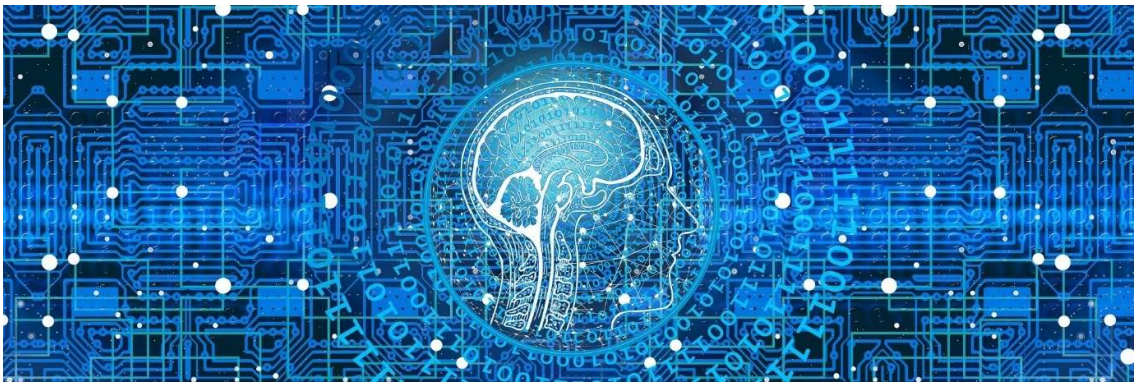
Network System Development

5CM510



2024-2025

Assessment Brief 1

Dr Suleiman Aliyu

UNIVERSITY OF DERBY

Network system development: 5CM510

Contents

Module Leader(UoD):
Dr. Suleiman Aliyu
Email: s.aliyu@derby.ac.uk
Phone: 01332 592962

Module Leader(Athens):
Dr Christos Katsigiannis
Email: c.katsigiannis@medcollege.edu.gr

Module Leader(Thessaloniki):
Dr Georgios Michoulis
Email: g.michoulis@mc-class.gr

Key dates and details

Assessment Type:	Individual Code Artefact
Assessment weighting:	70%
Word count/Length:	N/A
Learning Outcomes:	1
Submission Method:	Blackboard Assignment
Submission Date:	12:00 UK time, 24/01/2025
Provisional Feedback Release Date:	12:00 UK time, 10/02/2025

Description of the assessment

This assessment aims to introduce you to the concepts and implementation of implementing client-server (networked) systems programmatically. Additionally, this assessment looks to help you understand the implementation of information exchange and data security through the use of encryption and AAA mechanisms.

1. Understand the core principles of implementation of client-server systems
2. Understand and implement the concepts of internetworking using current networking technology.

UNIVERSITY OF DERBY

This assessment builds upon the programming knowledge that you gained in Programming 1 and Programming 2, and the networking knowledge from Networking Fundamentals.

Assessment Content

This coursework requires you to produce a small client-server network which demonstrates:

- ability to structure and comment code appropriately
- client-server model (networking) and/or peer-to-peer model principles
- information exchange (communication security) principles e.g. authentication, and encryption.

In negotiation with your tutor, select an client-server scenario that meets the learning outcome (1) and assessment criteria laid out within the rubric. If you cannot identify a suitable project that you want to implement then you should consider something like the Simple File Transfer Protocol (SFTP – RFC 913). This could be a smart home, industrial automation, healthcare, or any other relevant domain. Identify at least two IoT devices (e.g., sensors, actuators) that need to communicate securely. Define the communication requirements, including data types, frequency, and security considerations.

Deliverables:

Your assignment MUST be submitted electronically via Course Resources by the due date and time. You must submit it as ONE zip file that contains the following:

- The document detailing the testing of your application and any information needed to run your application.
- The full source code for your server(s), including any build files needed.
- The full source code for your client(s), including any build files needed.

Please note: submission in any form other than a single zip file (e.g. rar, multiple files, etc), will NOT be accepted and will be counted as a non-submission.

