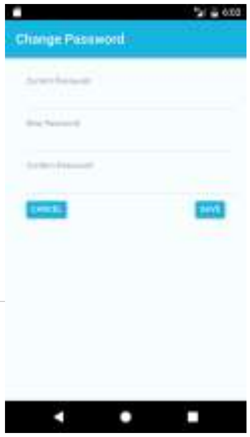
Profile Screen (signed-in)



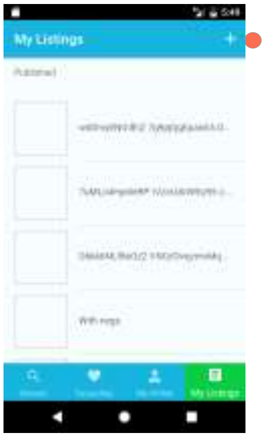
Profile Screen Continued (signed-in)



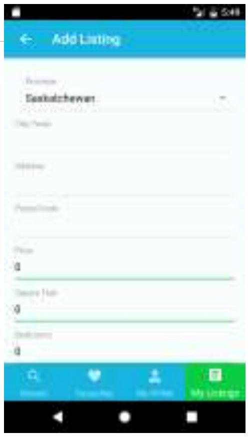
Change Password Screen



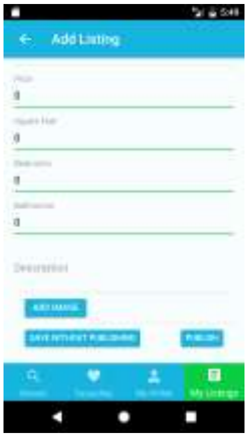
My Listings Screen



Add Listings Screen



Add Listings Screen ( Continued)



## 3.0 Code Base

Below is a link to the project’s GitHub:

https://github.com/CMPT371Team1/Project

- Front and back-end code is located in the /src and /server-gae folders, respectively

# 4.0 Installation Instructions

## 4.1 Front-end Setup

1. NodeJS can be installed at the following link:

   https://nodejs.org/en/download/

   NodeJS version 6 or higher is required. You can verify your currently installed version number by typing node -v in the command line.

2. Install Cordova and Ionic dependencies by typing npm install -g ionic cordova into the command line.
   If you are not running your command prompt with administrator privilege, you may need to use sudo npm install -g ionic cordova
3. Clone repository
4. Change current working directory to repository root
5. run npm install
6. run ionic state restore
7. Run the front-end code using ionic serve

## 4.2 Back-end Setup

1. In the command prompt, run curl https://sdk.cloud.google.com | bash
2. Run gcloud components install app-engine-python
3. Run gcloud components install app-engine-python-extras
4. Python 2.7 or higher is required, and pip must be for version 2.7. To verify your currently installed versions of Python and pip, run python -v and pip -v, respectively
5. Run sudo pip install webapp2
6. Run sudo pip install webob
7. Run sudo pip2 install pyyaml
8. Two Options:
   a. Modify the PYTHONPATH variable to the location where Google App Engine is installed
      OR
   b. In PyCharm:
      - Select Settings/Preferences
      - Search for “Project Structure” in the search bar
      - Click Add Content Root
        - Find Google App Engine in the pop up window
        - Select and click OK
9. Clone the project repository and import into PyCharm
10. Select the project, and open Settings/Preferences
11. Search “Project Structure” in the search bar and open
12. Select server-gae folder and select Mark as: Sources – click OK
13. To run the server, open the terminal and run dev_appserver.py server-gae/app.yaml

