Web Services

Assignment 1

**Date given:** 21/02/2019

**Date due:** 24/3/2019

**Grade weight:** 35%

**Description**

A company has contacted you to develop a REST interface for an existing weather station. The company has asked you to build the application on top of Flask, RabbitMQ and Python XML RPC.

The company has mentioned that they do not have a complete Python implementation and have asked that some components be developed using PHP to send information into the RabbitMQ queue.

**Assignment Specifics**

* Flask should be used to generate the web endpoint for the service. This endpoint should have three different defined URLs:
  + /justweather which is responsible for gathering JSON content from the <http://kylegoslin1.pythonanywhere.com/> address. The forecast should be extracted and returned back in JSON form.
  + /updates which is responsible for reading the updates.txt file that contains a list of all the times the service has been updated. This content is in plain text and should be returned back in complete JSON format.
* The company has a number of different clients who are going to interact with the service. Many of these users run local weather stations and need a method to send information into the service. Due to the cross-language capabilities of RabbitMQ, a local server should be created which can queue weather updates send in by clients. When the /read URL is called on the local web server, the queue of messages in the RabbitMQ server should be read and returned back to the caller in JSON format.
* The client has mentioned that they currently do not have any infrastructure for running Python and have suggested to you that PHP should be used to publish content to the queue. (Link: <https://www.rabbitmq.com/tutorials/tutorial-one-php.html>). The information that should be sent to the server is a single text string that contains the information “weather-rainy”.
* The client has mentioned to you that they would like a RPC interface developed which allows a client to call a function on the server. This server should allow the client to pass a variable containing the current temperature. If the temperature is between 0 and 10 the service should return back “cold” if the temperature is between 11 and 20 the service should return back warm.
* To allow the RPC service to be called, the RPC client should be called from the /callClient URL which should accept a single parameter called “temp”.
* To keep track of all the different calls that are made to the server, a log file should be created called “calls.log”. Whenever a request comes into the Flask server, a new record should be printed into the file, containing the time/date and the name of the function that was called.
* A URL should be added into the application that allows a user to specify who is currently using the service. This URL should be specified as /insert which accepts, the following parameters: *firstname, lastname, studentnumber*. This information should be saved into a text file called “users.log” along with the current time stamp.
* Every five minutes, a call should be made to the /ping URL. This URL will respond with the word “pong” followed by the current time stamp. A Cron / task schedular based utility should be used to automate this process.

**A-Grade**

* To achieve an A grade for this assignment, additional functionality should be included in the application in the form of HProse which should be used to call a function in Python from a JavaScript based application. The function needs to be called “ping” and the function should return back the machines IP address.

**Tips**

* To aid the development process, use Postman to emulate POST/GET requests.

**Deliverables**

* Python code as a single zip file uploaded to the “Assignment 1” uploader on the Moodle course page.
* Due to the number of interactions that the user will have with different environment specific setups, a demo will be held in class for a demo of the application you have developed along with the grading process.

**References / Further reading**

* <https://docs.python.org/3/library/xmlrpc.client.html>
* <https://www.rabbitmq.com/tutorials/tutorial-one-php.html>