BetterBNB

CS 30700

Design Document

Team 6:

Andrew Orlowski
Bradley Bottomlee
Sunil Green
Anudeep Yakkala
Christopher Lin

Index

Purpose

Consumers in the market for hotels, vacation rentals, or lodging have the potential to be misled about various aspects of an online booking. Sites that host travel listings such as Airbnb, hotels.com, and booking.com, among others, are not motivated to publish the many less than desirable or negative attributes of a booking. Important information including the safety of an area, available amenities in the area, and more are far too often not considered by booking services. What further complicates the booking process is that reviews for listings are scattered over the web amongst these various competing services.

The purpose of BetterBNB is to help online consumers wrap their heads around the overcomplicated process of booking a room. We will do this by providing booking website users a more complete, honest, view of the listings being recommended to them. We will combine data including but not limited to crime statistics, notable nearby attractions, ease of access to services like medical care, and reviews from all across the web to give consumers that complete picture they deserve. Existing solutions, such as trivago, hoteloogle, and hotels.com do not fill the need for a service like ours. They are either primarily focused on price, leave out many of these notable characteristics about a booking in order to push a listing they want to get rid of, or are fragmented themselves despite advertising themselves as all-encompassing. BetterBNB will work diligently to provide users with the complete profile of the hotels that they need by sourcing information from all across the web. We will also prioritize our users' experience over the price of a booking.

We are seeking to serve as a watchdog for some of the more nefarious and negligent booking sites that prioritize their profit over the safety and happiness of their customers. We believe that it is due time that the consumers of these websites receive the fair treatment they have been missing.

Functional Requirements

1. **User Account Setup and Management**
    a. As a user, I would like to register for a BetterBNB account so that I can easily access the information on the listings I've looked at.
    b. As a user, I would like to login to my BetterBNB account using OAuth so that I have access to all of the recent listings that I've looked at without having to mark them myself.
2. **Website Use**
    As a user,
    a. I would like to enter an address into a search bar on the home screen of the webpage and receive a detailed report in the form of a pdf so that I can easily find quick information about a listing.
    b. I would like to see other nearby listings to the one I am looking at so that I can compare them to each other and make the best choice for myself.
    c. I would like to categorize a set of listings as a "trip" so that I can set up multiple listings in a row that would be in different places based upon where I'm traveling in my trip.
    d. I would like to be able to customize the name of a trip so that I can easily differentiate between trips.
    e. I would like to be able to remove a listing from a specific trip if so I can change my mind and update a trip.
    f. I would like to be able to delete a trip so that I can remove it from my dashboard if I change my mind about a trip.
    g. I would like to have the option to download a pdf containing information provided by the service about a listing or trip so that I can access it offline and print it easily.
    h. I would like to visit the main website and use the information gathering services about a specific listing without having to register for an account so that I don't need to sign up if I don't want to.
    i. I would like to be prompted to sign up for an account and integrate a listing's information found from the home screen into a draft of a trip so I don't lose any information found from using the service.
    j. I would like to be able to get a safety rating for a listing on a residence booking website so that I'm assured that the listing is not in a bad area.
    k. I would like to be able to see a combination of reviews from different review sites including but not limited to Google Reviews, Yelp, alongside reviews from the site the booking was originally listed on so that I have centralized access to firsthand accounts of the listing.
    l. I would like to get a list of restaurants that are located near a listing on a residence booking website so that I know just how accessible food and dining is.
    m. I would like to see a dashboard containing all my saved listings so that it's not necessary to store my viewed listings in a file or other type of storage locally.

n. I would like to remove listings from my saved listings so that listings that are no longer relevant to me don't clutter up listings that I am considering for the future.
o. I would like to add a listing to my saved listing using a link to the residence booking website so that I don't have to install the extension and view the webpage in order to get the benefits of the software.
p. I would like to search for listings in my saved listings so that I save time by not having to sift through all the saved listings in order to find the one I'm looking for.
q. I would like to receive a notification if information about a saved listing changes so that I don't have to track listings myself and so that I don't get caught off guard after renting a listing.
r. I would like to be able to contact the service so that I can recommend ways in which it can be improved, or express any concerns I have.

3. **Browser Extension Use**

   As a user,

   a. I would like there to be a lightweight web browser extension so that I can easily get information about a listing.
   b. I would like the extension to display minimal default information about a listing on a site that I'm visiting so that I can get more information than the host site provides.
   c. I would like to be able to manually enter an address into the extension if the extension cannot scrape the address from the site I'm visiting so that I can still access relevant information about the listing.
   d. I would like the browser extension to redirect me to the website so that if I want a more detailed report on the listing found I can get one.
   e. (If time permits) I would like there to be support for a lightweight web extension for the browser of my choice.

4. **Personalized User Experience**

   As a user,

   a. I would like to be able to customize the qualities of a listing that appear on the browser extension so that I can use the browser extension as I see fit.
   b. I would like to be able to control the complexity of the pdf report so that I can get the most use out of the service as I need.

   (Dependent on APIs found) As a user

   c. I would like the application to take into consideration the availability of nearby available cuisine and dietary restrictions so that if I need or want to eat a certain type of food during my trip I can.
   d. I would like the application to take into consideration how important proximity to the center of town is for me so that I can choose a residence that suits my preference.
   e. I would like the application to take into consideration how important wheelchair accessibility is for me so that I know that I'll be able to get everywhere that I want to get to.

   f. I would like the application to take into consideration how important access to medical care is for me so that I can feel assured that if I run into any outstanding health complications there is sufficient care for me.
   g. I would like the application to take into consideration how important proximity to national parks and public outdoor areas is for me so that if that's the main goal of my trip, it's being met.
   h. I would like the application to take into consideration how important points of historical relevance are to me so that I know all of the local sightseeing spots if I desire to see them.
   i. I would like the application to take into consideration how important the forecast of the area is to me so that I can avoid areas that are expecting weather that I consider bad.
   j. I would like the application to take into consideration how important the prevalence of a specific language in an area is so that I'm not stuck somewhere where nobody can speak to me if I need help.
   k. I would like the application to take into consideration how important proximity to bodies of water is to me so that if I can plan my trip around scenic routes.
   l. I would like the application to take into consideration how important pet accessibility is, so if need be my pet can come with me.
   m. I would like the application to take into consideration how important the availability of exercise equipment is so if I need access to a gym I have it.
   n. I would like the application to take into consideration how important quietness is for me so that I can choose a residence that suits my preference.
   o. I would like to be able to see groups of listings ranked in order of importance of a specific category mentioned above (such as safety, wheelchair accessibility, etc) if I so choose so that I can see how different parts of a trip could pose challenges to me.
   p. I would like the service to recommend an overall best listing for me based on what I identified as important to a trip so that I can make the best choice.

5. **Content Management**

   As a user,

   a. I would like to be able to provide evidence and report listings that contain false advertising so that other users can avoid a bad experience I had at a location.
   b. I would like the service to give me a warning if a listing is known to contain false advertising.

   As an administrator,

   c. I would like to be able to view all evidence submitted against a listing from multiple users so that I can make a determination as to whether the listing contains false advertising.
   d. I would like to be able to identify listings that systemically contain false advertising so that users can know if a listing seems too good to be true.

Non-functional Requirements

1. **Aesthetics and UI Interface**
   As a user,
   a. I would like the app to be designed minimalistically so there is not a steep learning curve.
   b. I would like to have control over certain aspects of the interface so I can prioritize which characteristics of a listing I see first.

