Nottingham Trent University

School of Science and Technology

COURSEWORK ASSESSMENT SPECIFICATION

Module Code	**COMP20081**
Module Title	Systems Software
Module Leader	Dr Gadelhag Mohmed
Module Team	Dr Pedro Machado; Dr Isma Cheheb; Dr Pablo Lopez-Custodio; Dr Taha Osman; Dr Evtim Peytchev; Mr Feliciano Domingos

Nottingham Trent University

Coursework Title	Implementation and validation of cloud load balancer
Module Learning Outcomes assessed	**Knowledge:** **K2.** Demonstrate an understanding of the coding behind the main components of an operating system. **K3.** Demonstrate an understanding of Internet protocols and the theory of complex systems on the Internet **Skills:** **S2.** Independently research topics relating to Operating Systems, Internet protocols and programming of complex systems. **S3.** Demonstrate programming skills for adapting operating system code. **S4.** Demonstrate programming skills for producing client /server solutions using a relevant programming language. **S5.** Work effectively within a small team.
Apprentice Learner KSBs evidenced	N/A
Contribution to module (include contribution to element if appropriate)	80 %
Date work set	06/10/2025
Deadline for submissions	2.30 pm (UK time) - 09/02/2026
Method of submission	NOW Dropbox Folder
Deadline for feedback	02/03/2026
Method of feedback	Electronic via email and verbal feedback during the live demo.
Previous coursework that may support this submission.	Not Applicable

Late submissions and Considerations of Personal Circumstances

Work handed in up to five working days late will be given a maximum grade of Low Third whilst work that arrives more than five working days will be given a grade of Zero.

Please note, if you are repeating the work and are capped in your grade, then you will receive a Zero grade if the work is submitted after the initial deadline.

Work will only be accepted beyond the five working day deadline if you have asked for an extension in consideration of your personal circumstances. That means something has occurred, beyond your control, that has impacted your ability to complete the work. Details of the process of requesting extensions and what is considered valid reasons to request can be found here in the Students' Handbook

Group work
For this module, the Coursework is an individual project.

Presentations
As a student, you may have a statement of access concerning aspects of presentations such as eye contact or reading from notes. While these aspects of presentation will be part of the GBA grid used to assess work, it will not affect your grade.

Breaches of Academic Integrity
To ensure that you are not accused of any breaches of Academic Integrity, look at the NOW page Plagiarism and Academic Integrity at NTU for guidance.

The University views **plagiarism and collusion** as serious academic irregularities and there are a number of different penalties which may be applied to such offences. Section 17C of the Quality Handbook outlines the penalties and states on page 21-22 that **plagiarism** includes:

- "presenting someone else's ideas as your own [**including text, graphs, diagrams, videos, etc.**] in a substantial proportion of your work, with or without consent, by incorporating it into an assessment without full acknowledgement, including:
- **self-plagiarism**: reproducing or representing work for assessment without proper attribution and attempting to gain credit for this work where credit has already been received;
- **paraphrasing**: rephrasing a source's ideas without proper attribution;
- **mosaic plagiarism/patchworking**: weaving phrases and text from several sources into your own work; and/or adjusting sentences without quotation marks or attribution;
- **source-based plagiarism**: providing inaccurate or incomplete information about sources such that they cannot be found;
- **computer code plagiarism**: copying or adapting source code without permission from and attribution to the original creator;"

whereas **collusion** includes:

- "working with other students on an assessment meant for individual submission;
- sharing your work with other students enabling them to plagiarise your ideas."

In particular, please note that submitting portions of work already assessed for the same learning outcomes is **self-plagiarism** and is a serious breach of academic integrity.

Penalties for breaches of academic integrity range from capped or zero grades for elements of modules, to dismissal from the course and termination of studies.

