Miniproject Part 1 - Agile Cloud Automation (CO4217/CO7217)

1. Introduction

In this miniproject, you are tasked with creating a rapid prototype that utilises a specified dataset, processes it, and stores the data in a MongoDB Cloud instance via **MongoDB ATLAS**. Subsequently, the application will extract business intelligence from the stored data, employing both Groovy collection operations and the Java MongoDB API.

The objective of this exercise is to evaluate and contrast the standalone solution (exercise 2) and the cloud-based solution (exercise 3) along a chosen dimension: data evolution, scalability, or consistency. This task serves as an extension of the programming exercises encountered in the first part of this module, allowing you to integrate and apply the theoretical concepts in a practical setting.

2. Note on Academic Integrity

As a student, it is important to understand what good academic practice is and what academic integrity means. Go to Academic Skills Centre > Academic Integrity for further information.

Generative AI may be used in this assessment

AI can assist in the development phase. All AI-generated content must be documented appropriately (e.g., by explaining the key prompts you use and how the AI outputs were reused/adapted). Final submissions must be the student's original work.

Guidance on Plagiarism and Collusion

This is a **group project**. Plagiarism, including collusion, across groups is penalised. By submitting your solution, you are stating that you are aware of the consequences and that the solution provided for the worksheet is the result of your **sole work as a group**. For more information on plagiarism visit the link above on Academic Integrity.

3. Submission procedure

The submission link can be found under `Assessment and Feedback` on Blackboard, where you will find:

1. GROUP PROGRESS FORM: form to track progress of group contributions, task allocation and teamwork on a weekly basis (including answers to exercises);
2. INDIVIDUAL PROGRESS FORM: form to track your own individual performance for the mini-project;
3. GROUP EXERCISES FORM: form to submit textual responses in markdown format for the exercises together with any evidence of using AI tools.
4. Submission link (video, software):
   - Upload the video recording by following **the guide to submit a video assignment**.
   - The software must be submitted in a ZIP file containing the entire project folder, which contains the file `gradle.build`. Please add a short `readme.md` file explaining where the software artifacts can be found (scripts implementing exercises).

  - Please do `gradlew clean` **before zipping** your files in order to delete the binaries. This will result in a significantly smaller zip file, saving storage space. Additionally, you'll **reduce your environmental impact** by using only the necessary computational resources—an important step, as data centers contribute significantly to pollution.
  - Presentation slides (PDF,PPT,PPTX): in the presentation slides: add the username of the presenter in the footnote of each slide used by that presenter.
  - Any evidence you may want to provide regarding the use of AI: prompts, logs, etc (format MD - markdown).
  - Images referenced in your answers to exercises in the GROUP EXERCISES FORM.

Project supervision

The group allocation will be shown under `Assessment and Feedback > Coursework 1`. Each group must elect a **group rep**. The main tasks of the group rep are:

- to manage group tasks (acting as a project manager) and to submit group exercises/progress;
- to be the point of contact with the project supervisor.

Each group will be allocated a **project supervisor** and all communications regarding your project should be addressed to them. Our project supervisors have been trained to assist you in project tasks and team work.

Each group needs to submit the following information using the **group submission form** (the deadline for this will be advertised separately):

- title for the mini project, which represents the project topic (e.g. learning about Netflix watching trends)
- brief description of the application domain that you are going to analyse with data analytic queries (e.g. find out the average period of watching all of the available chapters of Squid Game) [max 100 words]

Presentation (video submission)

In the presentation, students should cover all five exercises from the mini-project, including the introduction with the problem, dataset, and query (Exercise 1), the implementation and results of the Groovy-based query (Exercise 2), the cloud solution using MongoDB (Exercise 3), a critical comparison of the two solutions based on scalability, consistency, or evolution (Exercise 4), and a deeper analysis with additional evidence or datasets (Exercise 5). The presentation should lead to clear, well-justified conclusions.

The group should divide tasks evenly, with each member responsible for presenting their specific contributions, ensuring that the content is cohesive and well-organized, reflecting collaborative teamwork. To maintain clarity and flow, ensure smooth transitions between sections and rehearse as a group to manage timing and consistency across the entire presentation.

Important points:

1. All participants must have their webcam on during the video recording, with their faces clearly visible.
2. Slides must be annotated with the University username (e.g., `ab123` if your email address is `ab123@student.le.ac.uk`) of the person speaking during that slide.
3. For each exercise, refer to the rubric for detailed expectations on what to cover in that section.
4. The presentation must be self-contained; do not assume that the audience has access to your code or textual explanations.

