School of Computer Science: Assessment Brief

Module title	Distributed Systems
Module code	XJCO3211
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	-
Use of GenAI in this assessment	RED: AI tools cannot be used.You must not use GenAI tools. The purpose and format of the assessments makes it inappropriate or impractical for AI tools to be used
Weighting	25%
Submission deadline	20/11/2025, 10:00
Submission method	Gradescope
Feedback provision	11/12/2025 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 functions, including *workflows*, running on such platforms is key. This coursework therefore aims to investigate the performance of workflow functions.

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

2. **Assessment tasks**

   (a) For the investigation, use *Microsoft Azure Functions* serverless platform.

   (b) Research into *serverless workflows* deployed on serverless platforms to provide insight into their capabilities and increase awareness of their potential;

   (c) Formulate and design a serverless workflow of your choice. The workflow should consist of *two* functions. You are free to implement more than two functions if your solution requires it.

   (d) Specify a cloud-based technical implementation for analysis of your serverless workflow, using one language runtime of your choice. You can choose, for example, among Python, Java and Node.js.;

   (e) Demonstrate the performance of the workflow in terms of *runtime* and *resource consumption*, e.g. CPU and memory utilisation.

   Requirements

   (a) Choose an application of a serverless workflow, e.g. image processing pipeline, machine learning, Internet of Things (IoT) etc.

   (b) An important aspect of the implementation of the serverless functions is *how they are triggered*, see lecture slides and examples on

the Azure portal. One useful mechanism is the *Azure SQL trigger for Functions*, see documentation. You may use the *Azure SQL database* as part of the implementation;

(c) For the evaluation, investigate the aspect of scalability by increasing the number of function invocations;

(d) You need to demonstrate your *ability to write code* to implement your own serverless functions.

(e) Report your performance results.

Example: An Internet of Things (IoT) framework collects environmental data, e.g. temperature, CO2 level etc from N sensors, which are simulated using code running in a *first* Azure Function to achieve high scalability whilst simplifying deployment. This code mimics the data that would be collected at *regular intervals* from sensors before it is stored in a database. Once the new collected data is stored in the database by the first function (simulated data), this will automatically trigger the *second* function thanks to a dabase change tracking mechanism. The role of the second function is to display various statistcis, e.g. output the minimum, maximum and average of the data. The aspect of performance relates to the number of sensors N as well as the volume of data generated and processed over time.

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 to 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**

Workflow originality	5
Solution Design	10
Implementation	15
Evaluation	10
Code/scripts	5
Video	5
	50

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