Michigan Technological University

Soil Infiltrometer Documentation

By:
Bryan Wandrych

If there is any question, feel free to email me: bdwandry@mtu.edu

Abstract

We have designed this Application/Website to help out for Soil Infiltrometer data collections by automating many different processes. This application is designed to take a 2 or 3 person job and condense it to one. This application is meant to be an aid for the tester, meaning a user is not to put all focus on this website, but this should be an afterthought. A user should pay attention to the soil infiltrometer first and wait for corresponding timers/alarms to go off on their device to collect data. But, they should be reviewing the infiltrometer while collecting data.

This application is structured in where App.js contains all of the functionality (i.e. functions, routing tables, and more) to keep track of all of the backend processing. All other JavaScript files are just meant to render pages to the website. There is no other purpose designed for them (meaning rendering pages). A problem with that initial design is that now App.js is over 2000+ lines of code. Which can be a bit confusing for another user to go in and wrap their heads around what is going on. A lot of different functions are just placed wherever and can also be a bit of a headache. Some functions are designed to operate when there is a transition from one page to another, while there are others that are designed to run during a certain page.

This application has a lot of features that are designed to add robustness to the website so it will not break. For instance, many of the pages are designed to have input from a prior page to be able to render, if a user refreshes the page, it will tell the user that they will need to revert back to the main page instead of just breaking. We also have added things along the lines of not allowing pull down refresh, page zooming (on chrome for mobile devices), and a prompt asking the user if they want to manually refresh the page.

Note Throughout the code, you will see a different naming convention for certain functions depending on the protocol being selected. These protocol names have changed over time from their initial renditions. But, these naming conventions were never changed in code in terms of names of certain functions or javascript pages. Just some of the text that gets rendered on Screen**

Naming conventions:

- Standard = Time interval collection
- BAER = BAER
- Experimental = Volume interval collection

FLOW OF DATA

This page will show the overall data process of all different pages being rendered. Please Review the Next Page for the Flow Chart.

Other Content Pages

Note*** All other Content Pages can route to each other from eachother

Start

App.js

Depending on What Protocol of Data Collection a user chooses to take. There are different Routes

Note*** All Data Collection Protocol Pages have resets to them. So they all have options to route back to the MainPage

