**Milestone 3 Part B: Group 2 Validating Group 3**

**Group 2 : Team G2**

Jacob Barrick,

Colin Blower,

Honming Chen,

Curtis Davis,

Victor Frandsen,

Sam Lanzo,

Andrew Miller,

Yu-Ching Wang

Table Of Contents

* Validation of Core Features
  + A.1 5 Proposed Features
  + A.2 Unified Search
  + A.3 Scrapped Content
  + A.4 Presentation of Content
  + A.5 Solution for Duplicate Content
* Testing with Selenium
* Behavior Modeling
* Report
  + Illustrative Flow of Website
  + Description of Setup and Use

Summary

|  |  |
| --- | --- |
| A.1 5 Proposed Features | 14% / 20% |
| A.2 Unified Search | 8% / 10% |
| A.3 Scrapped Content | 8% / 10% |
| A.4 Presentation of Content | 8% / 10% |
| A.5 Solution for Duplicate Content | 9% / 10% |
| Sequence Diagram | 6% / 10% |
| Illustrative Flow of Website | 7% / 10% |
| Description of Setup and Use | 7% / 10% |
| Free Points | 10% / 10% |
| Total | 77% / 100% |

**Validation of Core Features**

A.1 Do you see a working websites at least 5 of proposed features? List each feature and comment on what is and what is not working (4 x 5%= 20%)

* Unified Search Form - You can manually type incorrect dates and you can type a location that does not exist. The rest of the search form works fine. -1%
* Unified Output - Price is broken for homeaway and airbnb results are always on top. -1%
* Suggested Location - Works fine, good layout
* Local Restaurants - Need more information about the restaurants and sort the restaurants. -1%
* Comparator - Not enough room to compare the websites. -1%
* Google Map - Works the first time but it does not work again until a new search is performed. -2%

A.2 Do you see a unified input applicable to search on two websites that they are assigned to mashup? (10%)

* You can manually type incorrect dates and you can type a location that does not exist. The rest of the search form works fine. -2%

A.3 Do you see scrapped content from one or two given websites? Is the combined content presented with a uniform style? If not, offer improvement suggestions. (10%)

* Price is broken for homeaway and airbnb results are always on top. The title will display ‘&’ as ;amp -2%

A.4 Is the presentation style for merged content user friendly to users? How can it be improved? (10%)

* Not enough room to compare the websites in the comparator. If the window size is too narrow then the map gets dropped off the page. -2%

A.5 Is their suggested approach to avoid duplicated content from the two give websites reasonable? Comment on how it can be improved. (10%)

* We are going to compare the listing names between both sites and, if they match, only display one of them instead of the duplicate. In addition, we are showing the user where each result was obtained from. This should catch any duplicates from both of the sites Bed and Breakfasts and limit this slim chance even further.
* You can match hotel name in addition to listing name. -1%

**Testing with Selenium**

* Selenium is unable to test the comparator website because the comparator website does not have a title or a javascript variable associated with it when it is opened in a new tab, because of this Selenium is unable to select the comparator website and access its elements. An easy fix would be to give the comparator website a title.

**Behavior Modeling**

Does their sequence diagrams adequately describe the overall design of their website? (10%)

* There are not enough returns for sequence to be maintained and many sequences seem to happen concurrently. -4%
* Break into multiple sequence diagrams for increased clarity.

**Report**

Is the illustrated flow of their website equipped with enough snapshots and comments for one to understand? If not, what is missing? (10%)

* This report is well documented. Good Job!

For each of the features implemented in A1, is the description for software setup and used clear enough for someone like you to re-do it? Comment on which one is not well reported. (10%)

* Unified Search Form - More details would help
* Unified Output - More details would help
* Suggested Location - More details would help
* Local Restaurants - More details would help
* Comparator - More details would help
* Google Maps - More details would help
* Suggestions
* Loading full JQuery library causes homepage to load slowly
  + Homepage about 1MB
  + <http://www.websiteoptimization.com/services/analyze/>
  + May be possible to use a smaller library for cal (pickadate.js)
* Results page could be formatted better
  + Add more space and maybe lines/colors between each listing
  + Add more space between listings and images
  + Yelp feature covers listings on small windows
* -3%

**Bed, Breakfast, and Beyond**

**Milestone 3 Part A**

Tan Nguyen, Peter Wong, Brandon Rousey, Wilson Yee, Suzy Hanko, Carlos Pereira, Chi Koo, William McCarty, ChingYing Kuo

November 29, 2012

CS 160

Group 3

Assigned Domain: Bed and Breakfast

**Contents**

**Summary Table** …............................................................... 3

**A3** **Default Presentation Style** …............................................. 4

**A4 Approach to Avoid Duplicated Content** ............................... 4

**Workflow** …............................................................................... 5

**How Features Were Accomplished** …...................................... 18

1: Unified Search............................................................... 18

2: Unified Output............................................................... 18

3: Suggested Results........................................................ 19

4: Local Restaurants........................................................... 19

5: Google Map.................................................................... 19

6: Comparator.................................................................... 20

**Collaboration List**........................................................................ 21

**Risks, Gaps and Things to Build On**......................................... 22

**Sequence Diagram …........................**......................................... 23

**Sites Used** …..................…...................................................... 24

**Summary Table of Features :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Why is it useful to users | Why is it innovative | Applicable to/Similar to | Proposed by |
| g3.1 Unified Search Form | Users will be able to quickly search for a bed and breakfast venue. | It eliminates unnecessary distractions for an easy to use interface. | Car, job, and B&B /g1.3 | Tan Nguyen, Peter Wong, Brandon Rousey (group 3) |
| g3.2 Unified Output | Gives user a consistent result page making it easier to find a listing | Provides a simple to use way to view the results from airbnb, homeaway, and yelp | Car, job, and B&B |  |
| g3.3 Suggested location | This feature will help give the undecided users a starting point in their search. | This provide a quick one-click access to the website. | Car, job, and B&B (be sure to show a location useful to users) | Tan Nguyen, Peter Wong, Brandon Rousey (group 3) |
| g3.4 Local Restaurants | A listing of local restaurants can help users to plan their vacation | Provides additional information of the surrounding area | B&B | Carlos Pereira, Wilson Yee, Suzy Hanko (group 3) |
| g3.5 Comparator | Provides an easy way to compare one location with another | Gives user a visual way to compare prices, reviews, and other parameters | Car, job, and B&B | Chi Koo, William McCarty, Ching Ying Kuo (group 3) |
| g3.6 google map | give geolocation of search results | User can visualize the location to determine which hotel to book | car, job | Sam Lanzo (group 2), Thanh Au (group 1) |

**A.4 Default Presentation Style**

Our presentation style provides the User with any and all fields most Users would be interested in. When a User first begins to make a search, while the User does want to get the best deals, it is not job of the User fill in a price they're looking for. Our site will be sorting through both sites and give the best prices for any Bed and Breakfast they’re looking at. The User would only need to input the location where they want to find a place, the start and end dates, how many guests and how many rooms.are needed. So when a User comes to our site, it’s all we tell the User to give us. This style makes it a straightforward process for the User. It also helps our site to process exactly what they’re wanting, and with the best deals possible. This helps to avoid clutter on the search page and hide unnecessary information from the User.

