School of Computing: COMP3211 Coursework 2

Module title	Distributed Systems
Module code	COMP3211
Assignment title	Coursework 2
Assignment type and description	In-course Assessment
Rationale	The coursework demonstrates that you can discuss, reason and construct practical outcomes from learned knowledge. You need to review the material covered in the lectures and lab. sessions in week 4-6.
Word limit and guidance	-
Weighting	25%
Submission deadline	24/11/2023, 10:00
Submission method	Gradescope
Feedback provision	15/12/2023 on Gradescope.
Learning outcomes assessed	1. Design a high-level framework of a distributed system based on an advanced architecture. 2. Use a range of middleware tools to implement a distributed design.
Module lead	Karim Djemame
Other Staff contact	-

1. **Assignment guidance** This exercise aims to give you some practical experience of using serverless architectures. It will also give you the opportunity to research on the cloud computing domain of Function as a Service (FaaS), technologies and performance.

   Serverless computing is revolutionising cloud application development as it offers the ability to create modular, highly-scalable, fault-tolerant applications, with minimal operational management. In order to contribute to its widespread adoption of serverless platforms, the performance of FaaS serverless platforms is key. This coursework aims to develop a simple Internet of Things (IoT) simulation Framework for data acquisition and statistics as well as investigate its performance, see Figure 1. The IoT's real power today lies in harnessing data from diverse sources and converting it into actionable insights.

   You should review the lecture notes on the serverless architectures topic carefully before you start work.

2. **Assessment tasks**

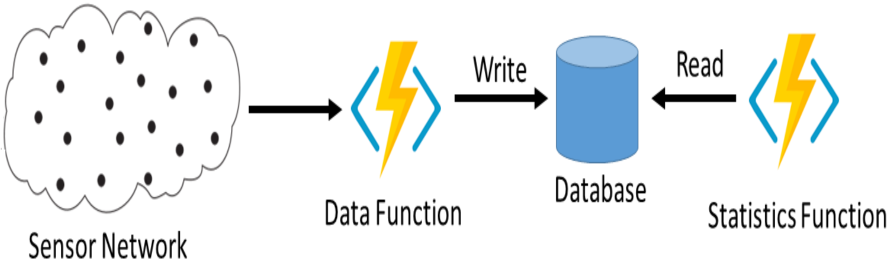

Figure 1: IoT Framework Components

Task 1: Simulated Data The framework collects environmental data from 20 sensors in the city of Leeds. This data includes the following weather information: sensor ID, temperature (in Celsius), wind (miles/hour), relative humidity (%), and CO2 level (ppm). All the sensors are simulated using code running in an Azure Function to achieve high scalability whilst simplifying deployment. This code mimics the data that would be collected at regular intervals from sensors placed throughout the city, measuring all the types of data mentioned above at the same time before it is stored in a database. The framework also incorporates random variations to ensure that the data mirrors real-world sensor data as closely as possible. A snapshot of the data generated by the sensors at a given time is:

Sensor ID	**Temperature**	**Wind**	**R.Humidity**	**CO2**
1	12	20	45	750
2	13	19	50	600
...				
20	11	21	60	1000

Data	**Range**
Temperature	8 .. 15
Wind speed	15 .. 25
Relative Humidity	40 .. 70
CO2	500 .. 1500

Implement the simulated data serverless function using a language runtime of your choice, e.g. Python, Java. Node.js. Demonstrate its performance in terms of scalability of the data. Generate a graph.

Task 2: Statistics This simulated data is subsequently processed through the analytical platform thanks to a second serverless function, where we can track patterns and carry out all kinds of evaluation to identify potential issues such as heatwaves or high pollution levels. The role of this function is to output the minimum, maximum and average of the data *per sensor*.

Task 3: Realistic Scenario Propose a technical solution for implementing the following realistic scenario:

(a) the data should be collected from sensors at a regular interval T, for example every 5 seconds.

(b) Once the new collected data is stored in the database by the first function (simulated data), this will automatically trigger the second function (statistics) thanks to a dabase change tracking mechanism.

(c) the second function should output the same results as in Task 2.

Notes:

- You should consider creating an *Azure SQL database* as part of the implementation, see details on MS Teams;
- You should consider using the Azure SQL trigger for Functions, see *Azure SQL trigger for Functions* in Task 3, see documentation and example.

3. **General guidance and study support** Learning resources and useful links are available in COMP3211 area on Minerva. Help is also available on the module Teams channel.

4. **Assessment criteria and marking process**

   The coursework will be marked out of 50. Coursework feedback and return of marks will be available on *Gradescope.*

5. **Presentation and referencing**

   The quality of written English will be assessed in this work. As a minimum, you must ensure:

   - Paragraphs are used
   - There are links between and within paragraphs although these may be ineffective at times
   - There are (at least) attempts at referencing
   - Word choice and grammar do not seriously undermine the meaning and comprehensibility of the argument
   - Word choice and grammar are generally appropriate to an academic text

6. **Submission requirements**

- **Questions**: address the questions in relation the implementation, discussion of results, and evidence of execution on *Gradescope*.
- **Code**: submit your code for the exercise. You can either: 1) provide a link to Git, or 2) create a Zip or tar archive of the files which make up your system and upload it on *Gradescope*. If you have organised your files into a directory hierarchy, then please package this as a single Zip or tar archive.
- **Demonstration**: produce a short video (maximum 2 minutes long) to demonstrate your results. You can either: 1) upload it on *Gradescope*, or 2) upload on any cloud platform of your choice, e.g. Youtube and provide the link.

7. **Academic misconduct and plagiarism**

   Academic integrity means engaging in good academic practice. This involves essential academic skills, such as keeping track of where you find ideas and information and referencing these accurately in your work. By submitting this assignment you are confirming that the work is a true expression of your own work and ideas and that you have given credit to others where their work has contributed to yours.

8. **Assessment/marking criteria grid**

Task 1	15
Task 2	10
Task 3	15
Code/scripts	5
Video	5
	50

A lateness submision penalty will apply (5% of the maximum available mark per day).