

Group Contract

1. the working policy for the group.

Version control: Github

IDE: VSCode (Rider if you dare)

Report: LaTeX

Diagrams: MIRO

Communication: Discord and Messenger for quick messages

2. Normal meetings times

Tuesdays: 12-16

Thursdays: 14-16

We are open to the possibility of adding additional meetings when needed.

3. Where you will work (normally).

We meet primarily on ITU, and usually work in groups of 2 and 3.

We can work from home if it is needed, but generally we work together at ITU.

4. How well as a group do you plan to do in this course.

We intend to learn a lot from this project.

We mostly wanted an option between 'have fun' and 'high grade', meaning we all want to do our best whilst having a good time with each other.

5. Code standards

Always make a *summary*, *param* and *return* description for every function before pushing to develop (the shortcut is *///*). Summary should contain a note if some of the code was stolen from the internet or if AI was used to write the code.

never use the “var” keyword. Try to use the explicit type of the variable. This makes the code easier to read, and helps avoid confusion.

don't commit irrelevant print statements.

Remember to add co authors, if you have help.

follow KISS as much as possible.

6. Github standards

Remember to do code review when reviewing pull requests.

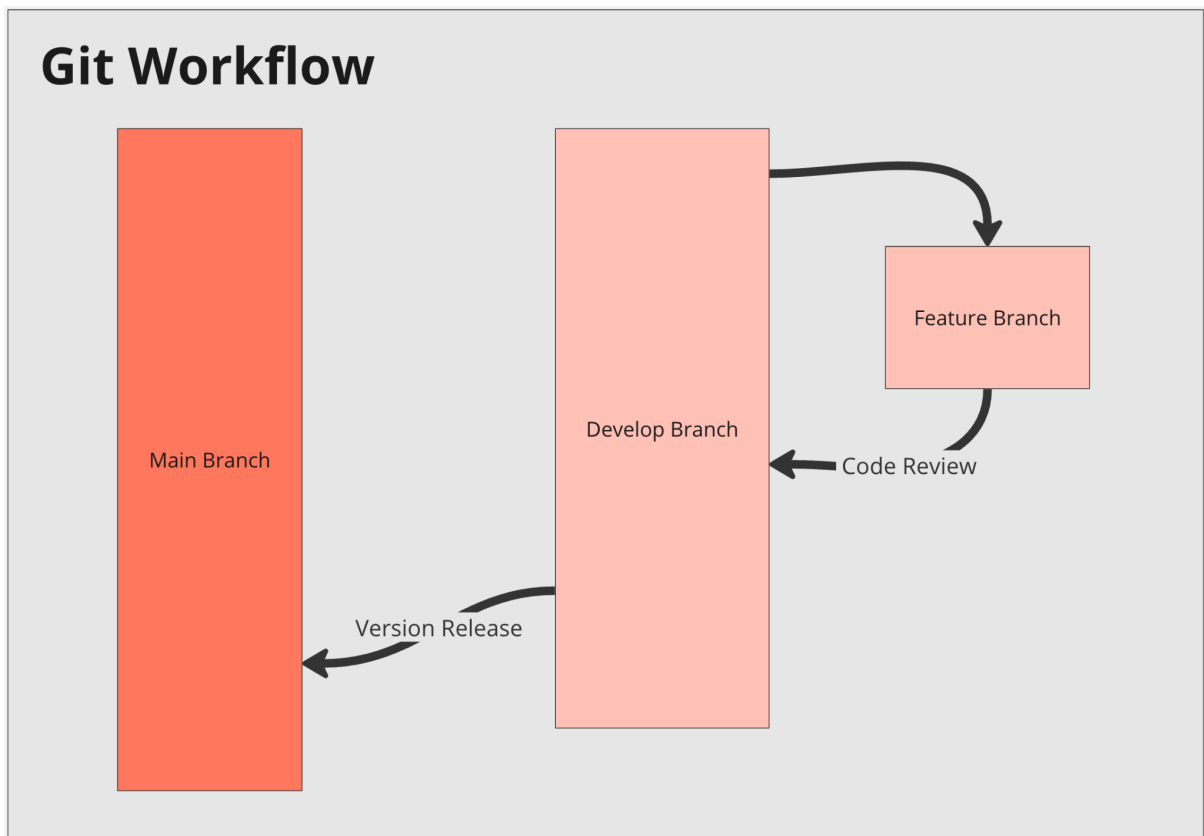
Delete branches after merging.

merge develop into main at the latest before wednesday afternoon.

Branch naming:

Feature branch: “feature/<name of feature>”

For bugfixes: bugfix/<name of bug or error>”



Code review checklist

Check for:

1. irrelevant print statements.
2. Methods without documentation.
3. the use of "var" instead of explicit typing.
4. Methods that do multiple things (a method should do one thing), or functions with side effects (if the function does something without the programmer using the functions knows about).
5. unreadable / not understandable code.
6. tests that failed in the Workflow.

remember to be nice. Nobody likes a know-it-all.

Code review best practise

<https://kellysutton.com/2018/10/08/8-tips-for-great-code-reviews.html>

1. We are human so remember to be constructive and nice.
2. everyone does reviews not just the same person over and over
3. make your review readable
4. leave at least one positive remark
5. provide alternatives to the problems you find if it is not a basic problem like the use of "var" instead of explicit typing.
6. latency is key
7. keep reviews as compact as possible

user authentication method

For this project we have chosen the b2c method as our authentication method of choice. We chose b2c because it presented itself as simpler and more beginner friendly than the other authentication methods.