Your mark and feedback will be given during a viva process in which you will demonstrate your code functionality to a tutor. You should provide a completed functionality checklist to your tutor at your viva. **Failure to attend your viva will automatically count as a non-submission for this assessment.**

Marking Criteria:

This coursework is marked against the following rubric, marks within a band are awarded based on the code and commenting quality. All components of a band must be completed for a mark to be awarded within that band. Academic judgement may be applied where all components are not completed but significant work has been carried out at a higher band level.

Assessment Rubric

All Requirements	Weight
Networking requirements (N1-N10)	30%
Application requirements (A1-A4)	25%
Security requirements (S1-S6)	25%
Submission Requirements (R1-R8)	20%
Total	100%

Submission requirements (all non non-submissions)

	Requirement	Level of Implementation/Details
	Submission requirements	
R1	Viva attended	
R2	Checklist provided to tutor at viva	
R3	Submission is a single zip file	

1. Basic Requirements

	Requirement	Level of Implementation/Details
	Submission requirements	
R4	Submission contains 'Client' and 'Server' folders inside the zip file	
R5	Code runs on any computer in MS214 or MS215. **Note:** you should not hardcode a path or IP address in the code that you write. Any path should be relative to the location of the executable	
R6	Solution interprets/builds without any errors or warnings	
R7	Implementation Log provided in zip file	
R8	Feature checklist provided in zip file	
	Networking Requirements	
N1	Synchronous bidirectional communication between client and server.	
N2	Connection and disconnection are handled without errors on the server side.	
N3	Error handling and message content verification are handled on the server side.	

	Security Requirements	
S1	Network traffic is encrypted using a standard algorithm using [a pre-shared, entered, or non-negotiated key].	
	Application Requirements	
A1	Implements at least 2 states that alter the behaviour of the system based on user input.	

2. Additional Features

	Requirement	**Level of Implementation/Details**
	Networking Requirements	
N4	Multiple clients supported by a single server.	
N5	Error handling and message content verification are handled on both the client and server side (replaces N3)	
	Security Requirements	
S2	Key is negotiated using an appropriate mechanism such Diffie-Hellman (modifies S1 component in [])	
	Application Requirements	
A2	Implements at least 1 complex state that allows data to be stored between sessions.	

3. Additional Features

	Requirement	**Level of Implementation/Details**
	Networking Requirements	
N6	Asynchronous bidirectional communication between client(s) and server (replaces N1).	
	Security Requirements	
S3	Packets implement mechanisms to prevent at least 1 kind of attack e.g. replay attacks.	
	Application Requirements	
A3	Implements at least 1 complex state that alters typical network communication pattern (e.g. client can send an arbitrary number of messages in a row) .	

4. Additional Features

	Requirement	**Level of Implementation/Details**
	Networking Requirements	

N7	Shared state information between multiple clients and server.	
N8	Network code is provided as a layered solution separated from the application logic.	
	Security Requirements	
S4	AAA implemented (authentication, authorisation, and auditing) e.g. user privilege levels.	
	Application Requirements	
A4	Implements an appropriate way to view or access AAA information.	

5. Additional Features

	Requirement	**Level of Implementation/Details**
	Networking Requirements	
N9	Single additional feature e.g. heartbeat, peer-to-peer implementation, or other functionality agreed with tutor.	
	Security Requirements	
S5	Single additional feature e.g. data encrypted on storage, user password resets, account lockouts.	

6. Additional Features

	Requirement	**Level of Implementation/Details**
	Networking Requirements	
N10	Multiple additional feature e.g. heartbeat, peer-to-peer implementation, or other functionality agreed with your tutor (replaces N8).	
	Security Requirements	
S6	Multiple additional feature e.g. data encrypted on storage, user password resets, account lockouts (replaces S5).	