Learn.js

MainPage.js

Or

Or

Volume Data Collection

Time Data Collection

BAER Data Collection

DataGathering.js

DataGatheringExperimental.js

DataGatheringBAER.js

LearnInfil.js

Routes back to

DataCompleted.js

DataCompleted.js

DataCompletedBAER.js

LearnBAERProtocol.js

Gets Saved To

Gets Saved To

About.js

Browser - IndexDB: **Database**

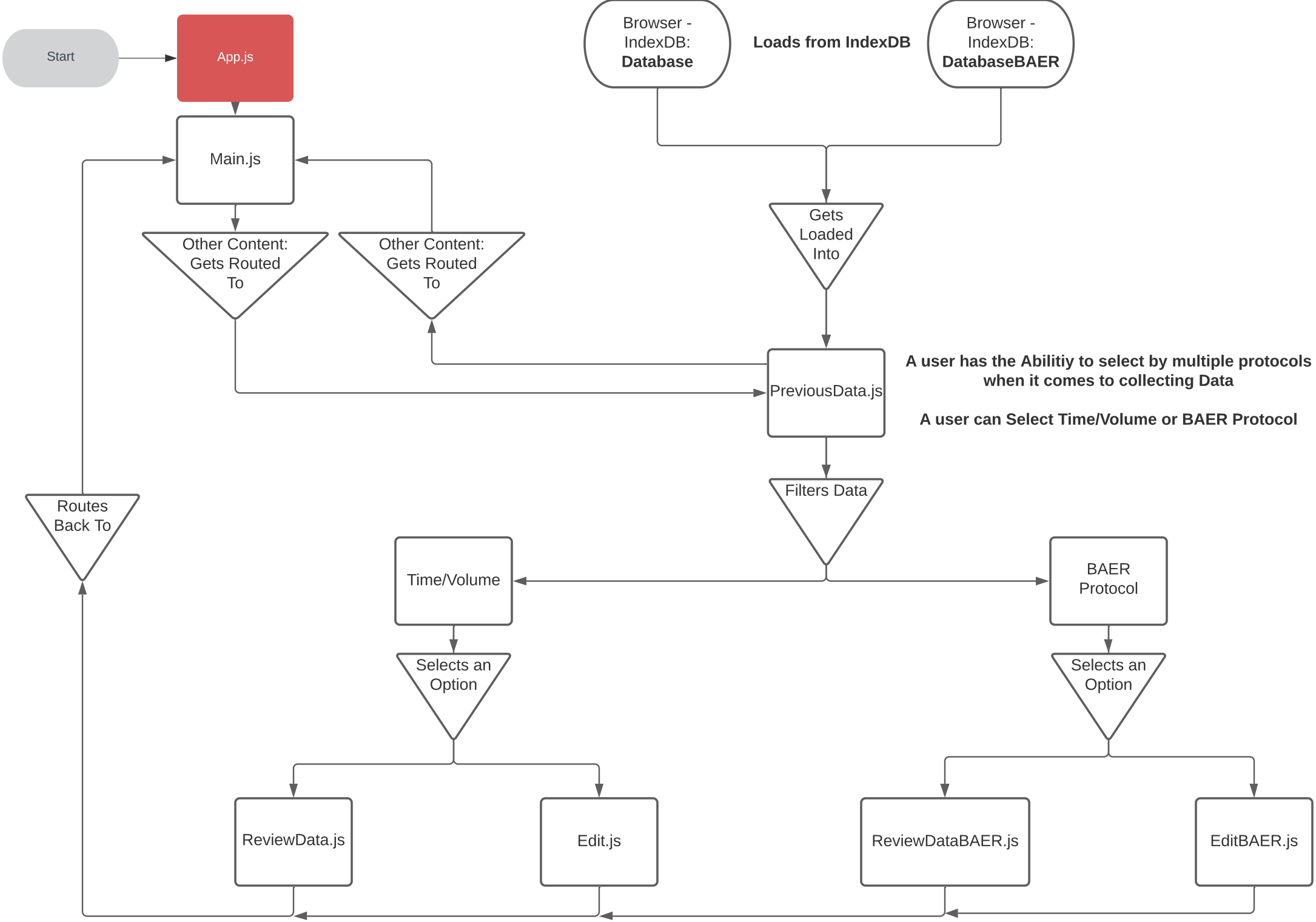
Browser - IndexDB: **DatabaseBAER**

downloadPage.js

PreviousData.js

FLOW OF DATA

This next page is designed to show you the flow of data in the Previous Data Segment (i.e. Reviewing old Test Cases). Please Review the Next Page for the Flow Chart.


Start
App.js
Main.js
Browser - IndexDB: Database
Loads from IndexDB
Browser - IndexDB: DatabaseBAER
Other Content: Gets Routed To
Other Content: Gets Routed To
Gets Loaded Into
PreviousData.js
A user has the Abilitiy to select by multiple protocols when it comes to collecting Data
A user can Select Time/Volume or BAER Protocol
Filters Data
Routes Back To
Time/Volume
BAER Protocol
Selects an Option
Selects an Option
ReviewData.js
Edit.js
ReviewDataBAER.js
EditBAER.js

Understand the State Variables

- **PageState**: "MainPage",
  - Not Currently Used anymore, it was used in an earlier iteration of our react code. It was used to switch the view of the website based on the string of the Pagestate

- **id**: 0
  - When collecting data, the ID will increment depending on the number of data points collected

- **time**: 0
  - This will keep track of the total time during Collecting Data for Time and Volume

- **timeInterval**: 30,
  - This will Set the Time intervals a user wants when collecting data
  - This is only used in Volume and BAER Protocol

- **initialVolume**: 95
  - This is the initial volume for when a user starts collecting data

- **audio**: ""
  - This is for the initialization of the audio that will play for the alarm.
  - Later on in the code, it will be initialized to the loaded audio

- **volume**: "",
  - This field is for the Volume Metric for data collection, it will get stored here

- **Data**: [],
  - This is the array that all tables from Data Collecting to Data completed will pull. It will dynamically be added too when a user adds more data points

- **title**: "",
  - This is for setting the title when a person is going to save a given test

- **gpsCoordinates**: "",
  - This will store the GPS coordinates in a Longitude + ", " + Latitude format

- **file**: “...”
    - File, below, is the picture stored. You can see that it is set to a default image. It will change depending if a user changes the images.

