

Chapter 7

Future Work

Snapstore is currently a minimal viable product. However there is more work that needs to be done before it is fit for version control and for distribution. These improvements fall into three general categories: implementation, user study, and release.

Implementation improvements include completing the implementation of the conceptual design and functionality goals of Snapstore, as well as improving the user interface. A user study needs to be done with real users to help improve the user interface and provide a quantitative evaluation of Snapstore's features, comparing them to features on existing systems. Finally, the release of Snapstore involves cross-platform testing for performance, functionality, and interface. To distribute the application, it will need to be packaged, deployed, and made available from some online source.

7.1 Implementation

7.1.1 Functionality

The first area of functionality that needs to be addressed is the grouping and tagging of snapshots. This area is pivotal for more managerial and administrative parts of version control. Beyond their data structure creation, they will need to be maintained by their branch in the same way snapshots are. Any changes must be propagated to all users with access to their branch. Front end components that interact with these

structures will also be needed.

The next **area** that needs to be implemented is the ability to clone and merge branches. This development is necessary for powerful parallel development. These cloned and merged branches are the same as regular branches, but they ability to merge and to clone them will need to have hooks in the front end.

A final, optional, feature is the ability to change the user's upstream location. This is currently just a **static value 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. Components have been aligned, and they are well labeled. However, no principles of good interface design have been consciously applied, and it overall has not been a strong focus. A redesign of the front end is necessary to garner a significant user base.

7.2 User Study

A user study, ideally multiple, should be performed for Snapstore. These studies would ideally span multiple demographics and industries to test Snapstore core goal of appealing to the masses. It would also help iterate on Snapstore and evaluate its success.

7.2.1 User Interface

Though a redesign of the front end is necessary, the ideas behind that redesign should come from this user study. Proper interface testing should be done on multiple designs to find the design with the best and most affordances. An interesting result of these studies will be the differences and overlap between the high tech and low tech

participants.

7.2.2 Quantitative Evaluation

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

7.3 Release

7.3.1 Platform Testing

Although Snapstore is built on a cross-platform framework, there will need to be testing done on those platforms. A common concern with cross-platform frameworks is their poor performance on those platforms. Snapstore will need performance testing for these platforms to make sure that it is compatible with them. Any issues in performance or function will be identified during this process and remedied with additional libraries and/or performance improvements.

Another common issue with cross-platform frameworks is the interface differences between platforms. Some front end components can look and behave differently on different platforms. Much like in a web app, these differences can be handled with logic in the front end that distinguishes between platforms and delivers the appropriate code.

7.3.2 Packaging and Deployment

To be able to run Snapstore, it must be packaged with Electron's package module. This will allow it to be downloaded to all supported operating systems. Some of the modules used in the development and testing of Snapstore will need to be replaced with

more end-user friendly modules before this is done, however. For example, the tool by which Snapstore can create a new Snapstore folder is handled via command line. To be truly user friendly, this should all be done in the applications.

The deployment of Snapstore to a persistent server is necessary for users to be able to use Snapstore right when they download it. As mentioned earlier, the default Snapstore server will be our own. This requires a persistent server that can scale as a user base is grown.

Finally, Snapstore needs an online source from which users can download it. This source should include tutorial data on Snapstore, information similar to that found in chapter 2. It will also include the link to the open sourced version of Snapstore and links to download Snapstore for every operating system.