AI-powered language models

At NTU, we use Microsoft Copilot. This is included within the institutional Microsoft Subscription and by logging into Microsoft Copilot using your NTU login credentials, you can be assured that your use of the tool will be more secure than that of other generative AI services that operate outside of the NTU digital ecosystem.
At its heart, Copilot is a chatbot that can assist you with an almost endless range of tasks. If you're new to Copilot, or Generative AI chatbots in general, Microsoft's three introductory videos are a great place to start:

- Meet Microsoft Copilot
- What you can do with Copilot
- Start using Copilot

We use Copilot over ChatGPT etc as NTU has a data security agreement with Microsoft, which ensures enhanced protection when working within the NTU digital ecosystem.
You can upload work documents to Copilot but only upload what is strictly necessary.

You may see instances of Copilot integrated directly into Microsoft apps such as Word, PowerPoint and Excel. NTU's licence doesn't currently provide this level of integration. For now, we only have access to the stand-alone Copilot app.

To ensure you are using Copilot within the safety of the NTU ecosystem, please login to Copilot using your NTU credentials

Other GenAI Tools
While other GenAI tools may be suitable for general purposes, but you must not upload NTU documentation or any other forms of intellectual property to these platforms.
This is essential to keeping everyone's data and resources secure.

Using AI Responsibly
General Guidelines:

- Transparency: Always be transparent about your use of AI. If in doubt, seek clarification from the author of this assessment and reference use appropriately.
- Academic Integrity: AI tools can support your learning, but they must not replace your own critical thinking and analysis. Never use GenAI to produce your submission for you.

- Ethical Use: Follow ethical guidelines when using AI, avoiding plagiarism and misrepresentation of AI-generated content as your own.
- Limitations: Never trust an output from GenAI without checking it first. They are poor at summarising resources correctly so you would need to check anyway.

The Traffic Light System is a simple framework used at many institutes to clarify when and how you can use generative AI in assessments. It categorises AI use into three levels: Green, Amber and Red

In this assignment:

Green We encourage the use of GenAI to aid with

Checking spelling and grammar but not in designing structure of the reports. There is a template for that. Reference and use of acknowledgement form needed

Amber We allow the use of GenAI to carry out the following tasks, but you must reference and keep outputs and mention in acknowledgement form.

- Research and data gathering (e.g., asking AI to summarise existing studies or generate ideas, but these must be critically evaluated as are often incorrect)
- Drafting essays or reports (e.g., using AI to help structure or rephrase your work, but with clear identification of AI contributions)
- Sense checks the tone of the reports to clients

Red You must not use GenAI to

- Complete your work entirely without critical input from yourself
- Complete your reflection aspect
- Generate fake resources for the client
- Generate fake data or to analyse data collected
- Generate code for your coursework

It is important to note when using any AI platform that they generate the most common responses to questions, not necessarily the correct ones. They also fabricate evidence. The material they produce is not your own words. Assessments require you to answer questions giving your own view and in your own words. The outputs from platforms such as ChatGPT do not provide that.

By presenting such material as your own words you are violating the Academic Integrity policy, a matter that NTU takes very seriously.

The skills you develop during your time with us allow you to interrogate material and evaluate it, which are important skills in all careers. Generative AI does not enable you to develop these skills. If you have utilised such platforms, **you must retain all outputs from them** to provide evidence that your work is your own in the case of suspected breaches of Academic Integrity.

Acknowledging use of GenAI

You must also acknowledge if and how you have used GenAI by completing the GenAI usage acknowledgement form and then submit it with your work to the same dropbox as the assignment here. There is no penalty to your grade for not submitting the completed form, but if there is a question about inappropriate use of GenAI then lack of completion will be considered as evidence in any investigation. **Dropbox for this assignment.**

Where GenAI is permitted, please follow the guide at the below link to correctly reference the use of GenAI as part of your assignment as you would any reference or resource. Referencing of GenAI Then click on **Quick start to Harvard referencing** and look at the bottom of the list.

I. Assessment Requirements

This coursework must be done individually. You will be **assessed individually** and therefore you will have to **submit all the files individually**. Given the problem scenario in described below, you will have to produce an individual presentation of the designed and implemented software solution. The presentation MUST be produced using the template that will be provided on the module page. Presentations with more than 6 pages will be marked down. **You MUST have a PRIVATE git repository on Olympus.**

An in-person live demonstration is required and must be conducted during the scheduled sessions. Failure to complete the demonstration will result in failing the module.

