Chapter 2: Detailed Test Plan

Updates since Deliverable 1:

Since **Deliverable 1**, we have scrapped the plan of building the entire desktop environment for `sugar` and instead have shifted our efforts to testing individual `Sugar Activities`. The first activity we have selected to test is the `calculate-activity`.

Summary:

We are drafting an initial test plan for individual `Activities` from `sugar` due to dependency issues with required packages. Our initial tests for the `calculate-activity` are focused on input validation and making sure that there are no errors that arise with specific inputs into the `add()` function of the python class.

Test Plan:

Process:

The tests for this software project will utilize the `bash` command line interface and using the `python` interpreter that `sugar` uses to run the `calculate-activity`. The `calculate-activity` consists of many methods that are used to compute results that the user would like to compute.

Requirements Traceability:

The requirements of a calculator with all its basic functions, for these cases specifically the addition function, further more clarify that we want to be able to do addition using all real numbers with the test cases testing natural numbers, whole numbers, integers, rational, and real numbers.

Tested Items:

The following functions are set to be tested with the possibility of more functions being added at a later date:

1. add(x, y)
2. sub(x, y)
3. mul(x, y)
4. div(x, y)
5. square(x)

Testing Schedule:

The following is our testing schedule:

Date	Deliverable Number	Task
Nov. 5	3	Design an automated testing framework
Nov. 17	4	Create 25 test cases and run them with our framework
Nov. 24	5	Create repeatable faults in the application and identify tests that are planned to fail

Hardware and Software Requirements:

Hardware: physical or virtual machine with appropriate resources that can run a Linux Distribution

Software: a functional terminal emulator with the `bash` environment, `python` interpreter, and the ability to install dependencies if needed through a package manager

Constraints:

The classes and methods must not rely upon the `python-sugar3` package or library, or any other broken dependency or package

Systems Tests:

Please see the chart below with the initial test cases for the `calculate-activity`

Test Cases:

Test ID	Requirement	Component	Method	Test Inputs	Expected Outcomes

Test ID	Requirement	Component	Method	Test Inputs	Expected Outcomes
1	Addition of Two Numbers	`functions.py`	add(x, y)	(1,2)	No errors expected with natural numbers, result should be returned as **3**
2	Addition of Two Numbers	`functions.py`	add(x, y)	(-1,1)	No errors expected with integers, result should be returned as**0**
3	Addition of Two Numbers	`functions.py`	add(x, y)	(1.1,2.1)	No errors expected with rational numbers due to casting with the `_d` function, result should be returned as**3.2**
4	Addition of Two Numbers	`functions.py`	add(x, y)	(pi,sqrt(2))	No errors should occur with irrational numbers, the result should be returned as **pi + sqrt(2)**
5	Addition of Two Numbers	`functions.py`	add(x, y)	(sys.maxsize,2)	**Note: the included `sys` library is needed to get the maximum integer** No errors occur when invoking the maximum size integer added with 2, the result will vary based on the system architecture and will add 2 with no overflow error