

TripTips: App for centralizing and uniformly formatting travel recommendations from friends and family



Members: Harry Chong, Paras Doshi, Willow Livengood, Hai Le
Supervisor: Dr. Mathew Burlick of Drexel University for CS375 Web Development

ABSTRACT

Our project idea is to create an application called TripTips, which stores, centralizes and uniformly formats travel recommendations from friends and family in one convenient location. A common situation when discussing travel plans with friends who have previously been to your vacation destination is that they will likely have recommendations of places to go and personal notes about their experiences. There are many ways to collect this information, e.g. email, text or notes when talking in person, and a myriad of ways to share this information, e.g. links to places on google maps or a simple list of names. The plethora of ways of sharing this information creates a challenge when trying to view recommendations on the go with limited cellular data. The person traveling, or collecting these tips from friends and family, will be able to register an account to begin crafting their own personal list. Each location that is entered by the user will be able to define the location name, address, approximate price, location type, and personal notes. Centralizing these recommendations based on location in an attractive user interface will create a stress-free method of quickly pulling up stored locations on the go while traveling.

FUNCTIONAL REQUIREMENTS

The application registers and stores multiple users. It has a custom user experience based on each account. Each user has a homepage that shows all saved locations on a map, and a quick overview of each location. The homepage also allows the user to log-out. If there are no locations, a prompt to add a location will be displayed to the user. Users can add locations to their saved lists through a form. Each location has the attributes: location name, address, approximate price, location type, and personal notes. Except for location name or address, each of these attributes are optional. If the location is not found in the map, the entry will not be added to the database. Once saved, the location is plotted on a map and their details will be shown in an information page. Users can also remove their saved locations on the detailed information page. See the architectural wireframe (Fig.2) for a detailed diagram of the user experience workflow. The main external component of the application is the use of the Google Maps Embed API and Google Geocoding API on the basic tier. Requests to the API are used to search for and display the locations using longitude and latitude information. The geolocations are then used to mark the location on a Google map. Another major component is database accessing and storage with MySQL. MySQL will store the users, and corresponding saved location information. For more information about the exact database used see the database schema in Figure 3. For a complete information flow, view Figure 3. This has all possible user interactions with the web application.

OBJECTIVES

- Create a full-stack web application
- Include User Sessions for personalized user experience
- Utilize jQuery, Ajax, and NodeJs for a endpoints to communicate data from the client side to the server side
- Include MySQL database accessing and management to hold user information and stored location information
- Utilize Google Maps Basic and Geolocation API calls to display saved locations

TECHNOLOGIES USED

Creating the appearance of the application relies on HTML5 and CSS. JavaScript, under the ES7 standard, and AJAX, for the endpoint calls, drives the functionality of the application. JSON formatting and parsing are vital in accessing and responding to endpoints and database requests. Node.js is the server running the application. A database was created using MySQL. Google Maps Embed API on the basic tier is used for location services. The Google Geocoding API was also utilized in order to obtain the longitude and latitude of the location to generate a map with markers. The use of the passport.js module simplifies the process of authenticating and validating the user's credentials. Lastly, the use of EJS (Embedded JavaScript Template) helps with generating HTML markup with plain JavaScript.