The result page is going to be sorted by showing the best deals, lowest prices, and best ratings first, while making it easy for the user to follow. By displaying the results in this way, it should make it very easy for the User to get the best deals they are looking for. Right now, both sites give their listings to our site effectively, and just a sorting algorithm is needed to be implemented in order to successfully complete our goal here.

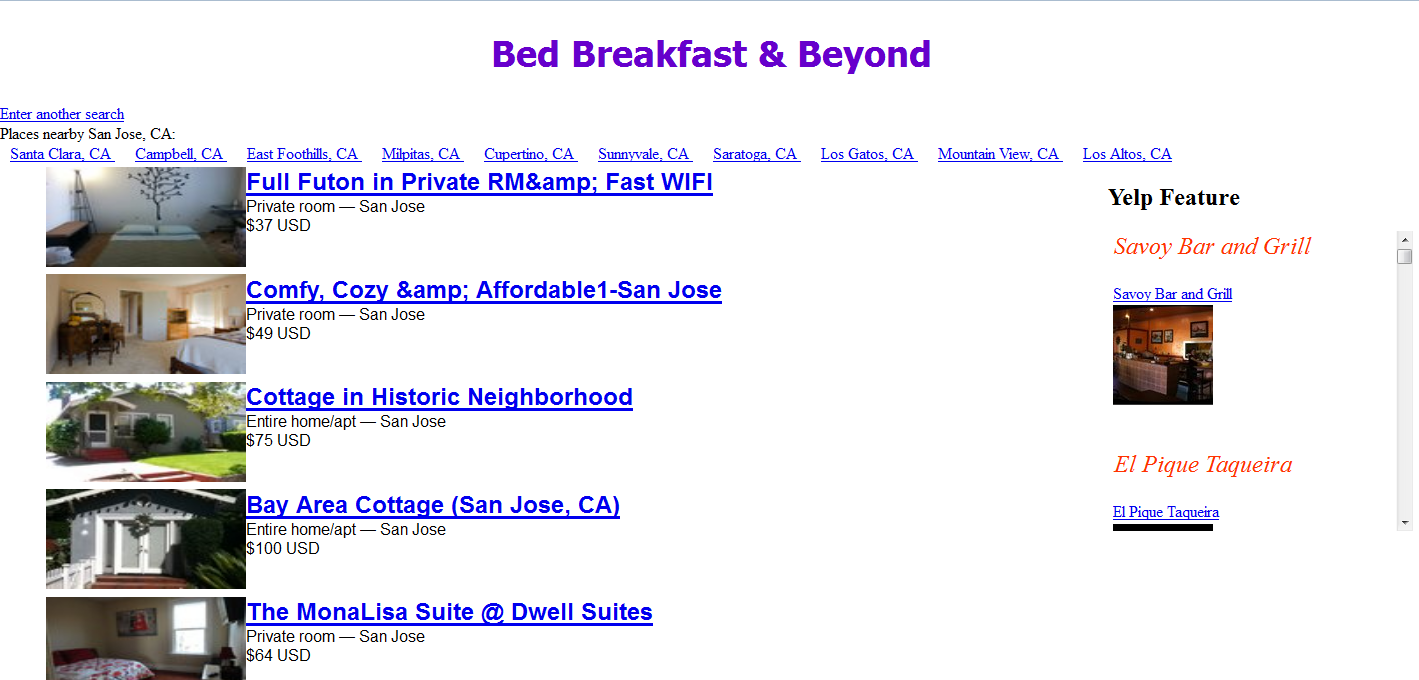
**A.5 Approach to Avoid Duplicated Content**

For our site, there is a very subtle and slim chance of duplicate content. The reason is because one is mostly US based while the other is UK based. This means the odds of duplicate information is very slim. However, duplicates are still present, so we are going to handle it as such. Both of these sites have different descriptions for each listing, with only little similarities, so the aspect of the duplicates we’re going to focus on are the names of the Bed and Breakfast listings. We are going to compare the listing names between both sites and, if they match, only display one of them instead of the duplicate. In addition, we are showing the user where each result was obtained from. This should catch any duplicates from both of the sites Bed and Breakfasts and limit this slim chance even further.

