

# Final Report

Team Correct

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

---

## Ch. 1:

Project: Glucosio

Tests Completed within Android Studio:

1. GlucosioApplicationTest
  - a. 3 tests were included within this java file
    - i. ShouldNotClearLanguage\_WhenAlreadyDone
    - ii. ShouldClearLanguage\_WhenSetFromHelloActivityAndNotFixedYet
    - iii. ShouldSaveLanguageClearedToPreferences\_WhenItIsDone
  - b. All tests were passed and completed in 887 ms
2. RobolectricTest
  - a. Included a variety of test builds, however, if the option “All Tests” was chosen, the return stated that the test suite was empty
    - i. If each individual test was run, then the code worked correctly
  - b. A1CCalculatorActivityTest
    - i. ReturnTrue\_WhenKeyboardActionDone
    - ii. Notify Presenter\_WhenGlucoseValueChanged
    - iii. ShouldAskPresenterToCheckUnit\_When Created
    - iv. ShouldBindViews\_WhenCreated
    - v. ShouldSetMmolToMolAsUnit\_WhenUserSettingsIsPercentage
    - vi. ReturnFalse\_WhenKeyboardActionOther
    - vii. All tests were passed in 1m 7s
  - c. AssistantAdapterTest
    - i. CallPresenter\_WhenExportAsked
    - ii. CallPresenter\_WhenFeedbackAsked
    - iii. CallPresenter\_WhenA1CalculatorAsked

- iv. CallPresenter\_WhenAddReadingAsked
  - v. CorrectlyBindRows
  - vi. All tests passed 42s
- d. BackupActivityTest
  - i. ShouldDelegateToBackup\_WhenDisconnectIsCalled
  - ii. ShouldDelegateToBackup\_WhenConnectIsCalled
  - iii. ShouldDelegateToBackup\_WhenActivityResultRecieved
  - iv. ShouldInitBack\_WhenCreated
  - v. All tests passed 41s
- e. HelloActivityTest
  - i. ShouldReportAnalytics\_WhenCreated
  - ii. ShouldInitLanguageSpinner\_WhenCreated
  - iii. ShouldSelectLanguage\_WhenCreated
  - iv. ShouldPassNLAsLocale\_WhenNLSelected
  - v. ShouldBindView\_WhenCreated
  - vi. All tests passed 59s
- f. MainActivityTest
  - i. ShouldReportAnalytics\_WhenCreated
  - ii. Test passed 37s
- g. PreferencesActivityTest
  - i. ShouldReportAnalytics\_WhenCreated
  - ii. Test passed 38s
- h. AssistantAdapterTest
  - i. CallPresenter\_WhenExportAsked
  - ii. CallPresenter\_WhenFeedbackAsked
  - iii. CallPresenter\_WhenA1CCalculatorAsked
  - iv. CallPresenter\_WhenAddReadingAsked
  - v. CorrectlyBindRows
  - vi. All tests passed 39s

- i. BackupAdapterTest
  - i. ReturnViewWithSizeFormatted\_WhenAsked
  - ii. Test passed 30s
- j. MigrationTest
  - i. ProperlyMigrateUserFromVersion4
  - ii. ProperlyMigrateGlucoseFromVersion4
  - iii. ProperlyMigratePressureFromVersion4
  - iv. ProperlyMigrateWeightFromVersion4
  - v. ProperlymigrateCholesterolFromVersion4
  - vi. All tests passed 314 ms
- k. UserBuilderTest
  - i. Builds user account
  - ii. All tests passed 1s
- l. AssistantFragmentTest
  - i. ShouldUnBindViews\_WhenDestroyed
  - ii. ShouldBindViews\_WhenCreated
  - iii. Tests passed 36s
- m. A1CCalculatorPresenterTest
  - i. 9 tests passed 433ms
- n. AssistantPresenterTest
  - i. CallFragment\_WhenUserAskedExport
  - ii. CallFragment\_WhenUserAskedSupport
  - iii. CallFragment\_WhenUserAskedToAddReading
  - iv. CallFragment\_WhenUserAskedToShowCalculator
  - v. All passed 92s
- o. ExternalViewPresenterTest
  - i. shouldLoadOpenSourceLicenses\_WhenLicenseParameters
  - ii. shouldThrowException\_WhenNoParameters

- iii. shouldInvokeShowNoConnectionWarning\_WhenNetworkIsNotConnected
    - iv. All passed 28s
  - p. HelloPresenterTest
    - i. 9 tests passed 28s
  - q. OverviewPresenterTest
    - i. ShouldAddZerosBetweenReadings\_WhenAsked
    - ii. ShouldSortReadingsChronologically\_WhenAsked
    - iii. Tests passed 402ms
- 3. AndroidPhoneAppTests
  - a. Required running emulators for test;
    - i. We are using an android emulator
  - b. HelloActivityTest
  - c. MainActivity Test

#### Test Speeds:

We noticed that as the test were run across multiple systems, the speeds were inconsistent. We have determined that there is nothing significant to be worried about and that the difference in testing speeds is most likely a result of being run on a specific machine.

