Code of Conduct – Group 49

Ipshit Raychaudhuri, Daniel Dumitru, Yana Mihaylova, Jake Nijssen,
Maria Zmpainou, Kasparas Savickis

1 SHARED TEAM VALUES

- As a team we have decided on a set of shared principles and values which we will abide by over the course of the project.
- We will be integrous by showing up for all planned meetings on time to the best of our abilities as well as by meeting the basic requirements, as outlined in the study guide, of writing at least a hundred lines of java source code, making at least three commits and approving at least one merge request of another member with a constructive comment each week.
- We will be respectful to each other's boundaries and as such will not press any member to work more than they are required to.
- We will be honest about our own strengths and limitations to each other during discussions so tasks can be divided accordingly based on each member's skill sets. Additionally, no member of the team is allowed to take credit for the contribution of another person.
- Finally, we expect that every team member will be accountable for their own actions and the consequences of said action. Each member should be responsible for all their contributions to the project and should not try to place the blame on others if something goes awry due to said contributions.

2 ASSIGNMENT DESCRIPTION

- What we aim to achieve working on this project is above all to get insight into how to work effectively in a group. It will teach us valuable skills such as giving honest and constructive feedback, chairing and leading meetings, distributing the roles in the team according to the members' strengths and weaknesses, coordinating our actions in a way that benefits the whole team and doesn't hinder anybody's personal work. Therefore, learning how to use Git is essential and something that will help us very much when working on any kind of project later on. It allows for the use of separate branches and merge requests that have to be approved by the other teammates, as well as evaluating and providing feedback on code contributions from other team members.
- We expect to create a full working application with a graphical client, that exchanges data through REST and web sockets, manages the lifecycle of an application through dependency injection and implements reactive REST clients with long-polling. It will encompass everything we've learned throughout the year so far in courses like OOP, Web & Database and IDM and allow us to put the knowledge to practise. This will involve utilising frameworks like Spring Boot for backend development and Java FX for building the graphical client interfaces.

3 TARGET OR AMBITION LEVEL

- Teamwork: Remember that everyone is working in a team. Consider their goals as well and help them achieve it.
- Grade 8.0: Everyone wants to pass the course, so getting a passing grade is bare minimum. As a group we have decided our target to be a grade of 8.0 and everyone should do their best to achieve this or an even higher grade.
- Backlog goals: Implement all the base features detailed in product backlog and as many additional features as possible in the timeframe.
- Adjustments to targets: Reflect on the progress weekly with the rest of the group. If we are unable to implement all target features before the deadline, consider which features can be dropped and which should remain.

4 PLANNING

- As a team, we believe in staying organized and efficient. To ensure smooth progress, we do regular check-ins every Thursday, one day before the deadline of each assignment, and we have meetings every Tuesday to make sure everyone is on track with the project. On those meetings we also focus on exchanging ideas and address any challenges we may face. We even extend them further to divide the tasks of the current week to the members of our team. When it comes to doing so, we try to focus on each person's strengths so we assign work based on people's skills. For this, we use GitLab's milestones and issues to have a clear view of each person's duties and progress. We value good time management, so we encourage everyone to plan their piece of work given by the team accordingly.
- To avoid confusions, the chair of each week takes charge of submitting the weekly assignment. Last but not least, we discourage the habit of leaving submissions and commits until last minute. This ensures higher quality of work and allows adequate time for review and improvement before deadlines. Overall, our approach emphasizes communication, collaboration, and efficiency to achieve our goals effectively.

5 BEHAVIOR

- We will treat each member of the team with respect and will not make any derogatory or offensive remarks against them.
- Whenever we encounter conflicts we would like to have a clear set of rules on how to handle them. For any disagreement both parties must write a formal explanation of their

opinion on the subsequent matter and present it to the rest of the team. After which a simple majority vote will be held to decide which direction is chosen.
- Whenever somebody is late they do not need to explain the reason if it's less than 5 minutes. Otherwise let your teammates know beforehand either in the Whatsapp group chat, Mattermost or by calling another teammate. If this continues as a pattern, it would be reported to the TA.
- Everyone should make sure to communicate with the others in the group chat to remain updated with all the ongoing changes and not miss any deadlines, meetings or merge requests that need to be accepted.
- If someone feels they cannot manage the task they were assigned or took on on their own, they should consult the other teammates for help or switch the task with somebody else.
- Humor is tolerated and even welcome as long as it isn't anything offensive and doesn't distract us from our work.