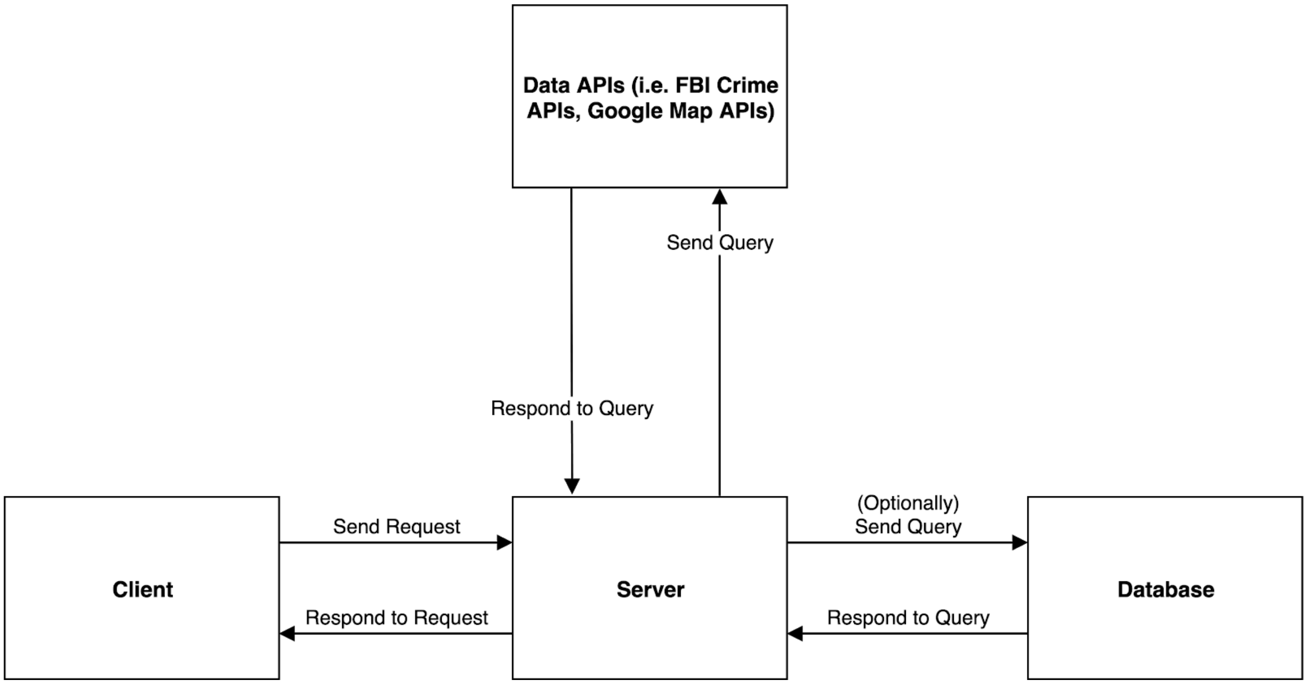
2. **Performance**
   a. I would like the web browser extension to find minimal information once an address is found on a booking site or manually entered into the extension in under 10 seconds.
   b. I would like the website to load in a reasonable amount of time when compared to other production level travel sites such as trivago.
   c. I would like the application to support over 10000 users

3. **Security**
   As a user,
   a. I do not want personal information about where I am traveling and how long I will be away to be made public to avoid an increased threat to my home security.
   b. I do not want personal preferences such as the importance of medical access, and dietary restrictions to be given to third parties or made public.
4. **Server**
   As a developer,
   a. I want the server to be able to return a list of statistics about a location based on a request.
   b. I want the server to be able to service multiple types of clients including a website and web browser extension.
5. **Usability**
   As a user,
   a. I would like the website to work and be designed for mobile too so that I can access it on the go.
   b. (If time permits) I would like there to be support for the extension among multiple different browsers
6. **Maintainability**
   As a developer,
   a. I want to be able to easily integrate use of more apis so that I can improve the quality of service
   a. I want to be able easily to add more characteristics to take into consideration into the service so that I can improve the breadth of the service.

Design Outline

High-Level Overview of the System

BetterBNB will be a web application (with an optional web browser extension) that will allow users to input an apartment booking listing into the site (or have a pop-up on the listing directing the user to the site's corresponding information). The site will display a wide range of information such as crime statistics, population density, and nearby restaurants to a varying level of specificity contingent on the data that is available for that particular listing. Additionally, users will have the option to register an account with the site in order to access functionality that allows them to save particular listings and view recommendations based on those listings. To accomplish these tasks, we plan on using the client-server model, wherein the client will communicate with the server and the server will respond with data related to the client's request. The server will concurrently serve a substantial number of clients with the sole restriction being the resources available to the server. To fulfill a client's request, the server will query publicly available resources for data that is localized to the listing, optionally save that listing to an account in the database, and return data to the webpage consisting of the discovered information.

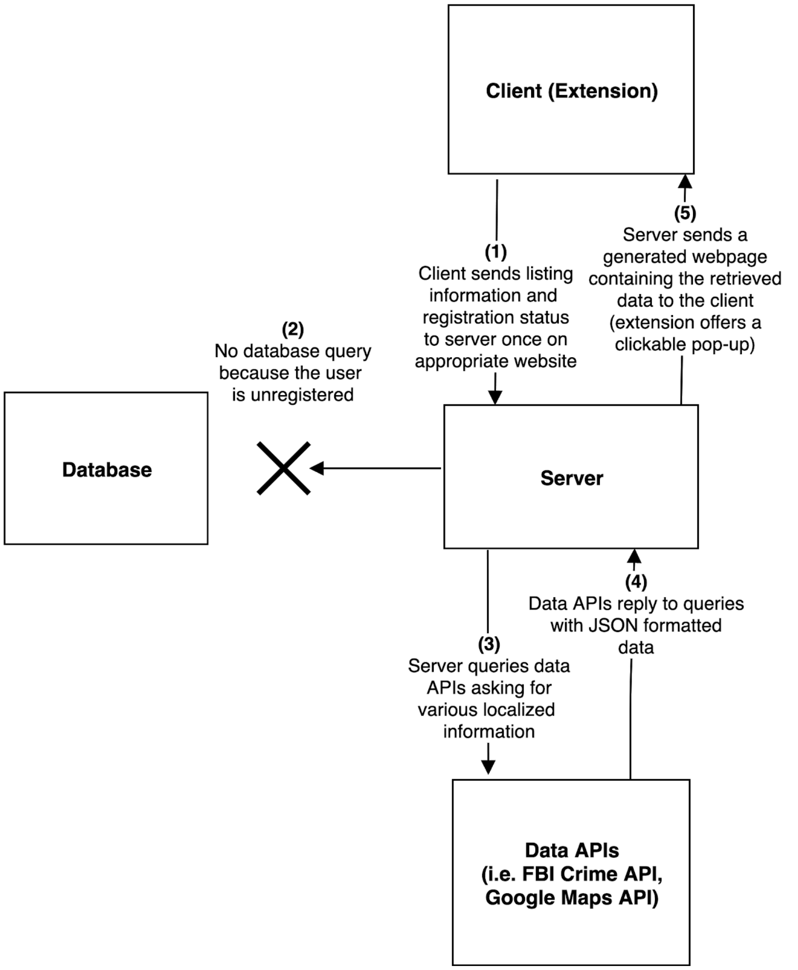


1. **Client**
    a. Serves as an interface between the user and our server.
        i. Two options for interfacing with the server:
            1. Web browser extension
            2. Directly the webpage
    b. Communicates with the server via HTTP requests.
    c. Receives webpage from the server with information corresponding to the HTTP request.
2. **Server**
    a. Communicates with the client and data APIs via HTTP requests.
    b. Serves as the congregating point for information from the data APIs and database pertaining to the client's request.
    c. Formulates a webpage personalized to the listing and delivers them to the client.

3. **Data APIs**
    a. Hosted by third parties.
        i. Google
        ii. The FBI
        iii. Other organizations (undetermined)
    b. Respond to HTTP requests with information pertaining to the request in JSON format (typically).
4. **Database**
    a. A document-oriented database storing user account information, saved listings, and potentially more.
    b. Responds to queries from the server, returning the specified information in JSON format.

Sequence of Events Overview

Below is a sequence diagram depicting the interactions between our components in servicing a typical user that has the web browser extension installed and does not have an account. The sequence is initiated by the user navigating to a listing on a website such as Airbnb, with the extension noticing this and pinging the server with listing information and the user's registration status. The server receives this request, then pings only the data APIs and does not ping the database since the user is unregistered. Upon retrieving the information from the data APIs, the server generates a webpage containing the data and sends it to the client. The extension then displays a clickable pop-up where the user can open the webpage. The precise sequence of events described below is ordered numerically above the text.


