Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: CMT 202
Module Title: Distributed and Cloud Computing
Lecturer: Padraig Corcoran
Assessment Title: CMT 202 Coursework 2
Assessment Number: 2
Date Set: Tuesday 12 March 2019.
Submission Date and Time: Tuesday 2 April 2019 at 9:30am.
Return Date: by Tuesday 23 April 2019.

This assignment is worth 15 % of the total marks available for this module. The penalty for late or non-submission is an award of zero marks.

Your submission must include the official Coursework Submission Cover sheet, which can be found here:

https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

All submission should be via Learning Central unless agreed in advance with the Director of Teaching.

Description		Type	Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
Solutions	**Compulsory**	One zip (.zip) file containing all Python code developed. This zip file should not contain any subdirectories.	[student number].zip

Any deviation from the submission instructions above (including the number and types of files submitted) may result in a mark of zero for the assessment or question part.

Assignment

You have been hired by an estate agency to build a distributed data storage system using a remote object paradigm that will allow their employees to access, store and update information relating to property (houses/apartments).

Each property has the following associated pieces of information which must be stored in the system:
1. Property unique ID (this should be an integer value).
2. Property owner.
3. Property postcode.
4. Property number.
5. Property year of construction.
6. Property status (currently for sale or has already been sold; this is a binary indicator).

Design and implement the above distributed data storage system using a remote object paradigm. The class name should be *estate_agency*; that is, use the expression *class estate_agency(object)* when defining the class. This class should be defined in a Python file entitled *estate_agency.py*.

The system should allow employees of the estate agency to perform the following seven tasks:
1. Display all six pieces of information relating to the set of properties currently stored. Note, this should be displayed in the client. Implement this using a method in the class estate_agency with the following header:
def display_ properties(self).

2. Add a property to the set of properties currently stored; a newly added property should have a status of for sale. The property unique ID should be returned after completion of this task. Implement this using a method in the class estate_agency with the following header:
def add_property(self, owner, postcode, number, year).

3. Display the subset of properties currently stored which have a construction date within a specified inclusive year range (e.g. 2000-2010). Implement this using a method in the class estate_agency with the following header:
def select_by_year(self, start_year, end_year).

4. Display the subset of properties currently stored which have a specified postcode. Note, this should be displayed in the client. Implement this using a method in the class estate_agency with the following header:
def select_by_postcode(self, postcode).

5. Set the status of all properties with a specified postcode and number to for sale. Implement this using a method in the class estate_agency with the following header:
def set_for_sale(self, postcode, number).

6. Set the status of all properties with a specified postcode and number to not for sale. Implement this using a method in the class estate_agency with the following header:
def set_not_sale(self, postcode, number).

7. Display all six pieces of information relating to the set of properties currently sorted by year of construction Note, this should be displayed in the client. Implement this using a method in the class estate_agency with the following header:
def display_properties_sorted(self).

The remote object created should be assigned the logical name *object.agency* and be registered with a name server.

To successfully complete all tasks, you are only required to stored data while your code is running; there is no requirement to write data to an external database or file.

Learning Outcomes Assessed

The following learning outcomes from the module description are specifically being assessed in this assignment.
Demonstrate and apply knowledge about the state-of-the-art in distributed-systems architectures.
Appreciate the difference between various distributed computing middleware and their communication mechanisms.

Criteria for assessment

Credit will be awarded against the following criteria.

Marks will be assigned to each of the seven tasks specified above as follows:
Successfully implement task 1 specified above. [2 marks]
Successfully implement task 2 specified above. [2 marks]
Successfully implement task 3 specified above. [2 marks]
Successfully implement task 4 specified above. [2 marks]
Successfully implement task 5 specified above. [2 marks]
Successfully implement task 6 specified above. [1 marks]
Successfully implement task 7 specified above. [2 marks]
Using the logical name *object.agency* and registering with the name server. [2 mark]

Feedback on your performance will address each of these criteria.

A student can expect to receive a distinction (70-100%) if they correctly implement all tasks.
A student can expect to receive a merit (60-69%) if they correctly implement most tasks without major errors.

A student can expect to receive a pass (50-59%) if they correctly implement some tasks without major errors.
A student can expect to receive a fail (0-50%) if they fail to correctly implement some tasks without major errors.

IMPORTANT – All code submitted must be written in Python and run on the computers in the teaching lab.

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria. Feedback and marks will be returned on Tuesday 23 April 2019. via Learning Central. Where requested, this will be supplemented with oral feedback.

Feedback from this assignment will be useful for the second coursework in this module.