Our Project

Choosing a project to test was difficult for our team, we looked through, downloaded, and attempted to compile seven to eight projects before finally settling on OpenMRS[1]. To see documentation on our experience with other projects check out our Evaluation report page on our Github Wiki[2]. We decided to go with OpenMRS for a few reasons. First, it is a program that provides an electronic medical record system and their mission is to, "Improve health care delivery in resource-constrained environments by coordinating a global community to create and support this software."[3] We thought this was a great cause and wanted to contribute to it. Secondly, this was one of the few projects we were able to compile and build making it the best choice for our team.

Building and Compiling OpenMrs

The Documentation for building OpenMrs were very clear and easy to execute. Before we did anything, we had to make sure we had the prerequisites to be able to build and compile OpenMRS. In this case we needed to make sure we had Java JDK, a minimum of Java 8, and Maven installed on our virtual machines. Once we had the prerequisites installed, we were free to begin the building and compiling process.

The first step was to clone the OpenMrs repository to our local computer by using the command "*git clone https://github.com/openmrs/openmrs-core.git"*. Once the repository was on our computer, we could start compiling. To do this we, used the command "*cd open-core"* to enter into the open-core

[1] https://github.com/openmrs/openmrs-core
[2] https://github.com/csci-362-02-2020/Team3/wiki/Evaluation-Report
[3] https://openmrs.org/

directory. From there we built the project by running the command *"mvn clean package."* This command built OpenMrs and ran a few of its tests as seen in Figure 1.

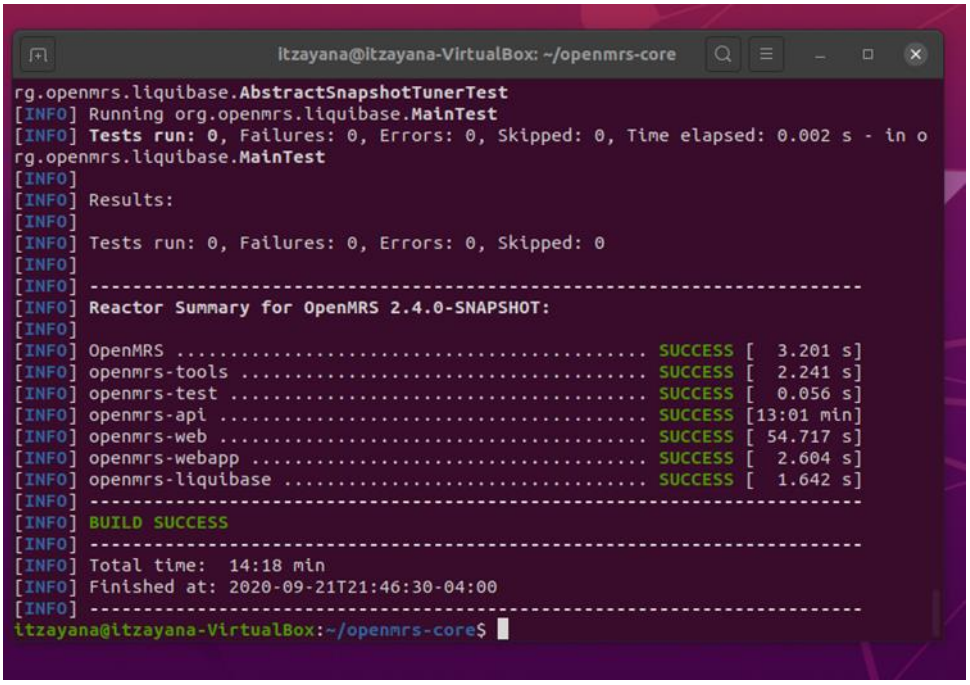

Figure 1: Successful Build of OpenMRS

In addition to compiling and building OpenMrs, we were also able to get it to deploy to FireFox. We simply ran the commands *"cd openmrs-core/webapp"* and *"mvn jetty:run."* This opened a localhost's page that could be accessed by typing this site handle in the search bar "*localhost:8080/openmrs."*

Testing OpenMRS

Running the tests on OpenMRS turned out to be a lot simpler than we had previously thought. Originally, we were trying to log into OpenMRS on the localhost site, but we kept running into issues with the connection between OpenMRS and MySQL. Luckily, we tried testing from the terminal by changing the directory to the OpenMRS directory and using Maven "*mvn"* test. After a little bit of trial and error, we were able to build and run tests on OpenMRS as seen in Figures 2 and 3. However since we

realized we should be testing in the terminal late in our project search, we weren't able to explore the tests thoroughly.

