

# Project: Lightning

Team: Winners

Matt May & Alan Zheng

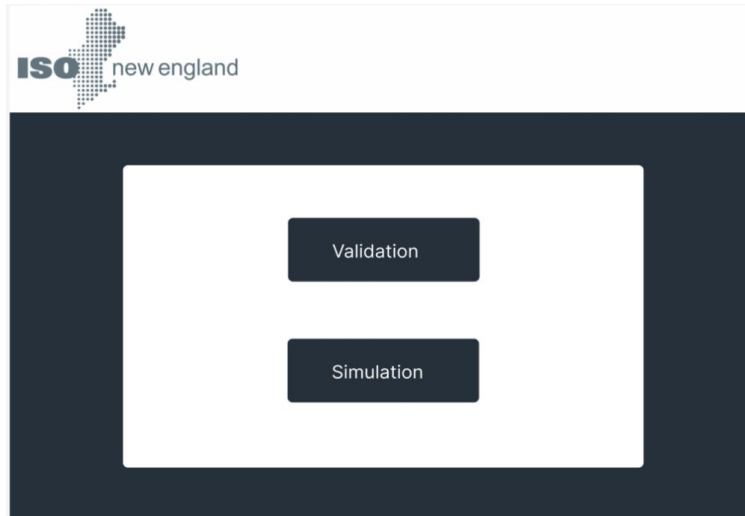




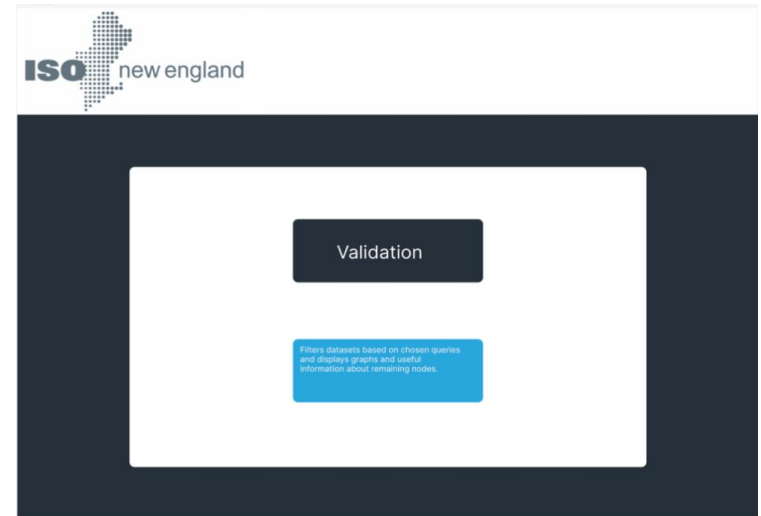
# Front-end Progress

- **Updated the filtering page** User Interface
  - Incorporated client's feedback from last week
  - Enhanced scalability and ability to customize
- **Added functionality** to Figma
  - Allowed us to have a better understanding of how the user will interact with the interface
  - Provided insights on how to code the interface
- Gained more **knowledge about React**
  - Familiarize ourselves with components
  - Explored stateless and stateful components

