

Table 1: Revision History

Date	Developer(s)	Change
Sept 21/22	Talha	Updating Workflow Plan
...

Development Plan

Flick Picker

Team 7, 7eam
Talha Asif - asift
Jarrod Colwell - colwellj
Madhi Nagarajan - nagarajm
Andrew Carvalino - carvalia
Ali Tabar - sahraeia

[\[Put your introductory blurb here. —SS\]](#)

- 1 Team Meeting Plan**
- 2 Team Communication Plan**
- 3 Team Member Roles**
- 4 Workflow Plan**

Git Workflow:

- *develop* branch will be the single source of truth, where changes on it are reviewed by the team before they get merged in. Thus the *develop* branch will have restricted permissions on it, preventing direct merges without admin overwrite and only one developer will be the admin, Talha
- Any changes have to be on their own branch, and a PR has to be cut with a full green checklist to get it merged into *develop*
- The checklist will grow as development on the application continues, currently the checklist is a single item, where the PR must have two approvals from the team. The future checklist items are as planned:
 - Entire test run has to be successful to ensure *develop* is in a healthy state
 - Test coverage delta should not be reduced, unless redundant tests are taken out

- Spotbugs must pass, enforcing healthy code practices
- Snyk checks must pass, ensuring the packages used do not have vulnerabilities
- Each PR must have at minimum a description filled out and a relevant feature ticket must have a Jira issue attached along with it

Issue Tracking - [Jira](#):

- All development changes need a descriptive ticket cut for it before it is ready for code review. Automation will link the PR to the ticket and vice versa as well
- The status of the ticket must be updated on the board, most importantly if it is in To Do so multiple developers do not start working on the same feature
- Points will be arbitrary for the ticket, based off how much work the developer working on it thinks it will be. It is going to be an indication of how complex the work is for the reviewers
- Utilize Jira's ticket types to have issue classifications:
 - Story: Ticket describing a new feature
 - Bug: Fixing existing code
 - Task: Small changes that do not fall in either of the above category

5 Proof of Concept Demonstration Plan

What is the main risk, or risks, for the success of your project? What will you demonstrate during your proof of concept demonstration to convince yourself that you will be able to overcome this risk?

6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

7 Coding Standard

8 Project Scheduling

[How will the project be scheduled? —SS]