

Deliverable #4

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

4 Deliverable 4

4.0 Documentation

During this deliverable, we utilized the same format for our test cases as mentioned in a previous section. We also added an input file for each test case to use. This input file can be edited to add additional inputs for one or all test cases and the output will automatically be updated.

4.1 Test Case Specifications

Test Number: 01

Requirement being tested: RS013 Set Reminders

Component being tested: Reminder.java

Method being tested: setMetric

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

0ml, 9ml, 10ml, 20ml

Expected outcome(s):

Metric returned = 0ml

Metric returned = 9ml

Metric returned = 10ml

Metric returned = 20ml

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):

10.0, 25.0, 50.0, 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

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

2.0, 10.0, 50.0, 100.0

Expected outcome(s):

Reading returned = 2.0

Reading returned = 10.0

Reading returned = 50.0

Reading returned = 100.0

Test Number: 04

Requirement being tested: RS001 Record user input - Body Weight

Component being tested: WeightReading.java

Method being tested: setReading

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

100.0, 50.0, 10.0 0.0

Expected outcome(s):

Reading returned = 100.0

Reading returned = 50.0

Reading returned = 10.0

Reading returned = 0.0

Test Number: 05

Requirement being tested: RS001 Record user input - Blood Pressure

Component being tested: PressureReading.java

Method being tested: setMinReading

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

110.0, 30.0, 15.0, 2.0

Expected outcome(s):

Min Reading returned = 110.0

Min Reading returned = 30.0

Min Reading returned = 15.0

Min Reading returned = 2.0

Test Number: 06

Requirement being tested: RS001 Record User Input - Cholesterol Level

Component being tested: CholesterolReading.java

Method being tested: setLDLReading

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

190.0, 50.0, 1.0, 0.0

Expected outcome(s):

LDL reading set to 190.0

LDL reading set to 50.0

LDL reading set to 1.0

LDL reading set to 0.0

Test Number: 07

Requirement being tested: RS017 Build User

Component being tested: User.java

Method being tested: setD_Type

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

1, 2

Expected outcome(s):

User's diabetes type set to Type 1

User's diabetes type set to Type 2

Test Number: 08

Requirement being tested: RS016 Provide Tips

Component being tested: ActionTip.java

Method being tested: set/getTipDescription

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

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

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

Expected outcome(s):

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

Tip: 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):

10, 50, 100, 0

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):

24, 8 , 4, 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: 11

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

Component being tested: InputFilterMinMax

Method being tested: isInRange

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

[5.0,8.0,10.0]

[1.0,0.0,100.0]

[4.0,5.0,6.0]

Expected outcome(s):

Range Tester [5.0,8.0,10.0] = false

Range Tester [1.0,0.0,100.0] = false

Range Tester [4.0,5.0,6.0] = false

Test Number: 12

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

Component being tested: GlucoseRanges

Method being tested: colorFromReading

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

69, 201, 75, 195, 100

Expected outcome(s):

reading: 69, color: "purple"

reading: 201, color: "red"

reading: 75, color: "orange"

reading: 195, color: "orange"

reading: 100, color: "orange"

Test Number: 13

Requirement being tested: RS019: Provide alerts

Component being tested: GlucosioNotificationManager

Method being tested: generateRandomNumber

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

[10,100], [50,500], [1,11], [0,90]

Expected outcome(s):

Number between the 10 and 100

Number between the 50 and 500

Number between the 1 and 11

Number between the 0 and 90

Test Number: 14

Requirement being tested: RS001: Record user input - Blood Pressure

Component being tested: PressureReading.java

Method being tested: setMaxReading

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

180.0, 100.0, 10.0

Expected outcome(s):

Max Reading returned = 180.0

Max Reading returned = 100.0

Max Reading returned = 10.0

Test Number: 15

Requirement being tested: RS001: Record user input - Blood Pressure

Component being tested: PressureReading.java

Method being tested: setId

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

1234, 146789

Expected outcome(s):

Id returned = 1234

Id returned = 146789

Test Number: 16

Requirement being tested: RS001: Record user input - HB1ACReading

Component being tested: HB1ACReading.java

Method being tested: setReading

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

2.0, 10.0, 15.0, 100.0

Expected outcome(s):

Reading Returned = 2.0

Reading Returned = 10.0

Reading Returned = 15.0

Reading Returned = 100.0

Test Number: 17

Requirement being tested: RS001: Record user input - Glucose

Component being tested: GlucoseReading.java

Method being tested: setReading, getReading

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

2.0, 10.0, 15.0, 100.0

Expected outcome(s):

Reading Returned = 2.0

Reading Returned = 10.0

Reading Returned = 15.0

Reading Returned = 100.0

Test Number: 18

Requirement being tested: RS001: Record user input - A1c reading

Component being tested: A1cEstimate.java

Method being tested: setMonth, getMonth

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

June, may, july, november

Expected outcome(s):

month returned = june

month returned = may

month returned = july

month returned = november

Test Number: 19

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

Component being tested: GlucoseConverter.java

Method being tested: glucoseToMmolL

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

60.0, 2.0, 10.0, 11.0

Expected outcome(s):

glucoseToMmolL: 3.33

glucoseToMmolL: 0.11

glucoseToMmolL: 0.56

glucoseToMmolL: 0.61

Test Number: 20

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

Component being tested: ReadingTools.java

Method being tested: parseReading

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

6, 1, 2.3, 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

Test Number: 21

Requirement being tested: RS004 - Display Graphically

Component being tested: DoubleGraphObject.java

Method being tested: setReading, get/setCreated

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

n/a

Expected outcome(s):

2018-11-28T11:05:27.951-05:00 10.0

Test Number: 22

Requirement being tested: RS004 - Display Graphically

Component being tested: IntGraphObject.java

Method being tested: setReading, get/setCreated

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

n/a

Expected outcome(s):

2018-11-28T11:05:28.222-05:00 10

Test Number: 23

Requirement being tested: RS003 - Provide a log of user inputs for the past days

Component being tested: SplitDateTime.java

Method being tested: SplitDateTime; DateFormat.getDateInstance

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

getMinute()

getHour()

getDay()

getMonth()

getYear()

Expected outcome(s):

Current DateTime

(05 11 28 11 2018 output when last tested)

Test Number: 24

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

Component being tested: NumberFormatUtils.java

Method being tested: getMinimumFractionDigits; getMaximumFractionDigits

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

n/a

Expected outcome(s):

0 3

Test Number: 25

Requirement being tested: RS017 Build User

Component being tested: User.java

Method being tested: setPreferred_unit_a1c

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

"mg/dL", "mmol/L"

Expected outcome(s):

User's A1c unit set to mg/dL

User's A1c unit set to mmol/L

4.2 How 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'

4.3 Evaluation and 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.