The Testing Process:

The bulk of our testing will occur inside of the `org.glucosio.android.tools` package. The classes therein act as tools and utilities that assist with other, larger sections of the product. Only about a third of the tools are comprehensively tested, so we should have our work cut out for us.
Our first step will be to identify the tools which do not have tests written for them. They are as follows:

1. `AlgorithmUtil`
2. `AnimationTools`
3. `FormatDateTime`
4. `GlucosioAlarmManager`
5. `GlucosioConverter`
6. `GlucosioNotificationManager`
7. `LabelledSpinner`
8. `NotDismissableEditText`
9. `RealmBackupRestore`
10. `SplitDateTime`
11. `TipsManager`

Some of these, such as `LabelledSpinner`, are composed of primarily getters and setters; not exactly things worth testing, within the bounds of this project. Others, like `FormatDateTime`, contain methods that return formatted results, which would require tests to ensure valid output. Those will be our focus.
Deadlines:

- The testing framework is to be completed by October 31st, 2017 (Deliverable #3).
- Twenty-five test cases must be implemented in the framework by November 14th (Deliverable #4).
- A report on five designed faults must be completed by November 21st (Deliverable #5).
- A presentation on the testing process will be given on November 28th, acting as the final deadline for testing (Final Report).

Tested Items:

Test Case Specification Template

- Test Number or ID - Number of the specific test case to be tested.
- Requirement being Tested - The condition being tested
- Component being Tested - A class within the application will be chosen for a method it possesses for testing.
- Method being Tested - A procedure within the class will be selected for the development and implementation of the test cases.
- Test Input(s) Including Command Line Arguments - This will include the parameters necessary to test the specified requirement, and the command to run this test from the command line.
- Expected Outcome(s) - This will specify the conditions necessary to determine whether the test has passed, after being run.

By the end of the testing period, we will have developed a total of twenty-five test cases. To date, we have the following five test specifications:

1.
   - **Test Number or ID** - 1
   - **Requirement Being Tested** - Tests that if the provided data's trend is above the up trend limit, an "up" `TrendArrow` is returned.
   - **Component Being Tested** - `org.glucosio.android.tools.AlgorithmUtil`
   - **Method Being Tested** - `public static TrendArrow getTrendArrow(Context context, GlucoseData data)`
   - **Test Input(s) Including Command Line Arguments** - `context = null`, `data = GlucoseData` where `GlucoseData.trend = TREND_UP_DOWN_LIMIT + 1`
   - **Expected Outcome(s)** - `TrendArrow.UP`
2.
   - **Test Number or ID** - 2
   - **Requirement Being Tested** - Tests that if the provided data's trend is below the down trend limit, a "down" `TrendArrow` is returned.
   - **Component Being Tested** - `org.glucosio.android.tools.AlgorithmUtil`
   - **Method Being Tested** - `public static TrendArrow getTrendArrow(Context context, GlucoseData data)`
   - **Test Input(s) Including Command Line Arguments** - `context = null`, `data = GlucoseData` where `GlucoseData.trend = -TREND_UP_DOWN_LIMIT - 1`
   - **Expected Outcome(s)** - `TrendArrow.DOWN`
3.
   - **Test Number or ID** - 3

- **Requirement Being Tested** - Tests that if the provided data's trend is above the down trend limit, but below the slightly down trend limit, a "slightly down" `TrendArrow` is returned.
- **Component Being Tested** - `org.glucosio.android.tools.AlgorithmUtil`
- **Method Being Tested** - `public static TrendArrow getTrendArrow(Context context, GlucoseData data)`
- **Test Input(s) Including Command Line Arguments** - `context = null`, `data = GlucoseData` where `GlucoseData.trend = -TREND_UP_DOWN_LIMIT + (-TREND_UP_DOWN_LIMIT - -TREND_SLIGHT_UP_DOWN_LIMIT)/2`
- **Expected Outcome(s)** - `TrendArrow.SLIGHTLY_DOWN`

4.

- **Test Number or ID** - 4
- **Requirement Being Tested** - Tests that if the provided data's trend is below the up trend limit, but above the slightly up trend limit, a "slightly up" `TrendArrow` is returned.
- **Component Being Tested** - `org.glucosio.android.tools.AlgorithmUtil`
- **Method Being Tested** - `public static TrendArrow getTrendArrow(Context context, GlucoseData data)`
- **Test Input(s) Including Command Line Arguments** - `context = null`, `data = GlucoseData` where `GlucoseData.trend = TREND_UP_DOWN_LIMIT - (TREND_UP_DOWN_LIMIT - TREND_SLIGHT_UP_DOWN_LIMIT)/2`
- **Expected Outcome(s)** - `TrendArrow.SLIGHTLY_UP`

5.

- **Test Number or ID** - 5
- **Requirement Being Tested** - Tests that if the provided data's trend is equal to the `TREND_SLIGHT_UP_DOWN_LIMIT`, a "flat" arrow is returned.
- **Component Being Tested** - `org.glucosio.android.tools.AlgorithmUtil`
- **Method Being Tested** - `public static TrendArrow getTrendArrow(Context context, GlucoseData data)`
- **Test Input(s) Including Command Line Arguments** - `context = null`, `data = GlucoseData` where `GlucoseData.trend = TREND_SLIGHT_UP_DOWN_LIMIT`
- **Expected Outcome(s)** - `TrendArrow.FLAT`

Test Recorded Procedures:

A test report will be generated and automatically displayed in the system default browser upon the completion of "runAllTests".

Hardware and Software Requirements:

The system will need to be running a UNIX-like operating system, and will need to have Java installed. The app itself requires the Android platform of at least version 4.1. The app is edited and tested in Android Studio.

Constraints:

Jake will be unavailable from October 24th through the 28th, conflicting with the period of time leading up to Deliverable #3, which means we will need to put in most of the group work required for that deliverable prior to October 24th.

Immediately following Deliverable #3, Tyler will be unavailable from November 1st through the 5th, limiting the amount of time we have to meet concerning Deliverable #4 (due November 14th). It would be wise of us to develop a plan of action before then, so when Tyler returns, we are able to hit the ground running.

System Tests:

System tests are out of the scope of this plan, as it is restricted to unit testing.