The following deliverables are mandatory:

1. **Presentation [max 6 pages]** to be submitted to the NOW Dropbox **Draft presentation folder** you to detect and reduce similarity of your presentation. The deliverable must be a PDF. **Ensure that the submitted file was not cropped.** Please use the provided presentation template, here.
2. **Presentation [max 6 pages]** to be submitted to the NOW Dropbox **Main presentation folder.** The presentation must be a PDF. Please ensure that the submitted file is not cropped**.**
3. A zipped file containing the NetBeans project(s) to be submitted to the NOW Dropbox **Source code** folder.

Note that:

- The code MUST also be stored in the git repository on https://olympus.ntu.ac.uk . Please refer to the labs to learn how to setup your git repository. Storing the source code in the git repository will enable you to recover quickly should the system fail and allow you to rollback.
- The source must compile and run in the module docker containers (**pedrombmachado/ntu_lubuntu:comp20081**)
- Keep evidence of the submission of your assignment, and a copy of your assignment in case of the unlikely event of any loss.

Assessment Scenario/Problem

For the COMP20081 Systems Software coursework, you will have to develop a load balancer and a simulated cloud infrastructure using Docker and/or Docker Compose. The cloud infrastructure will consist of; One file partitioning container that leading to two containers for file storage, one for the load balancer, one for a MySQL server, and one for a local SQLite database and the user GUI application. In addition to one support host manager docker-container (see Figure 1).

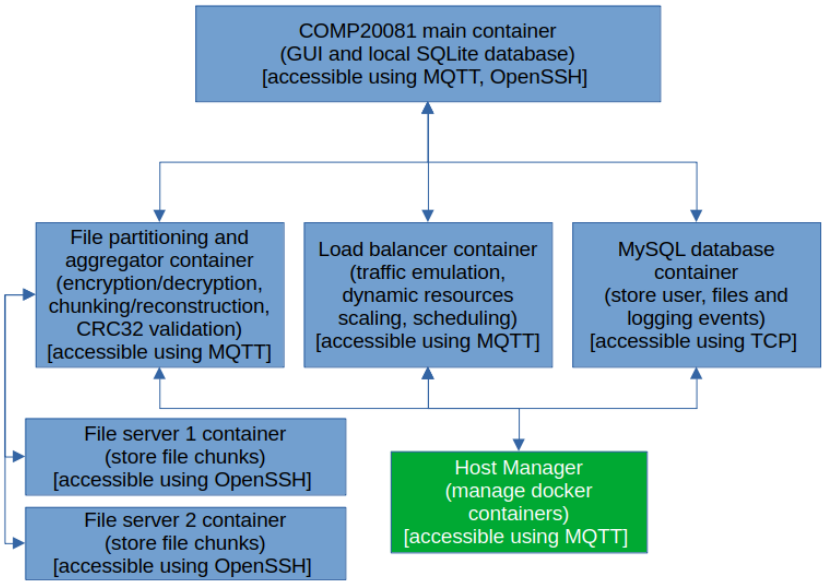

Figure 1 Distributed architecture of the minimal requested system.

Based on the idea presented in the Figure 1, the coursework requirements are summarised as follows:

- 1 file partitioning container
- 2 file storage containers.
- 1 load balancer (plus scheduler) container.
- 1 MySQL database container.
- 1 SQLite database (plus GUI) in main container.
- 1 support host manager docker-container.

Objectives:

- Implement a basic load balancer that distributes requests evenly.
- Develop simple microservices for file storage.
- Use Docker Compose to orchestrate the containers.

- Use Java/JavaFX
- Provide a basic user interface for interaction.
- Submit a presentation (max 6 slides) and a working codebase.