Client (Extension)
(1)
Client sends listing information and registration status to server once on appropriate website
(5)
Server sends a generated webpage containing the retrieved data to the client (extension offers a clickable pop-up)
(2)
No database query because the user is unregistered
Database
Server
(3)
Server queries data APIs asking for various localized information
(4)
Data APIs reply to queries with JSON formatted data
Data APIs
(i.e. FBI Crime API,
Google Maps API)

Design Issues

Functional issues

1. What user data do we need to store/collect?

Options:

1. Never store user data
2. OAuth ID for third party login and relevant information about listings
3. Account credentials and all information related to the user’s trip itinerary

Choice: Option 2

Explanation: We are going to have user’s sign in with a third party such as Google to make the experience much more seamless. Our entire philosophy is that BetterBNB should serve as a lightweight, unobtrusive extension to aid users in planning their trips. As a result, we don’t want them to have to create another account and sign in each time with a username/password. Lastly, we do not want to store massive amounts of user data. We want to remain privacy centric as the information regarding the users’ trip could be confidential. In a word, we do not intend to store anything beyond the generated reports for the listings and a simple OAuth ID to allow for the user to login.

2. How is the user notified by the browser extension that there is information about the listing available?

Options:

1. A pop-up appears from the extension menu (typically a “flag” like AdBlock)
2. An onscreen modal appears similar to “Honey”
3. The onscreen HTML is altered with a hyperlink to bring them to the web application

Choice: Option 1

Explanation: Once again, we want the browsing experience to be unobtrusive. Thus, we just need the user to be able to view ever so-slightly that there is more information available. For instance, if a user is searching Airbnb, they may view a large number of listings and it would be annoying to see a popup (modal) on every listing they view or something on the web page itself. That’s why we feel it will be best to delegate the notification to the extension toolbar.

3. How does the user create an account?

Options:

1. Sign-up with a username/email and password
2. OAuth Google/Facebook login
3. No account

Choice: 2

Explanation: While an account is not needed to use BetterBNB services, an account will enable users to save listings and reference them later. We see this as a benefit that will allow the user to compare and contrast their desired listings for a trip. As a result, this eliminates option 3. Option 1 is not a bad option, but we don’t want to store passwords and usernames. This is because a user’s account may contain sensitive information (travel plans, etc.), and we want to ensure the highest level of security. Naturally, this leads to option 2 which is delegating the login authentication to Google or Facebook. Presumably, these services have higher levels of security than what we can implement. Additionally, it makes the sign-in process easier as users don’t have to remember specific login details.

4. What immediate details does the extension provide when on a listing page?

Options:

1. Safety rating, average reviews of the area, and restaurant choices
2. Accessibility, languages spoken, and destinations nearby
3. Allow the user to customize their immediate results

Choice: 3

Explanation: Different users are bound to have different priorities. For instance, a family travelling to a location is going to have vastly different priorities than a group of college students. Thus, we want the most relevant information to be immediately available for any given user. So although options 1 and 2 might be important, a user may find a certain combination more important than another. As a result, we intend for them to have a straightforward, customization menu available through the extension.

5. What will the search results be when a user searches for a location from the web application homepage?

Options:

1. Airbnb listings
2. Hotels and Resorts
3. Combination of both

Choice: 3

Explanation: Having a combination of both is the best choice for our end users as it will provide the most variety. For instance, sometimes when travelling to a city, the city may not have many Airbnbs. As a result, hotels may be the only option for the user. Additionally, if the city is flush with Airbnb listings, simply listing hotels would not be representative of all the possible bookings in a city. Lastly, we want to provide the greatest and best number of options to the user.

Non-functional issues

1. How do we intend to host the web application and browser extensions?

Options:

1. Heroku
2. Google Cloud Platform
3. AWS
4. Azure

Choice: *Heroku*

Explanation: Heroku is the best choice for our needs since it is very robust for hosting small web applications. Additionally, there is native support for a React/Node app that makes continuous deployment very simple and just a matter of pushing commits to a production branch on GitHub. This is as opposed to something like AWS which is very verbose and may have functionality like this but involves a steep learning curve. Additionally, Azure poses the same issues as AWS (verbose) and deploying to Google Cloud Platform is just more involved than Heroku.

2. What is our backend framework?

Options:

1. Flask (Python)
2. Django (Python)
3. Node.js (Javascript)
4. Spring (Java)

Choice: *Node.js*

Explanation since we will be doing extensive data scraping and data processing with Python, Node.js pairs very nicely with our front-end framework and naturally lends itself to being the best option. Additionally, it is straightforward to develop robust API for all of our services (browser extensions and web application) to interface with. With Flask and Django, even though we could execute Python scripts easier, it would be more difficult to communicate with our frontend seamlessly. The Spring framework is great but doesn't necessarily meet our needs as we don't intend to use Java for this project.

3. What is our front-end framework?

Options:

1. React
2. Angular
3. Vue
4. React Native

Choice: *React*

Explanation: This choice simply comes down to how familiar each team member is with the above frameworks. Many individuals have extensive experience with React but have never used the others. Since we want to focus on building the best possible product, we want to reduce the time to learn new technologies. From preliminary research, Angular and Vue are similar but have a steeper learning curve than React and may not be as easy to deploy and interface with Node.js. React Native is an excellent framework for developing one application that can service web and mobile, but we don't intend to port our application to mobile. As a result, React Native would be overkill for our use case.

4. Should we use a SQL or noSQL database and which one?

Options:

1. MongoDB
2. PostgreSQL
3. MySQL
4. SQLite

Choice: *MongoDB (noSQL)*

Explanation: Since our data is going to be very oriented towards the real-world as opposed to tabulated, row-column data, MongoDB is a fantastic option. Mongo allows us to store data as BSON (binary JSON documents) and has an incredibly flexible schema. Additionally, it is very easy to retrieve data by simply using an API to interface with MongoDB Cloud Atlas. This is as opposed to the other three databases that are all SQL oriented databases that demand data be stored in a tabulated way.

5. How do we have all of our services communicate with one another?

Options:

1. Develop light-weight small backends for each service
2. Centralized RESTful API that each service can easily interface with
3. Develop robust backend for web application and small, separate backends for each browser extension

Choice: *Centralized RESTful API*

Explanation: Due to how many services we have, and the potential for further services, we intend to design a robust, secure, and fast API for all services to employ. By doing this, we can focus on the frontend without worrying about how to transform the data for each service. Option 1 would be a lot of additional code and testing to ensure everything works properly. Option 3 has the exact same downsides as option 1. In a word, option 2 is the best way to allow for all services to operate in a consistent way.

6. **Which API(s) should we use?**

Options:

1. Yelp API
2. Google Maps API

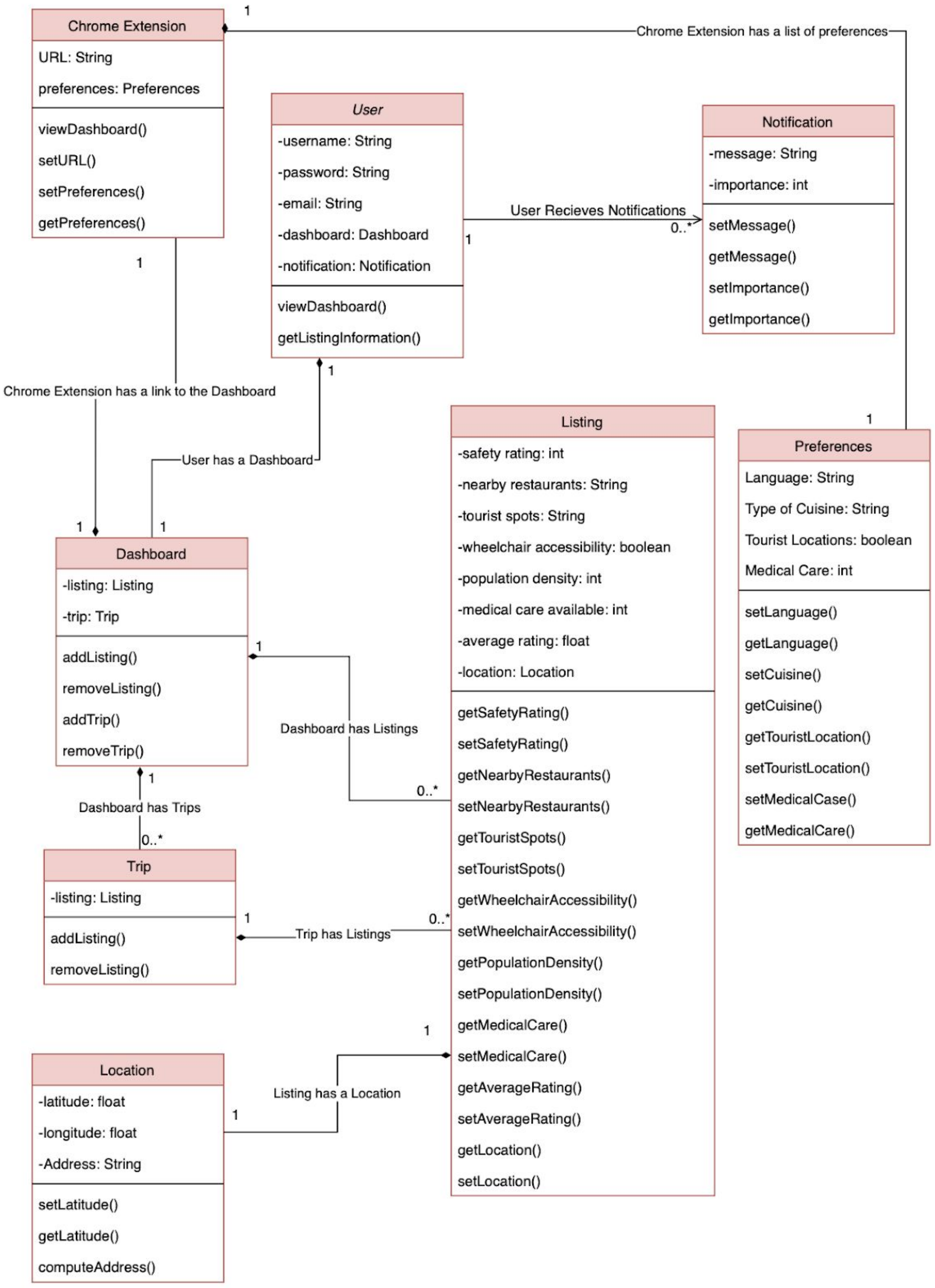
Choice: *Google Maps API*

Important Note: We intend to use many APIs. Most will have to be determined as we go along and tested to see if they provide good information. Despite this, we will have to use some API to get general location information, reviews, and landmarks nearby.

Explanation: Even though Yelp has been the de facto resource/platform for viewing and posting reviews about restaurants and other establishments for years, Google Maps provides the same level of reviews with the additional functionality of being able to get landmarks and establishments near a specified location. In a word, it has superior and more robust functionality that will allow us to drive more insight than using the Yelp API.

Design Details

Class Design



1. **User**
    - An instance is created when someone registers for the service
    - Each user will have a username, password, and an email that can be used to login to their account
    - Users will have access to a dashboard where they can see/edit their saved listings and saved trips
    - Users will have a notifications tab where they can see any updates to their saved listings/trips

2. **Dashboard**
    - An instance is created for each user that registers for the service
    - The dashboard will contain saved listings and saved trips
    - Listings can be removed or added from the dashboard
    - Trips can be added or removed from the dashboard
    - This will allow users to have a nice way of looking at all their saved listings/trips

3. **Trip**
    - A trip can be created from the dashboard
    - A trip contains a number of listings in a specific order
    - Listings can be added or removed from this trip
    - This is so listings can be organized in a nice way if a user wants to book multiple listings for one trip

4. **Listing:**
    - A listing is added to the dashboard or a trip when the user chooses to do so
    - A listing contains information regarding safety rating, nearby restaurants, tourist spots, wheelchair accessibility, population density, medical care accessibility, average rating of listing, and Location
    - This information can be retrieved or updated as necessary

5. **Location:**
    - An instance is created for each listing
    - This contains the longitude and latitude of a listing using information from a residence booking website
    - Using this longitude and latitude an address is computed
    - This address is associated with the listing that this location pertains to

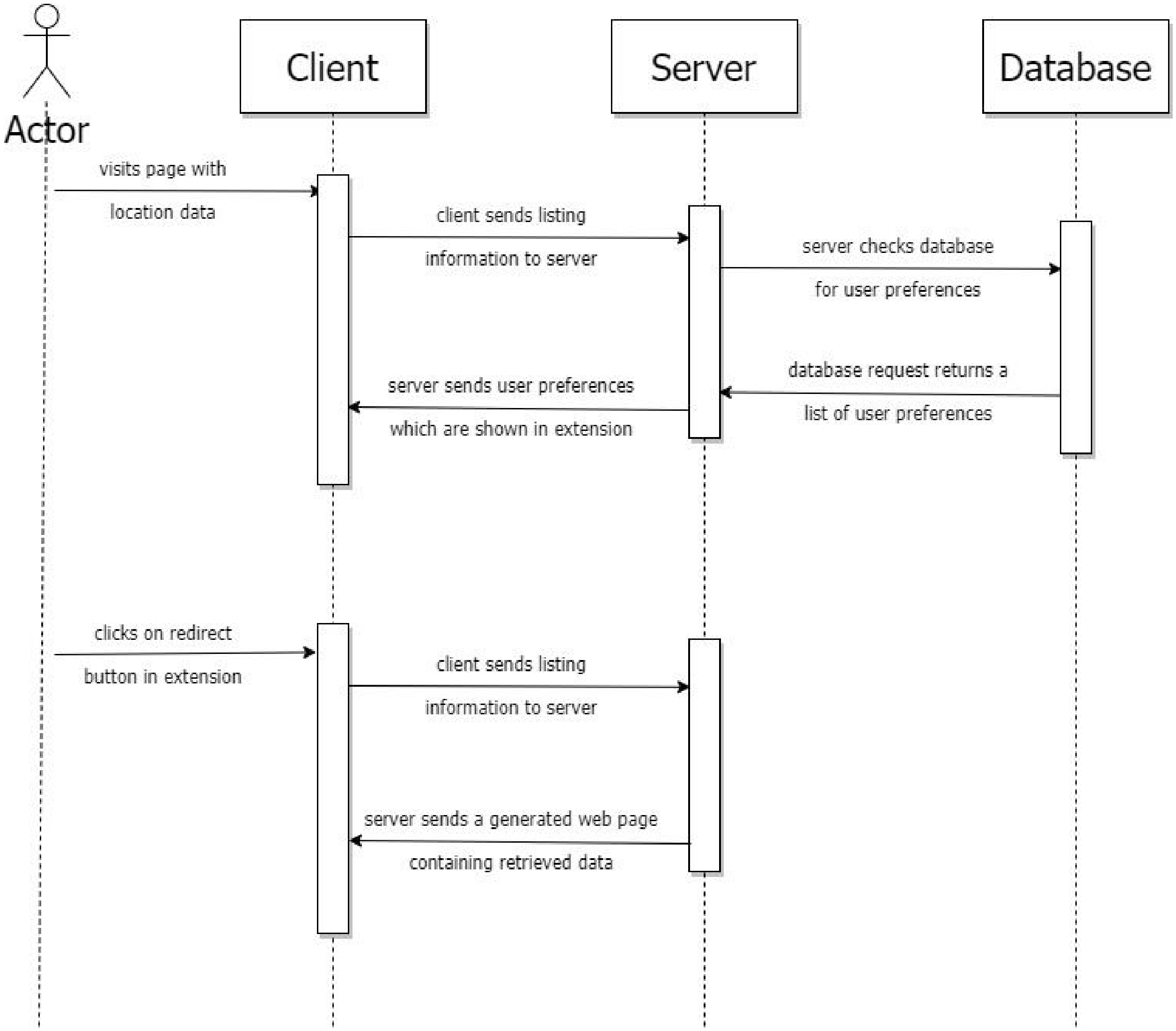
6. **Web Browser Extension:**
    - There is one instance of this class per user
    - This contains an url that will take the user to the BetterBNB website to gain more information about a listing on a residence booking website
    - It also allows for preferences to be selected so that recommendations for bookings can be created

7. **Preferences:**
    - One instance of this class is created for each user
    - This contains options that an user can select via the web browser extension to refine listings to match their preferences
    - They can select preferences pertaining to language, cuisine, tourist locations, and medical care accessibility.

