Deliverable #4

Team Members: Hannah Posch, Alex Thropp, Daniel Baczmaga, Daniel Lee

Project: Glucosio

TeamCorrect

The GitHub repo for TeamCorrect CSCI 362 Project

Team Members

Daniel Lee, Hannah Posch, Alex Thropp, Daniel Baczmaga

Project Description

This repository contains test materials for the project Glucosio. We have built a test script, runAllTests.sh, that automates the testing of the Glucosio system.

Experiences

Deliverable 4's production was difficult as we needed to revise our script to be more universal and modifiable. We wanted our script to run all of the tests without requiring hard coding the commands to compile our test java files. However, most of the java files in Glucosio must be compiled simultaneously and executed in a specific order which required us to build a creative solution to our script problem. After solving this, we had a good experience with building the test cases and using our script to run them. We split up the test case creation among us to ensure that we were all contributing to the testing and also testing different portions of Glucosio's requirements. Overall, we believe that our Glucosio testing framework provides a buildable solution to testing the functionality

and features of the Glucosio application and can be enhanced with further tests in the future.

Instructions to run tests

1. Open Terminal
2. git clone https://github.com/csci-362-fall-2018-01/TeamCorrect
3. cd TeamCorrect/TestAutomation/scripts (May need additional navigation, depending on where the repository was cloned to)
4. ./runAllTests.sh
5. Output will be stored in an html file which should open automatically in the default browser.

*If any of the above commands return permission denied, use 'sudo' in front of the command.

*If the ./runAllTests.sh returns command not found, use 'sudo chmod 755 runAllTests.sh', then './runAllTests.sh'

Test Cases

Test Number: 01

Requirement being tested: RS013 Set Reminders

Component being tested: Reminder.java

Method being tested: setID, setMetric, setActive

Test input(s) including command-line argument(s):

Id = 100

Id = 123456789

Metric = "0mL"

Metric = "9L"

Active = true;

Expected outcome(s):

Id returned = 100

Id returned = 123456789

Metric returned = 0mL

Metric returned = 9L

Is Active returned = true

Test Number: 02

Requirement being tested: RS001 Record user input - Glucose

Component being tested: Glucose Converter

Method being tested: glucoseToA1C

Test input(s) including command-line argument(s):

mgDl = 10

mgDl = 25

mgDl = 50.0

mgDl = 0.0

Expected outcome(s):

10.0 glucose level converted to 1.98

25.0 glucose level converted to 2.5

50.0 glucose level converted to 3.37

0.0 glucose level converted to 1.63

Test Number: 03

Requirement being tested: RS001 Record user input - Ketone

Component being tested: KetoneReading.java

Method being tested: setReading, setId

Test input(s) including command-line argument(s):

Reading = 2.0

Reading = 10.0

ID = 1234

ID = 146789

Expected outcome(s):

Id returned = 1234

Id returned = 146789

Reading returned = 2.0

Reading returned = 10.0

Test Number: 04

Requirement being tested: RS001 Record user input - Body Weight

Component being tested: WeightReading.java

Method being tested: setReading, setId

Test input(s) including command-line argument(s):

Reading = 100.0

Reading = 50.0

ID = 1234

ID = 146789

Expected outcome(s):

Id returned = 1234

Id returned = 146789

Reading returned = 100.0

Reading returned = 50.0

Test Number: 05

Requirement being tested: RS001 Record user input - Blood Pressure

Component being tested: PressureReading.java

Method being tested: setMinReading, setMinReading, setId

Test input(s) including command-line argument(s):

MaxReading = 180.0

MinReading = 110.0

ID = 1234

ID = 146789

Expected outcome(s):

Id returned = 1234

Id returned = 146789

Min Reading returned = 110.0

Max Reading returned = 180.0

Test Number: 06

Requirement being tested: RS001 Record user input - Cholesterol Level

Component being tested: CholesterolReading.java

Method being tested: setTotalReading, setLDLReading, setHDLReading, setId

Test input(s) including command-line argument(s):

totalReading = 190.0

LDLReading = 90.0

HDLReading = 70.0

ID = 1234

ID = 146789

Expected outcome(s):

Total reading set to 190.0

LDL reading set to 90.0

HDL reading set to 70.0

ID set to 1234

ID set to 146789

Test Number: 07

Requirement being tested: RS017 Build User

Component being tested: User.java

Method being tested: createUser, setName, setD_type, setCounty, seta1cUnit, setMinimum, setMaximum

