Project Goals:

The goals of this project were to experience the software development process by planning creating and documenting the creation of a testing framework for an H/FOSS open source project.

About OpenMRS

OpenMRS is a software platform and a reference application which enables design of a customized medical records system with no programming knowledge (although medical and systems analysis knowledge is required). It is a common platform upon which medical informatics efforts in developing countries can be built. The system is based on a conceptual database structure which is not dependent on the actual types of medical information required to be collected or on particular data collection forms and so can be customized for different uses.

Testing Framework

The testing framework we developed uses a script to take a series of test cases—generated by the user and saved in text files—and run the tests outlined therein. Once the test is run and the result is compared to the expected result as dictated by the requirement being tested. The test will then pass or fail based on whether the results match the expected output. The results of the test along with the details of each test case is then put into an HTML file and opened in a browser.

Directory Specifications

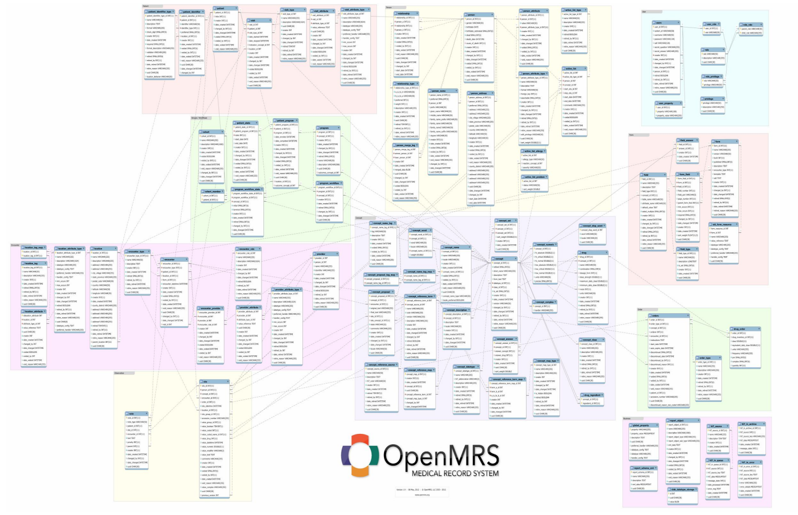
```
/TestAutomation
    /project
      /src
      /bin ...
   /scripts
      runAllTests.sh
/testCases
    testCase1.txt test, Case2.txt.
/testCasesExecutables
    testCase1...
 /temp
/oracles
    testCase1Oracle...
/docs
    README.txt
/reports
    testReport,html
```

Component Being Tested

For this project we chose to test the *Cohort* class of the OpenMRS system. A cohort is essentially a means of grouping patients together, assumedly based on some sort of commonality. To facilitate comparisons among groups and changes in groups, cohorts can have a series of set operations performed on them. These include unions, intersection, and subtractions. These should have the same result as those performed on sets. This functionality is what we concentrated our tests on.

OpenMRS

MEDICAL RECORD SYSTEM



How Does OpenMRS work? (Taken from OpenMRS.org)

OpenMRS is based on the principle that information should be stored in a way which makes it easy to summarize and analyze, i.e., minimal use of free text and maximum use of coded information. At its core is a **concept dictionary** which stores all diagnosis, tests, procedures, drugs and other general questions and potential answers. OpenMRS is a client-server application, which means it is designed to work in an environment where many client computers access the same information on a server. There are several layers to the system. (Warning, geek-speak ahead!)

- Our data model borrows heavily from the Regenstrief model, which has over a 30-year history of proven scalability and is based on a concept dictionary<.
- The API (application programming interface) provides a programmatic "wrapper" around the data model, allowing any developer to program against more simplified method calls rather than having to understand the intricacies of the data model.
- The web application includes web front-ends and modules that extend the core functions — these are the user interfaces and applications themselves built upon the lower levels.

An Example of one of our Test Cases

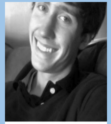
Test ID:	1
Requirement:	The union of two Cohorts has no repeated values
Component:	Cohort
Method:	public static Cohort union(Cohort a, Cohort b)
Inputs:	[1,2,3,4,5] [3,4,5]
Expected:	[1,2,3,4,5]
Driver:	CohortUnionDriver
Notes:	

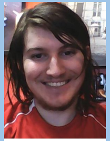
Below is a sample of the Test Results

Test Results

TestID	Requirement	Inputs	Expected Output	Test Result
1	The union of two Cohorts has no repeated values when the intersection of the inputs is non-empty	[1,2,3,4,5] [3,4,5]	[1,2,3,4,5]	Passed
2	The union of a non-empty Cohort and an empty Cohort is the non-empty cohort	[1,2,3,4,5] []	[1,2,3,4,5]	Passed
3	The union of two empty Cohorts is empty	[] []	[]	Passed
4	The union of two non-intersecting Cohorts contains the values of both Cohorts	[1,2,3] [4,5]	[1,2,3,4,5]	Passed
[illegible]	The union function [illegible]	[123456,234567,345678,456789] [illegible]	[illegible]	[illegible]

Our Team InsertNameHere


Daniel Hurlburt
Lead Programmer, Documentation


Benjaming Buckwalter
Secondary Programmer
Research

Avery Douglas
Code Review, Documentation

Personal Accomplishments

Before going into this assignment two of our three members had never programmed in bash and none of us had much experience in this type of programming. It was definitely as struggle at times, but we all came out knowing a great deal more that we had coming in.