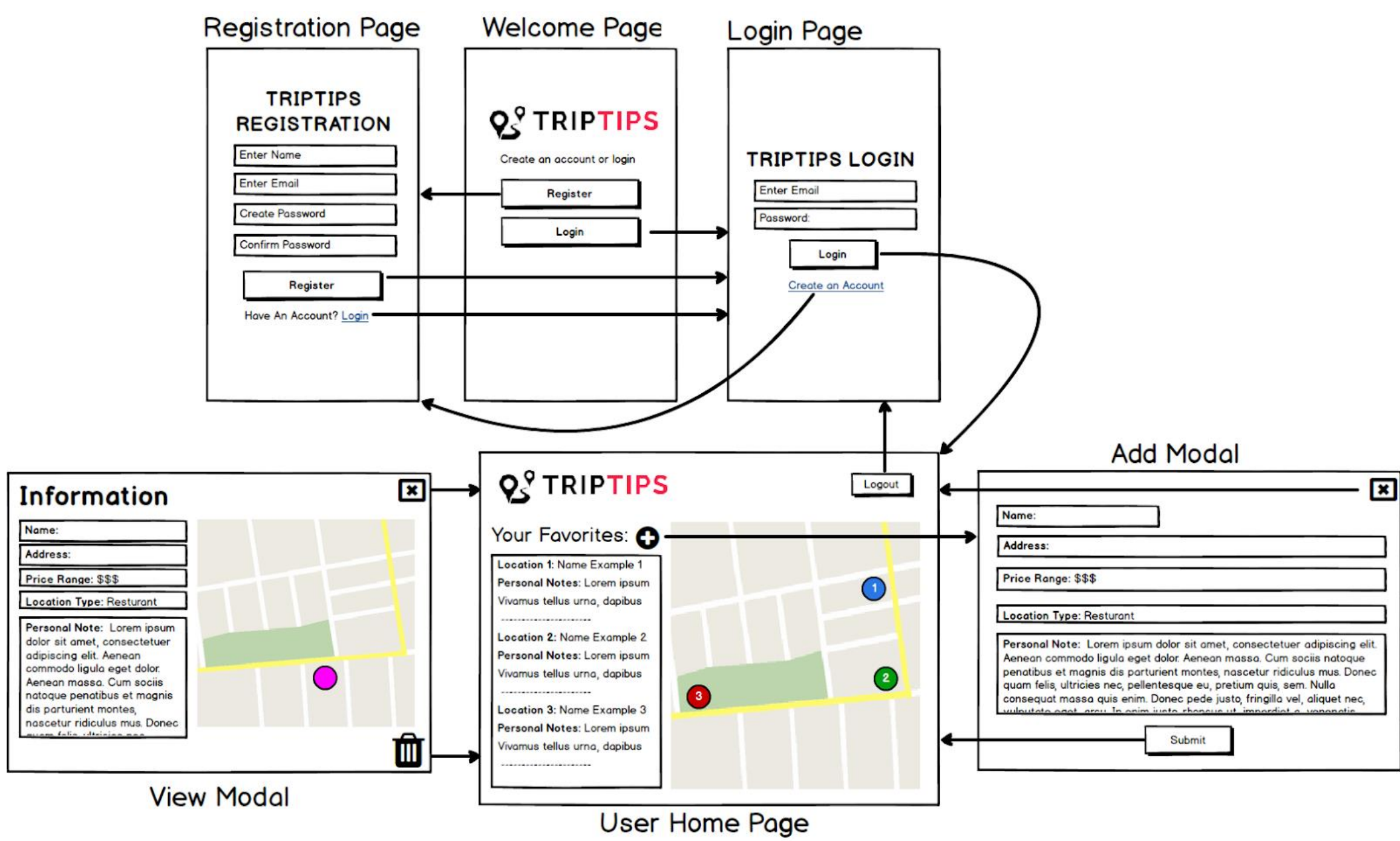
