Code of Conduct

Group 18b

November 2022

1 Meetings

- Agenda for the meeting is to be submitted by 18:00 on the day before the upcoming meeting; it should describe a clear outline of the discussed items, which on their own outline the goal of the meeting.
- Secretary of the team takes notes and publishes those in the repository
- Lab meetings have mandatory presence
- For other meetings, absence ought to be discussed beforehand

2 Product

- We are to develop a prototype of the application in the provided time window
- User story assignment using the MOSCOW principle; Must-have and a part of should-have features will be implemented. Could-have features will not be addressed.
- Development and Main branches - (if issues relate to documentation, they ought to be merged directly to main rather than dev if separate dev and main branches are agreed upon)
- Consistent code style, possibly enforced via Checkstyle, adjusted during the coding process
- Naming convention for issue branches - issue number followed by a short description of the issue in kebab case with lowercase letters
- Don't rename other people's variables and classes without consent
- Group Grade Aim: 9 or higher

3 Planning

- Recurring meetings on Mondays from 11:00 until 12:45
- Meetings with our TA are on Thursdays unless specified otherwise
- Communication - Usually conducted via Discord, Whatsapp for urgent matters, Mattermost for communication with our TA
- Chairmen and Notetakers are rotated on a weekl basis - previous Notetaker becomes new Chairman and new Notetaker either volunteers or is chosen by chairman

4 (Team)work

- The CoC can be updated if the entire team agrees with the amendment
- Time tracking in Gitlab is encouraged
- Templates for agendas for meetings defined in first meeting
- When the agendas are published on GitLab, initially they are just named “Agenda$_D D_M M_Y YYY$”$andislaterappendedwiththenotesto\backslash Agenda_N otes_D D_M M_Y YYY$”

5 Code Contribution

- Each merge request is approved by 2 others (check if the code is meaningful etc.)
- Every merge request should be reviewed by at least two people at latest 24 hours after the creation of the request.
- Every merge request should be reviewed by at least two people at latest 24 hours after the creation of the request.
- Test coverage for each class above 60 %$Everyteammembershouldfinishuptheirweekly Merge Request$ 59$andtheactual Merge Dayis Mondaytoresolvecodeconflicts, refactorings, etc.$
- Close to merge day, finishing existing MR’s should be prioritized over starting new features.
- Reassignment of issues is allowed in this case
- Constant meetings and project reviews to find out what needs improvement
- Accordingly, whenever a merge request is made, the one making it should inform the team by channels outside of Gitlab, such as Discord or Mattermost
- If the merge request is not approved, the one blocking it should explain it

- Merge request should not be approved without any feedback. In the case when there are no complaints or questions about a person's code, the reviewer should specify that in their comment
- The one making the merge request should put a short description of all the changes in their merge request. If there are stuff still to be added at a later date, the reviewer should describe what with "TO-DO" with a time limit they set for doing so with another merge request.
- Merge request reviews should be done in the style of conventional comments: conventionalcomments.org/
- When addressing TO-DO parts of a previous merge request in another, they should refer to that merge request explicitly in their comments
- If there is a known bug in the merge request, which is to be fixed, it should be denoted with the "BUG" label in a new issue and fixed as fast as possible. It should additionally be avoided to submit merge request with bugs unless the one submitting it feels it is absolutely necessary
- All commits should be atomic
- Avoid commits like "fixed pipeline" and "bugfix" by squashing it into the appropriate commit.
- You should do this before pushing:
    - git rebase -i HEAD [number of commits to go back]
    - If already pushed, git push –force for new history
- For every commit, the pipeline should pass
- One feature per commit
- The issues that reviewers mention in their comments should be at least addressed in 12 hours from the time of the comment (excluding the hours between 22:00 and 09:00 the next day to account for sleep)
- Never modify others' branches without consent

6 Conflicts

- Reaching consensus first, otherwise - majority vote, the chair is the tiebreaker
- In case of a group member not adhering by the CoC first try to address the issue within our group, but if that does not yield results, address higher authority without delays