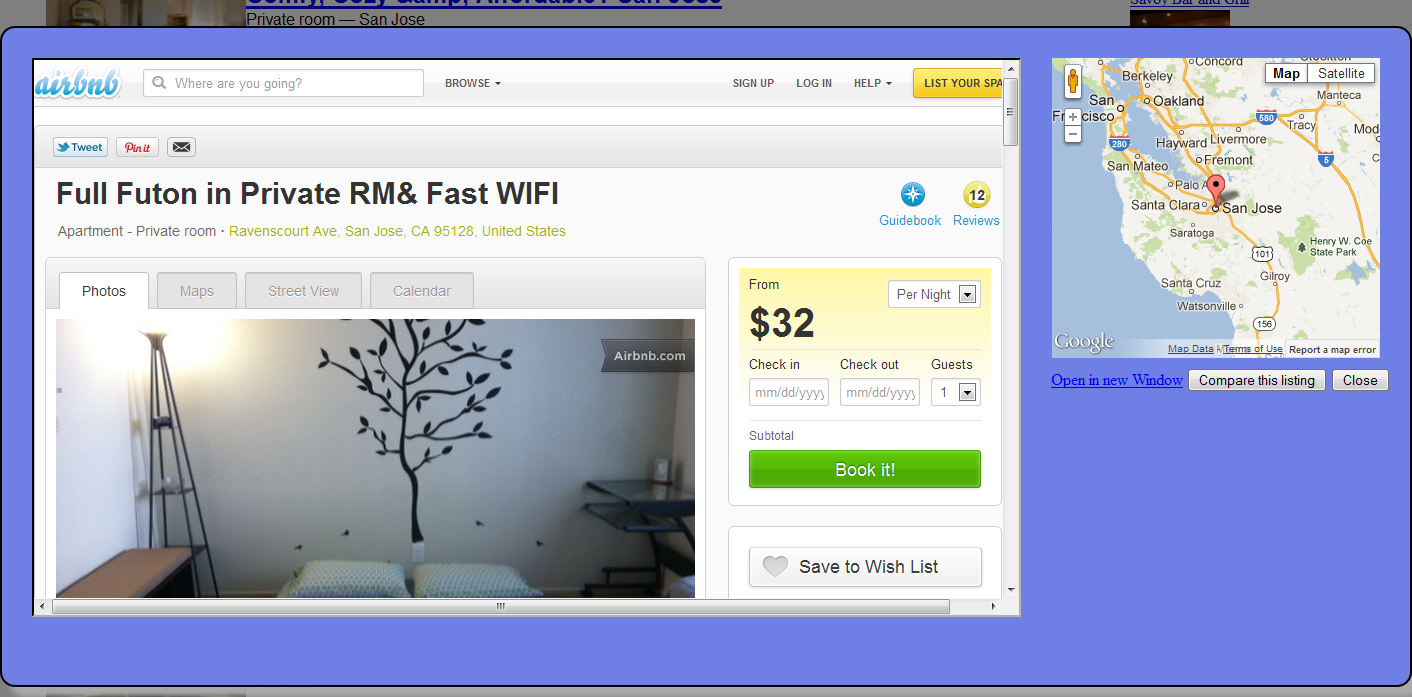
**Workflow**



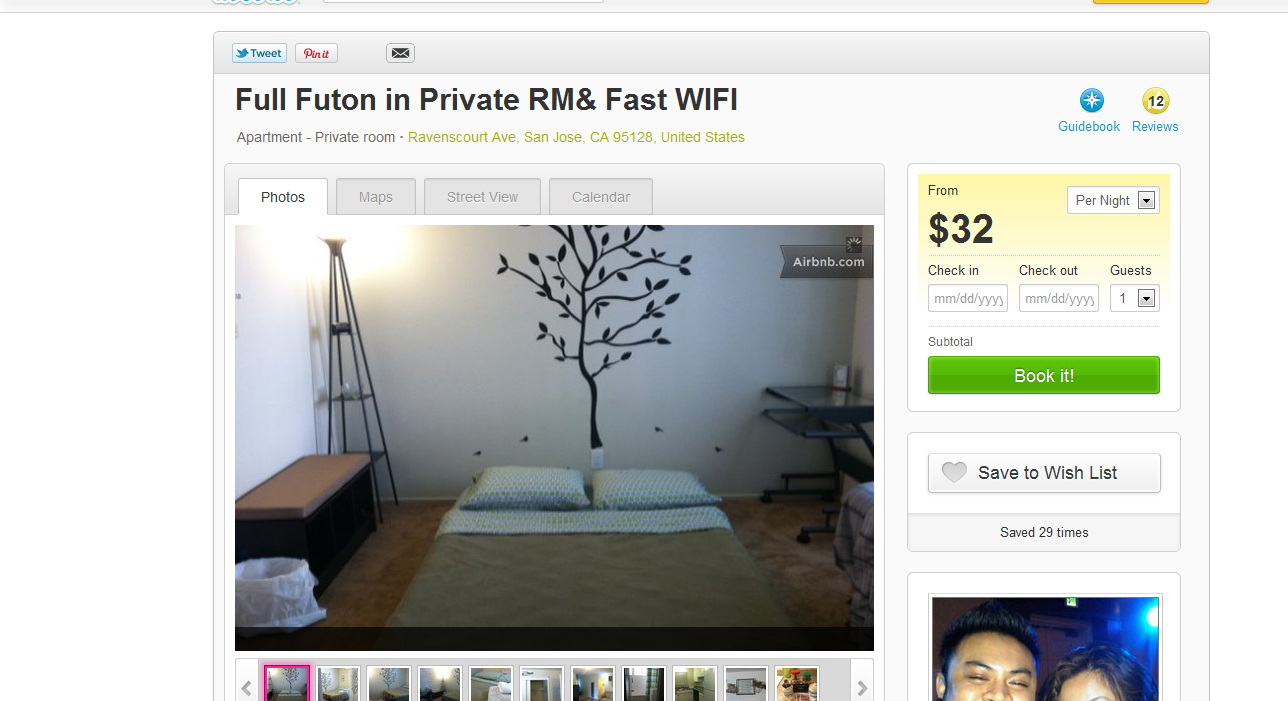
1. The User initially sees the starting search screen where they are prompted to enter in their starting search.



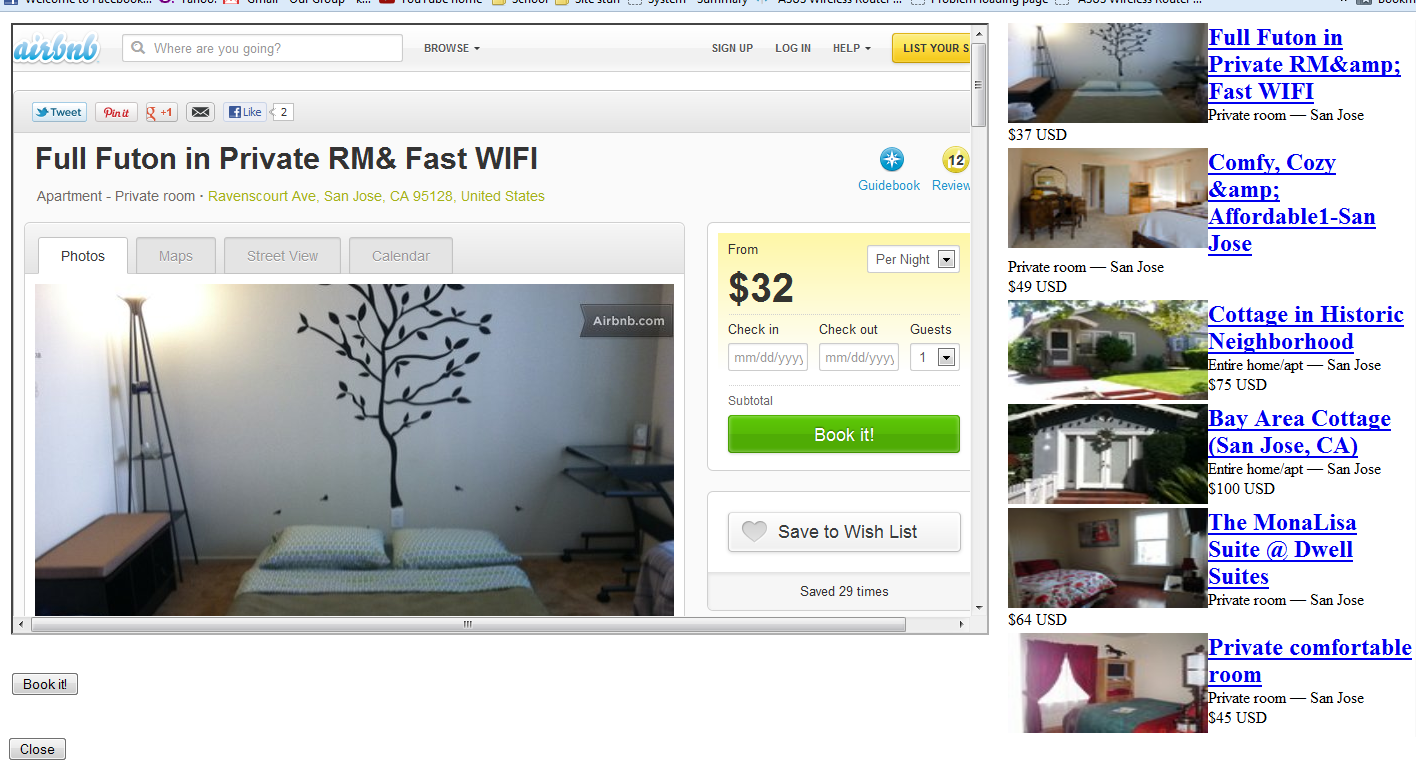
1. The User will then be redirected to the default result page (output.html) after the search button is clicked, which will show the results of the search they just made.



