

# COMP 3100

## Group 31 Iteration 2 Documentation

Ryan Hatcher (201916418)

Zachary Batten (201747524)

### Document Update:

Nearly all of the functionalities that we originally listed are implemented for client side. We have the coordinates of all major centres across Newfoundland along with other information such as their phone numbers. Our current databases are a little unpleasant because of the nature of the data. A lot of things in the field description is already in HTML format which makes it ready to implement on the front end. We also have some stretch goals like calculating routing distances. However, we will need to be farther into implementing the front end before we can begin implementing. With all that being said, we have covered most of the server side code which will allow us to begin implementing the client side.

### Models:

We have models for culture centers, ferries, hospitals, libraries, municipalities, and towns. Each of these models have Create, Read, Update, and Delete functions (CRUD) that allow us to edit each collection with ease. Reading functions are necessary for our application because we want to allow users to make a list of objects that they like so they can refer back at any time. We want our application to be a community effort, meaning that we want people to add their own sites, libraries, towns, etc. For people to be able to edit or add these sites, our application will need the create and update functions. If, for example, a hospital in an area shuts down, we want to be able to easily remove it. Delete functions allow us to do that easily.

### Data Models:

We set up our data models to be embedded to allow us to easily access them. The only issue we have at this point is that our description that has a host of different data in it, this is set up to be used as a description box. We kept it as such because there is no use for things such as phone numbers, when we can show the entire description instead. Also, it worked out well because it was already formatted in a HTML format.

### Testing Documents:

To test our functions, we used Mocha, a VS studio debugging software, along with in-line tests. We tested each create, read, update, and delete function for each model that included some passing and failing cases, but generally it worked out. We tested each CRUD function to ensure that they were working as we needed them. When it comes to testing, you can always be more thorough when it comes to performance and acceptance testing. With all of that being said, we believe that we did a thorough set of tests to ensure that our functions were working as planned.