#### Organization:

The project is organized in the following categories for which these tests were focused on.

- Activity
- Adapter
- Db
- Fragment

- Presenter
- Tools

#### Experience:

Overall, the project is fairly straightforward and easy to follow, Glucosio included plenty of documentation on how to set up and test/run their source code. The tests were thorough and easy to run, and I appreciated the pass/fail description for each subunit test. The ability to run all of the tests will need to be fixed for future runs, as that will allow testing to be more efficient, but I did not mind running the tests on their own as they are organized and easy to find in the directory. Of our group, Daniel L. owns a phone running an Android OS, so we hope to test on there.

#### Project:

The project is fairly robust and well organized. The developer made the code and terminology easy to follow with great documentation. I am personally excited to continue to work on this project and contribute to make the overall application better. We were able to run the app on an Android emulator, and hope to test on a physical Android device in the future.

---

## Ch. 2:

### **Test Plan:**

#### **The Testing Process:**

- Our testing process for this project is very closely related to the Agile process, specifically Scrum practices. Our current Scrum process has included short sprint planning, sprints, and sprint review meetings over skype. We have been following this process throughout the development of this project and will continue to do so to finish. Our scrum master is Professor Leclerc, who is responsible for setting up our team and guiding us in the process. This process is currently working well for us, as we are able to meet our delivery deadlines. The

scrum process also works well with our Glucosio project as we continue to build and test our project this semester.

- Our major phases of our testing process so far have included each deliverable, which are focused on a specific portion of the final process.
  - Phase 0: Select 3 H/FOSS projects to present to class and ultimately discuss with Professor Leclerc to decide which one we would work on.
  - Phase 1: Clone project and work on existing tests within the system. Report on experience.
  - Phase 2: Produce a test plan and create 5 of the eventual 25 test cases.
  - Phase 3: Build an automated testing framework.
  - Phase 4: Finish the 25 test cases which will be tested using the automated testing framework previously built.
  - Phase 5: Insert 5 faults into code to ensure 5 tests fail and discuss experience.
  - Phase 6: Compile our final report and present our project.

### **Requirements Traceability:**

- The main requirements of Glucosio include:
  - User Input
  - Log of input
  - Graphics
  - Export of data
  - Ability to connect to other devices
  - Provide diabetes information
  - Reminders/Schedule
- We will be focusing our testing on more specific components of these requirement categories.

### **Tested Items:**

- The items we will be testing include:
  - Glucosio GitHub repository code

- Glucosio Android App Folder
  - Via Android Studio
- Glucosio Android App
  - Via Android emulator
- Glucosio iOS App
  - Via iPhone

### **Testing Schedule:**

Test cases 1-5 date: 10/13

Deliverable 2 date: 10/15

Automated Testing Framework: 11/7

Deliverable 3 date: 11/9

Test cases 6-25: 11/17

Deliverable 4 date: 11/19

Deliverable 5 and Final Report: 11/28

### **Test Recording Procedures:**

- Our tests will be recorded through shared Google Drive Documents to ensure all team members have access and each member is using the most up-to-date document.
- Our tests will also be uploaded to our Team Correct Github Repository.

### **Hardware and Software Requirements:**

- Operating System: Linux Ubuntu
- Software Requirements: Android Studio, VirtualBox
- Hardware Requirements: PC, iOS phone

### **Constraints:**

- Time: For this project, we have a limited timetable of a semester. We have explicit deliverables that will need to be completed in a timely manner.
- Staff: We have a team of 4 members which is small for a normal software team, however, for the scope of our project, we have an appropriate team size.
- Cost: We do not currently have an associated cost with our project.

- Risk: We will be using VirtualBox for our system testing and project, so if there were to be a problem with a virtual machine, then we may need to start the process over. This is not a huge risk due to our team backing up files to Google Drive and uploading work to GitHub.
- Resources: Our resources include our personal computers and the lab, which means this should not be a constraint in our current project scope.

### **System Tests:**

#### **Test Case 01**

<b>Test Suite ID</b>	TS001
<b>Test Case ID</b>	TC001
<b>Test Case Summary</b>	To verify that Share Anonymous Data For Research checkbox allows user to share their glucosio data with researchers
<b>Related Requirement</b>	RS005
<b>Prerequisites</b>	1) User has downloaded Glucosio app 2) User has entered data into app
<b>Test Procedure</b>	1) Open Settings within Glucosio app 2) Enter user data 3) Select 'Share anonymous data for research'
<b>Test Data</b>	N/a
<b>Expected Result</b>	1) 'Share anonymous data for research' should have a check next to it



