

Group Working Guide

Using GitHub

In order to keep everyone's work up to date, facilitate code collaboration, and avoid code clashes, please review the following steps each time before working on the code until you're confident.

1. Open your local clone of the GitHub repository
2. In the project folder where the *.git* can be found, right click and select 'Open in Terminal'
3. Enter **git branch** in the terminal to ensure you're working on the correct branch, then **git status**
4. If there are any untracked or modified files (your local changes), enter:
 - **git add .**
 - **git commit-m "Required to write commit message here"**
5. To check for and pull any changes from GitHub, enter:
 - **git pull**
6. Finally, enter **git push** if you did step 4

Remember, it's best practice to commit any changes you make locally as soon as you can to ensure everyone is constantly up to date and to avoid conflicts. Always pull to update your local files in case of any changes on GitHub.

Additionally, keep the team further updated on task progression by continuously using the Kanban board and communicating in the WhatsApp group.

Code Consistency

- To ensure everyone's code is formatted the same way, Code Style rules have been configured on IntelliJ and copied to the project repository.
- The project root folder must use **kebab-case**, Java package names must be **lowercase** and Java class files **PascalCase**. Other non-code files and folders can be **Upper Case** for neatness and readability.
- Use blank lines to separate high-level blocks of code e.g. functions, classes, objects.
- For Java:
 - Variables, objects and functions should be in **camelCase**
 - Classes and interfaces should be in **PascalCase**

- For JavaScript:
 - Variables, objects and functions should be in **camelCase**
 - Classes should be in **PascalCase**
- For HTML/CSS:
 - Class names and IDs in **kebab-case**

Ensure to also comment your code where necessary, using the correct conventions for each language.