6 COMMUNICATION

- Respect: We will treat every group member with respect, courtesy, and kindness.
- Pay Attention: Listen to what others have to say during meetings and do not interrupt them. We expect that every member should participate in discussions and actively provide their own perspectives on the topic on hand.
- Feedback: Provide feedback during discussions, however, ensure that the feedback is constructive and not an attempt to put another member down. It is also important be open to receive feedback and suggestions yourself.
- Communication Platforms: Use WhatsApp to contact others and discuss parts of the project. Mattermost should be used if you need to communicate with the TA.
- Deadlines: Respect the deadlines set for the weekly contributions (Sunday) required on GitLab (i.e. one hundred plus lines of code, three plus commits and the creation, approval, and review of one plus merge request). Respond promptly to messages, on Whatsapp or Mattermost, to avoid delays in the project. In the case of inability to meet deadlines, inform other team members as soon as possible.
- Clarity: When communicating ideas or instructions, aim for clarity and conciseness. Use clear and understandable language and avoid ambiguity or unnecessary complexity. If something is unclear, do not hesitate to ask to avoid conflicts in the future.
- Privacy: Respect the privacy of individuals within the group. Do not share their personal information to any other people outside of the group.

7 COMMITMENT

Effective quality control is essential for a project to succeed, so to maintain said quality we have set some ground rules on how to handle issues and review them.

Issues:

Whenever an issue is created we expect said issue to be clearly defined using the following conventions:

- Always correctly label the issue in the most descriptive way possible, i.e. label them with Server if the issue has to do with the backend and Bug if it is to fix a bug.
- Concretely mention via a checklist or other means what needs to be done to resolve the issue.
- When giving feedback on an issue make sure it contributes to the resolving of the issue, i.e. do not make any meaningless comments.
- Issues should always be linked to an active development branch/merge request, this can be achieved by adding a link in the merge request to said issue and vice versa.
- Deadlines on issues must be taken seriously, so keep to them as much as possible and only deviate from the deadlines if there is a sufficient reason for it.

Commits:

The backbone of progressing through the project is individual commits. To make sure that commit messages do not become a meaningless mess, there are a couple of simple rules that must be adhered to:

- Keep commits clear and concise, make them clearly describe what has been changed in the commit, even if that requires making multiple commits at the same time with different messages. Examples that should be avoided include something along the lines of: "bug fixes".
- All commits must never break the ability of the application to run, as in that the application cannot start after the changes made.
- All commits except for documentation may never be directly pushed to the main branch, instead must be committed to a feature/fix branch and later be merged into the main branch.

Merge requests:

Merge requests are the last obstacle for code to be merged into the main branch, hence the quality control at this stage is of the utmost importance. So following that requires the following rules of control:

- A merge request may only ever be merged if it has met the approval of at least one other team member and two if the changes include conflicts or alter the codebase in any significant way.
- The code must have been fully tested via automated tests and make sure that it does not contain any known regression bug.
- Code that gets merged into the main branch must follow the coding conventions as much as possible to make sure the code stays readable, this includes but not all:
  - No unnecessary whitespace.
  - All methods even if internal must be fully documented by Java standards.
  - All classes and fields must follow the naming conventions.
  - Expressions should not exceed the line limit, if an expression is large split it up using tabs so it is fully visible on a single screen.

- During the approval process of a merge request the approvers must make sure all the rules above are followed and provide meaningful feedback if these are not as so it is clear what needs to be updated/changed. Furthermore, reviewers are required to write a concise but not meaningless summary of what they have taken into consideration when approving a request for documentation purposes.

