SOFTWARE ENGINEERING II
TEAM 5.
ASSIGNMENT 3.
SEPTEMBER 27TH, 2017

Executive Summary

This document contains all information required for COSC 4345 (Software Engineering II) project for Team 5. The project will be based on a Test Driven Development Each approach in which project requirements/feature requests are turned into very specific test cases and then the software is constructed to pass the test cases.

- Features running list: As the project evolves (e.g., version 1.0, 2.0, 3.0) the running features list will grow. The running features list is a way to keep track of the features that are implemented and are to be implemented.
- Priorities: The highest priority items are those that offer new or significant enhanced functionality or performance. Lower priority items are items that are improvements that do not significantly change or improve functionality.
- Milestones: throughout the project, features will be mapped to individual milestones.

Terminology

- SUT: System Under Test
- API: Application Program Interface
- CC: Command and Control
- M: Milestone

Description of Priorities

- P0 - Must have
- P1 - Should have
- P2 - Nice to have

Document Reviewer/Approval

Role	Team Member
Java	Christopher Inzana
Python	Paloma Samaniego
Web UI/PHP	Anthony Ofili
MYSQL/PHP	Abdulaziz Alibrahim

Initial Requirements

ID	Description
1	SUT requests tests from server
2	SUT runs test
3	SUT returns success/fail
4	SUT must test storage

Initial Features

Naming convention: Milestone, followed by requirement mapping and feature number (ie. feature 2.1.3 corresponds to M #2, requirement #1, and feature #3).

Implementation date: features not yet implemented will have a blank date.

	ID	Description	Priority	Owner	Date
M2	1.1.1	Create SUT Java Controller to manage all communication with the server to determine which test to run, when to run it and log results.	P0	Christopher	
	1.1.2	Tests should only interact with the Java client test controller. In other words, tests should never connect to the web server.	P0	Christopher	
	1.1.3	PHP web server should set test frequency and allow test client to run test and post results.	P1	Abdulaziz	
M3	2.1.1	The Java Controller is capable of fetching a test from the server, executing the test and returning a SUCCESS/FAIL result back to the server	P0	Christopher	
	2.2.1	Tests must be written in Python to run on Windows or Mac OS. Each test must have a run() method and return SUCCESS/FAIL.	P0	Paloma	
	2.2.2	There should be at least 5 Python tests for any of the subsystems of: memory, storage, networking, CPU, math operations and video.	P0	Paloma	
	2.2.3	Test multiple machines simultaneously	P1	Anthony	

	2.2.4	Automate the script execution	P1	Anthony	
	3.3.1	Test results must be logged to a MySQL database with either SUCCESS or FAIL with a PHP web service. Also log tests, system under test, and frequency, and other attributes.	P1	Abdulaziz	
	3.3.2	Web UI to report and aggregate test results. There must be one or more web pages to display test clients, results system details, and history for the managers and the engineers.	P0	Anthony	
	3.3.3	Create a database table for test with parameters (testId PR, testDesc, TestScriptName, testDailyFrequincyCount)	P0	Abdulaziz	
	3.3.4	Create a database table for test result with parameters (testResultId PK, testId FK, testResult, testStart DateTime, testEnd DateTime, sutId FK)	P0	Abdulaziz	
	3.3.5	Create a database table for SUT with parameters (sutId PK, sutOS, sutDescription, sutHardware)	P0	Abdulaziz	
M4	3.2.1	All python tests implemented. At least one test per subsystem.	P0	Paloma	
	3.2.2	Create test cases for database tables	P0	Abdulaziz	