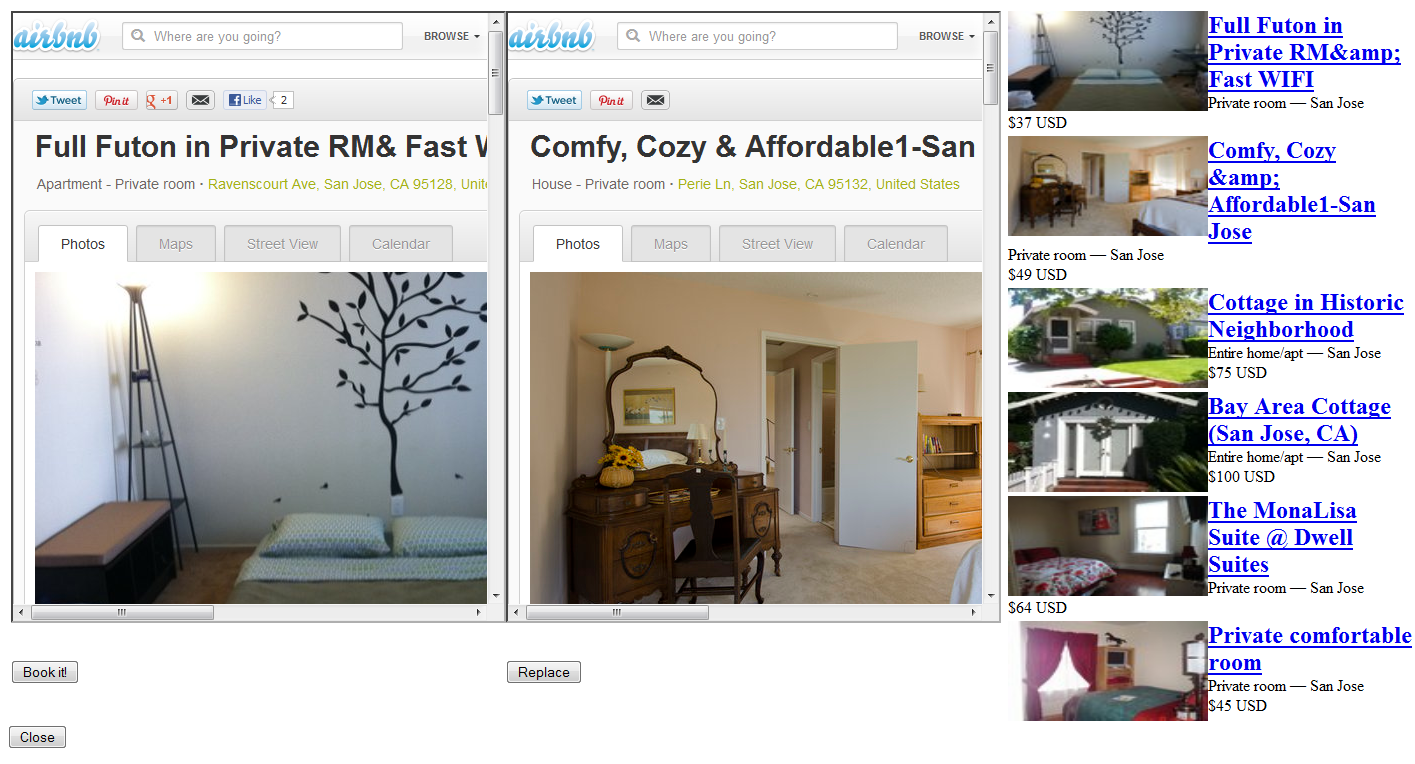
1. If a User clicks on a listing, a new window will come up to show the Google map of where it is located, and show information on the Bed and Breakfast.



