Version Control

The tools we are using for this project are: GitLab, IntelliJ, Papyrus.

Processes

GitLab:

- Provides online storage and branching to allow for independent work on different areas of the project, which is then merged with the master branch upon completion,stopping edits in one area interfering with edits in another, eg. changing documentation while another team member is editing code.
- Provides revert functions that allow errors to be removed that cannot be solved easily by editing the code, or removing the implementation of features that do not work as expected.
- The use of GitLab provides a convenient online editor for files so we can edit our documents online, and push directly, which can often be much more convenient than having to edit locally and push all the time.
- Merge conflicts in software should be rectified by a meeting between the PM and the programmer who "owns" the merging branch. They will work together to remove the conflict and ensure the program still works as expected, following the QA Plan.

IntelliJ:

- IntelliJ is Git compatible and works with a wide variety of file types that can be tracked for changes.
- This gives us freedom and trust in how we design the documents as IntelliJ provides a preview of the finished document as it is made.
- IntelliJ's Git functionality allows for quick retrieval of old versions of files should they be needed for reference or replacement.

Papyrus:

- Papyrus is also compatible with GitLab which ensures compatibility across platforms and ease of issue management.
- This was a useful feature as part of our last assignment was to create a UML diagram which we could easily link to GitLab without any issues, this provided us with a strong foundation to improve and edit our work.