# UI Updates



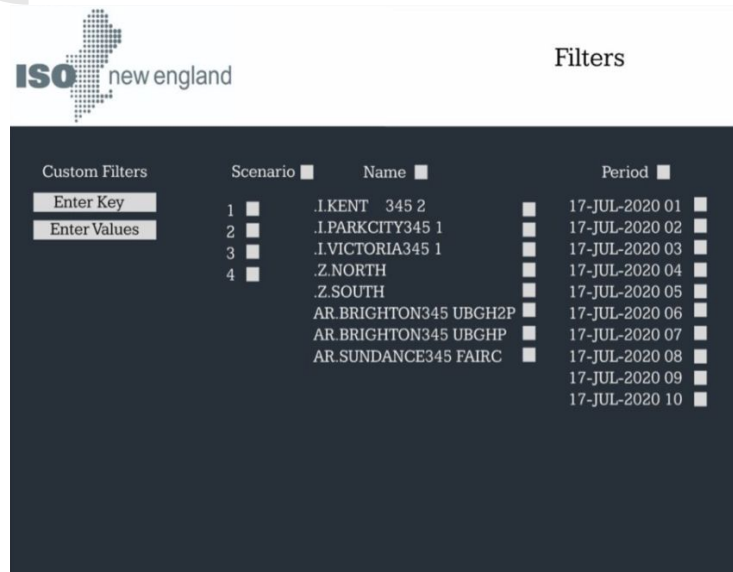
**Initial Interface**



**Updated Interface**

Figure 1: Use case and the description

# UI Updates



ISO new england

Filters

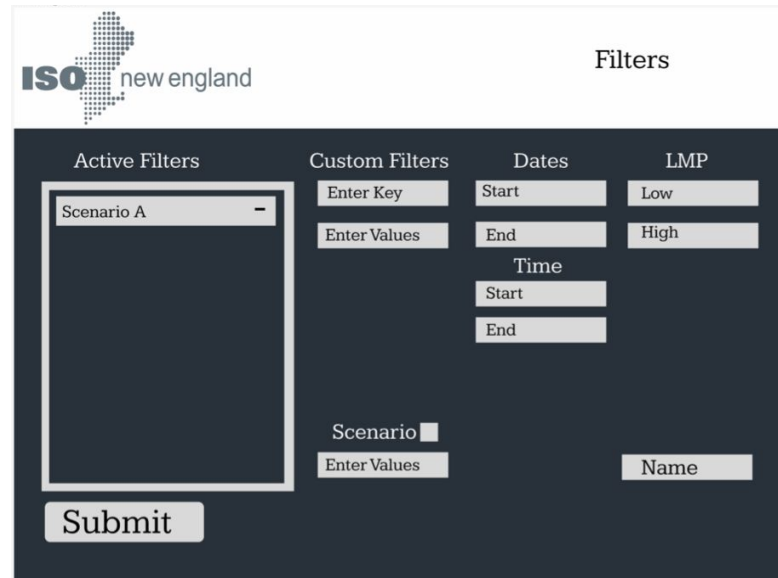
Custom Filters

Enter Key

Enter Values

Scenario	Name	Period
1	.I.KENT 345 2	17-JUL-2020 01
2	.I.PARKCITY345 1	17-JUL-2020 02
3	.I.VICTORIA345 1	17-JUL-2020 03
4	.Z.NORTH	17-JUL-2020 04
	.Z.SOUTH	17-JUL-2020 05
	AR.BRIGHTON345 UBGH2P	17-JUL-2020 06
	AR.BRIGHTON345 UBGHP	17-JUL-2020 07
	AR.SUNDANCE345 FAIRC	17-JUL-2020 08
		17-JUL-2020 09
		17-JUL-2020 10

Initial Interface



ISO new england

Filters

Active Filters

Scenario A

Submit

Custom Filters

Enter Key

Enter Values

Dates

Start

End

Time

Start

End

LMP

Low

High

Scenario

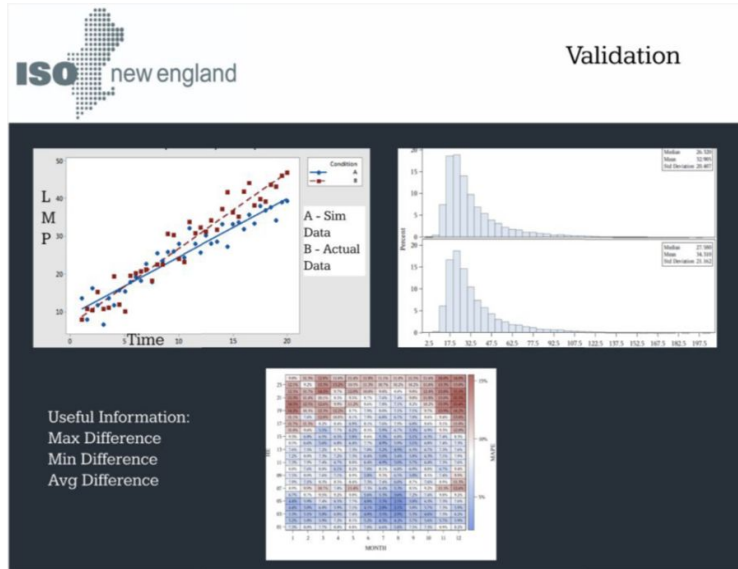
Enter Values

Name

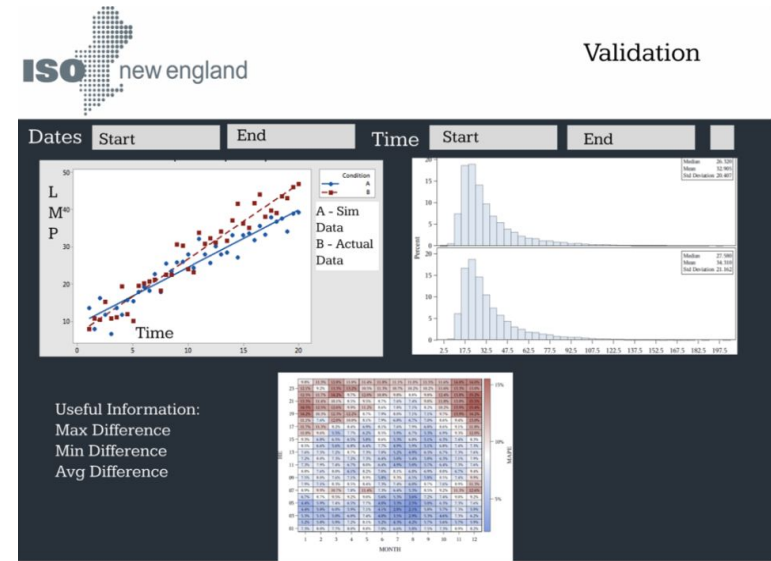
Updated Interface

Figure 2 : Added the ability to search to improve scalability and customize the filters