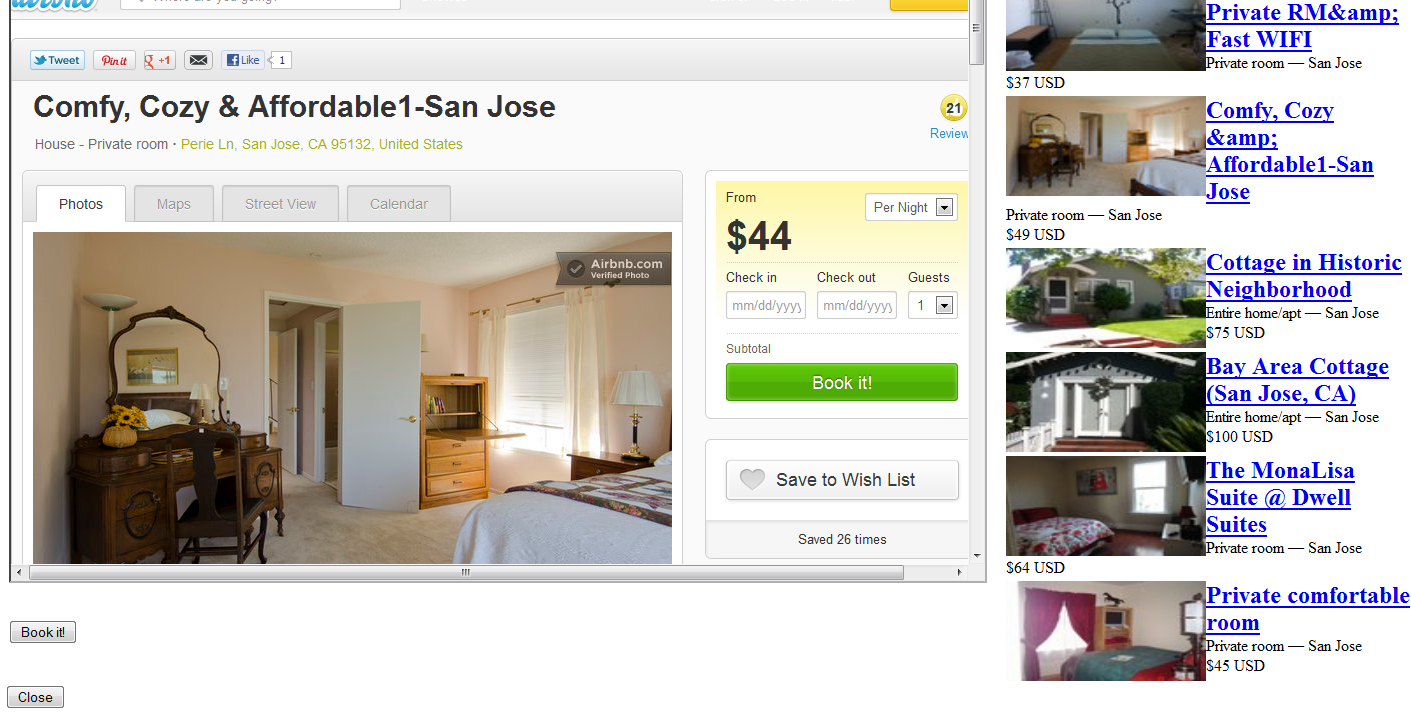
1. If they want to see the listing in a new window instead, they can click the ‘Open in new window’ link and it will redirect the user to the source in a new tab.



1. If the User clicks the ‘Compare this listing’ button, a new window will load and display the selected listing on the left and the selected listing on the right.



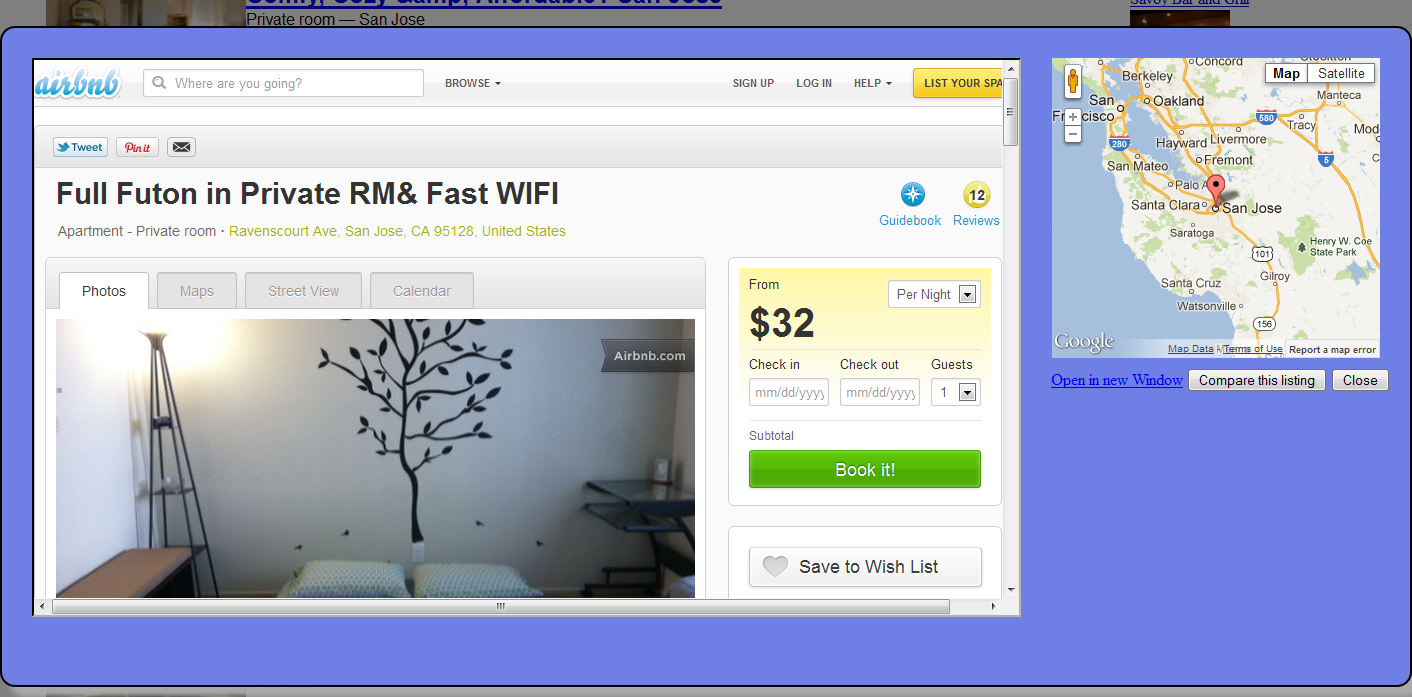
1. If a User selects a listing in the results, it gets enlarged to the right of the current selection so the user can compare 2 listings side by side.