## 4.3 Testing Set-up

- The two frameworks used for testing are Protractor and Jasmine. Protractor is an end-to-end test framework for angular and angularJS applications. Jasmine is a framework for testing JavaScript code with an easily readable syntax. Also used is the jasmine-spec-reporter, which aids in giving meaningful test output when running the E2E tests.
- Use npm to install Protractor with: `npm install -g protractor` (This project uses Protractor Version 5.0.0)
- Use npm to install jasmine with: `npm install -g jasmine`
- Use npm to install jasmine-spec-reporter with: `npm install jasmine-spec-reporter -save-dev`
- To run the tests, use the protractor command on the configuration files found in the base directory of the project. Each browser has its own configuration file (IE. protractor firefox-e2e-tests.conf.js, or `protractor chrome-e2e-tests.conf.js`).
- The tests can be found in tests/e2e-tests. All test files must end in tests.js, unless otherwise specified, within the configuration files.
- server-gae/extras/Random_Models.py contains the randomization of test database information
- server-gae/extras/Test_Database.py contains the hardcoded data for the test database
- Src/app/kapser-config.ts contains the database API_URl
  - Test database: http://cmpt371g1.usask.ca:4040
  - Main database: http://cmpt371g1.usask.ca:8080
- To reset the test database, use: `curl -v POST -d '{}'`
  - Curl must be installed: http://cmpt371g1.usask.ca:4040/initDBTESTERS

# 5.0 Future Programmers

## 5.1 Unimplemented Requirements

- Sign in / Sign up using Facebook
  - This feature is incomplete in the front-end, but complete in the back-end.
- Messaging Functionality
  - This feature is incomplete in the front-end, but complete in the back-end. It includes functionality for sending initial messages to sellers, marking read messages, and deleting messages. Once implemented, it will need to be integrated with the back-end and tested.
- Edit listing
  - This feature is incomplete in the front-end, but complete in the back-end
- Push notifications
  - The app should be able to track changes in the database and trigger notifications to interested buyers
  - Sellers should receive a notification if a prospective buyer sends a message regarding one of their Listings
  - Buyers can elect receive "hot list" notifications for one or more of the following:
    - A Seller publishes a Feature Listing
    - A new Listing is added that fits some predetermined criteria (i.e. neighborhood, price range, seller, etc.)
    - Price change on a Favourited Listing

        - A Listing is published that contains similarities to past searches
    - Sellers can elect to receive notifications regarding their personal Listings:
        - A Listing has been Favourited by a user
        - A buyer sets a price watch on a Listing
- Integration with Google Maps
    - This feature should show Listings with regards to their physical location on a map. The Google Maps functionality should be accessible from the Detail page.
- Forgot password
    - This feature should allow a user to request an email or text message containing reset information
- Re-order a Listing's images.
    - This feature should allow a seller to select a primary image to be displayed on the browse page, and to re-order a Listing's images.
- Super Admin User
    - A Super Admin User can access the following additional functionality:
        - Publish a Listing under any user
        - Edit any Listing
        - Remove any Listing
- API Versioning
- Edit User

## 5.2 Known Bugs – Defect Report

| | |
|---|---|
| **Bug:** | Images don't appear for listings |
| **Reproducibility:** | 100% |
| **Steps taken:** | I. Navigate to Browse page (Browse page must not be empty) |
| **Variations:** | None |
| **Resolved:** | Yes |
| **Screenshot:** | None |

| | |
|---|---|
| **Bug:** | Repeated presses of "Publish" while adding a listing will result in multiple instances of that listing being created |
| **Reproducibility:** | 100% |
| **Steps taken:** | I. Sign in on any account<br>II. Attempt to add a listing with correct information<br>III. When publishing the listing, continue to press "Publish" |
| **Variations:** | Depending on the speed of the server, can create a lot of listings (slow) or only a few (quick). |
| **Resolved:** | Yes |
| **Screenshot:** | None |

| | |
|---|---|
| **Bug:** | Changing password doesn't change password |
| **Reproducibility:** | 100% |
| **Steps taken:** | IV. Sign in on any account<br>V. Attempt to change password on My Profile page -> Change password<br>VI. Enter valid credentials<br>VII. Sign out and attempt to sign in with new password. |