A **load balancer** is a cloud service that distributes network traffic across multiple servers to improve performance, reliability, and scalability. In the context of the COMP20081 project, the load balancer will play a crucial role in managing the workload across the two file storage containers within the simulated cloud infrastructure.

Key functions of a load balancer in this coursework:

- Receive incoming requests (e.g., file uploads, downloads) and distribute them evenly to the available file storage containers.
- Help to prevent any single server from becoming overloaded, ensuring better overall performance and responsiveness.
- Redirect traffic to the remaining healthy containers, minimising downtime and ensuring service continuity.
- The load balancer can be configured to handle increasing workloads by adding more file storage containers to the infrastructure. This feature allows the system to scale up or down based on demand.

The load balancer will employ at least three of the following scheduling algorithms: First Come, First-Served (FCFS), Shortest-Job-Next (SJN), Priority Scheduling, Shortest Remaining Time, Round Robin (RR) and Multiple-Level Queues. The scheduling algorithms will be used to optimise resource allocation and handle concurrent file uploads, downloads, and deletions.

Users will access the system through a Java and JavaFX-based portal. Standard users can create accounts, update their details, upload, download, share with read and/or write permissions and delete files. Admin users have additional privileges, including creating, updating, and deleting users and promoting standard users to admin.

To simulate real-world scenarios, the load balancer will introduce artificial delays ranging from 30 to 90 seconds to emulate the varying response times experienced by multiple users accessing cloud resources. Additionally, file containers will be locked during upload, download, and deletion operations to prevent concurrent access and ensure data integrity.

As part of this coursework and to pass the module, you will have to do an in-person live demo to showcase the system's capabilities by simultaneously performing file uploads, downloads, and deletions, highlighting the load balancer's effectiveness in managing resource allocation and maintaining system performance.

The GUI app must deliver the following functionality and behaviour:

#	Functionality/Behaviour	Target Grade
1	**User Authentication & Roles** • Create, delete, update accounts • Login/Logout • Standard vs Admin roles	3 or above

2	**Session Management:** Implement session management to keep users logged in during their interactions with the system	3 or above
3	**Files Management:** Create new file: allowing the user to add new content, Update file content and Delete files.	3 or above
4	**Access Control Lists (ACLs):** Enable users to set permissions on files for different users (e.g., read and/or write).	3 or above
5	**File Sharing:** Allow users to share files with other users and set specific permissions (read/write).	2.2 or above
6	**Encryption/Decryption:** Use strong encryption algorithms (e.g., AES) to encrypt passwords and file chunks before distribution and decrypt them upon retrieval.	2:1 or above
7	**Load Balancer Functionality:** The load balancer should distribute incoming requests evenly among the file storage containers.	3 or above
8	**Traffic Management:** The load balancer should handle traffic spikes without degrading performance, maintaining a responsive user experience.	2.2 or above
9	**Health Check Mechanism:** Implement health checks for file storage containers to ensure only send traffic to healthy containers.	2:2 or above
10	**Artificial Delay Simulation:** Introduce artificial delays ranging from (e.g., 30 to 90 seconds) to simulate real-world latency.	2:1 or above
11	**Concurrency Control:** Lock files during upload, download, and deletion to ensure data integrity and prevent conflicts	2:2 or above
12	**Local SQLite Database:** Implement a local SQLite database to store user session data and temporary file metadata, ensuring fast access and reduced latency.	2:2 or above
13	**Remote MySQL Database:** Configure a remote MySQL database to store user profiles, file metadata, and access control lists, enabling centralised management.	2:2 or above
14	**Database Synchronisation:** Simple implement synchronisation mechanisms between the local SQLite database and the remote MySQL database to ensure data consistency across both databases.	2:2 or above
15	**Logging:** Implement logging for all user actions and system events for audit and debugging purposes, this means record user/system events.	2:1 or above
16	**Scalability and Elasticity:** The system should allow the addition of more file storage containers to handle increased workloads seamlessly (e.g., using docker compose, Kubernetes, Terraform).	2:1 or above
17	**User Interface (JavaFX):** Simple, Provide a user-friendly JavaFX interface for users to interact with the system intuitively.	3 or above
18	**Performance Metrics:** Implement metrics to monitor performance and usage statistics of the load balancer and file storage containers.	2:1 or above
19	**Emulation of a terminal:** Enable users running the following commands: • Mv • cp • ls	3 or above