<b>Actual Result</b>	1) If the checkbox is working correctly, a check will display 2) If the checkbox is not working correctly, an error message will display
<b>Status</b>	Pass
<b>Remarks</b>	N/a
<b>Created By</b>	Team Correct (Hannah P.)
<b>Date of Creation</b>	10/11/18
<b>Executed By</b>	Team Correct (Hannah P.)
<b>Date of Execution</b>	10/11/18
<b>Test Environment</b>	OS: iOS 12.0.1 App: Glucosio Application on iPhone

---

## **Test Case 02**

<b>Test Suite ID</b>	TS001
<b>Test Case ID</b>	TC002
<b>Test Case Summary</b>	To verify that the user data is properly displayed graphically
<b>Related Requirement</b>	RS004
<b>Prerequisites</b>	1) User has downloaded Glucosio app 2) User has entered readings into app
<b>Test Procedure</b>	1) Open Overview within Glucosio app 2) View day, week, month graphs
<b>Test Data</b>	At least one input of each type (Glucose, HbA1c, Ketones, Body Weight, Blood Pressure, and Cholesterol)
<b>Expected Result</b>	1) Readings are displayed correctly on graphs

<b>Actual Result</b>	1) Readings are displayed correctly 2) If there are no readings for a certain type, the message "No chart data available" will be displayed
<b>Status</b>	Pass
<b>Remarks</b>	How to automate?
<b>Created By</b>	Team Correct (Daniel L.)
<b>Date of Creation</b>	10/12/18
<b>Executed By</b>	Team Correct (Daniel L.)
<b>Date of Execution</b>	10/12/18
<b>Test Environment</b>	Emulator: Pixel API 28 OS: Android Version 9 (Pie) App: Glucosio Application on Android

---

### **Test Case 03**

<b>Test Suite ID</b>	TS001
<b>Test Case ID</b>	TC003
<b>Test Case Summary</b>	To verify that user data can be input properly
<b>Related Requirement</b>	RS001
<b>Prerequisites</b>	1) User has downloaded Glucosio app
<b>Test Procedure</b>	1) Open Overview within Glucosio app 2) Press + in lower right hand corner 3) Select input type 4) Complete input form
<b>Test Data</b>	Any kind of input

<b>Expected Result</b>	1) Readings are displayed in 'History'
<b>Actual Result</b>	1) Readings are displayed correctly
<b>Status</b>	Pass
<b>Remarks</b>	Can you delete readings?
<b>Created By</b>	Team Correct (Daniel L.)
<b>Date of Creation</b>	10/12/18
<b>Executed By</b>	Team Correct (Daniel L.)
<b>Date of Execution</b>	10/12/18
<b>Test Environment</b>	Emulator: Pixel API 28 OS: Android Version 9 (Pie) App: Glucosio Application on Android

---

#### **Test Case 04**

<b>Test Suite ID</b>	TS001
<b>Test Case ID</b>	TC004
<b>Test Case Summary</b>	To verify that the app successfully logs users past entries
<b>Related Requirement</b>	RS001
<b>Prerequisites</b>	1) User has downloaded Glucosio app 2) Users have previously entered data
<b>Test Procedure</b>	1) Open Overview tab within Glucosio app 2) Press the History Tab located next to the Overview tab 3) Verify past entries to be correct
<b>Test Data</b>	Past Data Points

<b>Expected Result</b>	1) Readings to be displayed in 'History'
<b>Actual Result</b>	1) Readings displayed correctly
<b>Status</b>	Pass
<b>Remarks</b>	Can you delete readings?
<b>Created By</b>	Team Correct (Alex T)
<b>Date of Creation</b>	10/12/18
<b>Executed By</b>	Team Correct (Alex T)
<b>Date of Execution</b>	10/12/18
<b>Test Environment</b>	Emulator: Pixel API 28 OS: Android Version 9 (Pie) App: Glucosio Application on Android

---

#### **Test Case 05**

<b>Test Suite ID</b>	TS001
<b>Test Case ID</b>	TC005
<b>Test Case Summary</b>	To verify that the app exports spreadsheet data effectively
<b>Related Requirement</b>	RS001
<b>Prerequisites</b>	1) User has downloaded Glucosio app 2) User has entered readings into app
<b>Test Procedure</b>	1) Open Settings within Glucosio app 2) Select "Export to CSV"
<b>Test Data</b>	Any kind of input
<b>Expected Result</b>	1) User inputs and calculated readings are converted to a

	.csv file properly
<b>Actual Result</b>	1) Charts are converted to zip file 2) 11 individual .csv files for each reading are produced within the zip file
<b>Status</b>	Pass
<b>Remarks</b>	Composite csv containing all chart readings?
<b>Created By</b>	Team Correct (Daniel Baczmaga)
<b>Date of Creation</b>	10/12/18
<b>Executed By</b>	Team Correct (Daniel Baczmaga)
<b>Date of Execution</b>	10/12/18
<b>Test Environment</b>	OS: iOS 12.0.1 App: Glucosio Application on iPhone