| | |
|---|---|
| | The password is unchanged. |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Username does not trim white space when signing in |
| Reproducibility: | 100% |
| Steps taken: | I. Attempt to sign in using any registered account, but include whitespace in the username field |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Change password closes the page even if the credentials are wrong |
| Reproducibility: | 100% |
| Steps taken: | I. Sign in on any account<br>II. Navigate to My Profile page<br>III. Attempt to change the password |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Change password doesn’t prompt the user that their old password is incorrect |
| Reproducibility: | 100% |
| Steps taken: | I. Sign in on any account<br>II. Navigate to My Profile page<br>III. Attempt to change the password, using the incorrect “Current Password” |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Updating Province in User info doesn’t keep province selected |
| Reproducibility: | 100% |
| Steps taken: | I. Sign in on any account<br>II. Navigate to My Profile page<br>III. Change the province of the user |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Wrong error message for favourites tab when not signed in |
| Reproducibility: | 100% |
| Steps taken: | I. While not signed in, navigate to the favourites tab. |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | No internet connection not handled |
| Reproducibility: | 100% |
| Steps taken: | I. Use the application without Wi-fi and witih Airplane mode turned on (on device), or disconnect computer from internet (on browser). |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Like/Dislike on “Browse” page breaks cycling of Listings on details page |
| Reproducibility: | 100% |
| Steps taken: | I. Like/Dislike any listing by swiping on the “Browse” page |

| | |
|---|---|
| | II. View detailed information of any listing, and attempt to cycle listings with arrow buttons |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Fields while registering user scroll up and out of view |
| Reproducibility: | 100% |
| Steps taken: | I. On a mobile device, navigate to the registration page and attempt to register a new user |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | No feedback when liking/disliking a listing |
| Reproducibility: | 100% |
| Steps taken: | I. Like/dislike any listing while signed in |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Password fields clear on edit |
| Reproducibility: | 100% |
| Steps taken: | I. On either registration page, sign-in page, or change password page, enter information into the password field.<br>II. Select out of the field<br>III. Select back into the field |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | On the filter page, all range bars extend and shrink when  the range on the end of the bars goes from text to a number |
| Reproducibility: | 100% |
| Steps taken: | I. Navigate to the filters page<br>II. Slide the dual sliders. The range bars incorrectly shrink and extend. |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Like/Dislike on “Browse” page breaks cycling of Listings on details page |
| Reproducibility: | 100% |
| Steps taken: | I. Like/Dislike any listing by swiping on the “Browse” page<br>II. View detailed information of any listing, and attempt to cycle listings with arrow buttons |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | While adding a listing, no feedback given when inputting invalid input for postal code, bed, bath, price, square feet. |
| Reproducibility: | 100% |
| Steps taken: | I. Sign in<br>II. Navigate to Add Listing page<br>III. Attempt to add a listing without entering information in postal code, bed, bath, price, square feet fields. |
| Variations: | None |
| Resolved: | Yes |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Error checking on front end and back end are inconsistent |
| Reproducibility: | 100% |
| Steps taken: | IV. Attempt to register a new user<br>V. Enter a phone number with spaces, enter an email with 2 periods.<br>VI. User cannot continue registration |
| Variations: | None |
| Resolved: | No |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Error checking on front end and back end are inconsistent |
| Reproducibility: | 100% |
| Steps taken: | I. Like/Dislike any listing by swiping on the “Browse” page<br>II. View detailed information of any listing, and attempt to cycle listings with arrow buttons |
| Variations: | None |
| Resolved: | No |
| Screenshot: | None |

| | |
|---|---|
| Bug: | While signing in with an email address created through registration, using incorrect password won’t give an error prompt. |
| Reproducibility: | 100% |
| Steps taken: | I. Register a new user<br>II. Sign out from that user<br>III. Attempt to sign back in to that user, but with an invalid password, but valid e-mail address |
| Variations: | The hard-coded users in the test database always give the error prompt. |
| Resolved: | No |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Error checking on front end and back end are inconsistent |
| Reproducibility: | 100% |
| Steps taken: | I. Like/Dislike any listing by swiping on the “Browse” page<br>II. View detailed information of any listing, and attempt to cycle listings with arrow buttons |
| Variations: | None |
| Resolved: | No |
| Screenshot: | None |

