HELPFUL FRONT-END DEVELOPMENT

BY JOSH BRUNNER, B.S. CSSE — MARCH 13, 2015

INTERNSHIP

From the summer of 2014 into the Fall Quarter at UW Bothell, I interned at Helpful Human as a Front-end developer. Helpful Human is a young, interactive consultation and development company in Seattle.

When I began my internship, I was their first employee aside from the CEO, Mark Sandeno, and the Lead Developer, Nick Glenn. Since then, Helpful Human has grown to 9 employees. We are capable of producing complete, interactive web applications for our clients.

Helpful Human follows a hybrid of Agile and Scrum for it's project planning methodology. Sprints typically last two weeks, and a typical contract with a company tends to span over 3-4 months.

PROJECT

My internship involved:

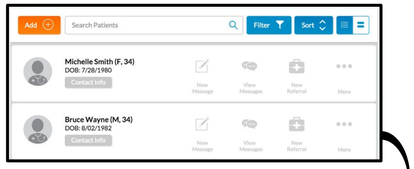
- ➔ Learning new programming languages
- ➔ Working directly with clients
- ➔ Complex problem solving

With the fast-paced, technological environment that we live in today, the demand for fast, reliable, and easy-to-access systems has never been greater. The project that I worked on came from a client in need of a software solution for their proposed Health Care management system. The value this proposed software offers is to provide a means to connect patients and providers in a faster, more reliable, and accessible way than current software solutions afford.

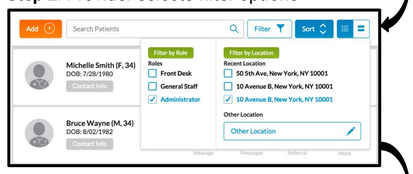
The proposed system needed to allow direct communication between healthcare providers and their patients. In addition, patients needed a way to search for and find providers, and visa versa. Due to the high demand for mobile access to everything, this system needed to be able to support mobile, tablet, and desktop design.

DROPDOWN SEARCH FILTER

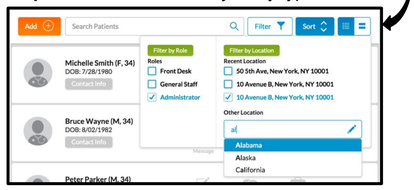
Step 1: Provider clicks "Filter" button



Step 2: Provider selects filter options



Step 3: Provider can filter by a city typeahead



Functionality

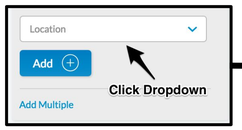
- ➔ While searching for patients, a provider should be able to filter the search results
- ➔ Filtering can be done by selecting checkboxes or searching in a text field
- ➔ If a provider clicks outside of the dropdown, it closes

Challenges

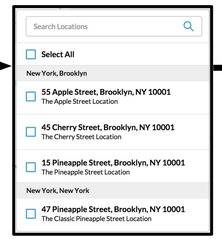
- ➔ Handling click-events using jQuery to behave as a user would expect
- ➔ Applying filters to provider's search results
- ➔ Creating modularized functionality in code

SEARCHING FOR PROVIDER LOCATION

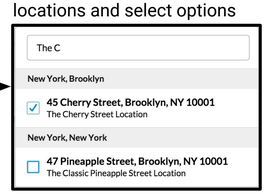
Step 1: Provider clicks dropdown button



Step 2: Provider views relevant office locations



Step 3: Provider can search locations and select options



JSON data must be structured

```
$scope.providerLocations = [
  {
    cityState: "New York, Brooklyn"
    address: "55 Apple Street, Brooklyn, NY 10001"
    location: "The Apple Street Location"
    selected: false
  },
  {
    cityState: "New York, Brooklyn"
    address: "45 Cherry Street, Brooklyn, NY 10001"
```

Challenges

- ➔ Mapping JSON data into a suitable set of objects, so that you can reason about the data
- ➔ Writing CSS to display the iterated JSON data in the HTML elements properly

HTML template (in Jade)

```
.input.as-filter-input(type="text" placeholder="Search Locations" ng-model="filterBy")
p text: {{ filterBy.length }}
.location-picker
  .input.as-select-all(ng-hide="filterBy.length")
    .checkbox-box.width-100
      input(type="checkbox"  id="selectAll" ng-model="selectedAll")
      label(for="selectAll") Select All

  .location-category(ng-repeat="(key, value) in ngModel | filter:filterBy | groupBy: 'cityState'")
    p.category-label {{key}}
    .location-item(ng-repeat="loc in value")
      .input
        .checkbox-box.width-100
          input(type="checkbox"  id="location-{{ $parent.$index }}-{{ $index }}" ng-model="loc.selected")
          label(for="location-{{ $parent.$index }}-{{ $index }}")
            .location-address {{loc.address}}
            .location-info {{loc.location}}

i.icon-search.as-filter-icon.size-18(ng-show='!filterBy')
```

COURSE WORK

CSS 360 - Software Engineering
During CSS 360, I was provided with a glimpse into the lifecycle of software in the workplace. Throughout this class were shown how software businesses are run, so I knew what to expect when I began working.

CSS 350 - Management Principles for Computing Professionals
I was taught about teamwork and the importance of clarity in communication. In addition, I learned to be comfortable to say "No" when "No" was appropriate.

CSS 480 - Principles of Human-Computer Interaction
I learned about building user interfaces (UI) while thinking about the experiences those interfaces afforded. Additionally, I learned to consider overall accessibility when designing components.

CSS 343 - Data Structures, Algorithms, and Discrete Math
This is where I learned how to debug software while considering a multitude of other factors that may be causing a single bug. I learned to be persistent and to step back and think about what is going on "under the hood."

LESSONS LEARNED

Things Change, Very Quickly
Be comfortable being uncomfortable. Software, especially in web dev, changes quickly and I've learned to accept and embrace this notion.

Estimation Is Difficult
It can be difficult to estimate the time a task will take to complete. This is due to an array of factors that can affect any given project's deadline.

Fail Fast, Fail Often
Recognize failure early on and work towards an agile state where plan changes are encouraged rather than discouraged.

Plan Retrospectives
Purposefully set aside time to reflect upon quality/quantity of work that was completed. Doing so fosters growth in the ability to complete tasks on time.

TOOLS

ACKNOWLEDGEMENTS

Many thanks to:

- ➔ Capstone advisor, David Socha, Ph.D.
- ➔ Sponsor, Mark Sandeno
- ➔ Mentor, Nicholas Glenn