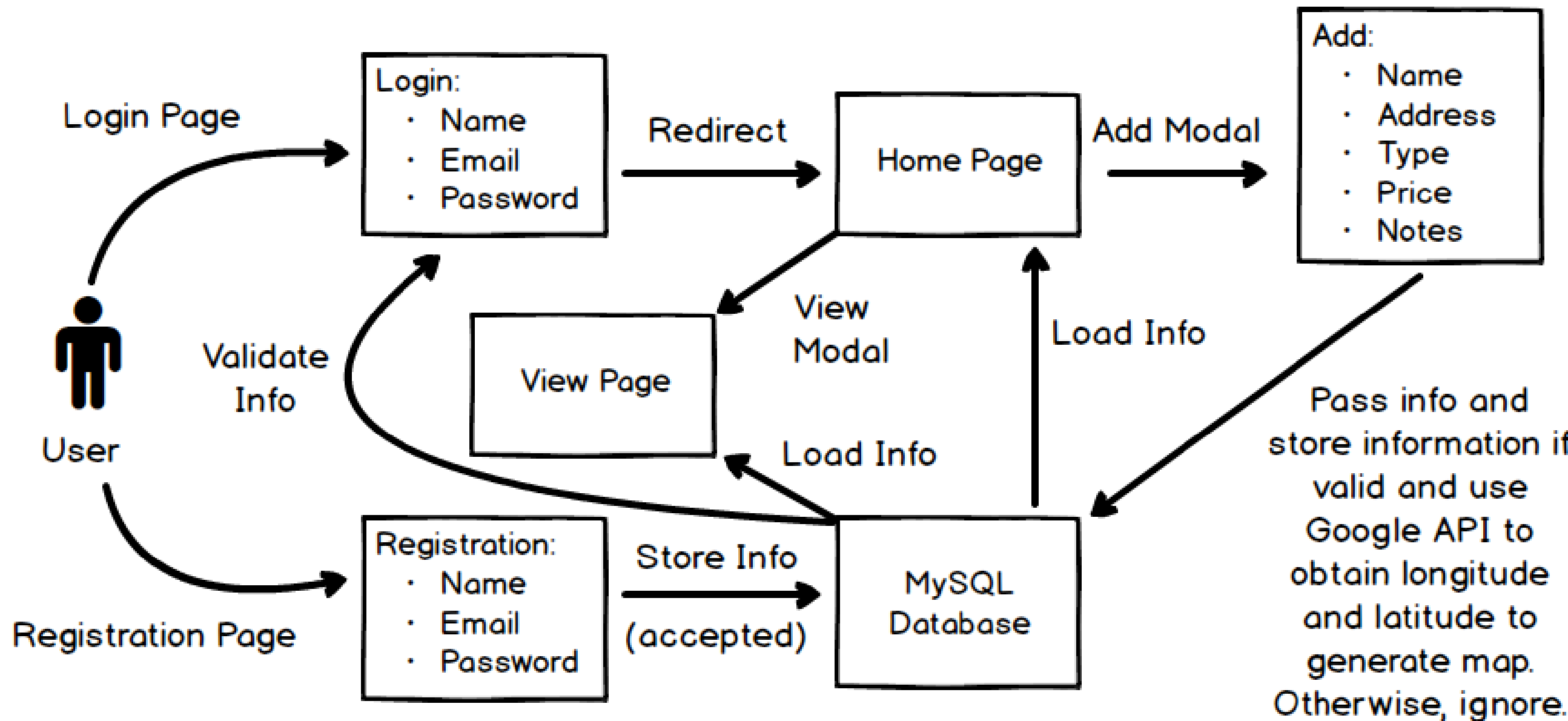