# UI Updates



Initial Interface



Updated Interface

Figure 3 : Added the ability to allow the users to customize the graph by time



## Backend, Database Progress

- Explored MongoDB Atlas in depth using sample code
- Successfully imported the data into MongoDB!
- Currently working on a script to implement a GET pull query to/from the database to finalize REST architecture

QUERY RESULTS: 1-20 OF MANY

```
_id: ObjectId('6346f14103dc8d659ea468f1')  
SCENARIO_ID: "1"  
PNODE_NAME: ".I.KENT    345 2"  
PERIOD_ID: "17-JUL-2020 01"  
LMP: "30.18"
```

Figure 4 : Sample entry in the database

# System Architecture

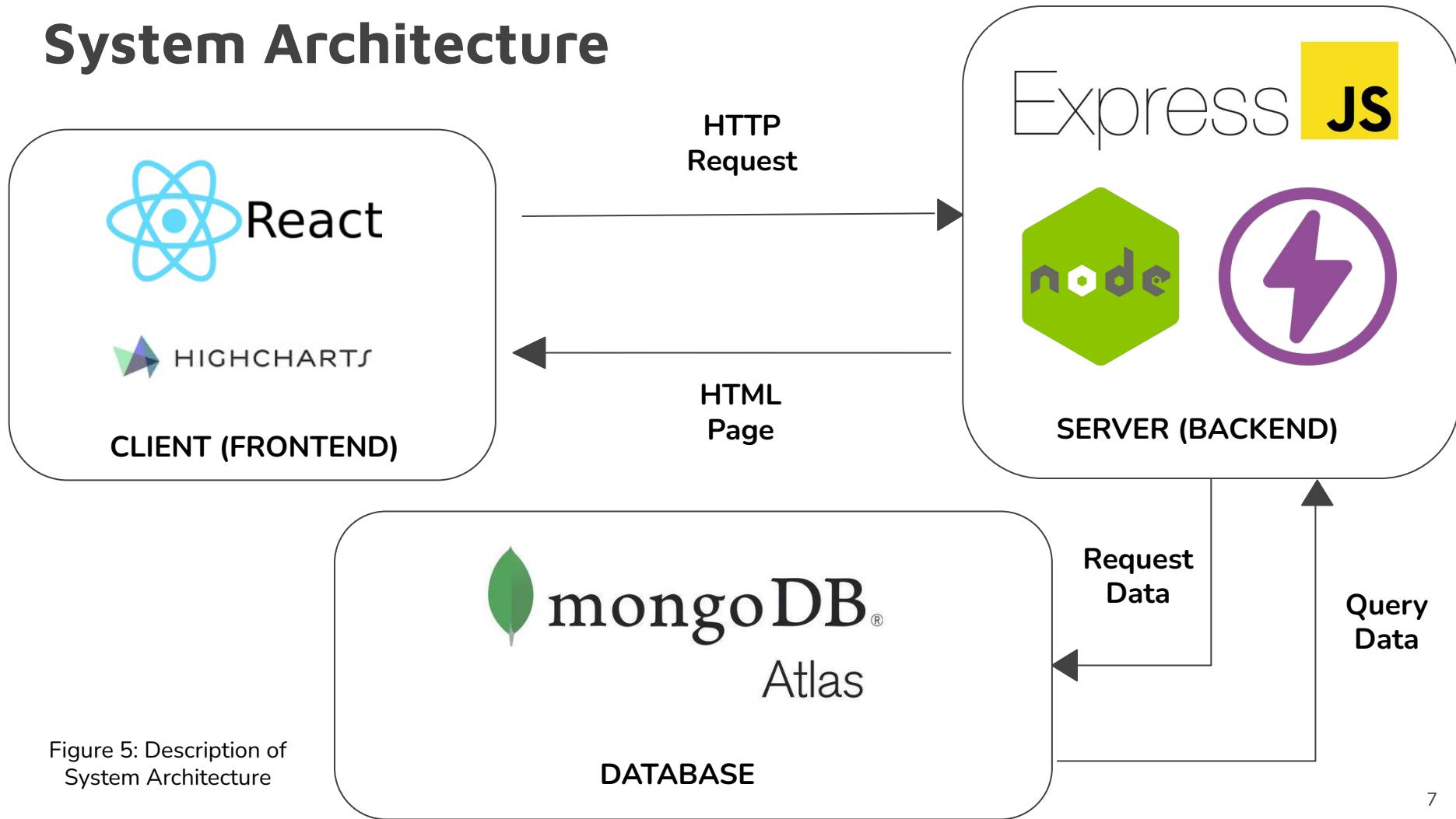


Figure 5: Description of System Architecture

# REST API

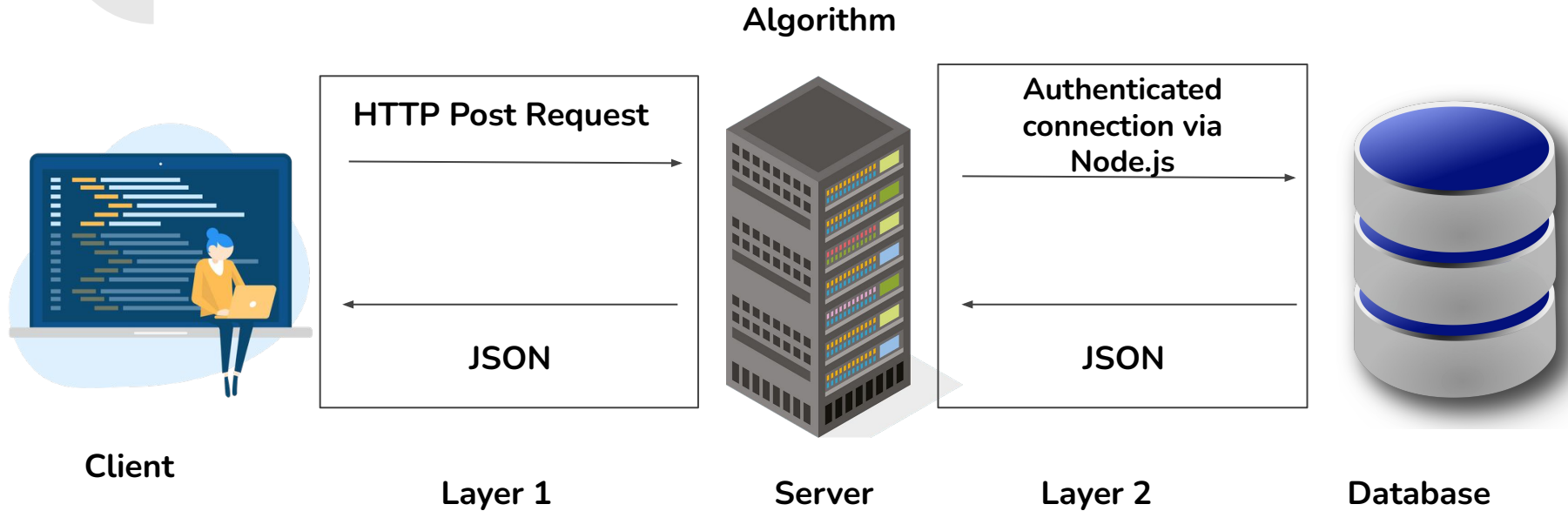


Figure 6 : A high level overview of the two-layered RESTful API





# Frontend Challenges

- Steep learning curve of React components essential to our proposed implementation
- Division of page-specific tasks among team members
- Identification of the complexity of each feature
- Deciding the API endpoints that we need from the back end

# Backend Challenges

- VSCode was not securely connecting to the database
  - Resolved by adding IP addresses inside the database
- Comments in the client's CSV are automatically converted into database fields
  - Do we need to account for comments in our implementation?

```
error: MongoNetworkError: connection <monitor> to 34.233.191.195:27017 closed
  at Connection.onClose (/Users/aaditbhatia/node_modules/mongodb/lib/cmap/connection.js:135:19)
  at TLSSocket.<anonymous> (/Users/aaditbhatia/node_modules/mongodb/lib/cmap/connection.js:62:46)
  at TLSSocket.emit (node:events:513:28)
  at node:net:313:12
  at TCP.done (node:_tls_wrap:587:7) {
  [Symbol(errorLabels)]: Set(1) { 'ResetPool' }
},
```

Figure 7: Connection - Authentication error

# Goals for next week



## Frontend

- Implement a basic framework of the User Interface
- Implement a mock up of the visualization graphs using some fake data

## Backend

- Implement a test query on the database to ensure smooth and streamlined access
- Initiate the construction of the REST API

## General

- Establish a contract between frontend and backend teams that defines terms of access to the data, data flow and transfer between the application components.



**Questions?**