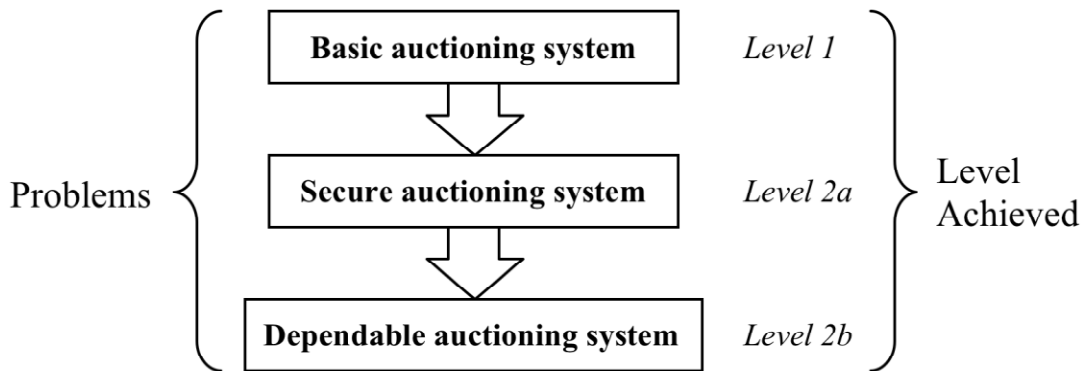


## Coursework for SCC 311: Auctioning System

### *1. Description of the Problem*

You are asked to design and implement a simple *distributed auctioning system* using Java RMI. Crucially, this overall problem is organised into three sub-problems of increasing difficulty: basic auctioning system (level 1), secure auctioning system (level 2-a), and dependable auctioning system (level 2-b). Each problem builds on the previous one. The level you reach determines the maximum mark you can possibly obtain for the coursework. How well you reach a level determines your actual mark. Achieving level 1 satisfactorily is sufficient to obtain a reasonably good 2.2 mark if you completed the project retrieval exercise. Precise marking details are provided at the end of this document.



### *2.1 Basic Auctioning System (Level 1)*

You are asked to implement a distributed auctioning system. This auctioning system should consist of a central auctioning server and two separate client programs:

- The *first client* program should enable a seller to create a new auction for an item offered for sale. The seller should be able to provide a starting price and a minimum acceptable price (reserve price). Creating the auction will return an auction id. At some point in the future, the seller should be able to close the auction using the same client program, by quoting the auction id. When the seller closes an auction, the client program should either indicate who the winner is, along with their details (see below), or else indicate that the reserve price has not been reached.
- The *second client* program should enable potential buyers to bid for auctioned items. Firstly, the program should enable buyers to browse the list

of currently active auctions with their current highest bid (but not the reserve price, which is secret). The client program should then enable a buyer to bid for a selected item, by entering the buyer's details (name and e-mail).

- The *auctioning server* should then deal with requests from the two client programs and maintain the state of the ongoing auctions.

## ***2.2 Advanced Features for Auctioning System***

Now, suppose this Auctioning system is a big success, and attracts a dramatic increase in traffic, along with threats from a powerful global crime organisation. You are asked to *consider* and *implement* techniques to enhance the *security*, *scalability* and *availability* of your system

### **Level 2a: Secure Auctioning System**

The implementation of level 1 is insecure: for instance, it has no mechanism to protect the system against external attacks such as the interception of messages (thus violating the confidentiality of the reserve price for instance), tampering with bids (one buyer modifying or stopping another buyer's bid), or bidding in another buyer's name.

Modify your system to provide (i) authentication; (ii) secure communication, and (iii) encryption mechanisms to ensure that users cannot tamper with other users' bids and auctions. Note that for authentication you are not expected to implement any key distribution mechanism (you may assume that any identification keys you might find necessary have already been securely distributed).

You are advised to use the lecture notes from "Technical Stream 2: Security in Java RMI". This explains the foundations that the systems should be based on for authentication. "Fundamental Stream Session: Security" for descriptions of the encryption methods which should be used for your implementation.

### **Level 2b: Dependable Auctioning System**

To ensure a larger scale use of the auctioning system you are required to enhance the *availability* of your system by using replication techniques. You should implement an active replication system to meet these requirements (thereby increasing both dependability and scalability). You are advised to use the *JGroups* framework to complete this part of the exercise. This can provide a useful foundation for active replication through its reliable group communication service (think about what problems this solves for an active replication system).

**Note:**

1. *You are not expected to provide a sophisticated user interface for the two client programs (a simple text based interface will suffice).*
2. *Crucially, this overall problem is organised into three sub-problems of increasing difficulty: basic auctioning system (level 1), and secure & dependable auctioning system (levels 2a & 2b).*
3. *Each problem builds on the previous one. The level you reach determines the maximum mark you can possibly obtain for the coursework. How well you reach a level determines your actual mark.*

**3. Marking Scheme**

The coursework mark for SCC 311 will be calculated as follows:

Overall Component	Additional Information	Weighting (%)
<b>Coursework Retrieval</b>	<b>Week 4</b>	<b>20%</b>
	<i>Use of username/password authentication</i>	10%
	<i>Or Use of key-based authentication</i>	20%
<b>Basic Server (Level 1)</b>	<b>Week 7</b>	<b>35%</b>
	<i>Completion of clients and server as specified</i>	30%
	<i>Quality of the solution</i>	5%
<b>Secure &amp; Dependable Server (Level 2a &amp; Level 2b)</b>	<b>Week 10</b>	<b>45%</b>
<i>Level 2</i>	<i>Secure communication + authentication</i>	22.5%
<i>Level 3</i>	<i>Implementation of Replication</i>	22.5%
<b>TOTAL</b>		<b>100</b>

This mark will then be combined with the exam mark in a ratio of 60% exam: 40% coursework to provide an overall grading for the course.

**4. Deadline, marking, and submission:**

**IMPORTANT NOTE:**

Your solution must be formally submitted through Moodle by **Tuesday (midnight) of week 7 for level 1 and of week 10 for levels 2a and 2b.**  
You will be asked to demonstrate your work from your Moodle submission.

1. Level 1 will be marked in the practical sessions of **week 7**. Levels 2a and 2b will be marked in the practical sessions of **week 10**.
2. **You need to demonstrate your work** to a member of the course team. Failing to do so will result in a grade of 0.
3. The demonstration must take place during the session you are allocated to. You cannot come to an earlier or later session for marking.
4. You need to come to the marking session **with your exercise completed**. We will not provide any support during marking sessions.
5. You should be able to explain what you have done clearly, to show that you understand the concepts introduced.
6. Checks for plagiarism and collusion will be carried out on all work (you are advised to see the university student handbook if you're not sure about the potential consequences of plagiarism).