Chapter 7

Future Work

Snapstore is currently a minimal viable product as a file syncing system. However, there is more work that needs to be done before it is fit to be a quality version control system. These improvements fall into two general categories: implementation and user study.

7.1 Implementation

7.1.1 Functionality

The first area of functionality that needs to be created is the grouping and tagging of snapshots. These data structures must be maintained by their branch, as snapshots are. Any changes must be propagated to all collaborators on their branch.

The next feature that needs to be implemented is the ability to clone and merge branches. One potential interface problem we foresee is choosing which snapshots to clone over to the new branch. Users will be given the option to choose any subset of snapshots from every file to clone. Choosing these snapshots, from multiple cluttered snapshot graphs, might prove difficult.

The user should also be able to change the frequency with which snapshots are taken. This will help reduce the total amount of snapshots taken and clean up the snapshot graphs, making them easier to navigate.

A final feature is the ability to change the user's upstream location. This location is currently just a static value, pointing to a specific IP address, in the application, so it can be easily modified. However, this process will take work on the part of the end user to set up their own Snapstore server.

7.1.2 User Interface

The user interface, up until the writing of this paper, has been designed with function in mind, not aesthetics. No principles of good interface design have been consciously applied, and the interface overall has not been a strong focus. A redesign of the front end is necessary to garner a significant user base.

User interface design and conceptual design are related. Conceptual design dictates what concepts will appear in the interface and how they will interact with each other. User interface principles can be applied to simplify the interaction with these concepts.

However, user interface principles can also be applied in a way that confuses users. Concept overload (having one concept fulfill more than one purpose) might be applied to simplify the interface at the expense of confusing the user. This is the case with Dropbox's shared folder deletion [8]. It is important that future interface work on Snapstore not obscure the conceptual design that it is trying to show the user. It is the goal, after all, that this conceptual design becomes the user's mental model of the system.

Future iterations of the user interface should also minimize the amount of functionality accessible through only the Snapstore application. Like Dropbox, Snapstore functionality should be accessible from native file managers like the Mac OS Finder. This should include file-specific functionality like reversion and accessing a file's snapshot graph.

Snapstore would also benefit from another Dropbox feature, a menubar icon. The Snapstore menubar icon would allow users to easily see recent changes to a branch, elect to work offline, and perform other network-related tasks.

7.2 User Study

A user study will test how well participants like the functionality provided by Snapstore and the interface used to provide it. An interesting result of this study will be the differences and overlap between the high tech and low tech participants.

Thus far, we have evaluated Snapstore from a conceptual design theory perspective and from a personal, subjective perspective. However, in these user studies, we could gather more quantitative data. We could measure the time it takes for participants to perform actions they are used to performing with their file syncing or version control system. With those numbers, we could quantify the benefits that Snapstore brings the end user.