Fig 1, 2 Information Flow diagram and User Experience Wireframe

DATABASE SCHEMA

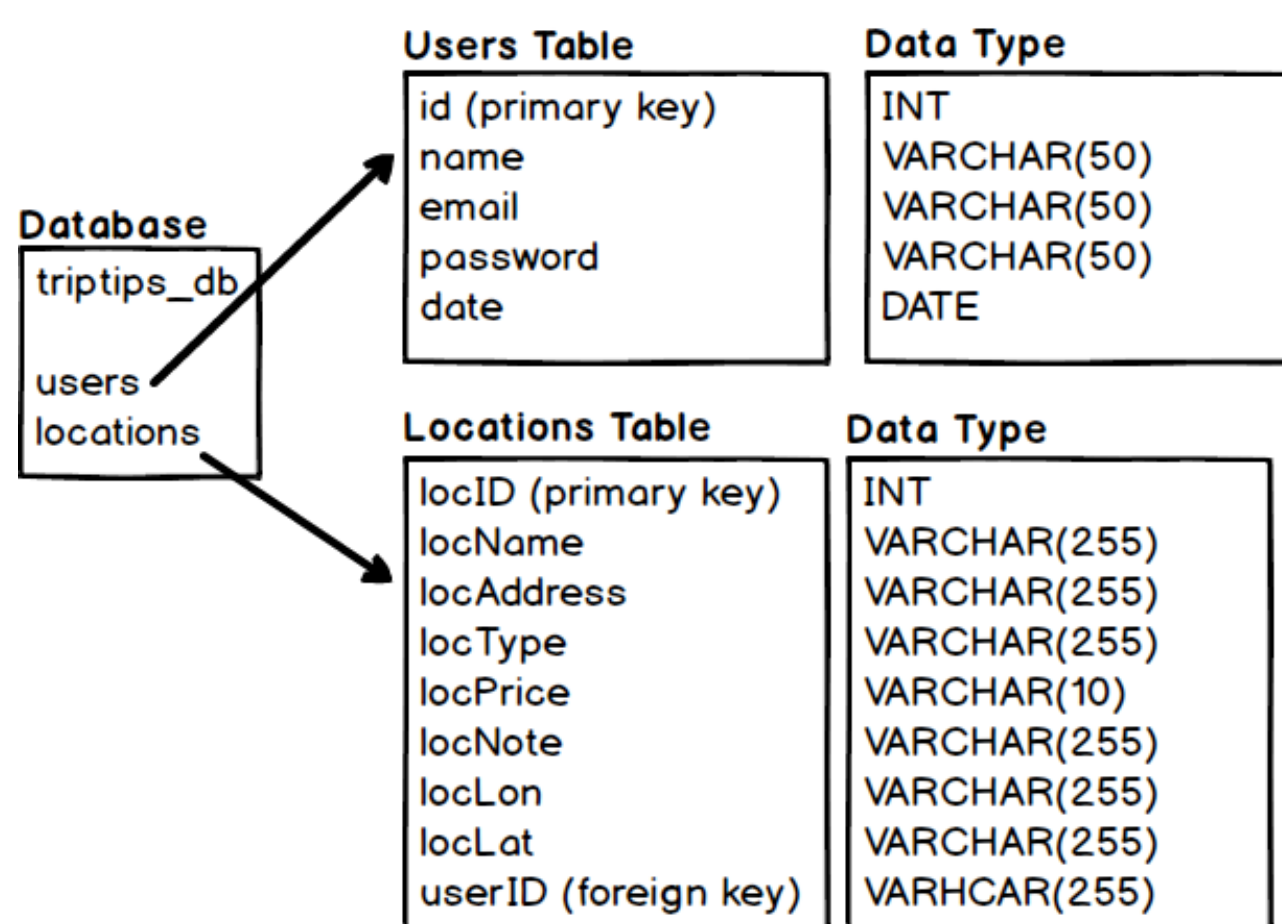