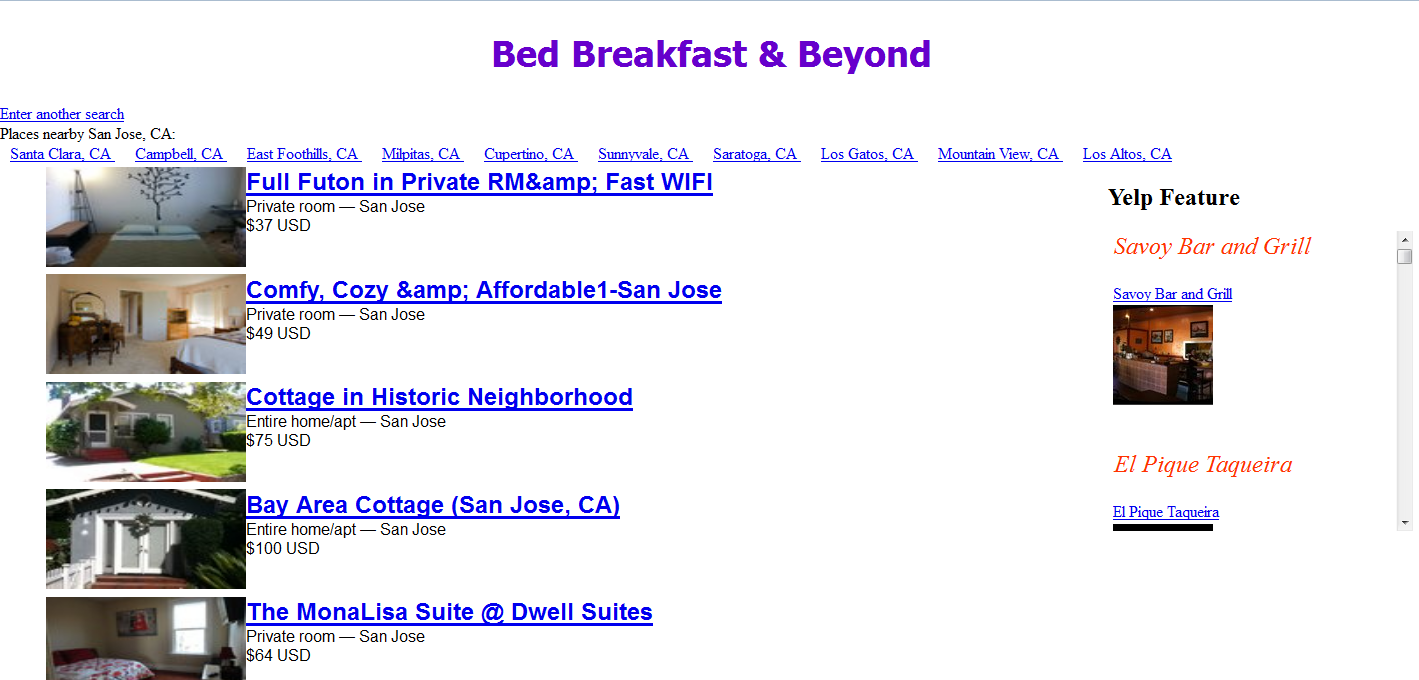
1. The User may then click the ‘Replace’ button, and then the new listing selected will replace the current selected listing.



1. If the User clicks the ‘Book it’ button, it will redirect the user to the source and the place to make the booking.



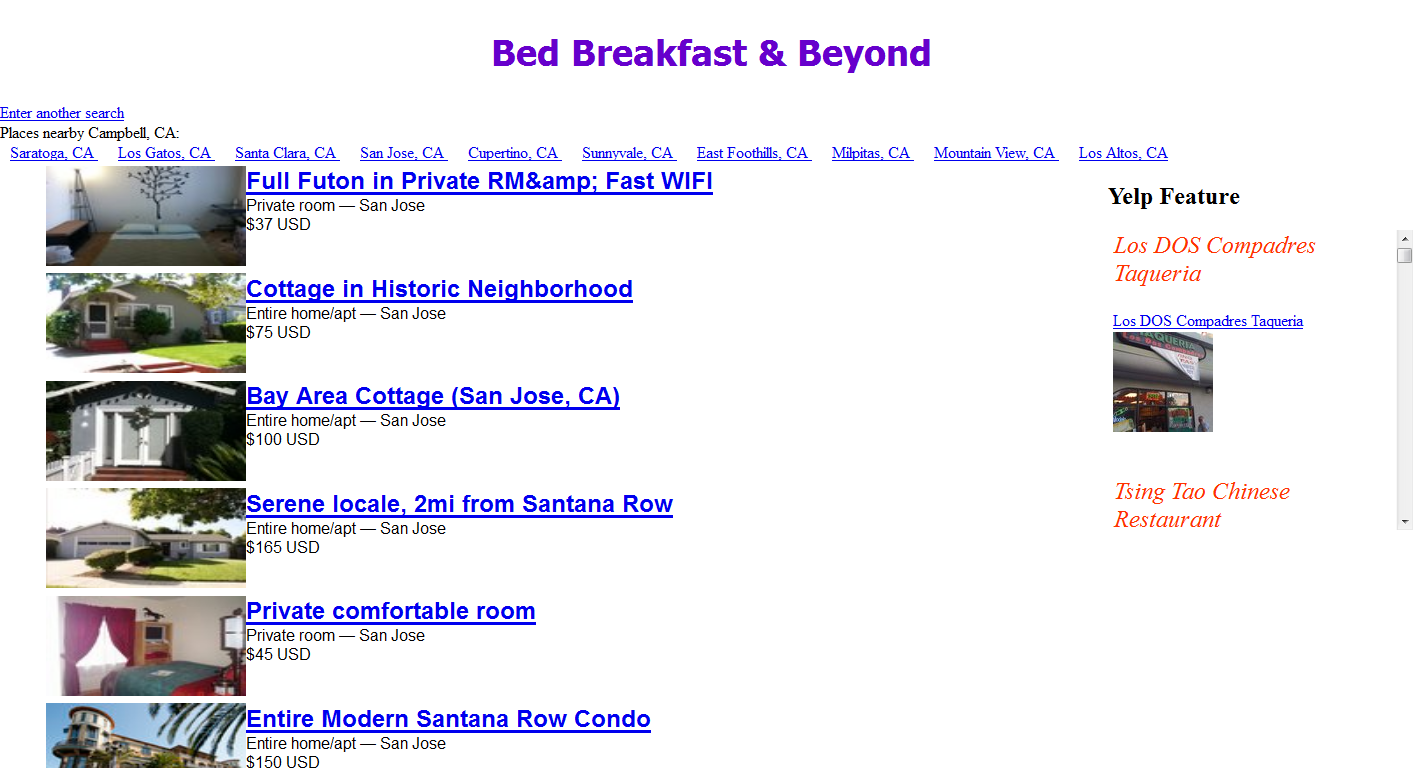
1. If the user then clicks the ‘Close’ button, it will close the tab and revert back to the previous tab.



