TEAM 7

This is the GitHub repository for Team 7 CSCI 362 Project

TEAM MEMBERS

Jayton Fee, Violet Smith, Thomas Marshall

PROJECT DESCRIPTION

This repository contains test materials for the OpenMRS project. Our team has built an automated test script that runs 25 test cases.

REQUIREMENTS

Ubuntu
One of the requirements for the class was that our project runs on a Linux Operating System. We have chosen to use Ubuntu

Java
OpenMRS is a Java application which is why you need to install a Java JDK.
If you want to build the master branch you will need a Java JDK of minimum version 8.

Maven
Install the build tool Maven.
You need to ensure that Maven uses the Java JDK needed for the branch you want to build.
To do so execute: *mvn -version*

Git

Install the version control tool git and clone this repository with:

git clone https://github.com/csci-362-01-2020/Team-7

and the OpenMRS repository with:

git clone https://github.com/openmrs/openmrs-core.git

INSTRUCTIONS TO RUN TESTS

1. Download the TestAutomation folder and the openmrs-core repository from the openmrs project on github: https://github.com/openmrs.
2. Make sure the TestAutomation folder and the openmrs-core folder are both located in the same directory.
3. Copy all of the files located inside the TestAutomation/testCasesExecutables directory and paste them inside the openmrs-core/api/src/main/java/org/openmrs
4. Go into the OpenMRS code and comment out all instances of Log and for the security class comment out the first private method in openmrs-core/api/src/main/java/org/openmrs/util/naturalStrings java
5. Open the terminal and use the cd command to enter the top level of the openmrs-core directory
6. Once there, make sure you have maven and a java jdk of version 8 or higher installed and then compile the project with the command: mvn clean install (As a tip, once the system begins running the tests, it is finished compiling the project and the built in tests can be skipped by using ctrl+c. This is recommended because the tests can take up to 30 minutes to complete)
7. Once the compilation is complete, use the cd command to navigate to the TestAutomation/scripts directory
8. To run the automated tests script use the command: bash runAllTests.sh
9. The script should be fully automated from this point and automatically open your preferred browser with the results of each test once completed.