Definition of Done and Style Guide,

Training Control Center

# Definition of Done:

* A feature must have all functions outlined by its user story specification
* The feature must work for all test cases including edge cases
* The code for the feature must conform to the style specification
* The feature must be merged into the main branch without any conflicts
* The feature must be integrated into the app without consequences to other features and such that it is of the same visual style as the other features

# Style Guide

**Repository/Project Structure**

Frontend

* src/Components
  + JSX files that make up part of a page, such as the manual activity entry, appBar, goal setting, etc.
* src/Pages
  + JSX file that is displayed to a user
* images/
  + Example images outputted from backend to be display to user (mainly used for graphing functionality)

Backend

* db/
  + Files to handle database fetching directly
* src/
  + General backend functionality such as login, authentication, and graphing
* api/
  + Guideline for database api calls

**Coding Guide (general JS formatting)**

General Guidelines

* Avoid magic literals
* Avoid code duplication
* Avoid negative conditionals
* Use explanatory variables/functions (simple and purposeful)
* Use constants instead of magic numbers
* Functions should do one thing, name should be self explanatory
* Define variables and functions close to where they are used
* Define private functions just below its usage

Programming Guidelines

* Spacing
  + Indentation: 2 spaces
  + 1 space in between all operators i.e (x + 1), not (x+1)
  + Limit horizontal spacing
  + 1 vertical space in between functions
  + 1 vertical space in between imports and functions
  + 1 space after a (:)
  + 1 space after a (,) or (;), not before
  + No space at start or end of functions
* Comments/JSDoc
  + /\* \*/ for multi line, using \* for lines in between
  + JSDoc string should be used for all classes, fields, and methods

Example:

/\*\*

\* Multiple lines of JSDoc text are written here,

\* wrapped normally.

\* @param {number} arg A number to do something to.

\*/

function doSomething(arg) { … }

* Use markdown format for JSDoc strings
* Use @param for describing parameters

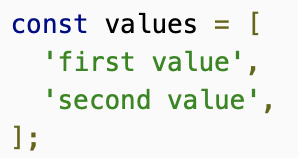
Example: @param (type) variableName description

* Use @return for describing function return value

Example: @return (type) description

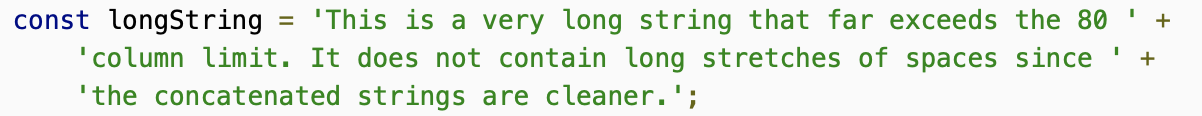
* Language
  + Use const (default) or let, do not use var
  + Use trailing commas

Example:



* + Don’t use Array(x1), use [x1] instead
  + Don’t use Object(x1), use {x1} instead
  + Subclasses must call super() before accessing or setting fields
  + All imports should be defined at the top of a file
  + Nested functions should be defined by a local const
  + Use single quote for string literals rather than double quote (‘template’)
  + Template literals using (`${template}`)
  + No line continuations, line length should be 80

Example:



* + No empty catch blocks
* File Naming
  + File encoding in UTF-8
  + File names should be camelCase, so first character lower case, following words with firstLetterUppercase
  + Files should end in .JSX, unless otherwise specified
* Naming
  + Class names should be UpperCamelCase
  + Methods names should be lowerCamelCase
  + Variable names should be in camelCase
  + Constants should be ALL\_CAPITALIZED