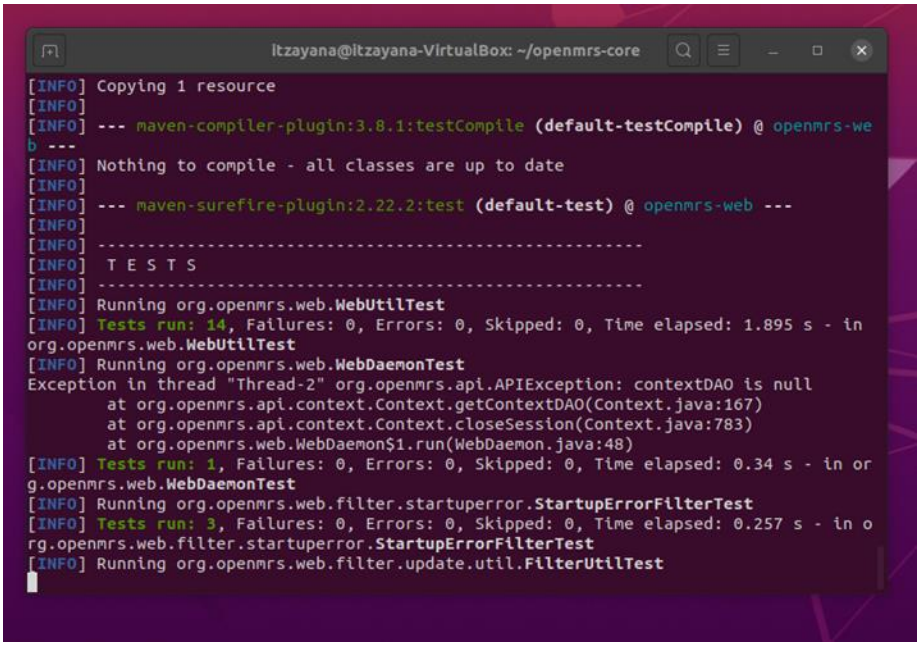

Figure 2: Testing OpenMrs

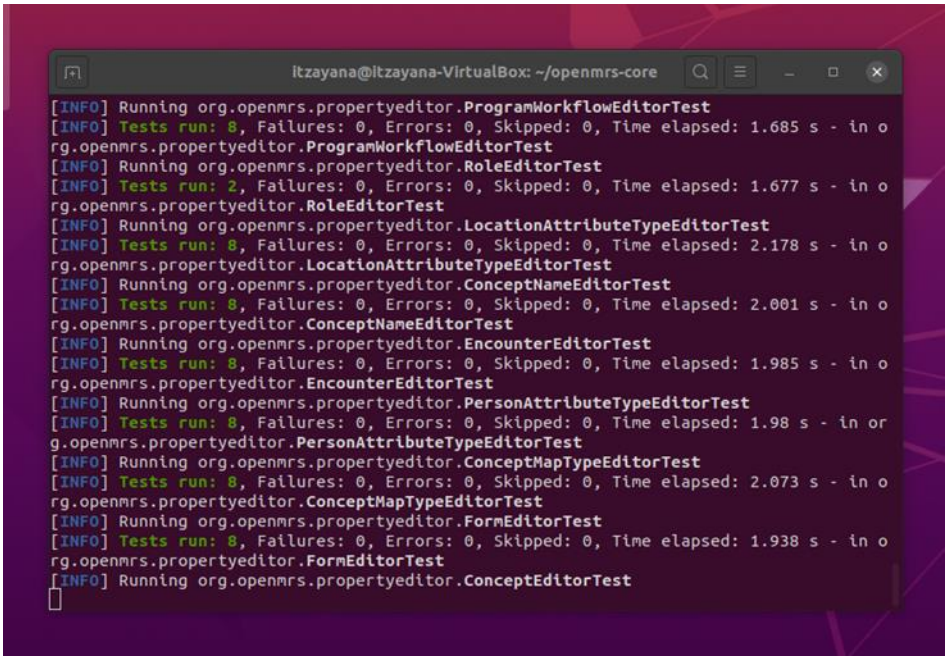

Figure 3: Testing OpenMrs

Final Thoughts and Experience Report

This experience has been difficult so far. Our biggest issue as a team has been knowing what exactly we were to be looking for. Our first major struggle was looking for a project with clear building instructions, and even when we thought we understood the instructions it seemed every time we tried to build and compile a new project we seemed to run into more issues and errors. Our second struggle seemed to be knowing what a compiled and built program looked like. We seemed to think that to successfully build and compile a project we needed to get it up and running in a browser. While this is certainly a step in the right direction, it seems we may have been overthinking it a bit. With the use of Maven mvn test we were able to build and test OpenMRS. Deliverable One has proven to be a challenge for our team, but we have decided to continue with OpenMRS for Deliverable Two.