Team HighTech GeyserApp (I.P.A)

Coding Standards

The following document will be discussing the coding standards for our software code specifically.

For the programming languages used in the software. Please find the corresponding coding standards below.

NodeJS -https://docs.npmjs.com/misc/coding-style.html

HTML- https://docs.ckan.org/en/2.8/contributing/html.html

Ionic framework- https://ionicframework.com/docs/theming/basics

Reliability

For the development of the software rigorous tests were conducted using unit and integration tests to insure that the final algorithms and functions used would work with little to no error at all and that all expected outcomes were accounted for. Where security and validation were required checks to see if there was no exploits was done to ensure the software will be secure and reliable.

Quality and validation

As we are still learning a lot about the programming languages we are using for the project. The quality of the software is ensure by using code validation tools such as TsLint which is preinstalling in the ionic framework. This allows us to ensure that everything done is as good as it can get. While there may be other alternative ways to implement functions and algorithms which may improve on the quality, what we have done and the way we did will still produce the same out come at the end. Future developments of the software may improve the current quality of it however. The same style of coding was used to ensure that everyone could understand how any part of the software would work in the need of changing anything.

Compliance of code/review

In order to ensure coding compliance of the software. The 5 steps of compliance were used where when a piece of code was submitted for review it was…

1. Checked to see if any potential coding compliance problems could be **detected** such as a miss match of information or a security vulnerability or violates how code can be used according to its own standards.
2. **Corrections** were made if any existed.
3. The team was notified about the issue and on how to avoid/**prevent** it.
4. The same piece of code was run again to insure **verification** and that no other errors or issues arose.
5. **Comparisons** were made on how we have used the functions and procedures compared to how the general use to insure it is inline with it.

For the reviewing of the code. GitHub was used as it provides a easy to understand and easy to interact with UI that highlights all potentials errors or mistakes in red and allows for easy commenting on the line where you see an issue.

Code comments and variable names

Comments are used through out the source code in order to explain how a piece of code works if it cannot be understood by just looking it. Comments are used also to explain the types of data expected to be stored in variables since our variables can be set to any data type. Commenting by using detailed variable names have also been used to make the understand of the flow of code better in the case where a function can be followed by just understanding the variable names. Functions with detailed names are also used for easier understanding of what the function will do by just looking at the function name.