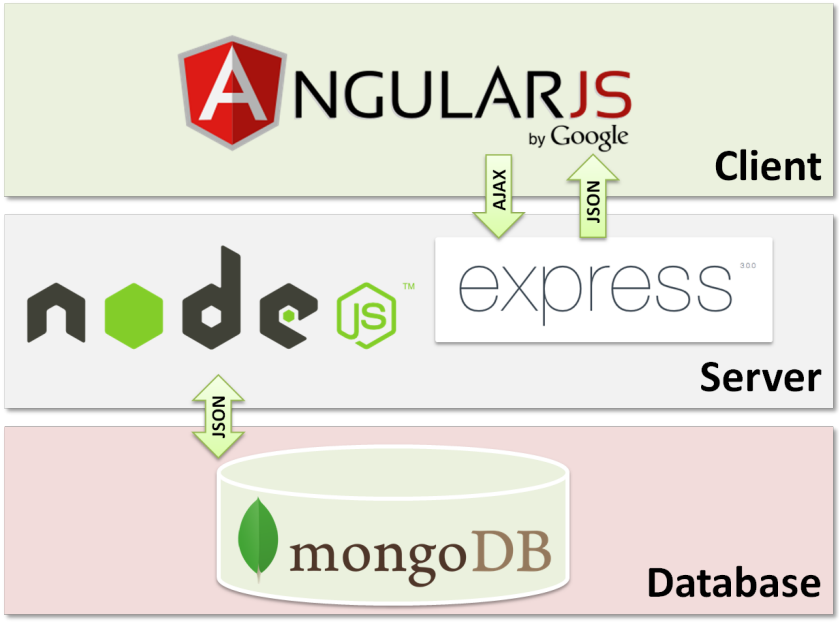
Team 4: Ibrahim, Kristel, Joanne, Shreya

Waitster will be heavily reliant on the very large database which contains all the wait-times for each restaurant and its attributes. Because of this, we will be using the MEAN platform and a three-tiered architecture with client, server, and database (see diagram below). mongoDB will allow us to manipulate the database efficiently, and using Angular we can implement a good user interface that is integral to our application since it is very user centric. When considering which architecture and platform to use, we dabbled with the idea of using Python with django and Flask. However, because a Python backend comparatively has a slow run-time, it would not be ideal in returning search queries due to our expectedly large database.



UPDATE (4/23/16)

Over the semester, the developing platform we used for Waitster changed to accommodate our developing and testing needs. For the purposes of this project, we essentially used frameworks and libraries that “worked” without sacrificing efficiency and reliability.

Database: IndexedDB

We changed our database system from MongoDB to IndexedDB because of its convenience for testing and development purposes. Over time, Waitster’s functionality became more client-side heavy than server-side. IndexedDB was a good compliment to this structure because it is a Javascript-based, object-oriented database that lives on the user’s browser. This makes it

unique to each user's' queries, keeping it light. It also performs transactions asynchronously, ensuring speed and concurrency.

Routing & Back-End: Javascript, AJAX, JQuery, Express

To communicate with the delivery.com and Google Maps APIs, we used vanilla Javascript, JQuery, and AJAX calls. Using these Javascript libraries was beneficial because the API calls were also performed asynchronously. Javascript also worked well with passing information to the HTML templates, displaying data seamlessly to the front-end.

Front-End & UI: HTML, CSS, Bootstrap, AngularJS

As commonly practiced, we used HTML and CSS for the front-end user interface of Waitster. We also used Bootstrap classes and templates to easily polish the look and feel of the application. AngularJS was also used minimally to display the header on each page of Waitster because it was a recurring UI element.