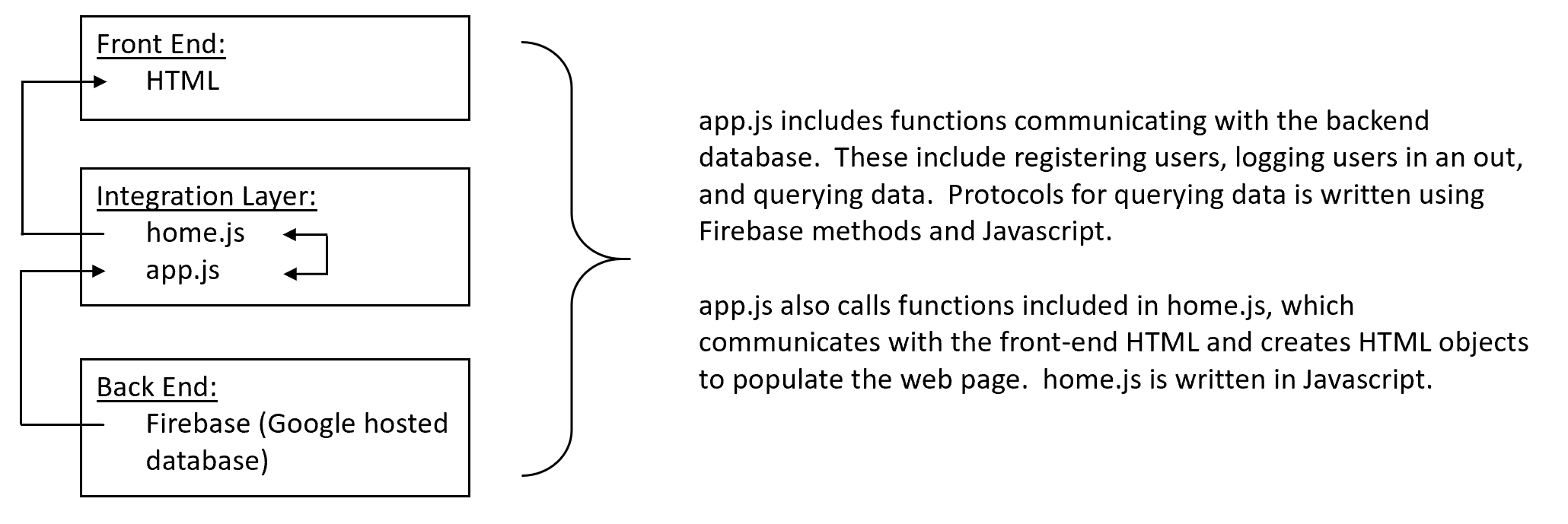
**Latalattes (104-7)**

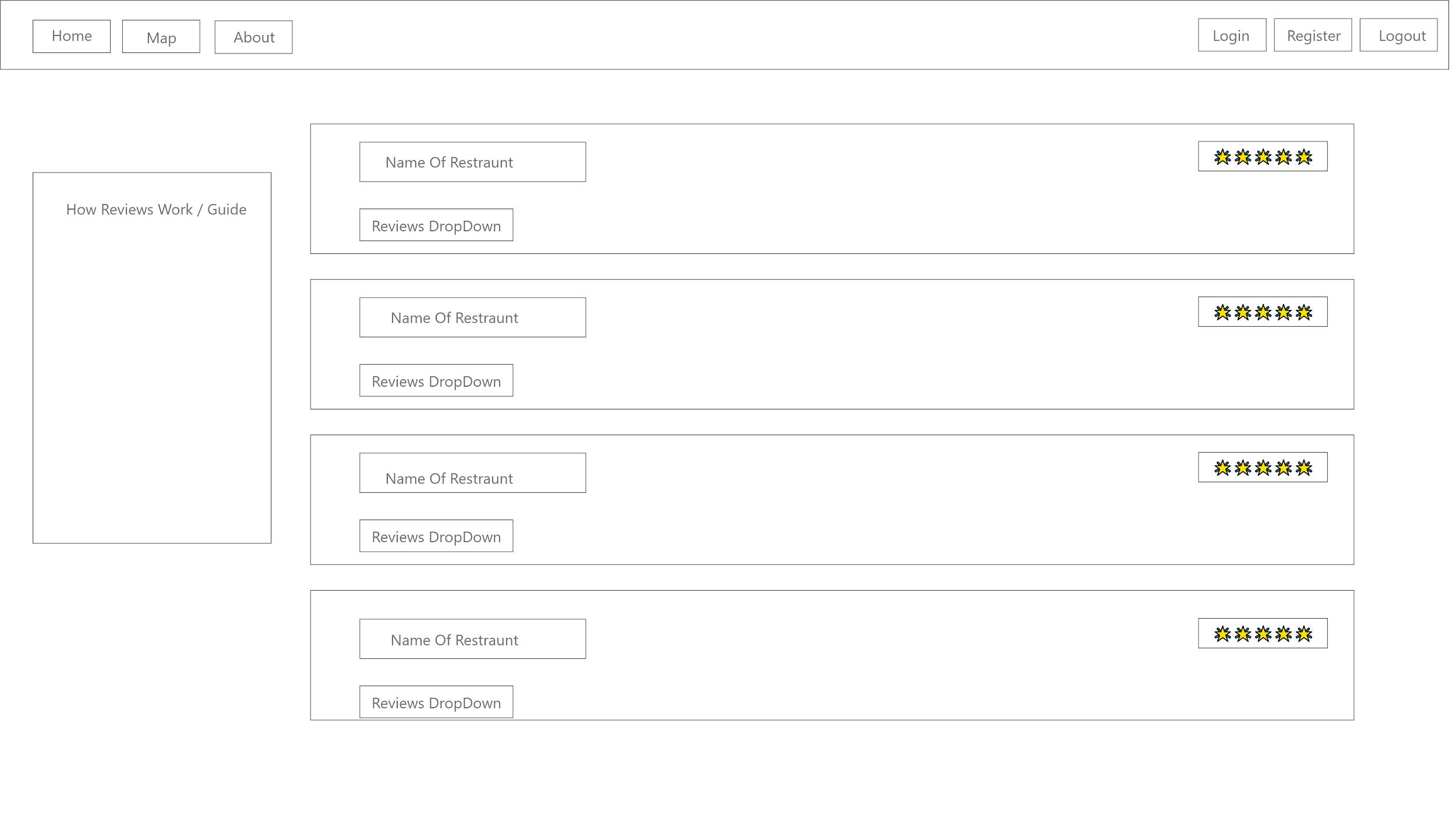
**Revised List of Features:**

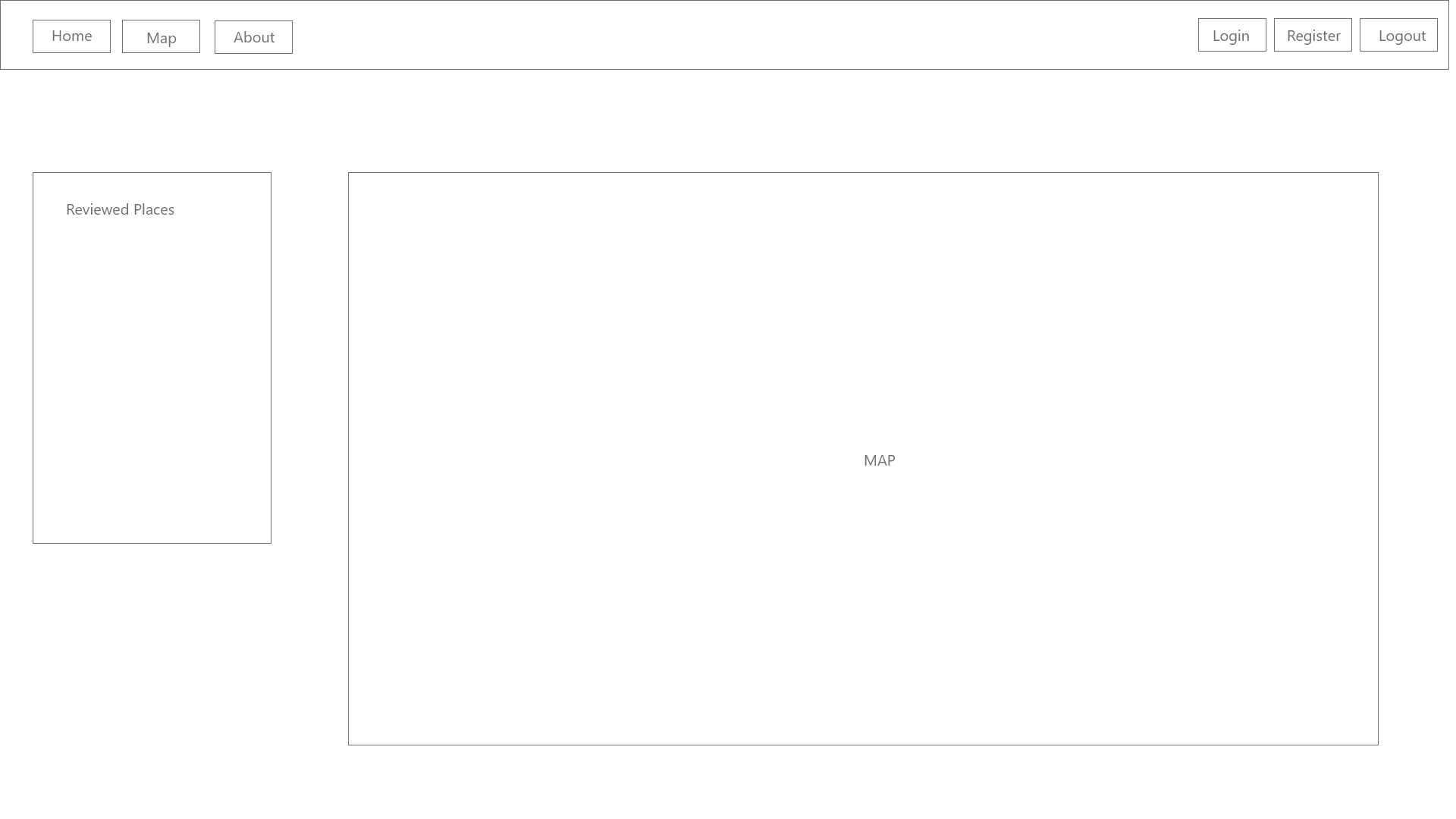
1. Login / Logout / Register
   1. Users can register an account on the website which then auto-populates the backend Firebase database
   2. Users can log in / logout of the application and the database can track authentication of users
   3. When a user is logged in, the only buttons they should see on the nav bar are logout / account info
   4. When a user is logged out, the only buttons they should see on the nav bar are register / login
2. Reviews can be added to coffee shops.
   1. Users that have created an account can write reviews on different coffee shops and the web page will automatically update when a review is added
   2. If a user is not logged in, the application will display an error message to the user
3. Star system-review
   1. Our system will append the correct number of stars to each review, as well as calculate the average rating and number of reviews for each coffee shop to be displayed on the web application
4. Get directions
   1. A button on our web page that will be linked to our maps page and sync with Google Maps
5. Store Menu
   1. A button on our web page that will link to the store menu (or online order form) depending on what is available on the coffee shops website
6. User types
   1. There will be admin / non-admin users
   2. Only admins can add coffee shops and make other users admins
7. Adding Coffee Shops
   1. Will be able to add coffee shops via front-end
   2. Only available to site admins

