Fantastic-Four_deliverable3.md

Chapter 3: Initial Test Automation

Executive Summary:

Our automated testing framework of the `sugarlabs calculate-activity` is 25% complete with five test cases developed to test the `add(x, y)` function within the activity. The test cases follow the table below:

Test Cases:

Test ID	Requirement	Component	Method	Test Inputs	Expected Outcomes	Text file
1	Addition of Two Numbers	`functions.py`	add(x, y)	(1,2)	No errors expected with natural numbers, result should be returned as **3**	Test Case 1
2	Addition of Two Numbers	`functions.py`	add(x, y)	(-1,1)	No errors expected with integers, result should be returned as **0**	Test Case 2
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**	Test Case 3
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 `math.pi + math.sqrt(2)`	Test Case 4
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	Test Case 5

We will look to add in the following methods:

- `sub(x, y)`
- `mul(x, y)`
- `div(x, y)`
- Additional method TBD

Technical Summary:

Directory Descriptions:

1. `docs` : Home to the documentation for the project as well as a `README` for running the project
2. `oracles` : Home to the outputs of our test cases
3. `project` : Home to the `sugar-activity` code that we will be testing
4. `reports` : Home to all the reports generated by our test cases

5. `testCases` : Home of all the test case files with the following format:

- [Test Suite ID]
- [Test Case ID]
- [Requirement]
- [Driver]
- [Component]
- [Method being tested]
- [Inputs (comma separated)]
- [Expected Oracle]

6. `testCaseExecutables` : Home to the drivers for the methods being tested for low coupling and high cohesion in the testing framework.

Instructions on how to run our Testing Framework

Prerequisites:

1. A Linux based system with the `bash` command line interface
2. Python3 interpreter
3. The `git` program is installed on the system

Setting up the environment:

1. On the linux system, open a terminal emulator
2. Clone our repository with the following command: `git clone https://github.com/csci-362-01-2020/Fantastic-Four.git`
3. Navigate to the propper directory: `cd Fantastic-Four/TestAutomation`
4. Run the test driver: `./scripts/runAllTests.py`