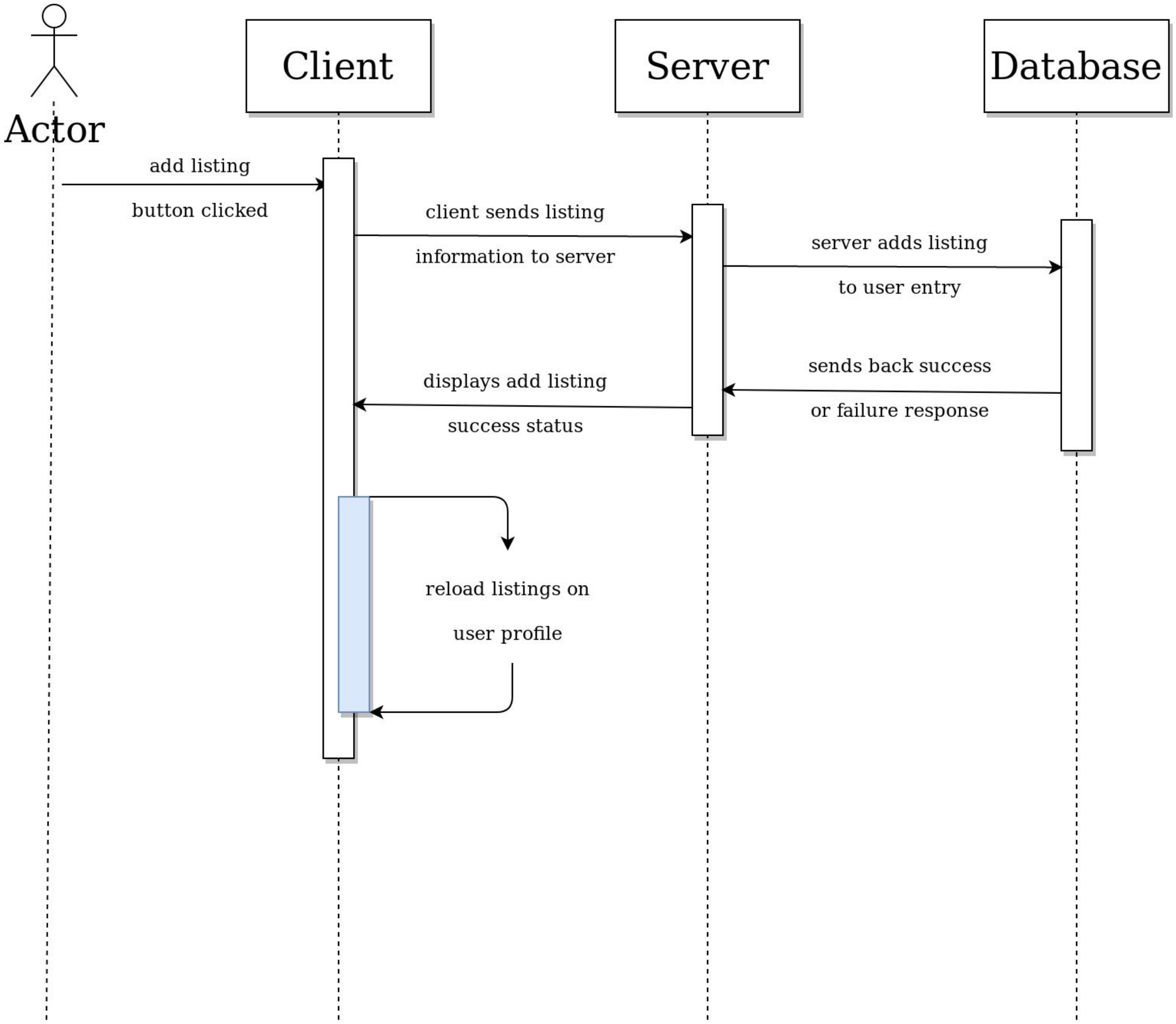
8. **Notifications:**
    - An user can have any number of notifications
    - A notification contains a message and an integer representing its importance
    - It gets stacked in the notifications tab in the dashboard and is ordered by importance
    - A new notification is created every time there is an update to any information pertaining a saved listing

Sequence Diagrams

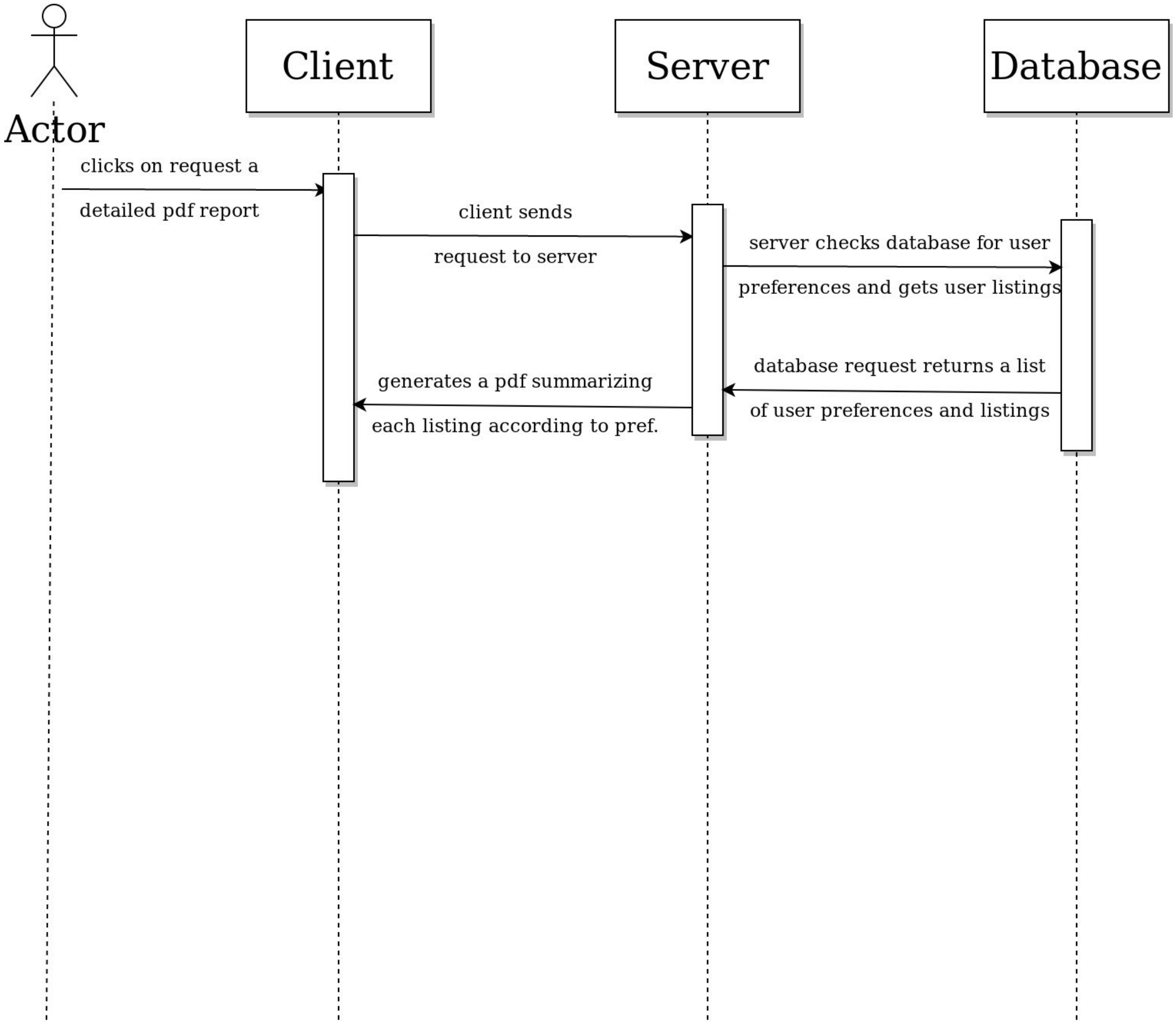
Sequence of events when user visits page with location data


