

Chapter 1

Reasoning behind this app

Chapter 2

Workflows

This app works with two predetermined workflows and a unopinionated mode for more advanced users.

2.1 Basic workflow

In this basic workflow, the user modifies files, and those files show up on a staging area where the user can choose to not add some files. The app by default disregards hidden and temporary files from the selection.

Those files are permanently shown and updated as needed.

Then the user adds the commit message and commits the changes. When this happens, under the hood the selected files are added to the staging area, committed with the chosen message and then pushed onto the predefined server on the configuration wizard.

2.2 Auto branched workflow

In this slightly more advanced workflow, the user modifies files, and those files on a staging area where the user can choose to not add some files. The app by default disregards hidden and temporary files from the selection.

Those files are permanently shown and updated as needed.

Then the user adds the commit message and commits the changes. When this happens, under the hood a new temporary branch is created, the files added to the staging area and committed. Further commits are kept under this branch, and when the user chooses to, the branch is merged/rebased into the main branch. The master branch cannot be worked on, and only receives merge commits.

2.3 Unopinionated workflow

In this workflow, the full spectrum of commands are available, with the power user/developer in mind.

Chapter 3

Core features

3.1 Staging Area

The app should have a staging area where the modified files show up and can be seen, selected and deselected by the user.

This staging area should include a list of the modified files in the working dir, alongside the status of each one and some kind of interaction (namely, a button) to be able to add/remove items from the staging area.

Elements present in the staging area shall be added to Git's staging area when the user commits its changes, as to keep on with the workflow and make things easier.

Therefore, no action shall be done on the elements shown here until the commit box fulfills successfully the commit promise.

The working dir is permanently screened so to update the staging area on any detected file change.

3.2 Actions space

The actions space is the place where the action occurs. In here there should be buttons to execute the actions needed according to the workflow used.

For basic workflow:

- Undo commit
- Change repo

For auto branched workflow:

- Undo commit
- Merge branch
- Change repo

For unopinionated workflow:

- Undo commit
- Redo commit

- Merge branch
- Push
- Pull
- Sync
- Rebase
- Change repo

It should be noted that every push/commit executes a sync and resolves possible conflicts. Also if too much time passes (i.e. 10-20 minutes) the app should perform a checkout/pull to grab possible changes upstream.

3.3 Commit box

This is the simple commit box where the user writes its message and performs the commit by pushing a button.

3.4 Branch Viewer

This is the main interactive area of the app. Here the user can view the state of the branches, its pointers and who worked on which commit, commit info and status of everything. From here they can also select another branch or commit to update the workspace up to that point. The system automatically stashes, updates and restores points and files as needed.

The branch view can be navigable in 2D by zooming in/out and displacing the view.

The view is horizontal and color coded for branches, status and commits.

3.5 Workflow switcher

This is a simple button in the actions space that allows the user to update the UI and change the workflow to one of the three available.

3.6 Conflict solver

This is an under the hood feature that automatically solves merging issues arisen with the basis of the date (most recent has priority), stashing changes and allowing them to be restored if needed.

This feature as well preemptively updates the working dir and the repo to avoid such conflicts if possible.

As a last resort it may query the user to select a version.

3.7 Configuration wizard

The system prompts a wizard that allows the user to configure remote/user/git service and connect to the appropriate repos.

Chapter 4

Design mock-ups

Chapter 5

Tech stack