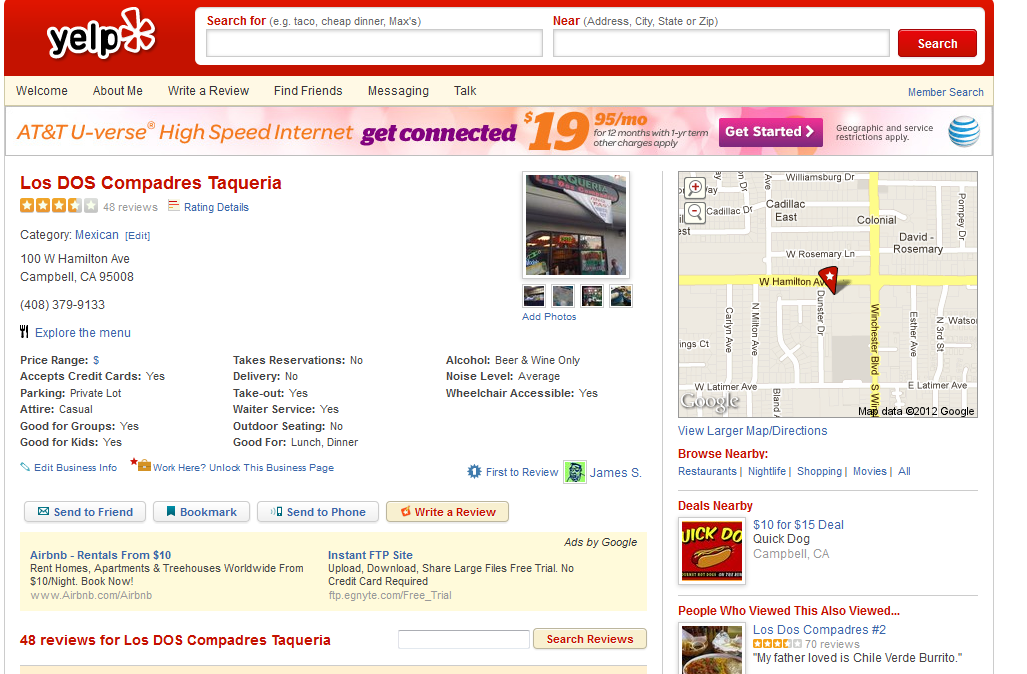
1. If the User clicks the ‘Close’ button, it will return to the default search page.



1. If the User clicks on the “Enter another search”, it will return to the main page.



1. If the User clicks on a Suggested Location, it is the same as doing a new search with the location that was clicked.



1. if the User clicks on a local restaurant listing, they will be redirected to yelp.com’s listing of the restaurant.

**How Features Were Accomplished**

**1: Unified Search**

Inputs are the crucial information that this smash up website (Bed, Breakfast and Beyond) needed. To create a Unified Search, studies and analysis are done on both sites, Airbnb.com and homeaway.co.uk. Information is gathered and used to design the Unified Search. From the analysis we saw that both sites want the user to provide a location with a start date and an end date. We notice that Airbnb does not require user to enter the number of rooms the user is seeking for. However, the number of room options are offered after the search is performed. Also, the user does not require to enter start/end dates if the user does not wish to. As for the number of guests and rooms, we assume that the user will have a need of at least one room for themselves. Therefore we decided to have the default to one Guest and one room. The user will input a location, a valid start date and end date, or no date at all, the number of guests, and number of rooms if the user requires more than one. The date cannot be earlier than current date. These search criterias will be passed to Scrap.php to scrape data from Airbnb, Homeaway, and Yelp. The results will be store in a json files, and will be displayed to the output.html. We used JavaScript to implement error checking for user’s input. The Javascript will check for Location and Valid date, if the user decides to give a start/end date. The JavaScript will check to make sure that the location is not empty or null. If user did not enter a Location to search for, we will alert the user with a message, “Location cannot be empty!” For the date input, we used Jquery to populate a calendar for the user to pick the date. For the convenience of error tracking we disabled the past dates so user cannot select it. We used javascript to check if start date is not after end date. We did this by using an open source date function written by stackoverflow user. For the room and number of guest fields, we implemented error checking by using html5 input number type Tag. With error checking implemented, we can be assured that the user will give us correct input to feed it to scrap.php to scrape data from Airbnb and homeaway.co.uk.

**2: Unified Output**

After the user provides a valid input to perform a search from input.html, it is POST to the scrap.php. The data which we pass to the scrap.php is the location the user searched for, the start date and end date, the number of rooms, and number of guest. Any empty input is stored as an empty string. We build a GET request to both airbnb.com and homeaway.co.uk to obtain a listing of bed and breakfast venues based on the user input. We used Simple HTML Dom library to obtain a DOM tree from both websites. Using this tree, we scraped all the relevant data from each site and store it in a php array. The data is stored as a JSON object which is then serialized and written to file called results.json. Also included in the JSON object is the input the user entered in the site. We then load the output.html and, in the body of the website, a script is executed which loads the JSON file created by scrap.php and is deserialized and stored in the global variable: resultJSON. We then build a table and populate it with the listing from both the airbnb and homeaway in the same script. Each listing is given an index attribute which corresponds to the listing’s place in the array stored in the JSON array found in the resultJSON (under the key “searchresult”). This is done for easy access to the listing’s data when needed.

**3: Suggested Results**

This feature relies on the input the user inputed for their destination in the index.html. The code for this feature is located in the scrap.php near the end of the file. How this feature works is similar to how the Airbnb and Homeaway data is scrapped. By using SimpleHTMLDom, we use the destination input as a search term for travelmath.com. It will scrap the first ten results of nearby cities and store the result in the result.json file along with the result listing and user input. When the output.html is loaded, a function in suggestedlocation.js gets executed which will get the JSON object created in scrap.php and create a link which will perform a new search when clicked. The search will take from user’s initial search parameters and replace their destination with the new location and passes it through the scrap.php as if the user is performing a new search.

**4: Local Restaurants**

**Background:**

The idea behind the local Restaurant feature is that it will show the User restaurants nearby the Bed and Breakfast. This way Users can see the kinds of restaurants that are around the area they are searching.

**Implementation:**

The best way we found to retrieve this data was from Yelp.com. We originally were going to actually scrap from yelp as well but later realized we could in fast utilize the Yelp api and get the result that much easier.

**5: Google Map**

After the scrap.php is done scraping data from Homeaway and Airbnb, output.html gets loaded onto the page. In the header DisplayMap.js gets loaded and it contains one function. The function will load the map, place a pin with an infowindow, and center the map on that pin. The map canvas is located in the listing overlay and gets loaded when the user selects a listing from the results. The JSON object correlating with the listing contains a key value pair with the key as ‘address’ and the value contains the address of the listing. Passing the map canvas element and the address, the function in the DisplayMap.js will use the Google Map API to pin and center the map based on the address.

