**File Management Plan**

Refer to textbook pages 24-25, 169-174, 282-83 to answer the following.

**Outline the naming conventions you have used for files and folders in your project.**

My variables and code will follow the Camel Case naming convention, as is the convention with development in Dart. My files/folders within the software repository will follow the Snake Case convention, while my files and folders in my project not pertaining to code (planning, reflections, word documents, etc) won’t follow any specific convention, rather just using whole words with spaces.

**Outline the structure of files and folders in your project.**

Files within the project itself (not code) are organised by the outcome, with individual folders for individual criteria. Within each of these folders will be a collection of markdown files, such as READMEs that provide extra information about my work along with the multimedia in the same directory. In the global scope of the project (the “SAT” folder) lies a README that links to a directory of all files and folders by criteria/outcome.

Files within the software (code) will be organised by the Flutter file structure. This will include an “assets” folder for static assets, a “cloud\_functions” folder for cloud functions, a “lib” folder for a library of code, “screens” within this folder for individual screens, etc. For more information on how Flutter projects are structured, you can view the folder structure [here](https://github.com/GeekyAnts/flutter-folder-structure).

**Outline the backup procedures you have implemented. How will you back up? How often? Which files and folders? Storage of backup? Location of backup?**

Backups will take place on OneDrive, with the primary location for files being GitHub. GitHub also takes backups of all files every single commit, which means almost every change I make will be backed up in two locations, both GitHub and OneDrive. As such, if anything goes wrong, I can just roll back the git repo and have the software project from before.

This means that the backup process is automated, and all backups will be stored via git.