| | |
|---|---|
| Bug: | User activity is not reset to default settings when switching users on the same device in the same session. |
| Reproducibility: | 100% |
| Steps taken: | I. Sign-in on any account<br>II. Open a listing on the browse page<br>III. Sign out from that account<br>IV. Sign in with a different account than the previous one<br>V. Navigate to Browse page, the listing is the same. |
| Variations: | When the app is closed on the device, everything is reset. |
| Resolved: | No |
| Screenshot: | None |

| | |
|---|---|
| Bug: | Able to input negative values when adding a listing. |
| Reproducibility: | 100% |

| | |
|---|---|
| **Steps taken:** | I. Sign in to any user and navigate to Add Listing<br>II. Add a listing with valid information, but make the numerical fields negative. |
| **Variations:** | None |
| **Resolved:** | No |
| **Screenshot:** | None |

| | |
|---|---|
| **Bug:** | Phone number can contain letters |
| **Reproducibility:** | 100% |
| **Steps taken:** | I. Register a new user<br>II. In the phone number field, use characters rather than numbers.<br>III. Registration is able to proceed. |
| **Variations:** | None |
| **Resolved:** | No |
| **Screenshot:** | None |

| | |
|---|---|
| **Bug:** | Browse page doesn't refresh after coming back from detailed listing page when liking/disliking. |
| **Reproducibility:** | 100% |
| **Steps taken:** | I. While signed in, navigate to a listings details page.<br>II. Like/dislike that listing<br>III. Go back to the main Browse page<br>IV. The listing currently liked/disliked still appears in the list (It should not) |
| **Variations:** | None |
| **Resolved:** | No |
| **Screenshot:** | None |

| | |
|---|---|
| **Bug:** | Editing and publishing non-published listings is not functioning. |
| **Reproducibility:** | 100% |
| **Steps taken:** | I. While signed in, attempt to edit an unpublished listing of that user. |
| **Variations:** | None |
| **Resolved:** | No |
| **Screenshot:** | None |

| | |
|---|---|
| **Bug:** | Error checking on front end and back end are inconsistent |
| **Reproducibility:** | 100% |
| **Steps taken:** | I. Like/Dislike any listing by swiping on the "Browse" page<br>II. View detailed information of any listing, and attempt to cycle listings with arrow buttons |
| **Variations:** | None |
| **Resolved:** | No |
| **Screenshot:** | None |

## 5.3 Tested/Untested Areas

- Edit listing has not been tested because it is not implemented in the back-end
- Firefox and Chrome Browsers are tested, but no other browsers
- Manual iOS testing has not been completed

## 5.4 Tricky Areas

- Get Listings using Filter Functionality:
    - It is not possible to query multiple variables at once with Google App Engine, so each variable needs to be queried separately. The intersection of the queries yields the correct results.

## 5.5 Integral Areas

- The check_required_valid function in server-gae/extras/Required_Fields.py
    - This function is used by every API call and checks the JSON for correctness
- Get user by ID/password/token in server-gae/web_apis/Change_Password.py
    - User.get_by_id: returns a User object
    - self.auth.get_user_by_password: returns a dictionary of the user's information
    - User.get_by_auth_token: returns a tuple (Boolean, User) where Boolean represents whether the user exists or not, and User represents the user data
- In the Initialize_DB_Testers file in server-gae/web_apis:
    - This file is responsible for reinitializing the test data base by deleting all data and creating dummy data for testing

# 6.0 Deployment and Maintenance

## 6.1 Deployment

For detailed information on deploying a Python app, see the following link:

https://cloud.google.com/appengine/docs/standard/python/tools/uploadinganapp

## 6.2 Maintenance

As time progresses, packages get updated and dependencies change. It is important to ensure that the build is functioning properly throughout development.

**Build Maintenance**: Travis CI is free, open source software that is easy to with any repository. Only slight modifications to the ".travis.yml" file are needed to use it in a personal repository.