- **DatabaseData**: [],
    - This is the state variable that will store the data that gets pulled from IndexDB. This will store Time/Volume protocol. I made it kind of stupid, but the global users variable gets saved first, and then that variable will be stored to this one. Should've change it around to get rid of the global variable

- **DatabaseDataBAER**: [],
    - This is the state variable that will store the data that gets pulled from IndexDB. This will store the BAER protocol. I made it kind of stupid, but the global baerDB variable gets saved first, and then that variable will be stored to this one. Should've change it around to get rid of the global variable

- **infiltrometerData**: [{Radius: 0, Alpha: 0, NperH0: 0, Suction: 0}],
    - Keeps track of the Soil Infiltrometer Settings
    - Will prepare to save data into the IndexDB

- **newTitle**: "",
    - This is for editing old data, its used to compare a new title with an old title or if the new title is left blank

- **sqrtTime**: 0,
    - This is for Time/Volume data collection, it will be used for calculating the square root of time.

- **Radius**: "",
    - This will display on the main page, it will change when a user changes the soil infiltrometer radius. Its used for later calculations

- **Alpha**: "",
    - This will display on the main page, it will change when a user changes the soil infiltrometer Alpha. Its used for later calculations

- **NperH0**: "",
  - This will display on the main page, it will change when a user changes the soil infiltrometer N/h0. Its used for later calculations

- **Suction**: "",
  - This will display on the main page, it will change when a user changes the soil infiltrometer Suction. Its used for later calculations

- **MLPerMin**: "",
  - This is used for the BAER Protocol, it will be changed for BAER Calculations

- **infilt**: 0,
  - This is for the all Data collection Protocols, it will be used for calculations for soil infiltration

- **longitude**: "",
  - For GPS Coordinates

- **latitude**: "",
  - For GPS Coordinates

- **A**:0,
  - This is a calculated constant for the Time/Volume calculation protocol

- **C1**:0
  - This is a calculated constant for the Time/Volume calculation protocol

- **K**:0
  - This is a calculated constant for the Time/Volume calculation protocol

- **InfiltrometerCalculations**: []
  - This will be used to store the soil infiltrometer calculated constants in the IndexDB Array

- **ReviewOldDataArray**: [{Title: 0, Picture: 0}]

    - This is simplified view used when a person reviews the previous data page - time/volume protocol

- **ReviewOldDataArrayBAER**: [{Title: 0, Picture: 0}]
    - This is simplified view used when a person reviews the previous data page - BAER protocol

- **indexNum**: 0
    - This is used when a person is going to review, edit, delete an old test case

- **play**: false
    - For the audio play/disable purposes

- **isOpen**: false,
    - No notes on this

- **visible**: false,
    - No notes on this

- **position**: 'left',
    - No notes on this

- **calculatedTime**: 0
    - This is for having a calculated time logged when a user enters in volumetric data when collecting data for Time

- **notificationFlag**: true
    - This is for a notification to be pushed when a user is a computer/tablet

- **audioFlag**: true,
    - No notes on this

- **resetValue**: 0,
    - No notes on this

- **CSVArray**: [],
  - This is for CSV to be exported when a person is done collecting data or reviewing old data

- **DataCollectingProtocol**: "StandardProtocol"
  - This is for specific rendering of different fields on the main page or previous editing page. Different options will render depending on the text that is saved here

- **NumberOfRuns**: "3"
  - This is for the replicates of the BAER Protocol

- **runIndex**: 0,
  - No notes on this

- **AverageMLPerMin**: 0
  - This will be used in averaging the ml/min calculated in the BAER Protocol

- **LoadPreviousData: 0,**
  - No notes on this

- **SwitchDataGathering: 0,**
  - No notes on this

- **SwitchDataCompleted: 0,**
  - No notes on this

- **SwitchReviewData: 0,**
  - No notes on this

- **SwitchEditData:0,**
  - No notes on this

- **VolumeInterval: "",**
    - No notes on this