	• mkdir • ps • whoami • tree • nano	
20	**File Chunking:** Develop a file chunking mechanism to split large files into smaller, manageable parts.	2:1 or above

Overall, you are expected to use:

- The module container if you are targeting up to a 3 Mid;
- The module container and extra containers for storing files and run the load balancer if you are targeting a 3 High and up to 2:2 High;
- The module container and four extra containers for storing files and run the load balancer if you are targeting a 1st.

Use the following project and source code to implement your coursework.

Refer to week 13 to remember how to create docker containers. **NOTE that you will have to create built-in commands to store the current directory and to change the value of the current directory (i.e. same behaviour as cd) and to visualise the current directory (i.e. same behaviour as pwd).**

Tools and frameworks:

1. Environment: you **MUST** use the module docker containers (**pedrombmachado/ntu_lubuntu:comp20081**) and create new file storage container(s) in case you are targeting a High Pass or above.
2. Java IDE: Netbeans **MUST** be used, and the project **MUST** compile using Maven.
3. GUI design: Scene Builder **MUST** be used to designing GUIs.
4. Java GUI library: JavaFX **MUST** be used for creating GUIs. Please note that projects using Java GUI tools (e.g. Swing) will not be accepted.
5. Databases: SQLite and MySQL **MUST** be used.
6. Git repository: All the projects **MUST** be **privately** stored on https://olympus.ntu.ac.uk/.

Implementation

You are expected to implement the above software system using Java. The final application MUST meet the following requirements:

- error handling must be used when needed.
- at least a simple UI that allows users to interact with the software system.
- the NetBeans project(s) must compile using Maven.
- The power point template that is provided on the module page MUST be used and the presentation cannot contain more than 6 pages.
- the source code MUST be stored in https://olympus.ntu.ac.uk

II. II. Assessment Criteria (contextualised GBA Grid)

12

Class	General Characteristics
1st (Excellent)	The presentation and implemented solution demonstrate outstanding/excellent knowledge and understanding of the chosen area. The presentation and developed code show that you are typically able to go beyond what has been taught (particularly for a mid/high 1st). Evidence of extensive and appropriate selection and critical evaluation / synthesis / analysis of reading / research beyond the prescribed range of information given in lectures and labs. The submission shows an excellent demonstration of relevant skills and knowledge. The application performs according to its specifications, effectively managing and displaying errors and exceptions in the logs pane. The student adeptly showcased their ownership of the project during the in-person live demonstration. The presentation and live demonstration were exceptionally comprehensive, covering all functionalities up to and including those targeting a 2.1, with a minimal absence of up to two expected behaviours. An executable *.jar file has been provided, ensuring the application's compatibility with JRE 20 or higher. The presentation was thoughtfully crafted using the provided PowerPoint template.
2:1 (Very good)	The presentation and implemented solution demonstrate particularly good knowledge and understanding of the area of study. The presentation shows evidence of appropriate selection of reading/research, some beyond the prescribed range of information given in lectures and labs in the development of the solution. The application functions precisely as specified, effectively handling and displaying errors and exceptions within the logs pane. The student demonstrated a strong sense of ownership during the in-person live demo. Both the presentation and the live demonstration were exceptionally thorough. A shell is accessible, delivering the requested functionality seamlessly. All functionalities, including those targeting a 2.2 and below, are fully operational, with only a maximum of four behaviours or one functionality missing. An executable *.jar file has been provided, ensuring compatibility with JRE 20 or higher. The presentation was skilfully crafted using the PowerPoint template.
2:2 (Good)	Good knowledge and understanding of the area of study balanced towards the descriptive rather than analytical (i.e. more evidence of more complete descriptions rather than more in-depth understanding of the process behind the given area and its implementation). Both the developed code and presentation show evidence of an appropriate selection and evaluation

