Coding Standards Document

Team Name: devs101
Client Name: Mark-Anthony Fouché

Team Members

- Brendan Bath
- Dylan Draper
- Charles Claassen
- Dail Nathan Jonker
- Njaravani Christopher

1 UML DIAGRAM

2 Coding Conventions

2.1 Naming Conventions

2.1.1 Folder Names

- All folder names does start with a small letter and the name is descriptive enough to provide infromation on what exaclty they contain eg components, contains all components files

2.1.2 Class Names

- All class names had to begin with capital letters and new name within the class name had to begin with capital letters aswell.

2.1.3 Variables and Attributes

- variableName - variables and attributes had to start with a lowercase letter and each new word within the same name had to begin with a Capital letter, meaning to say they have to be exactly the same as the methods.

2.1.4 Accessors

- *eg getAttribute()* - had to follow the same rules as the methods and had to always be named after the variable which they intent to access.

2.1.5 Mutators

- eg setVariable(.....) - had to follow the same naming conventions as methods i.e they had to start with a lowercase and any new name within the name should begin with a capital letter.

2.1.6 Packages

- eg display.package - names of the packages had to always be in lower-case letters and no upper-case.

2.1.7 File names

- Vuetify.js - file names had to be descriptive enough to tell what they contain inside and name had to always begin with a small letter except only App.vue which loads initially which is like the main.eg rules.js shows that it contains rules, store.js thats the store file which stores files

2.1.8 Exceptions

- Exceptions had to be applied and used wherever necessary and these should the following had to always be checked.
- Exception classes had to be avoided if neccessary and more meaningful classes with meaningful names had to be used.No more functionality had to be in these classes except only catch statements so that they clearly show what exactly have happened without any confusion.
- Appropriate catch statements had to be used which had to allow the program to continue if need be rather than crashing.

2.1.9 Methods or Functions

- Methods had to be given a descriptive name of what exactly they do and the name had to begin with a lower-case letter and any new name within the name had to begin with a Capital letter for easier readability.

2.2 Layout Rules

2.2.1 Indentation and Line spaces

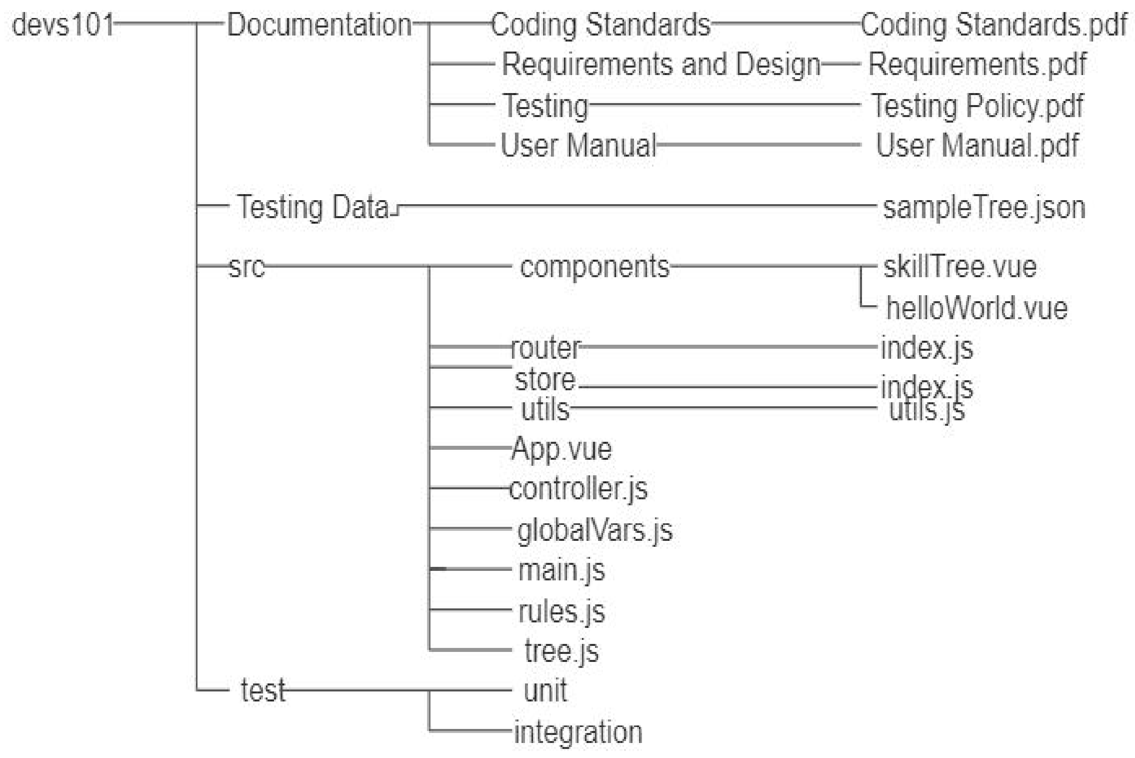
- Lines had to be kept of very small length so as to ensure readability.
- Once a line had reached a certain length(mostly 8 words), it had to be broken and indented further than the previous line for readability.
- Indentation had 4 spaces for each level, blank lines to seperate methods and also to emphasize or make it clearer where each method ends.
- attributes had to be be defined mostly after the methods in a class.
- public final variables being the exception which had to be defined first.

2.3 Commenting Practises

- It was a mandatory to comments so that the code becomes very easy to read and understand
- Comments were added in blocks and not in lines as it might cause some confusion and the code wont be clean hence no readability.
- For line comments we used *// and multiline use /* ..*/*

  for example int index = 1; // index of starting point.

3 File Structure



4 Code Review Process

4.1 Pair Programming

- Any 2 group volunteers will work in groups of two, code together, one coding while the other is reviewing the other's code.That follow up check made it sure that all coding standards rules are followed eg Variable naming conventions, bracketing and indentation

4.2 Peer reviews

- Any group member who get the chance can get to look at someone's code once the member is done.After each assignment of work, issues are created and before an issue is closed, that person's code shou