OpenMRS
MEDICAL RECORD SYSTEM

Violet Smith, Jayton Fee, and Thomas Marshall

Introduction:

OpenMRS is a patient based medical record system focusing on giving providers a free customizable electronic medical record system.
The mission of OpenMRS is to improve health care delivery in resource-constrained environments by coordinating a global community that creates a robust, scalable, user-driven, open source medical record system platform.

Chapter 1:

We started off with the project Martus which proved to give our team many problems. From Martus we found the readme file with the code needed to be well documented in order to succeed. We then made the decision to switch to OpenMRS.

Chapter 2:

We created five test cases that run successfully in InteliJ although they could not run in the command line just yet.

Chapter 3:

We designed and built an automated testing framework that would work with our five original test cases. We were still having issues at this point getting the code to run through the command line instead of InteliJ.

Test Cases:
Each test case is seven lines:
1. Test ID
2. Requirements tested
3. Component being tested
4. Driver being used
5. Method being
6. Test inputs
7. Expected output

Chapter 4:

We were able to get past the error from the previous chapter. We also reworked the first 5 test cases and created 20 more. Ran the code through the script.

OpenMRS Test Results:

Test ID 01
- Requirements: Subtracts the ASCII values of s and t
- Component: NaturalStrings.java
- Driver: Test01
- Method: compareNaturalAscii(String s, String t)
- Test Input: "adc" "abc"
- Expected Outcome: 2
- Actual Outcome: 2

Passed

Test ID 02
- Requirements: Subtracts the ASCII values of s and t
- Component: NaturalStrings.java
- Driver: Test02
- Method: compareNaturalAscii(String s, String t)
- Test Input: "abc" "adc"
- Expected Outcome: -2
- Actual Outcome: -2

Passed

Test ID 03
- Requirements: Subtracts the ASCII values of s and t
- Component: NaturalStrings.java
- Driver: Test03
- Method: compareNaturalIgnoreCaseAscii(String s, String t)
- Test Input: "ABC" "abc"
- Expected Outcome: 0
- Actual Outcome: 0

Passed

Script:

- Completely relative
- Runs all files in the testCases directory
- Gets the necessary driver and inputs needed from each test case
- Compare expected and actual results for a pass/fail
- Prints results in an html file and automatically opens in your preferred browser

Fault Injections:

We were able to inject small changes into the code that would cause our test cases to fail. Example: Changing an == to a != in OpenMRS' code.

Test ID	Requirement	Component	Method	Tested Input	Expected outcome	Pass/Fail
01	Tests the method compareNaturalAscii when the value for s is greater than the value for t	NaturalStrings.java	compareNaturalAscii(String s, String t)	s: "adc" t: "abc"	A value greater than 0	Pass (returns 2)
02	Tests the method compareNaturalAscii when the value for t is greater than the value for s	NaturalStrings.java	compareNaturalAscii(String s, String t)	s: "adc" t: "abc"	A value less than 0	Pass (returns -2)
03	Tests the method compareNaturalIgnoreCaseAscii when the value for s is equal to the value for t	NaturalStrings.java	compareNaturalIgnoreCaseAscii(String s, String t)	s: "ABC" t: "abc"	0	Pass
04	Tests the method compareNaturalAscii when the value for t is greater than the value for s	NaturalStrings.java	compareNaturalAscii(String s, String t)	s: "Abc" t: "adC"	A value less than 0	Pass (returns -32)
05	The person can be reached through the user	User.java	Public person User(Person)	"Male"	"Male"	Pass
06	Tests the compareTo method on how it handles when the first low inputted is greater than the second low and the first high is greater than the second high	DoubleRange.java	CompareTo(DoubleRange other)	Range1 = low 78.9, high 90.1 Range2 = low 34.0, high 56.0	1	Pass
07	Tests the compareTo method on how it handles when the first low inputted is less than the second low and the first high value is greater than the second high value	DoubleRange.java	CompareTo(DoubleRange other)	Range1 = low 34.0, high 90.1 Range2 = low 34.0, high 56.0	-1	Pass
08	Tests the compareTo method on how it handles when the first low value inputted is equal to the second low value and the first high value is equal to the second high value	DoubleRange.java	CompareTo(DoubleRange other)	Range1 = low 34.0, high 34.0 Range2 = low 34.0, high 34.0	0	Pass
09	Test the from Specification method when a string is given as an argument	LocaleUtility.java	fromSpecification(String localeSpecification)	"eg_US_56"	"eg_US_56"	Pass
10	Test inputs a role and confirms the role was added	User.java	hasRole(String r)	"CEO"	True	Pass
11	Test inputs a role and confirms if the role makes the user a superuser	User.java	IsSuperUser()	"System Developer"	True	Pass
12	Test inputs a role and confirms if the role makes the user a superuser	User.java	IsSuperUser()	"Graphic Designer"	False	Pass
13	Test if the method contains can detect if the double value is in the range	DoubleRange.java	contains(double d)	20.9	True	Pass
14	Test inputs two allergens and sees if they're the same	Allergy.java	hasSameAllergen(Allergy allergy)	"DRUG"	True	Pass
15	Test inputs two allergens and sees if they're the same	Allergy.java	hasSameAllergen(Allergy allergy)	"DRUG", "FOOD"	False	Pass
16	Test if the method contains can detect if the double value is in the range, if the range is inclusive	DoubleRange.java	contains(double d)	10.0	True	Pass
17	Tests if the method supportsPropertyName returns false when a designated property name is not entered	LocaleUtility.java	supportsPropertyName(String propertyName)	"hello"	False	Pass
18	Tests if the method	NaturalStrings.java	compareNaturalAs	s: "abc"	0	Pass

How to get the Framework Running:

1. Clone the Repository
2. Navigate to the driver directory
3. Copy this into the same directory as OpenMRS
4. Compile the project through maven
5. Run the script

Lessons Learned:

- How to work with github
- How to run code from the command line
- BASH
- Some HTML
- Improved teamwork
- Testing Frameworks