5. Show your expertise by speaking naturally and confidently about the content, avoiding reliance on reading from the slides. Demonstrate that you have thoroughly thought through the material and are in command of your contributions.
   - In programming exercises, short demos may help in the programming exercises but **do not read the code, explain it!**
   - In research/analysis exercises, explain how you researched the topic, show the analysis from different perspectives (charts/illustrations may help). Explain concepts in depth, focusing on clarity (ensuring the explanation is easy to follow and logically structured), relevance (directly addressing the question or problem, avoiding unnecessary details), and correctness (using accurate data, valid methods, and citing reliable sources).

The group should do the presentation online using Blackboard Collaborate or Microsoft Teams and **record it**. A guide to **recording a video on MS Teams** is available under `Assessment and Feedback > Coursework 1`. The presentation should be in a single slide deck, with one team member sharing the screen as each person explains their part.

Finally, download the video and check that it has been recorded correctly. Sample mini-project presentations are available on Blackboard (`Assessment and Feedback > Coursework 1`).

Rubric and marking criteria

A more detailed description of what is expected in each exercise, accompanied by the corresponding mark and grade band can be found in the rubric available under `Assessment and Feedback > Coursework 1`.

4. Group exercises - 70%

Exercise 1: Planning (9 marks) max. duration 2 minutes

Choose a JSON dataset that is of your interest. For example you can choose one dataset from the lists below but feel free to find an interesting JSON data set:

- awesome JSON datasets
- Webster English dictionary

Other datasets, which may be present in other formats (CSV, TSV, etc):

- Interesting datasets
- Data world datasets: find data sets related to your country (e.g. Covid-19)

Define a query that will require at least one of each of the following type of operations (SQL statements included for reference only as we are working with NoSQL technology):

1. **data selection** (typical `SELECT * FROM ...` in SQL),
2. **data projection** (typical `SELECT COL1, COL2, COL3 FROM ...` in SQL),
3. **data filtering** (typical `condition` in `SELECT * FROM ... WHERE condition`),
4. **data combination** (like sums) **and/or data grouping** (like `GROUP BY` in SQL).

In the presentation, explain the chosen data set (schema, context) and the intented query.

Exercise 2: programming with Groovy collections (9 marks) max. duration 2 minutes

Implement your query in Groovy, execute it to test its functionality, and elaborate on the Groovy code. The explanation should be self-contained and include an interpretation of the results. Align your task with the objectives set forth in the introduction, particularly in relation to either data evolution, scalability, or consistency.

Exercise 3: programming with MongoDB (9 marks) max. duration 2 minutes

Implement the cloud-based solution using the MongoDB Java Driver within a Groovy environment. Execute the code to test its efficacy, and provide a self-contained explanation, interpreting the results in the context of your chosen focus—be it data evolution, scalability, or consistency.

Exercise 4: critical analysis (15 marks) max. duration 2 minutes

Critically compare the standalone and cloud-based solutions with respect to one of the following aspects:

- Evolution/Maintenance
- Scalability
- Consistency

Your presentation must articulate clearly defined comparison criteria, which will then be systematically applied to evaluate the two solutions. Refer to the grading rubric for additional details on what is expected. Align your comparison with the specific objectives stated in the introduction, ensuring a comprehensive analysis.

Exercise 5: In-depth Critical Analysis (20 marks) — Max. Duration: 2 minutes

Deepen the analysis initiated in Exercise 4 by providing additional evidence and insights. This can include, but is not limited to:

- Utilising additional datasets from the same domain to reinforce your points.
- Performing performance measurements for quantitative backing.
- Extending your research on the topic through consultation of relevant academic literature.

While Exercise 4 aims at a preliminary analysis of the query scrutinised in Exercises 2 and 3, the objective here is to validate your claims through more extensive experimentation and rigorous scrutiny. This component is open-ended; success will be assessed both considering the ingenuity of the additional experiments conducted and the validity of the conclusions drawn. Optionally, you may also explore datasets in formats other than JSON, like CSV, employing Groovy for data importation.