Team roles:

During each week there is a chairman/woman and a minute taker who have certain responsibilities during the weekly meetings. As such there are a couple of metrics that define if they have done a sufficient job:

- • Agenda points are meaningful and describe something appropriate during the current state of the project.
- The chairman/woman should make sure that meetings do not derail into conversations that are not on the agenda and must take decisive action to keep everything on track. There are going to be some moments when something important pops up during the meeting in that case the chairmen/woman should use their common sense to judge that.
- The chairman/woman should make sure that every single agenda point is handled during a meeting.
- The minute taker should keep a copy of the week's agenda and properly time how long an agenda point takes.
- The notes of a minute taker of a meeting must be concise and clear and try to follow chronological order and only deviate from that order if deemed necessary.
- No topic that is discussed may be missing from the notes, as that is detrimental if anyone wants to refer to such a moment later on.

8 DECISION-MAKING

- We will make decisions as a group by using a majority voting system. So, for a decision to pass at least four of the six members need to agree to it.
- If the impact of an action by a member is extremely minor, said member is allowed to go ahead with it immediately without having to consult the rest of the team. The consequences of this will then be reviewed later at the next scheduled meeting.
- If an action has more significant consequences, a discussion with the rest of the team will need to take place before proceeding any further to review any potential issues or changes that could be required due to it. The team member can only go ahead with the action if they can convince a majority of the other members of the team of the benefits of it.
- In a meeting, during the distribution of the workload, if there is a task that nobody is willing to undertake, a member will be randomly assigned to it by means of a generator.

9 DEALING WITH CONFLICTS

As a team of six students developing a project, we understand that conflicts may arise and we're committed to resolve them. First of all, we strongly believe that prevention of disagreements is key. This is why, in our community, each one of us has clearly defined roles and responsibilities within the project. This ensures that everyone understands their contribution, reducing misunderstanding that can lead to fights. Unfortunately, conflicts are almost inevitable, but once they occur, we prioritize constructive dialogue that will optimistically lead to their resolution. We try to create a culture of respect and empathy that enables us to consider each other's perspectives and collaborate efficiently even when we don't fully agree with each other's opinions. Therefore, we recognize the importance of compromise and we aim to find intermediate solutions to problems that may arise. We are aware however that some conflicts may require outside assistance and in such cases, we seek help from our team's Teaching Assistant or the lecturers in most extreme scenarios. Nevertheless, our goal is to address conflicts efficiently and achieve a positive and harmonious atmosphere that will lead to the completion of the project and course success.

10 CONSEQUENCES

- Results based on the severity: The repercussions for not following the rules correspond to how bad the action is. We are taking this approach to ensure a fair and proportional response to any violation of the rules. The severity of a mistake will be decided by the entirety of the team who are not part of the issue itself.
- Small Mistakes: In instances of minor errors, we will issue a warning. These warnings have the purpose of inspiring corrective actions for those who make small mistakes. Although one warning may not be significant, not correcting it may generate harsher consequences.
- Moderate Mistakes: For medium-sized mistakes, a three-strikes rule shall be implemented. If an individual reaches three strikes, a thorough evaluation will take place, which may very well lead to excluding that person from the project entirely. Given that there are three strikes, everyone will have a chance to correct their mistake and thus remove the corresponding strike.
- Big Mistakes: Significant errors will be met with zero tolerance, as they can make or break the effort and dynamic of the entire group. These types of issues have to be dealt with quickly, to ensure regaining stability and progress in the project. There will be no chance to correct these actions, as their existence is not without malice or intent. Removal from the project will take place immediately.

11 OUTSIDE COLLABORATION

- Before Meetings: Before our required meetings, team members talk separately to plan what they want to discuss. This helps us be ready for anything regarding the agenda and organized for the main meeting, ensuring everyone is contributing.
- Weekly Coding Talks: We have regular meetings to discuss our coding work every week, outside of the regular team meeting on Tuesday. We talk about the tasks we need to do, who is responsible for what (Assignee), and other important things. This helps us keep track of our progress and

ensures the contribution is split equally. (both in terms of the amount of code, as well as front-end/back-end)

- How We Talk: We mostly use WhatsApp to talk to each other. It's easy to use, and everyone can access it. We might try other ways of talking in the future as our app development goes on.
- Urgent Meetings: If something really urgent comes up, we have a plan for quick meetings to solve the problem. This helps us handle important issues right away and keeps our workflow smooth.