A Free Open Source Software Experience with OpenMRS

Aaron Gaynor, Jake Marotta, Paige Peck, Colton Williams, Dr. James Bowring

CSCI 462: Software Engineering Practicum, Dr. James Bowring, Spring 2017

What is FOSS?

Free Open Source Software (FOSS) is software that is both freely licensed for use, copy and study, and also has source code that is openly shared. This allows for a community to grow so long as there is a desire from people to contribute. Having the code available publicly also ensures that the software is more secure, as multiple eyes are on the code at any point in time. This means that no one is able to commit malicious code to the project, and anything that is not acceptable is fixed quickly.

What is OpenMRS?

OpenMRS is designed to provide comprehensive record-keeping for medical organizations which don't have the resources for more expensive systems, and is used all over the world. OpenMRS is FOSS, which means that it is 100% free to use, and that anyone who wishes to contribute to the project can. It has been active since 2004, and its community consists of, software engineers, medical professionals, and more.

Issue Tracking

The OpenMRS development community organizes itself using a tool called JIRA, which organizes each issue into separate tickets. Each ticket contains detailed information about an issue, where it was found, what version, etc. Tickets are also used to propose new functionality. When someone wants to work on an issue, they must first claim the ticket on the JIRA. All of the issues that we worked on were selected from a section called "Introductory Tickets", which was filled with issues meant for newcomers who may not be familiar with the inner workings of the project.

Experiences

Familiarizing ourselves with the project structure took a lot of time, because the project is separated into dozens of modules, each of which has dependencies that must be delicately set up in order to build successfully. The bulk of our learning at this stage came from attempting to implement a convenient feature allowing a user to press 'Escape' to cancel a form entry. While we eventually moved on from this issue, it provided us with the insight necessary to tackle further contributions.

A team member was able to solve an inconsistency in one of OpenMRS' forms where a field that was required was not marked so with a red asterisk (see Figure 1). We submitted the solution, and, much to our disappointment, a main developer told us that the solution was no longer needed for the current version of OpenMRS.

Our final chosen ticket dealt with a problem in the reporting module, in which dataset keys with an ampersand would cause inconsistent behavior, namely the inability to edit or delete those dataset definitions. We deduced that this must be an input sanitization issue; '&' is a special character in SQL, which is used to manipulate OpenMRS' databases, and including such a character caused unexpected behavior in the reporting module. Our solution was to replace every attempted instance of an ampersand with the word "and", the code of which is shown in Figure 3. Figure 5 shows a dataset definition that we attempted to save with the name "test&", but was instead changed to "testand".

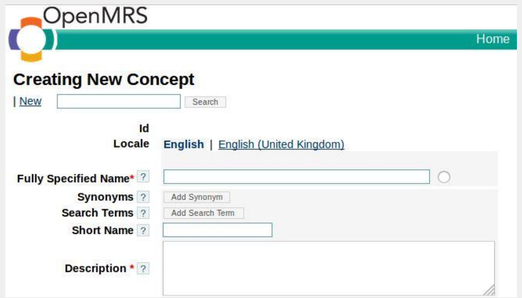

Figure 1: Adding red asterisk next to description field

Figure 2: Submitting a pull request

```
/**
 * @see DefinitionPersister#saveDefinition(Definition)
 */
public T saveDefinition(T definition) {
    definition.setName(definition.getName().replaceAll("&", "and"));
    return getService().saveDefinition(definition);
}
```

Figure 3: Fix for reporting module bug

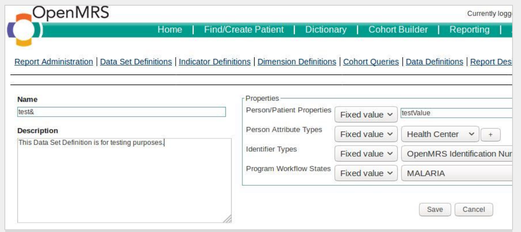

Figure 4: Before saving a dataset definition

Figure 5: After saving a dataset definition

Contributions

- Description Field Bug - Paige Peck
- Reporting Module Documentation Error - Paige Peck and Aaron Gaynor
- Dataset Definition Bug - Jake Marotta and Aaron Gaynor
- Wiki Contributors - Paige Peck and Colton Williams
- Poster Contributions - Paige Peck, Colton Williams, Jake Marotta, Aaron Gaynor

Conclusions

Our group learned a number of lessons over the course of the semester. One of the lessons we learned is that you often need more time than you expect to accomplish a task like fixing a bug. Another lesson we learned is that you must be adamant about communicating with teammates in order to utilize your time so that your work doesn't overlap. In conclusion, our group was able to learn about what contributing to a FOSS project entails, will have valuable experience going forward, and will be able to work on large software projects as a result.

Works Cited

1. Downey, Michael, and Ivo Ulrich. "Getting Started as a Developer." OpenMRS Wiki, OpenMRS Inc., 2004, wiki.openmrs.org/display/docs/Getting Started as a Developer. Accessed 11 Apr. 2017.
2. "Gnu.org." [A GNU Head], Free Software Foundation, Inc., www.gnu.org/philosophy/free-sw.html. Accessed 11 Apr. 2017.

Checkout our GitHub

OpenMRS

COLLEGE of CHARLESTON
COMPUTER SCIENCE
1770