**6: Comparator**

When a user clicks on a listing, an overlay is displayed which shows an iframe. The iframe gets loaded with the listing detail from its respective website and on the side there is a button which is labeled as “Compare this listing”. This button is linked to a javascript function, enterComparator(), and stored in the comparator.js file. The parameters of the comparator function are the JSON object created by the scrap php, the JSON object for the current listing, and a hidden form element from the output.html. In this function, both JSON objects are serialized using JSON.stringify() and is stuffed into the from as a hidden input. The form is then submitted to comparator.php which is opened in a new tab. This php file loads the comparator.html. Comparator.php then recreates the JSON object from the strings created by comparator.js. Using the JSON object of the listing the user selected to compare, we get the url for the listing and load it into an iframe. The resultJSON variable contains the JSON object created by scrap.php and is used to load the listing again. A hidden div which contains an iframe is stored in the comparator.html. This div gets shown when the user clicks on listing from the results from airbnb and homeaway. The iframe is then gets loaded with the listing details the user selected. Both iframes are displayed side by side with the current listing on the left and the new listing on the right. A replace button is also displayed and when the user clicks on it, the user’s current listing iframe gets loaded from the 2nd iframe and the 2nd iframe is then hidden away. The close button is linked to window.close() which will close the comparator window. The book it button directs the user to the current listing url within the same window.

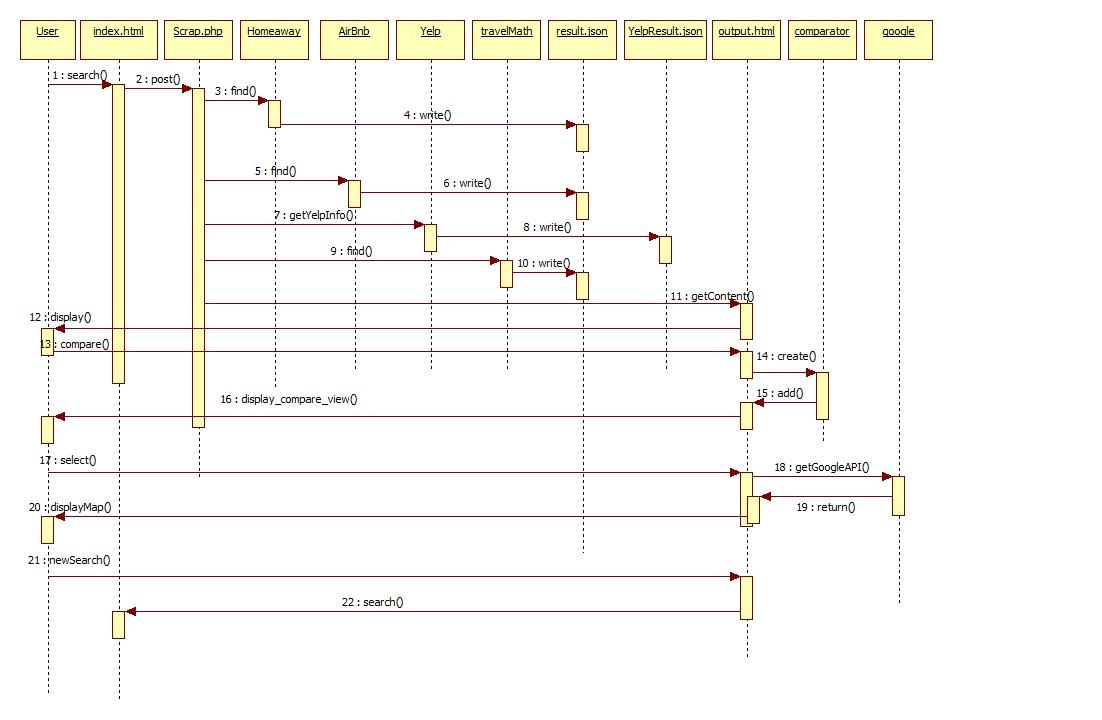
**Collaboration List**

|  |  |
| --- | --- |
| Task | Collaboration |
| Unified Search | Peter Wong |
| Unified Output | Tan Nguyen |
| Suggested Results | Chi Koo |
| Local Restaurants | William McCarty |
| Google Map | Carlos Pereira |
| Comparator | Tan Nguyen |
| PowerPoint | William McCarty, Tan Nguyen |
| Doc: Default Presentation Style | William McCarty, Tan Nguyen, Carlos Pereira, Wilson Yee, Brandon Rousey, Peter Wong, Chi Koo, ChingYing Kuo |
| Doc: Approach to Avoid Duplicate Content | William McCarty |
| Doc: Workflow | William McCarty, Chi Koo |
| Doc: Collaboration List | William McCarty, Tan Nguyen, Chi Koo, Peter Wong |
| Doc: Risks, Gaps, and Things to Build On | William McCarty |
| Doc: Sequence Diagram | Chi Koo, ChingYing Kuo |
| Doc: Spelling and Grammar Checking: | Brandon Rousey |
| Doc: How Each Feature Were Accomplished |  |
| 1. Unified Search | ChingYing Kuo, Peter Wong |
| 2. Unified output | Tan Nguyen |
| 3. Suggested Results | Tan Nguyen |
| 4. Local Restaurants | William McCarty |
| 5. Google Map | Tan Nguyen |
| 6. Comparator | Tan Nguyen |

**Risks, Gaps, and Things to Build On:**

* Limit of how many Yelp calls per day
* Slim chance of duplicate content
* Sorting results from lowest costs with best ratings
* Comparator able to compare against same listing
* Google Map Panning Issue
* Comparator is not big enough to display whole page
* The websites used to scrap the data are down

**Sequence Diagram:**



**Sites used:**

* SimpleHTMLDOM -<http://simplehtmldom.sourceforge.net> - This library parses a website into a DOM tree which can be used to extract important information.
* OAuth - <http://oauth.net/> - Used for authentication for Yelp API.
* [www.travelmath.com](http://www.travelmath.com) - Used to get Suggested Locations. Data is scraped using SimpleHTMLDOM and using the user destination as the input.
* [www.stackoverflow.com](http://www.stackoverflow.com) - Excerpts of source code needed for the project were taken from this site.
* <http://www.gliffy.com> - used to create UML diagram

**Tools used:**

* StarUML - Used to create sequence diagram
* GitHub - Used to store the website code for branching and editing. Used for version control
* XAMPP - Used to run the web server
* PHP/jQuery/Javascript/HTML - The languages used to provide the backend functionality for the website, primarily Javascript.
* Yelp API - Used to display the restaurants and other dining facilities in the area around the location.
* Google Map API V3 - Used to display the locations of the output on a map, to give the user a sense of locality.
* Firebug - Firefox plugin used for debugging