Test input(s) including command-line argument(s):

Name = "Bob"

Type = 1

Country = "United States"

A1c Unit = mg/dL

Minimum = 4.0 mmol/L

Maximum = 9.0 mmol/L

Expected outcome(s):

Bob's name set to Bob

Bob's diabetes type set to Type 1

Bob's country set to United States

Bob's a1c unit set to mg/dL

Bob's range minimum set to 4.0 mmol/L

Bob's range maximum set to 9.0 mmol/L

Test Number: 08

Requirement being tested: RS016: Provide tips

Component being tested: ActionTip.java

Method being tested: getTipTitle, getTipDescription

Test input(s) including command-line argument(s):

TipTitle = "Diabetes Tip #1"

TipDescription = "Use whole foods in cooking like brown rice and whole wheat pasta."

TipTitle = "Diabetes Tip #2"

TipDescription = "Try to eat at least 8 grams of fiber per meal."

Expected outcome(s):

Tip title: Diabetes Tip #1

Use whole foods in cooking like brown rice and whole wheat pasta.

Tip title: Diabetes Tip #2

Try to eat at least 8 grams of fiber per meal.

Test Number: 09

Requirement being tested: RS018: Provide correct Glucose calculations based on User's data

Component being tested: GlucoseData

Method being tested: glucose

Test input(s) including command-line argument(s):

mgdl: 10, mmol: true

mgdl: 50, mmol: true

mgdl: 100, mmol: true

mgdl: 0, mmol: true

Expected outcome(s):

mgdl: 10, mmol: true, Glucose: .6

mgdl: 50, mmol: true, Glucose: 2.8

mgdl: 100, mmol: true, Glucose: 5.6

mgdl: 0, mmol: true, Glucose: .0

Test Number: 10

Requirement being tested: RS007: Export data to various formats and services

Component being tested: ReadingTools

Method being tested: testHourToSpinnerType

Test input(s) including command-line argument(s):

Hour: 24

Hour: 8

Hour: 4

Hour: 19

Expected outcome(s):

Hour: 24, SpinnerType: 8, Time of day: night

Hour: 8, SpinnerType: 1, Time of day: after breakfast

Hour: 4, SpinnerType: 8, Time of day: night

Hour: 19, SpinnerType: 4, Time of day: before dinner

Test Number: 16

Requirement being tested: HB1ACReading

Component being tested: HB1ACReading.java

Method being tested: setID, getId, setReading, getReading

Test input(s) including command-line argument(s):

Id = 1234

Id = 146789

Reading = 2.0

Reading = 10.0

Expected outcome(s):

Id returned = 1234

Id returned = 146789

Reading Returned = 2.0

Reading Returned = 10.0

Test Number: 17

Requirement being tested: GlucoseReading

Component being tested: GlucoseReading.java

Method being tested: setID, getId, setReading, getReading

Test input(s) including command-line argument(s):

Id = 1234

Id = 146789

Reading = 2.0

Reading = 10.0

Expected outcome(s):

Id returned = 1234

Id returned = 146789

Reading Returned = 2.0

Reading Returned = 10.0

Test Number: 18

Requirement being tested: A1cEstimate

Component being tested: A1cEstimate.java

Method being tested: setValue, getValue, setMonth, getMonth, Average

Test input(s) including command-line argument(s):

Value = 8.5

Month = june

Expected outcome(s):

value returned = 8.5

month returned = june

Test Number: 19

Requirement being tested: GlucosioConverter

Component being tested: GlucosioConverter.java

Method being tested: glucoseToMmolL, glucoseToA1C, a1cToGlucose, a1cNgspToIfcc, a1cIfccToNgsp

Test input(s) including command-line argument(s):

value = 60

value = 60

value = 6.0

value = 6.0

value = 64

Expected outcome(s):

value returned =3.3

value returned =~3

value returned =126

value returned =42

value returned =8

Test Number: 20

Requirement being tested: ReadingTools

Component being tested: ReadingTools.java

Method being tested: parseReading

Test input(s) including command-line argument(s):

value = 6

value = 1

value = 2.3

value = 8.1

Expected outcome(s):

str to number "6" = 6

str to number "1" = 1

str to number "2.3" = 2.3

str to number "8.1" = 8.1