Here you have a few examples to explore each dimension in more depth - feel free to explore additional ones:

Scalability

- Pagination
  - Objective: Introduce the concept of scalability by illustrating how pagination can improve query performance in MongoDB.
  - Task: Create a MongoDB collection with a simple dataset, say, 100 records. Implement a pagination feature in the application code to fetch records in batches of 10.
  - Reflective analysis: Discuss the difference in performance and user experience with and without pagination. Explain how pagination could be vital as the database scales.
- Caching Mechanisms

    - Objective: Investigate the impact of caching on query response time.
    - Task: Implement a simple caching mechanism for frequently queried data.
    - Reflective Analysis: Measure the query performance before and after the implementation of caching. Discuss the implications of caching for database scalability.
- Query Optimisation
    - Objective: Analyse the influence of query optimisation techniques on database scalability.
    - Task: Take a complex query and optimise it using indices in MongoDB.
    - Reflective Analysis: Compare the performance of the optimised query against the non-optimised version. Discuss how query optimisation can play a critical role in scalability.

Consistency

- Basic Consistency Checks
    - Objective: To understand the principle of consistency by implementing basic data validation.
    - Task: Add a basic validation rule to a MongoDB collection, for instance, a mandatory field. Try inserting documents that violate this rule.
    - Reflective analysis: Observe what happens when inconsistent data is attempted to be inserted. Discuss how such simple rules can aid in maintaining consistency.
- Eventual vs Strong Consistency
    - Objective: Understand the difference between eventual and strong consistency.
    - Task: Configure your MongoDB instance to operate under both eventual and strong consistency models, then perform the same query under each configuration.
    - Reflective Analysis: Discuss the differences observed in each scenario and the trade-offs involved in the choice of a consistency model.
- Data Redundancy
    - Objective: Explore the role of data redundancy in maintaining consistency.
    - Task: Create multiple replicas of a MongoDB database and perform concurrent read and write operations.
    - Reflective Analysis: Evaluate the behaviour of the system under concurrent operations. Discuss how redundancy can both aid and challenge data consistency.

Maintenance and Evolution

- Simple Maintenance with Field Addition and Deletion
    - Objective: To understand the need for database maintenance by modifying the MongoDB collection schema.
    - Task: Add a new optional field to the collection. Run queries that specifically look for this new field.
    - Reflective Analysis: Discuss how the addition of a new field affects previous records and queries. Reflect on how this is indicative of maintenance concerns in database management.
- Versioning
    - Objective: Examine the role of versioning in database evolution.
    - Task: Introduce a version field to the documents in your MongoDB collection and manipulate it when updates occur.
    - Reflective Analysis: Reflect on how versioning can assist in evolving a database schema without disrupting existing functionality.
- Data Migration

  - Objective: Understand the complexities of data migration in database maintenance and evolution.
  - Task: Migrate a subset of your MongoDB data to another collection with a different schema.
  - Reflective Analysis: Evaluate the challenges encountered during migration. Discuss how frequent data migrations might affect maintenance and evolutionary paths of a database.

Exercise 6: Project management (8 marks)

In this exercise, your group will demonstrate effective project management by organizing tasks, managing workload, resolving conflicts, and maintaining clear communication.

What to do:

1. **Task Allocation**: Assign tasks early and ensure workload is balanced across the team. Keep a shared document to track progress and update it regularly. State contributions early as expectations and refine them before submission.
2. **Meetings**: Hold regular meetings to track progress. Take notes on key decisions and update the task allocation accordingly.
3. **Final Submission**: submit responses to **group progress form**, **group exercises form** and software artifacts (video, project source code, slides, document with additional evidence/information).
4. **Conflict Management**: If conflicts arise, resolve them within the group or escalate to your supervisor if needed. Document any conflicts and resolutions.

5. Individual assessment - 30%

5.1. Individual Reflection Exercise - 20%

Complete a brief individual reflection **each week** to track your progress and contributions to the mini-project.

1. **Weekly skill focus**: Choose one or two skill from the rubric (e.g., NoSQL concepts, teamwork). Over the mini-project, reflect on **five different skills**.

2. **Rate your performance**: For the chosen skill, rate yourself as **ChatGPT passenger, Novel, Competent, Master, or Expert** based on the rubric.

3. **Justify your rating**: In **50-100 words**, explain why you selected that level by:

   - Describing your **specific contributions**.
   - Highlighting any **challenges** and how you addressed them.
   - Discussing how your efforts contributed to the **overall project progress**.
   - Mentioning, if applicable, how AI tools supported your work.

4. **Submit weekly**: Ensure timely submission of reflections each week.

This process keeps you engaged, helps you adjust your approach, and connects your work to the **module learning outcomes**. Your reflections will be compared with the group's submissions for accuracy and fairness.

Refer to the **rubric** for skills and performance expectations.

5.2. Individual Reflection in Lab Exercises (weeks 13 and 14) - 10%

This mark will be obtained by submitting a meaningful justification of your work in the lab sessions.