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.