	of reading/research but is reliant on set sources to advance the work and implementation. Both the presentation and developed code may be limited in range and complexity. The presentation and code show clarity but structure may not always be coherent. The application largely adheres to the specified requirements, with effective error and exception handling displayed in the logs pane. The student demonstrated a clear sense of ownership during the in-person live demo. Impressively, the presentation and in-person live demo collectively offer a comprehensive understanding of the project. A functional shell is available and delivers the requested functionality seamlessly. All functionalities targeting a 3rd are fully implemented, albeit with a maximum of six behaviours and one functionality yet to be addressed. Users can readily execute the application through the provided *.jar file, ensuring compatibility with JRE 20 or higher.
3 (Sufficient)	Knowledge and understanding are sufficient to deal with terminology, concepts and algorithms but fails to make meaningful synthesis. Some ability to select and evaluate reading/research for the development of the code is present, however work may be more generally descriptive and not provide detail into how the chosen area algorithmically works or the potential complexities. Both the presentation and code show strong reliance on available sources to advance the work and content/code may be weak or poorly constructed. The submission shows adequate demonstration of relevant skills over a limited range but contains some weaknesses. Adequate implementation of the application is observed. Errors and exceptions are appropriately displayed in the terminal. File storage is effectively managed within the module container. A user manual is provided, and the student effectively demonstrated ownership during the in-person live demo. The quality of the presentation and in-person live demo meets acceptable standards. Nevertheless, up to eight behaviours and two basic functionalities remain outstanding. The presentation adheres to the PowerPoint template guidelines.
FAIL (Insufficient)	Insufficient knowledge and understanding of the area. Concepts, algorithms, and the development of code is descriptive and fails to address requirements of the assessment.
ZERO	Work of no merit OR absent, work not submitted, penalty in some misconduct cases.

III. Feedback Opportunities

The module content from Weeks 5 - 14 are particularly relevant for understanding what you are expected to do for this assignment. The weekly practice activities are designed to prepare you for this assignment, and you are strongly encouraged to complete Weeks 5 - 13 tasks from before attempting this assignment. The module content from Weeks 14 and 15 also supports achieving a high grade on this assignment.

During the lab sessions, you can request verbal feedback on the practice activities from your tutor.

Your tutor can also provide you with help and guidance if you get stuck on any of the practice activities.

You should receive written feedback on your submitted assignment within three weeks of the submission deadline, together with your awarded grade. After this, you can request further verbal feedback on your submission from your tutor during a lab session.

Resources that may be useful:

Remember to use Outlook or physical calendars to block out time between lectures and labs to work on this coursework.

- NetBeans, SceneBuilder, JavaFX tutorial - this tutorial is a good starting point for students who want to get start with these 3 tools.
- SQLite using java tutorial - This is a good tutorial to learn how to work with SQLite in java
- Guide to Docker volumes - tutorial explaining how to use volumes to store containers data.

IV. Referencing Styles

General Reference guidance
Referencing in the style of Harvard
Referencing in the style of Vancouver.
Referencing in the style of Science and Justice Journal
Referencing in the style of IEEE
Referencing in the style of the Royal Society of Chemistry
Referencing of GenAI Then click on **Quick start to Harvard referencing** and look at the

bottom of the list.
A guide to planning your time is here and a range of study skills can be found here

V. Moderation

The Moderation Process
All assessments are subject to a two-stage moderation process. Firstly, any details related to the assessment (e.g. clarity of information and the assessment criteria) are considered by an independent person (usually a member of the module team). Secondly, the grades awarded are considered by the module team to check for consistency and fairness across the cohort for the piece of work submitted.

VI. Aspects for Professional Development

This assignment will give you experience using version-control software and development of applications using micro-services. This assignment will also develop your skills at developing network applications. This is an important skill that employers claim is often lacking among software development students. Many of these are useful transferable skills for employment applications.