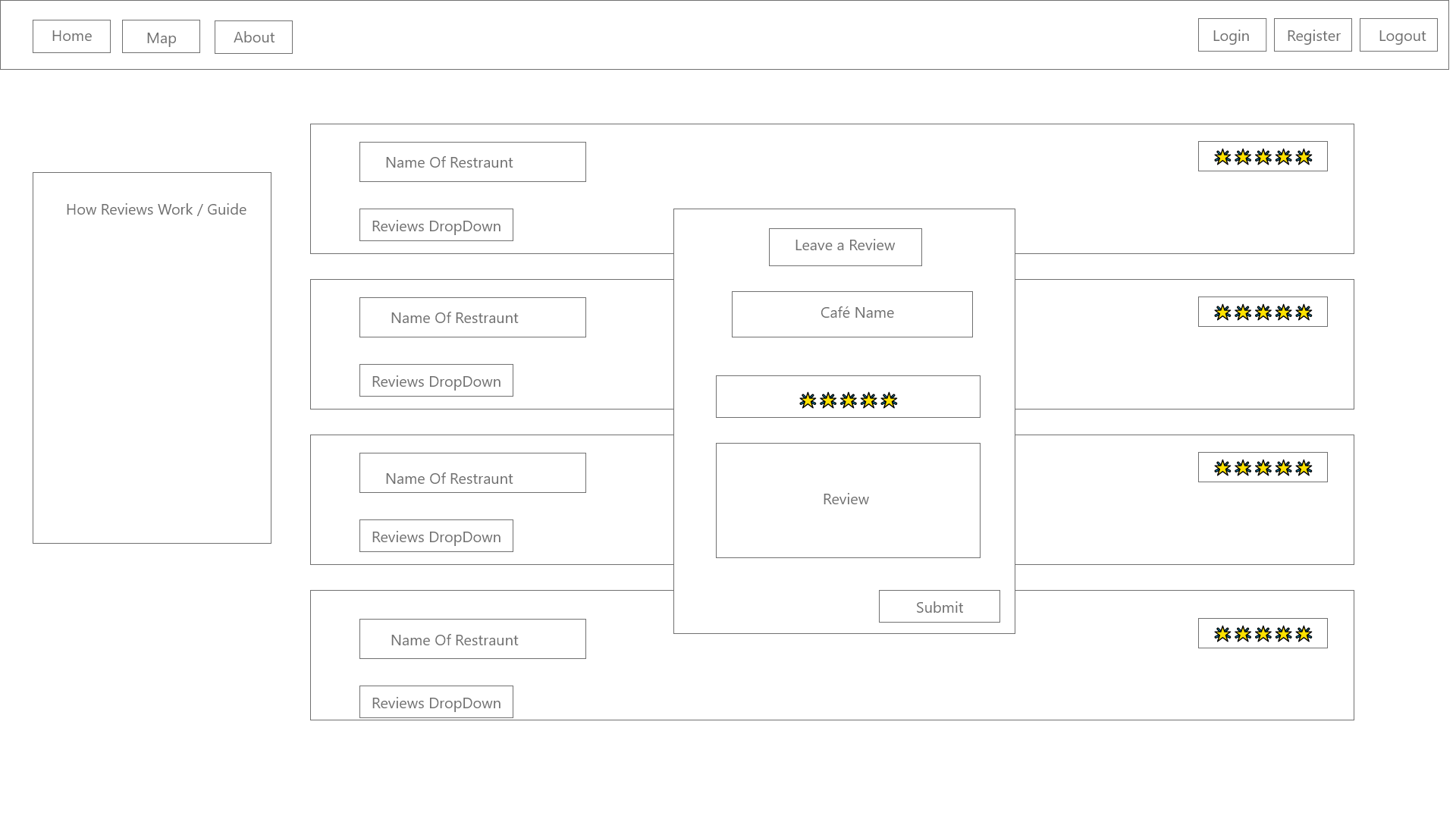
**Architecture Diagram:**



**Front End Design:**



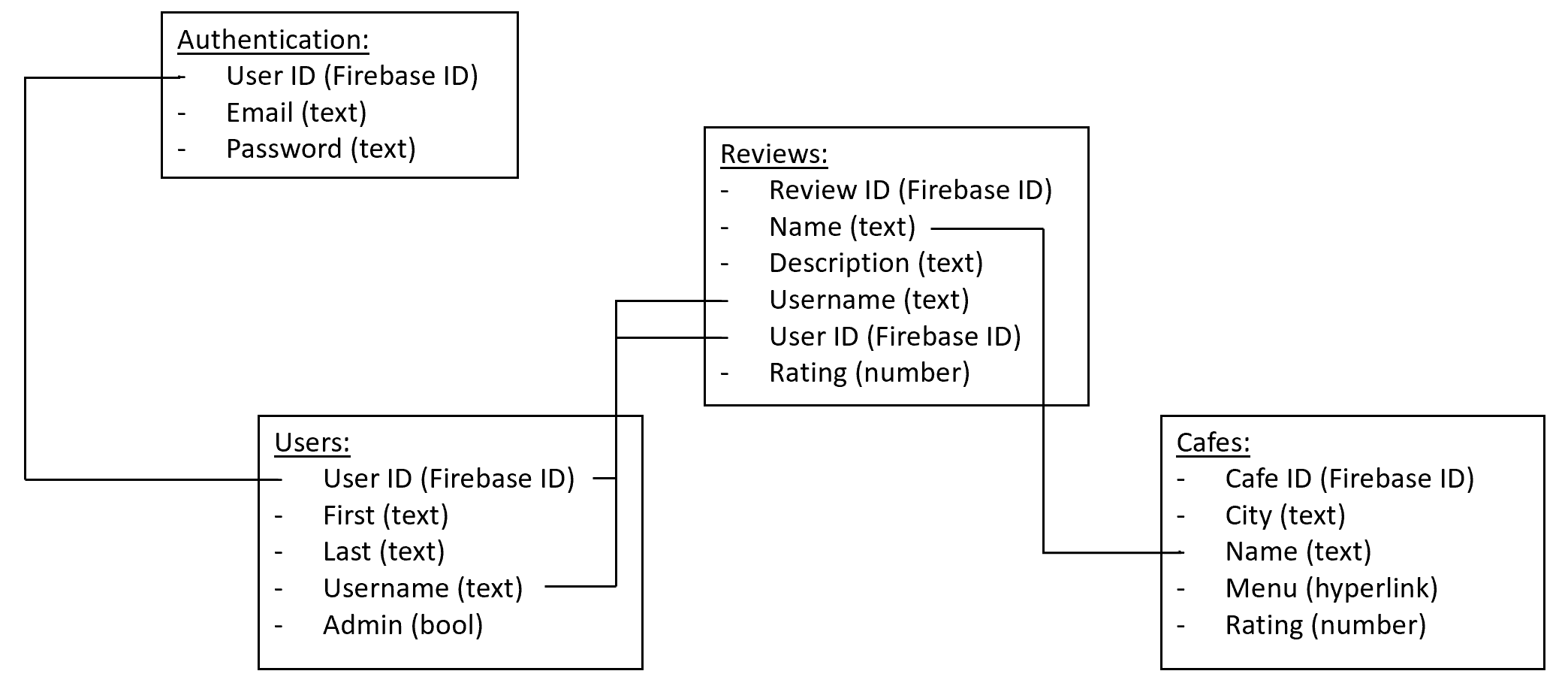




**Web Service Design:**

For our project, we are using a Firebase database to store several tables for information accessed on the website. Most of the database table information is being passed to the website (data in the review table, data in the cafes table, and certain parts of the authentication and users table) to display information to those who are on the website. The data passed from the website to the database is primarily authentication and users table-based, as well as information that the users put into their reviews so that this can be added to the table and displayed back onto the website.

**Database Design:**

****

We chose to use Firebase for our application for a few reasons. It was originally suggested as an option from our TA, Shravan. It’s hosted online for free, providing ease of use for all project members and is relatively easy to work with after some short tutorial videos. The authentication features for registering users, logging in, and logging out are also a big help when developing the app. While it isn’t set up as a relational database, you are able to pass in fields from one collection to another which can then be used to mimic the behavior of a relational database.

Our database holds three collections (Firebase term that’s essentially equivalent to tables) for different tasks: one for cafes, one for reviews, and one for users. Note that when a user first registers, they are authenticated with a User ID (generated by Firebase), email, and password. The User ID is automatically passed to the users table.

The user table stores the user’s ID, first and last name, username, and a boolean value indicating whether or not they are a site admin.

The cafe collection stores the name of the cafe, city, a hyperlink to the menu of the cafe, and an overall rating for the cafe (intended to be the average of all individual user ratings).

The reviews table stores the ID and username of the user writing the review (providing a relation to the users table), the name of the cafe (providing a relation to the cafes table), description, and rating.