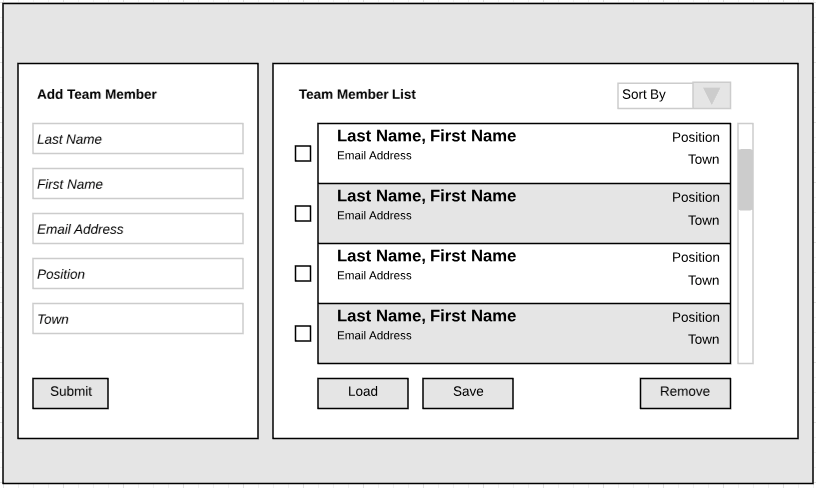
You will find in this document two challenges. When doing these challenges, please create a public git repo on Github. When you start the challenge, please push your first commit and add [**MJGTwo**](https://github.com/mjgtwo)as a collaborator within the 1st hour of commencement of each challenge; the commit doesn’t need to be working code but rather to see how much you progress in an hour time period. **You should be the only other listed collaborator of this repo.**

## UI Challenge: Team Member Directory

You are a part of a budding startup that needs to keep track of the information of its membership. This product will use a ReactJS frontend **and** a SQL database to contain the data.



### Requirements

The application should:

* Load the data below into the existing data section on page load.
* Have the ability to create team members that will immediately prepend to the list on submit.
* Have the ability to remove selected team members from the list.
* Have the ability to sort the list by any of the fields, both ascending and descending.
* Type ahead searching is a *nice to have* (**not** required).
* Last Name, First Name and Email Address are **required fields**.
* Match the spirit of the mock up but artistic freedom is encouraged.
* All data exchanges should be in JSON format inside the frontend.
* Any design decisions or alterations to this mock up made should be discussed in the README.md document included in your repo and you should be prepared to discuss them if need be.

### Existing Data (JSON)

[

{"last\_name": "Gardner", "first\_name": "Michael", "email\_address": "michael.gardner@ippsecpro.com", "position": "CTO & Co-Founder", "town": "Troy"},

{"last\_name": "Shulman", "first\_name": "Alexander", "email\_address": "alexander.shulman@ippsecpro.com", "position": "CEO & Co-Founder", "town": "Boston"},

{"last\_name": "Volk", "first\_name": "Ben", "email\_address": "ben.volk@ippsecpro.com", "position": "lead architect", "town": "Troy"},

{"last\_name": "Kloss", "first\_name": "Peter", "email\_address": "peter.kloss@ippsecpro.com", "position": "COO", "town": “Framingham”},

{"last\_name": "McCurry", "first\_name": "Peter", "email\_address": "peter.mccurry@ippsecpro.com", "position": "android developer", "town": "Troy"},

{"last\_name": "Giron", "first\_name": "Paulina", "email\_address": "Paulina.Giron@ippsecpro.com", "position": "CMO", "town": "Boston”}

]

## Prime Number Generator Coding Exercise

Overview:

Your task is to use test driven development to implement a prime number generator that returns an ordered list of all prime numbers in a given range (inclusive of the end points). You must implement the interface specified below. You may also create any other methods, interfaces and/or classes that you deem necessary to complete the project. You should also develop a small main program to drive your generator and to allow the user to specify the prime number range via the command line. To successfully complete the exercise, all unit tests must pass as well as provide 100% code coverage. You must use **Java** for this assignment, and again document any design choices made in the README.md document within the repo.

Note:

* The code should handle inverse ranges such as 1-10 and 10-1 are equivalent.
* Ensure that you run a test against the range 7900 and 7920 (valid primes are 7901, 9707, 7919).
* Expect the user input to be *robust* (e.g. symbols outside of integers) so design your code accordingly
* Coding Style should be standardized to section 4 of the [Google Java coding standards](https://google.github.io/styleguide/javaguide.html#s4-formatting)

Interface:

Interface PrimeNumberGenerator {

List<Integer> generate(int startingValue, int endingValue);

Boolean isPRime(int value);

}

Definition (from wikipedia):

In mathematics, a prime number (or a prime) is a natural number which has exactly two distinct natural number divisors: 1 and itself. The first twenty-six prime numbers are:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101