Fig 3 Database schema for SQL data storage

USER EXPERIENCE

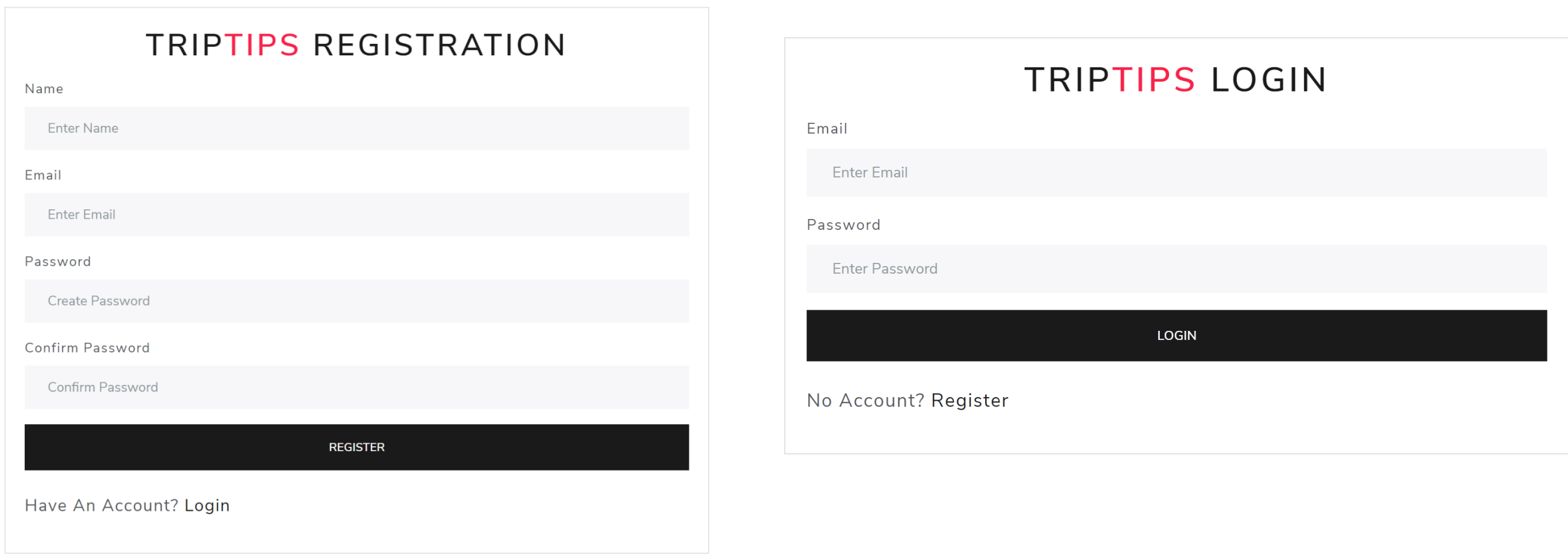


Fig 4, 5 New User Registration screen and User Login screen

