

Git Conventions (Group 3)

This document specifies the instructions and guidelines for the team to follow regarding the git repository workflow.

1. Naming conventions

This section contains the list of conventions that the team has to follow during the course of this project.

1.1. Backlog items

1.1.1. User Story Backlogs

User story backlogs represent the user stories that are to be implemented in the project. The format for those backlogs is:

```
``text
USxx-BIxx: <Backlog title>
Example: US01-BI01: Create abcxyz
```

1.1.2. Bugs Backlogs

Bugs backlogs represent known bugs with the existing codebase. The format for those backlogs is:

```
Bug-xx: <Problem description>
Example: Bug-01: abc is not working
```

1.2. Branches

Branches name must follow the standard branch naming convention of GitLab.

```
<GITLAB_ISSUE_NUMBER>-<usxx>-<bixx>-<Backlog title>
Example: 1-us01-bi01-Create abcxyz
```

1.3. Commit message

All commit messages must follow the [Conventional commits](#) specification.

```
<type>[optional scope]: <description>

Example:
```

- feat(frontend): created x component
- docs(readme): fixed typo in y.md
- fix: user unable to do z

2. Branches

This section states the official branches in the project and their access level.

Branch	Merge access	Push access	Force push access	Remark
main (default)	Maintainer	Maintainer	No	This is the default branch. It is used for the main development of the project.

3. GitLab labels

3.1. Scrum Board Categories

3.1.1. In progress

This category is used to label the backlog items that are currently being worked on.

3.1.2. To be reviewed

This category is used to label the backlog items that are finished and waiting to be reviewed.

4. Git workflow

1. Open a task from the **Open** issues of the Scrum board.
2. Assign yourself to the chosen task.
3. Create a *Draft* merge request using the **Create merge request** button, select **Create merge request and branch** from the drop menu, and adjust the target branch to be the **week** branch for that workweek.
4. Add yourself as an *Assignee* on the merge request page.
5. Adjust the Milestone for the merge request to match the Sprint number.
6. (*Preferably*) Select 'Delete source branch when merge request is accepted.' checkbox in the **Merge options** section.
7. Click the **Create merge request** button.

Notes:

- For any conflicts in the merge request, the author should resolve the conflicts locally and then push the changes to the target branch.