

Contribution Guide and Code of Conduct

The following contribution guide has been created for the **Focus** messaging application. It highlights the responsibilities of the members whilst noting the contributions. Alongside this, a code of conduct and project standards have also been created.

Responsibilities

There are a total of 4 members working on the development of Focus, each member holds different responsibilities that are vital to the success of the project. These responsibilities naturally affect the contributions to the overall project as well as the repository made. Below we can see the different responsibilities and their contributions:

- **Project manager** is responsible for organizing and scheduling the work, enforcing regular git repository pushes as well as assuring effective communication takes place. Also responsible for taking action when there is a failure to communicate and work appropriately.
- **Software architect** is responsible for creation of UML diagrams and aiding in the development of the application, assuring the diagrams translate into implementation.
- **Software developer** will be the main developers of the actual application, they hold the right to control the git repository in terms of adding, removing and editing any contributions made that don't adhere to the coding standards created.
- **Software tester** is responsible for effectively communicating concerns to the software developer that could affect a successful reception of the application. Holds the ability of editing and adding additional comments to the git repository.

Git Repository Contributions and Pull Requests

There are a few steps that must be considered when making pull requests and commits to the git repository. This is to avoid any confusion and disorganization through out the development. Each member of the project should be aware of the requirements for making git repository contributions.

- Each pull request should include thorough and relevant comments based on the work added and a review of any changes.
- Software releases should correspond with schedule made.
- New software versions should be named appropriately by increasing the version numbers each update – this should also resemble with software releases. For example, working on the first software release with multiple different versions would be: "Focus 1.4" implying that there have been 4 different updates on the first software release.

Standards

In order to create a helpful development environment there are a few standards that should be enforced. Below is a list of this project's standards as well as circumstances which are not tolerated.

- Communication should be suitable and appropriate.
- Contemplation towards everyone's opinions and ideas.
- The project's best interest should not be clouded by personal and group issues.

- Consideration for personal issues such as illness or extenuating circumstances.

The project will not tolerate:

- Negative and inappropriate communication that affects development.
- Sharing conversations whether private or group to anyone other than the group itself.
- Sharing any of the project development before official release.