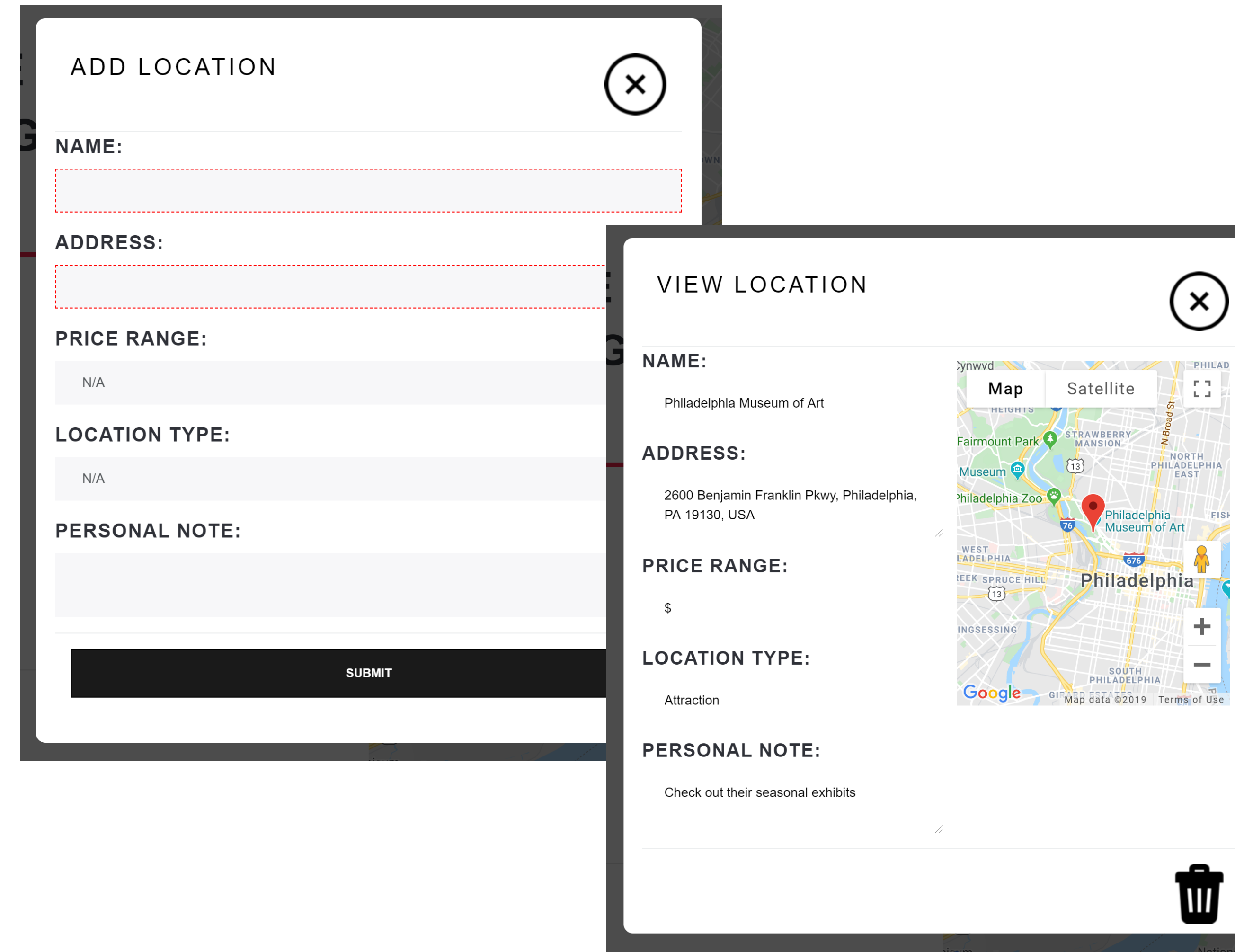
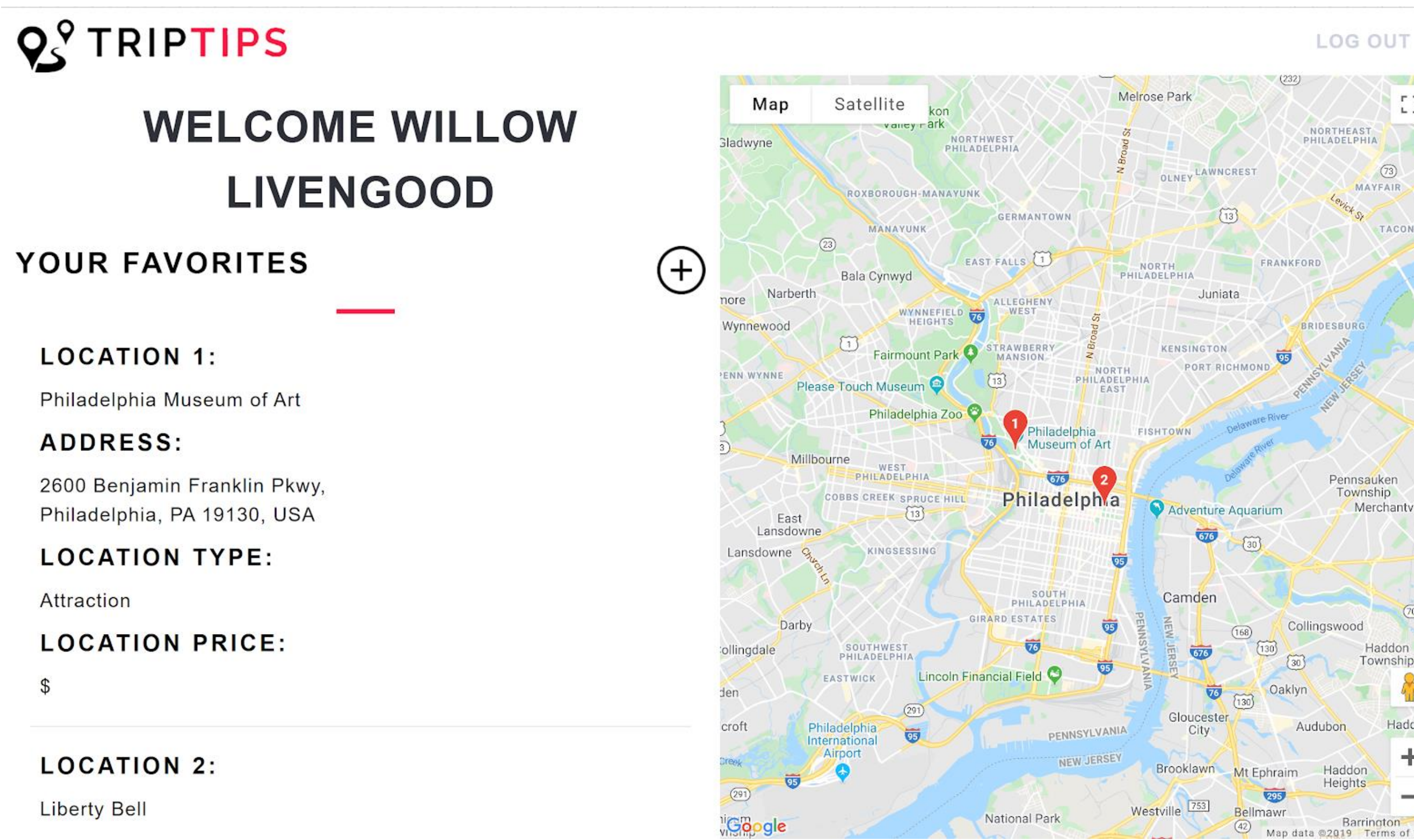


Fig 6, 7, & 8 Custom Home page, Add Modal and View Modal