Actor
Client
Server
Database
visits page with location data
client sends listing information to server
server checks database for user preferences
database request returns a list of user preferences
server sends user preferences which are shown in extension
clicks on redirect button in extension
client sends listing information to server
server sends a generated web page containing retrieved data

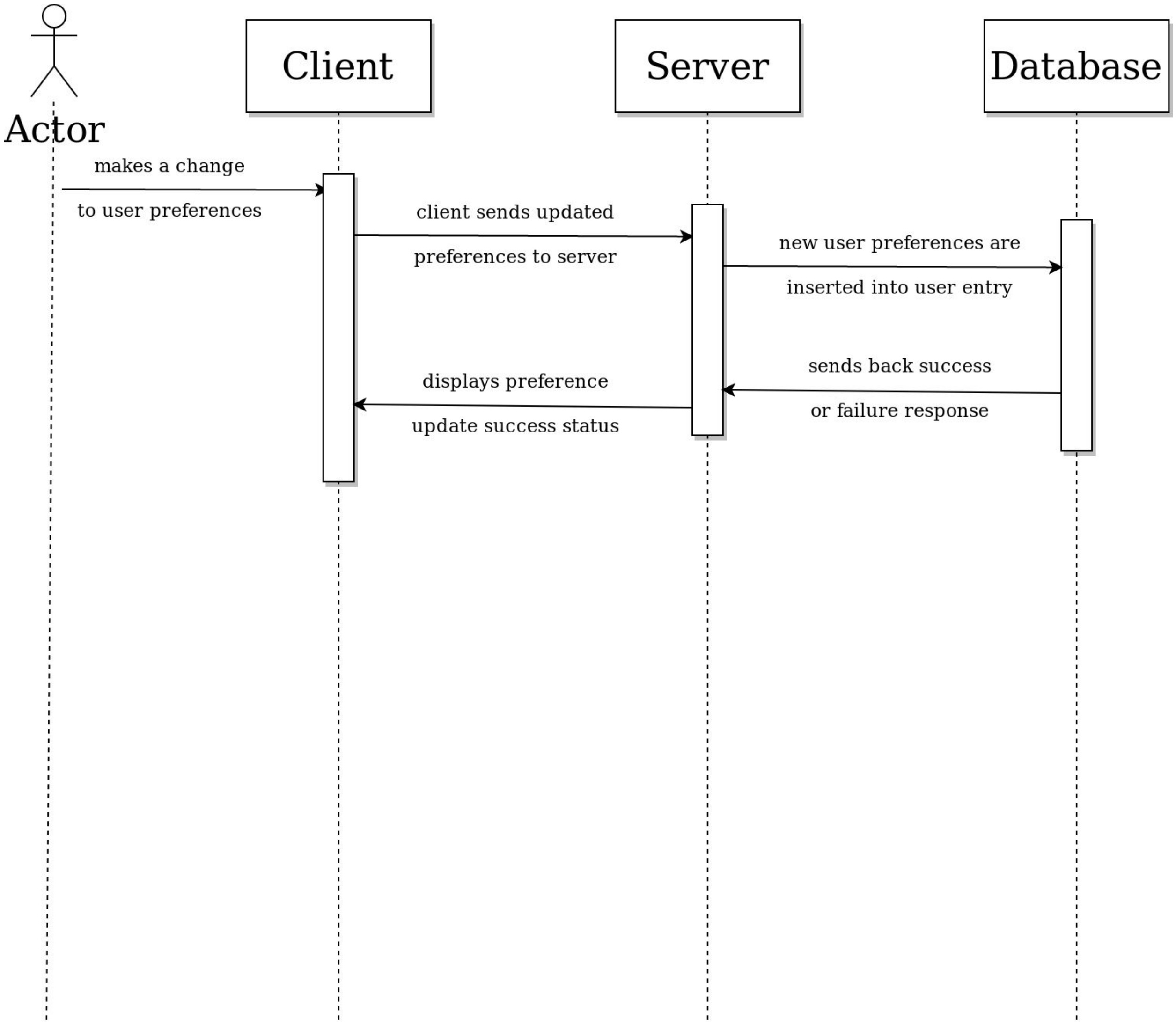
Sequence of events when the user adds a listing



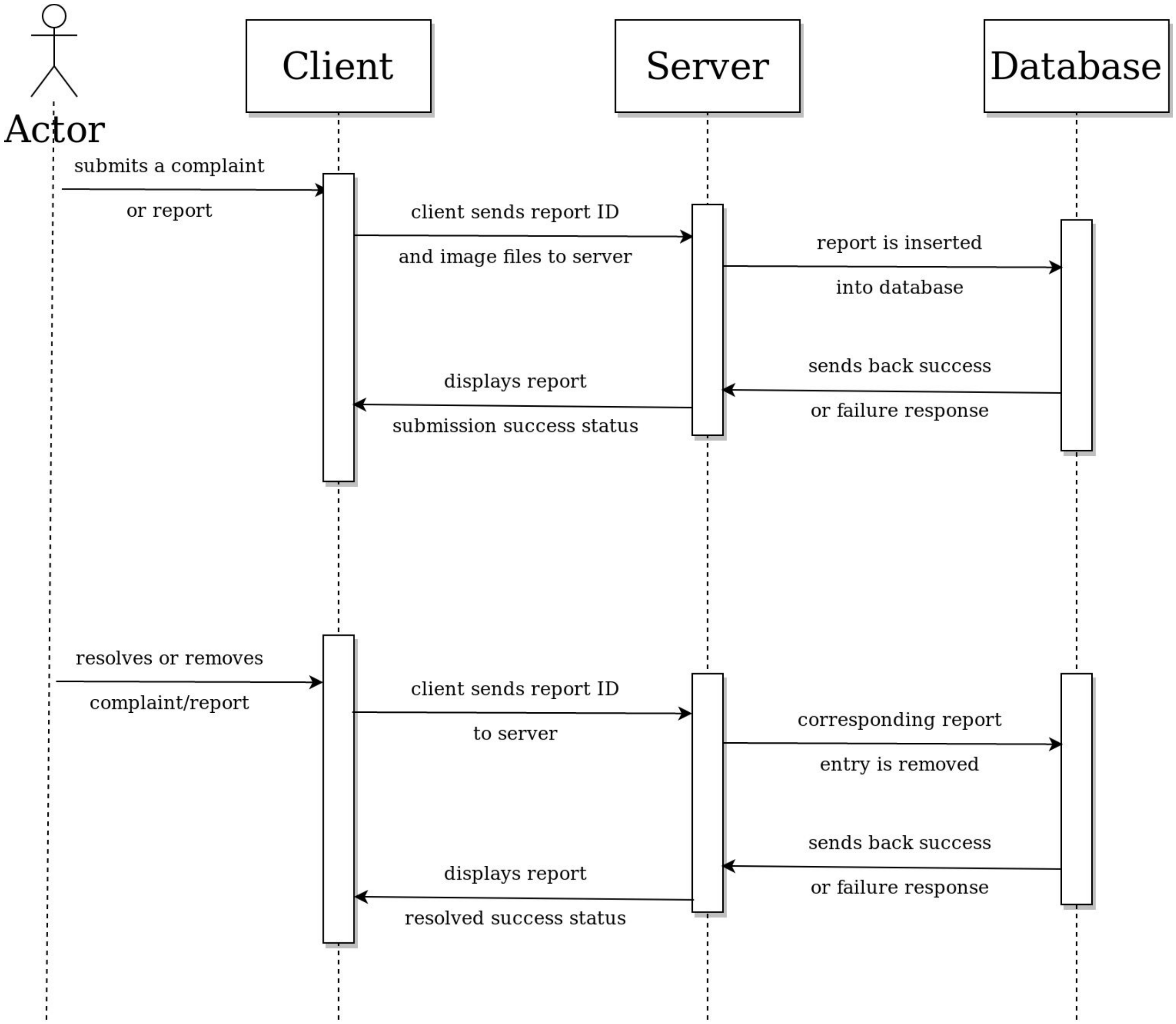
Sequence of events when the user requests a detailed pdf report


Actor
Client
Server
Database
clicks on request a
detailed pdf report
client sends
request to server
server checks database for user
preferences and gets user listings
database request returns a list
of user preferences and listings
generates a pdf summarizing
each listing according to pref.

