Sprint Report

Jackie Law (jjl284) *jjl284@cornell.edu*
Nishad Mathur (nm594) *nm594@cornell.edu*
Ning Wang (nw265) *nw265@cornell.edu*
Zhan Zhao (zz444) *zz444@cornell.edu*

March 9, 2017

1 Activity Breakdown

1.1 Jackie Law

In this sprint, I worked on the GUI and incorporating it with the system through methods in the DirectoryController class that allow the user to create and delete keychains and passwords. Specifically, I wrote the KeychainViewer, PasswordViewer, and GUIKeyService classes. In total, I spent about 10 hours in the past 2 weeks on this project. Creating a GUI was a valuable use of time because having a GUI was something that we had set to do in the first sprint, even though it was not originally listed as M in our functional requirements; having a user-friendly interface for those who are less tech-savvy would be effective.

1.2 Nishad Mathur

In this sprint I developed the crypto subsections and protocols of the system for this milestone and planned out protocols for future milestones (e.g. sharing). I also implemented the directory controller system (transparently handles file systems and crypto duties for the keychain) and associated testing for the module. I also implemented the client side code for the ServerController, which is responsible for addressing synchronization (and in the future sharing) of keychains. I also implemented a rudimentary CLI for testing the system (as well as a potential alternative client for the system). More generally I implemented the backend, CLI, controller and keychain packages. I spent approximately 16 hours on this project in the two weeks for this project.

I think that we all used out time correctly, as we focused on the 'M' class requirements for the project in this sprint and implemented fully (or partially) all of the M requirements, which i think is crucial for delivering this project in time.

1.3 Ning Wang

In this sprint I worked on the server-side functionalities and client-server communications. On the server side, I have set up database schema and implemented controller logics for interacting with database regarding account and keychain information. On the client-server communications, I designed message format of requests and responses between the client and server, and implemented logics for handling received requests on the server side. I spent approximately 15 hours on this project in the two weeks for this project.

I think that the tasks I did were a good use of time, since the implementation of server is crucial when there are more than one user of the application. Our goal is eventually making the application usable over the network, which requires the existence of a server side application.

1.4 Zhan Zhao

In this sprint, I first worked on creating wrapper classes on the postit server side. I worked on setting up communication channels to connect the postit server and the postit client. I spent around 8 hours on this project so far. It is a valuable use of time because the postit needs to communicate with clients and the postit clients need to talk to the server as well. In the future, sync will also need this communication channel.

2 Productivity Analysis

In this sprint, we used GitHub to create issues to complete and assign them. Our minimum goal was to have the system work locally and store encrypted passwords on disk. In relation to our Backlog, this would be to complete all of the 'M' items where the user was specifically "customer". We were able to almost complete this goal, but we missed two items:

- As a customer, I can create a new account to use the password manager
- As a customer, I can synchronize keychains across multiple devices for convenience

We did not do the former because our focus was on being able to set up the keychain and password system and the functionalities with those assets. As for the latter, we needed to complete the set up of servers before we could accomplish the second, which took longer than intended. We did not complete all items we had sought to do also because there were some issues that were not originally in our system backlog that took time out of our sprint. These items were:

- Creating a wrapper class for passwords, keychains, and directories
- Creating a GUI for the system
- Creating a Controller class that connects the GUI to the model
- Setting up the database for the system
- Fixing design differences between client and server side

Some items were necessary for some must-have items in our backlog, while some were unaccounted for completely. We did not anticipate that these items would take as much time as they did; as a result we actually had less time than we thought.

In future sprints, we should allocate more buffer time within tasks in case any aspects take longer than intended. Additionally, we should be more coordinated. Since there was some design differences between client and server side, it took time out of our sprint to fix them. To do this, at the beginning of the sprint, we should discuss clearly what we intend to complete and how, so that there is little to no ambiguity between what people exactly have to do and how they will do it.