Software Requirements Specification

Introduction

The application is an autograder that will take in a folder of zipped files from each student for a specified assignment.This application will be used for an introductory course to Python to check the basic function of the program and syntax of the language. There will be a grade for each student and an overall average for the class.

Purpose

This SRS document is used to give an overview of the software. We will detail the intended use for the software along with requirements for our clients to run and utilize the software. This document will assist in future software delivery lifecycle (SDLC) processes.

Scope

This program will be run by the professor of the course, therefore will be limited to the files in that professor's local machine. Files outside of the unzipped file for an individual student cannot be accessed and will be not considered in final grading. All output files containing individual grades and overall average will be compiled into a text file for professor to reference.

Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project and the functional and data requirements of the system. Functionality gives the functional requirements, data requirements and constraints and assumptions made while designing the autograder. It also gives the user viewpoint of product. This section also gives the specific requirements of the system. Functionality also discusses the external interface requirements and gives a detailed description of functional requirements.

Overall Description

This document will give an overview of how the autograder works. The system will prompt the user through the command line throughout the whole process of attempting to grade each assignment for each student. Criteria will be graded based on user input for specifications.

Functionality

Provide analysis of criteria met

1. The system shall output the criteria which was met during grading.
2. The system shall indicate where the user failed.
3. The system will cache the results of the assignment for overall statistical use.
4. The system will analyze overall students' performance and output the class average to a file.

Grade specific assignment based on criteria

1. The system shall grade a specific assignment.
2. The system will parse the whole python script.
3. The system will evaluate each criteria to see if it is met.

Unzip mass directories

1. The system shall unzip a large directory containing folders for each student.
2. The system will then unzip each folder and run the evaluation of criteria.

Criteria is based on but not limited to

1. The system will check for sufficient comments
2. The system will check for function declarations/identifier names
3. The system will check for fork bombs
4. The system will check if the python script compilation success
5. The system may check # of lines of code
6. The system will check for redundancies (i.e. if x == True)

Command line interface

1. The system will be entirely command line interface prompted for user.
2. The system will require the user to pass in a directory of zipped folders (student assignments)
3. The system will prompt the user through every step.

Accessibility

1. The system will be limited to faculty and teaching assistant use only.