**QUALITY ASSURANCE   
POLICY**

# STATEMENT OF INTENT

|  |
| --- |
| This document servs to outline the obligatory procedures to be followed when committing work to the riipen/unified-lms repository. The intent is to deliver an open-source Ruby client to Riipen that can be extrapolated upon by different development teams in the future. The work done will be the baseline for a tool that can be developed further in the future. |

## 

# policy aims

|  |
| --- |
| * Keep branch nomenclature clear and consistent. * Implement a standardized commit process. * Implement a standardized review process. * Establish open communication between the development team and Riipen. * Implement a standardized nomenclature for file names. * Establish unified code formatting for the code base. |

# strategies/actions to reach goals

|  |  |
| --- | --- |
| • Adhere to the branching policy listed in section 1.0.  • Adhere to the commit policy listed in section 2.0.  • Adhere to the naming policy listed in section 3.0.  • Adhere to the review policy listed in section 4.0.  • Adhere to the code format policy listed in section 5.0. |  |

# 1.0 Branching policy

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * 1. Branch structure   2. Creating a branch for a new user story * When creating a branch for a new User Story the branch must originate from the Dev branch in the repository. * The branch must be named as follows: US#<user story number>   1. Creating a branch for a new task * When creating a new branch for a new task, the branch must originate from the User Story branch that the task belongs to * The branch must be named as follows: Task#<number of task>\_<name of task>   1. Resolving review findings * Review findings must be resolved in the task branch they originated from   1. Merging branches * Every merge into a parent branch should be a fast-forward. * First merge the parent branch into the child branch and resolve any conflicts. * The child branch can then be merged into the parent branch. * Reviews/PR’s are required for merges into the Dev and Master branch. * When A User Story is approved and merged into the dev branch the User Story branches and task branches will be deleted. (Any issues found in the future will require opening a new task).  2.0 Commit policy  |  |  | | --- | --- | | * When committing changes a commit message must be made including the User Story number and Task number along with a description of the change made. |  |  3.0 naming policy  |  |  | | --- | --- | | * Filenames must be written in snake case. * Class names must be written camel case. |  |  4.0 Review Policy  |  |  |  |  | | --- | --- | --- | --- | | Reviews will be constructed using the following check-list  [] Branching for the task is structured correctly.  [] No failures in the testing cases  [] No commented-out code is present.  [] All functions have the appropriate documentation.  [] No rubocop flags are present. 5.0 Code Format Policy  |  |  | | --- | --- | | * The formatting of the code will be enforced by robocop. |  | |  | |  |

# monitoring

|  |
| --- |
| To monitor the project is meeting the quality assurance guidelines, the team will conduct bi-weekly audits on the code base to ensure no policy violations are present. Any findings will be documented and resolved. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SIGNATURE** | Anthony Passetti | | **DATE** | 01/31/2024 |
|  | |  |  |  |
| **SIGNATURE** |  | | **DATE** |  |