List any other features implemented	**Details**

UNIVERSITY OF DERBY

Example: Simple DNS System

Submission requirements (all non non-submissions)		
	Requirement	**Level of Implementation/Details**
	Submission requirements	
R1	Viva attended	Yes XX/YY/ZZZZ-AA:BB
R2	Checklist provided to tutor at viva	Yes
R3	Submission is a single zip file	Yes
Basic Requirements Required for Pass (40% or greater)		
	Submission requirements	
R4	Submission contains 'Client' and 'Server' folders inside the zip file	Yes
R5	Code runs on any computer in MS214 or MS215.	Yes, no hardcoded paths
R6	Solution interprets/builds without any errors or warnings	Yes
R7	Implementation Log provided in zip file	Yes
R8	Feature checklist provided in zip file	Yes
	Networking Requirements	
N1	Synchronous bidirectional communication between client and server.	Yes, asynchronous see N6
N2	Connection and disconnection are handled without errors on the server side.	Yes, errors are handled gracefully, client and server do not crash
N3	Error handling and message content verification are handled on the server side.	Yes, see N5
	Security Requirements	
S1	Network traffic is encrypted using a standard algorithm using [a pre-shared, entered, or non-negotiated key].	RSA implemented using library and simple pre-shared key "DERBY"
	Application Requirements	
A1	Implements at least 2 states that alter the behaviour of the system based on user input.	Retrieve A records (success: A record, fail: error) Add new A record (success: A record, fail: error)
Additional Features		
	Networking Requirements	
N4	Multiple clients supported by a single server.	Separate thread supports each new client connection
N5	Error handling and message content verification are handled on both the client and server side (replaces N3)	Server: success/fail on messages, invalid message Client: test for valid A record format
	Security Requirements	

S2	Key is negotiated using an appropriate mechanism such Diffie-Hellman (modifies S1 component in [])	Diffie-Hellman key exchange implemented – key negotiated on connection, option to still fall back to “DERBY”
	Application Requirements	
A2	Implements at least 1 complex state that allows data to be stored between sessions.	Add record saves new A record, retrieved on server restart, instantly accessible to other clients
Additional Features Required for Grade of 60% or Greater		
	Networking Requirements	
N6	Asynchronous bidirectional communication between client(s) and server (replaces N1).	Yes, threaded implementation model
	Security Requirements	
S3	Packets implement mechanisms to prevent at least 1 kind of attack e.g. replay attacks.	Unique ID added to each packet sent
	Application Requirements	
A3	Implements at least 1 complex state that alters typical network communication pattern (e.g. client can send an arbitrary number of messages in a row) .	Implementation of iterative C-Name lookup (no further user input required between entries)
Additional Features		
	Networking Requirements	
N7	Shared state information between multiple clients and server.	Administrative mode allows visibility of connected clients, and identifies queries they have submitted this session.
N8	Network code is provided as a layered solution separated from the application logic.	Yes, network layer has “getMessage”, “messageExists”, and “sendMessage”
	Security Requirements	
S4	AAA implemented (authentication, authorisation, and auditing) e.g. user privilege levels.	Default mode, authenticated user, admin Authenticated users/admin can add new records All interactions are recorded to an interaction log
	Application Requirements	
A4	Implements an appropriate way to view or access AAA information.	Admin level has “view <user>”, or “view”
Additional Features		
	Networking Requirements	
N9	Single additional feature e.g. heartbeat, peer-to-peer implementation, or other functionality agreed with tutor.	Heart-beat implemented (30s timeout from initial connection)
	Security Requirements	
S5	Single additional feature e.g. data encrypted on storage, user password resets, account lockouts.	Reset password uses SMTP lib to send email to registered account with a unique code.
Additional Features		

	Networking Requirements	
N10	Multiple additional feature e.g. heartbeat, peer-to-peer implementation, or other functionality agreed with your tutor (replaces N8).	Auto-reconnect implemented after timeout using same login details + new key.
	Security Requirements	
S6	Multiple additional feature e.g. data encrypted on storage, user password resets, account lockouts (replaces S5).	Data is encrypted on disk