Sequence of events when the user makes a change to their preferences

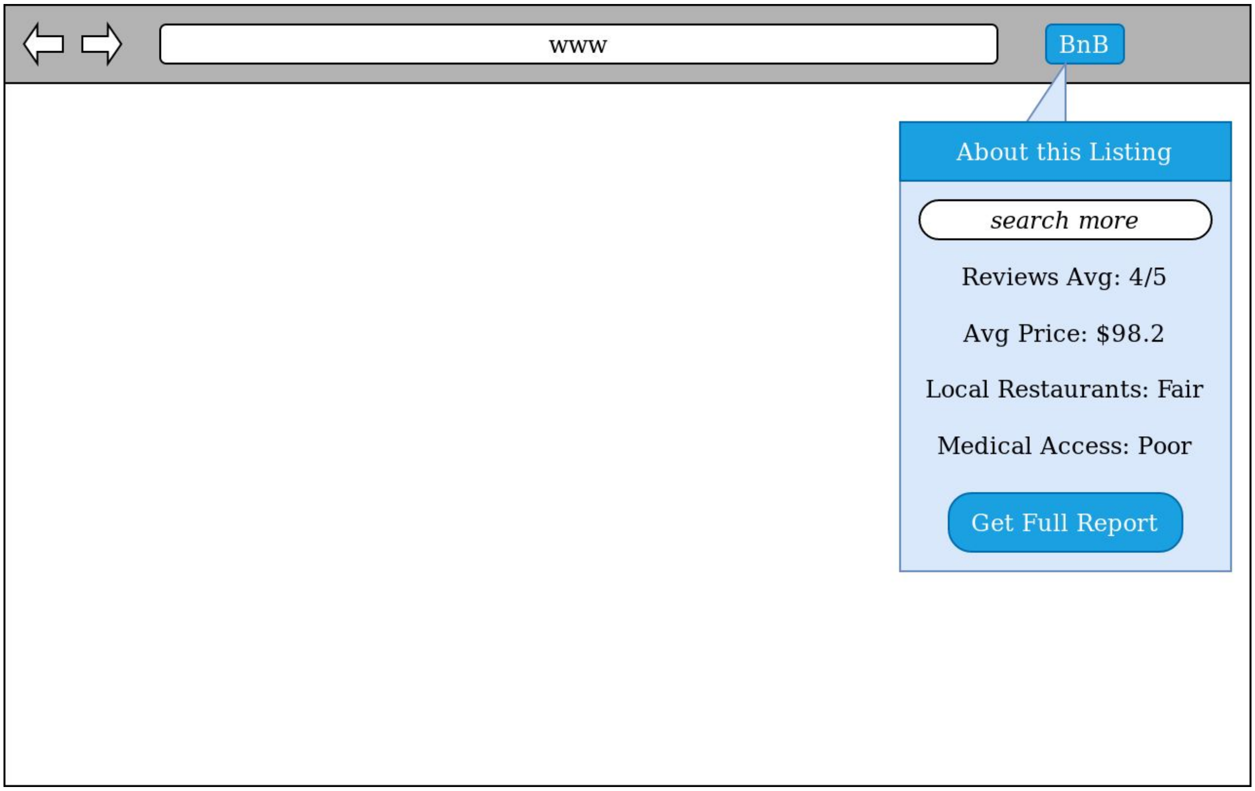


Sequence of events when the user submits a complaint/report or removes a complaint/report


Actor
Client
Server
Database
submits a complaint or report
client sends report ID and image files to server
report is inserted into database
sends back success or failure response
displays report submission success status
resolves or removes complaint/report
client sends report ID to server
corresponding report entry is removed
sends back success or failure response
displays report resolved success status

UI Mockups

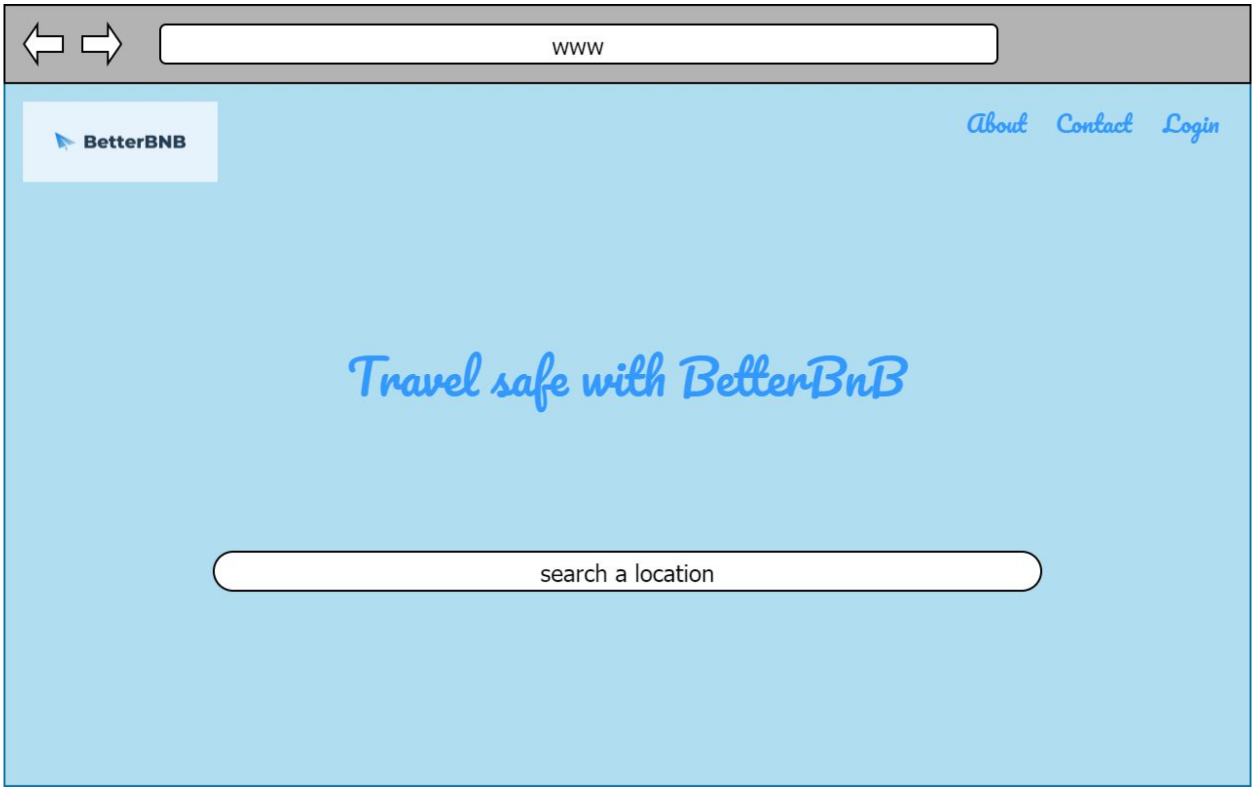
Browser Extension



This is the UI for the extension. The extension will display information according to the user's specific preferences and will also display an option to get the full report, redirecting the user to the website. If the extension doesn't detect a listing or location on the page, the user can also manually search for one within the extension.

