Assurance

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

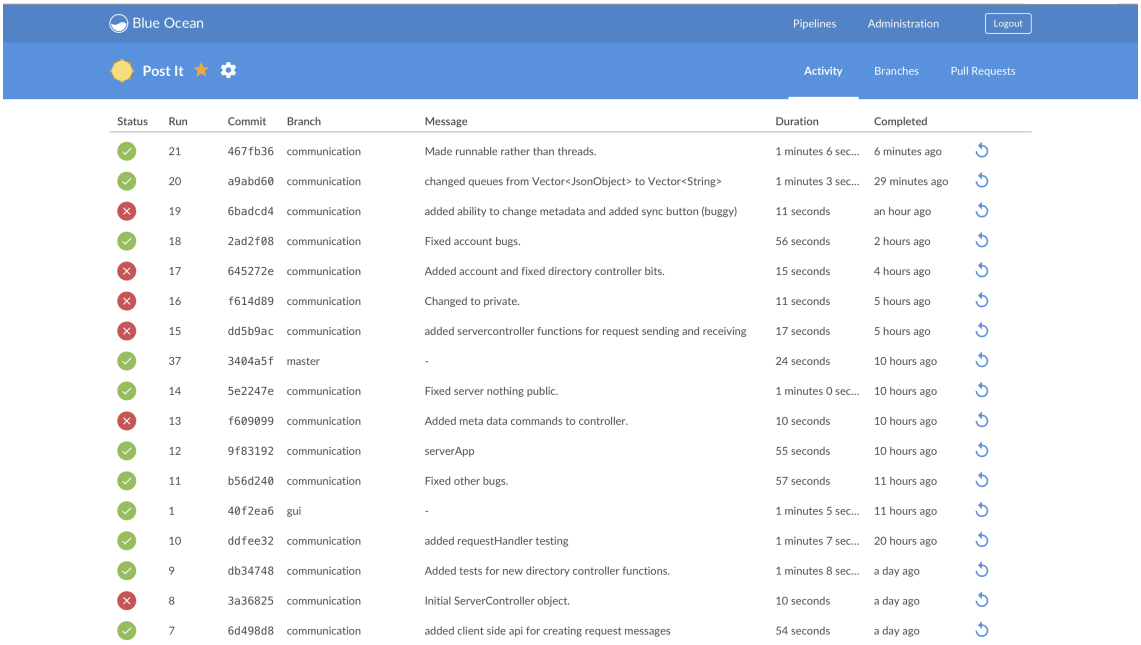
1 Testing

1.1 JUnit Tests

Currently we have JUnit testing for the core directory-controller class on the client. This class is responsible for mediating all interactions the applications have with the keychains and directory, transparently handling authentication, cryptography and persistence for them. Thus it was vitally important to test these functions to ensure they behave correctly in the standard cases and as a result we now have automated testing which runs for this class and validate its behavior.

The underlying classes aren't tested individually at the moment but testing the overarching controller should provide sufficient coverage as to ensure that any operations which can be performed by the user code are correct (as the user code only accesses the underlying libraries through the unit tested controller). The mutation testing provides some evidence as to the sufficiency of this testing strategy.

1.2 Jenkins



Status	Run	Commit	Branch	Message	Duration	Completed
✓	21	467fb36	communication	Made runnable rather than threads.	1 minutes 6 sec...	6 minutes ago
✓	20	a9abd60	communication	changed queues from Vector<JsonObject> to Vector<String>	1 minutes 3 sec...	29 minutes ago
✗	19	6badcd4	communication	added ability to change metadata and added sync button (buggy)	11 seconds	an hour ago
✓	18	2ad2f08	communication	Fixed account bugs.	56 seconds	2 hours ago
✗	17	645272e	communication	Added account and fixed directory controller bits.	15 seconds	4 hours ago
✗	16	f614d89	communication	Changed to private.	11 seconds	5 hours ago
✗	15	dd5b9ac	communication	added servercontroller functions for request sending and receiving	17 seconds	5 hours ago
✓	37	3404a5f	master	-	24 seconds	10 hours ago
✓	14	5e2247e	communication	Fixed server nothing public.	1 minutes 0 sec...	10 hours ago
✗	13	f609099	communication	Added meta data commands to controller.	10 seconds	10 hours ago
✓	12	9f83192	communication	serverApp	55 seconds	10 hours ago
✓	11	b56d240	communication	Fixed other bugs.	57 seconds	11 hours ago
✓	1	40f2ea6	gui	-	1 minutes 5 sec...	11 hours ago
✓	10	ddfee32	communication	added requestHandler testing	1 minutes 7 sec...	20 hours ago
✓	9	db34748	communication	Added tests for new directory controller functions.	1 minutes 8 sec...	a day ago
✗	8	3a36825	communication	Initial ServerController object.	10 seconds	a day ago
✓	7	6d498d8	communication	added client side api for creating request messages	54 seconds	a day ago

We use Jenkins to run our automated test suite every time a commit is made to Github and it then automatically reports the results of these tests to Github where it is displayed in-line to the user, allowing us to track the stability of the system for each branch and avoid or detect regressions in our rather branch heavy process, whether they be due to user error or due to merge issues.

Jenkins also runs our more time consuming processes, such as Mutation testing (to test the coverage of our tests) and aggregates the results, allowing us to observe trends and properties of the program which might otherwise be missed.

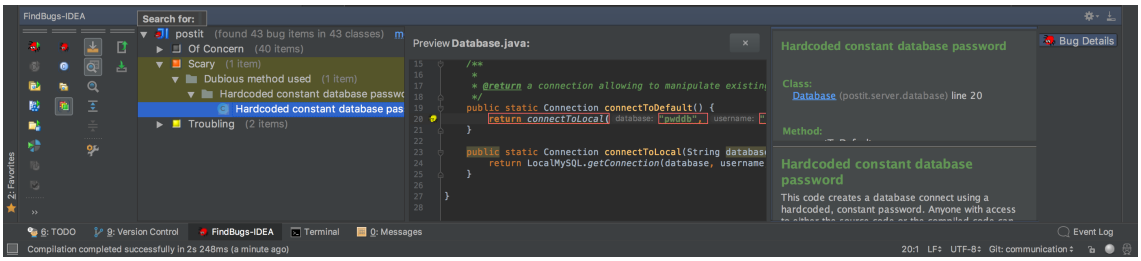
1.3 Manual User Tests

With the GUI, we manually ran user tests to check for bugs or unplanned features. We would test that our GUI implemented the anticipated functionalities (adding and deleting keychains and passwords) and that any updates and changes would be stored after the app was closed. We also manually tested to ensure that the system would run only when the correct master password was entered, aligning with our security goal: The system shall prevent unauthorized users from viewing the passwords.

We used manual testing to test the server side functionalities, such as database connections and updates, and request handling. We manually created requests to send to the server side and check if the response is as expected. We went through the pipeline of for the following use cases: add account, authenticate account, get keychains of a user, add, remove and update a keychain. In future milestones, however, we are considering switching to using JUnit tests since that way can be automated through Jenkins.

2 FindBugs

Our code currently does not contain any Scariest Bugs (Rank 1-4). We managed to whittle down the bugs we had to only one Scary bug with a rank of 7.



Our 1 Scary bug is due to the hard-coding of the database credentials (which we are using for now before we connect to a database); this will be soon resolved through the use of configuration files in the next sprint.