Example 2: Choose your own adventure

Submission requirements (all non non-submissions)		
	Requirement	**Level of Implementation/Details**
	Submission requirements	
R1	Viva attended	Yes XX/YY/ZZZZ-AA:BB
R2	Checklist provided to tutor at viva	Yes
R3	Submission is a single zip file	Yes
Basic Requirements Required		
	Submission requirements	
R4	Submission contains 'Client' and 'Server' folders inside the zip file	Yes
R5	Code runs on any computer in MS214 or MS215.	Yes, no hardcoded paths
R6	Solution interprets/builds without any errors or warnings	Yes
R7	Implementation Log provided in zip file	Yes
R8	Feature checklist provided in zip file	Yes
	Networking Requirements	
N1	Synchronous bidirectional communication between client and server.	Yes, asynchronous see N6
N2	Connection and disconnection are handled without errors on the server side.	Yes, errors are handled gracefully, client and server do not crash
N3	Error handling and message content verification are handled on the server side.	Move to room (success: room, fail: error) Combat (success: combat result, fail: error) See N5
	Security Requirements	
S1	Network traffic is encrypted using a standard algorithm using [a pre-shared, entered, or non-negotiated key].	RSA implemented using library and simple pre-shared key "DERBY"
	Application Requirements	
A1	Implements at least 2 states that alter the behaviour of the system based on user input.	Retrieve A records (success: A record, fail: error)

		Add new A record (success: A record, fail: error)
Additional Features		
	Networking Requirements	
N4	Multiple clients supported by a single server.	Separate thread supports each new client connection
N5	Error handling and message content verification are handled on both the client and server side (replaces N3)	Server: success/fail on messages, invalid message Client: cannot use items that do not exist in current inventory.
	Security Requirements	
S2	Key is negotiated using an appropriate mechanism such Diffie-Hellman (modifies S1 component in [])	Diffie-Hellman key exchange implemented – key negotiated on connection, option to still fall back to "DERBY"
	Application Requirements	
A2	Implements at least 1 complex state that allows data to be stored between sessions.	Add record saves new A record, retrieved on server restart, instantly accessible to other clients
Additional Features		
	Networking Requirements	
N6	Asynchronous bidirectional communication between client(s) and server (replaces N1).	Yes, threaded implementation model
	Security Requirements	
S3	Packets implement mechanisms to prevent at least 1 kind of attack e.g. replay attacks.	Unique ID added to each packet sent
	Application Requirements	
A3	Implements at least 1 complex state that alters typical network communication pattern (e.g. client can send an arbitrary number of messages in a row) .	Combat now allows target selections and attack sequences e.g. Attack troll with knife, dodge, attack goblin with hammer
Additional Features		
	Networking Requirements	
N7	Shared state information between multiple clients and server.	Messages can be shared between players in the same room using "Push <message>".
N8	Network code is provided as a layered solution separated from the application logic.	Yes, network layer has "getMessage", "messageExists", and "sendMessage"
	Security Requirements	
S4	AAA implemented (authentication, authorisation, and auditing) e.g. user privilege levels.	Default mode, authenticated user, admin Authenticated users/admin can add new records All interactions are recorded to an interaction log
	Application Requirements	
A4	Implements an appropriate way to view or access AAA information.	Admin level has "view <user>", or "view"

Additional Features		
	Networking Requirements	
N9	Single additional feature e.g. heartbeat, peer-to-peer implementation, or other functionality agreed with tutor.	Heart-beat implemented (30s timeout from initial connection)
	Security Requirements	
S5	Single additional feature e.g. data encrypted on storage, user password resets, account lockouts.	Reset password uses SMTP lib to send email to registered account with a unique code.
Additional Features		
	Networking Requirements	
N10	Multiple additional feature e.g. heartbeat, peer-to-peer implementation, or other functionality agreed with your tutor (replaces N8).	Auto-reconnect implemented after timeout using same login details + new key.
	Implement the designed protocol on IoT devices (e.g., Raspberry Pi, Arduino, or any suitable hardware).	Write code for both the sender and receiver sides of the communication. Ensure that the implementation adheres to secure coding practices and follows security best practices.
	Security Requirements	
S6	Multiple additional feature e.g. data encrypted on storage, user password resets, account lockouts (replaces S5).	Data is encrypted on disk

Anonymous Marking

You must submit your work using your **student number** to identify yourself, not your name. You must not use your name in the text of the work at any point. When you submit your work in Turnitin you must submit your student number within the assignment document and in the *Submission title* field in Turnitin. A video showing how to do this can be found here (link)

Assessment Regulations

The University's regulations, policies and procedures for students define the framework within which teaching and assessment are conducted. Please make sure you are familiar with these regulations, policies and procedures.