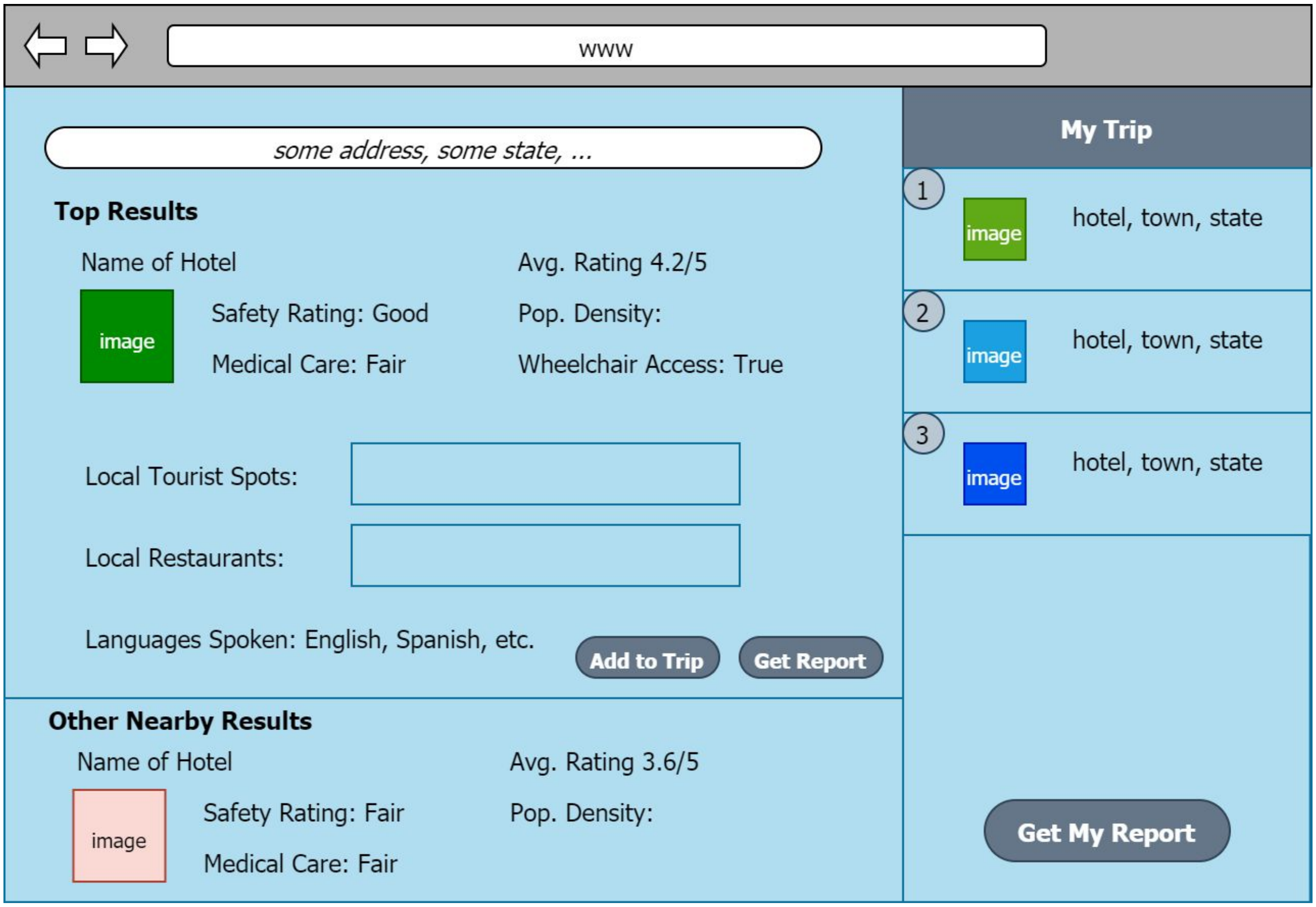
Website

Home Page



This is the home page for BetterBNB, where users will land if they search for our site or click on the logo after finishing planning for a trip. Users can search for a location and see a similar user interface as our results page (shown below) without the dashboard on the side. This is in an effort to make the site as accessible as possible especially for users who are not willing to at first sign up for an account.

Results Page For Searching a location



Top Results

Name of Hotel

Avg. Rating 4.2/5

Safety Rating: Good

Pop. Density:

Medical Care: Fair

Wheelchair Access: True

Local Tourist Spots:

Local Restaurants:

Languages Spoken: English, Spanish, etc.

Other Nearby Results

Name of Hotel

Avg. Rating 3.6/5

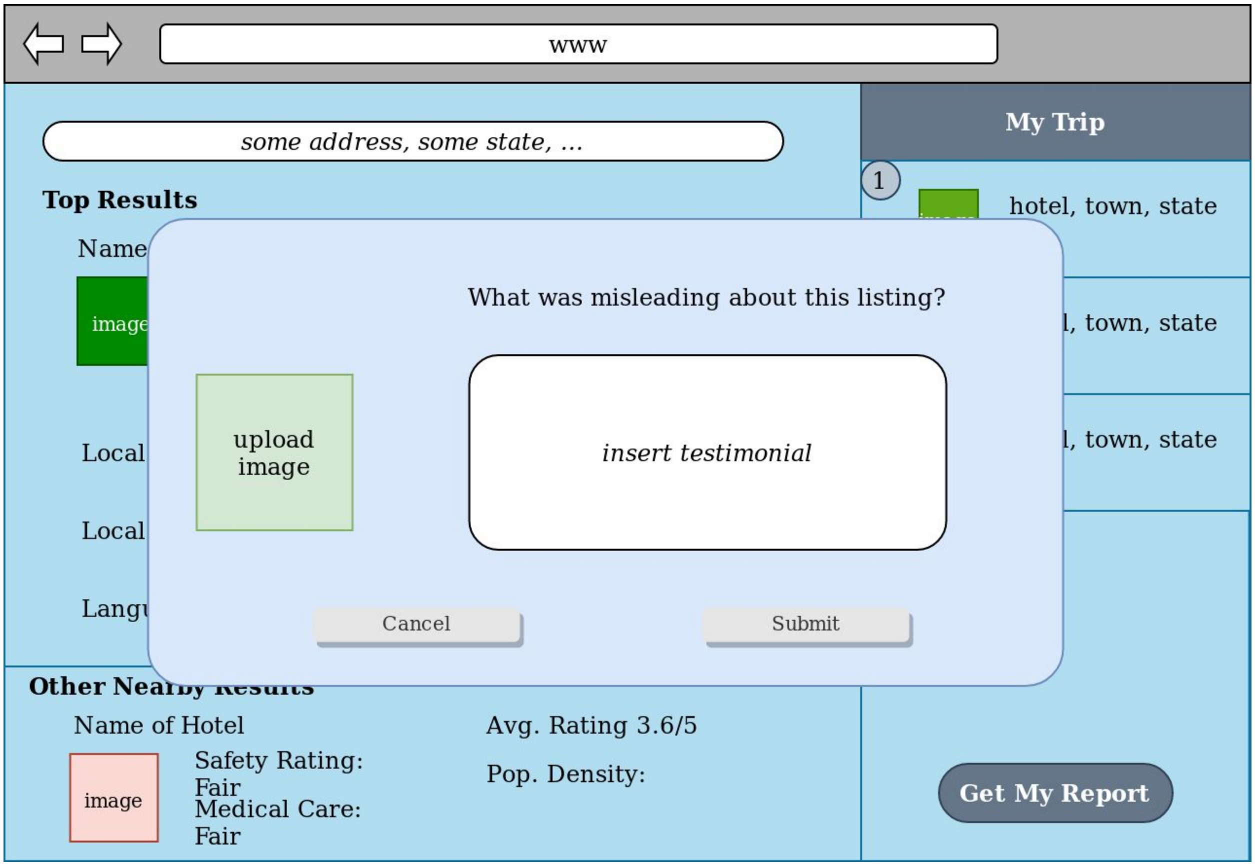
Safety Rating: Fair

Pop. Density:

Medical Care: Fair

This is the results page for when the user wants to search for a location. On the left, the results are ordered in terms of which locations appeal directly to the user's preferences. Information such as general safety, medical care, average rating, etc. will be displayed underneath each result. If you want to add a listing, you can also see a short summary of your trip on the right-hand side.

Report a Listing Window



If the user wants to report a false or misleading listing, there will be a report button that creates a small popup window. The user can upload images and an explanation/reason for the report. A compilation of all reports from multiple will be taken into account when considering if a listing should be flagged as misleading or not.