

5 Further Java (arb33)

A programmer designs a client-server booking system for a meeting room. The role of the server is to distribute bookings between clients when they connect. Clients open a socket connection to the server regularly for a short period. When a client connects, the client first sends to the server an instance of `Message` which contains any new bookings made by the client; in response, the server sends an instance of `Message` containing all bookings made by other clients since the client last connected; the server then closes the connection. The key parts of the `Message` and `Booking` classes are defined as follows:

```
public class Message implements Serializable {
    private final String uniqueClientId;
    private final java.util.List<Booking> bookings;
    ...
}
```

```
public class Booking implements Serializable {
    private final String uniqueClientId;
    private final java.util.Date startTime;
    private final java.util.Date endTime;
    private final String description;
    ...
}
```

- (a) Write a Java implementation of the server, using a single thread to serve each client in turn. You may assume the existence of a static method `processBookings`, which accepts a list of new bookings from a specified client and returns a list of bookings to be sent back to the client. You may assume suitable accessor methods for `Message` and `Booking`; you do not need to handle exceptions. [8 marks]
- (b) The programmer decides to extend the booking system with *vector clocks*.
- (i) Write down a suitable data structure for a vector clock in Java. [2 marks]
- (ii) Describe in words how the system can be modified to incorporate vector clocks and allow clients to compute a partial order of `Message` objects. Discuss how vector clocks are initialised and updated. [6 marks]
- (iii) The programmer wants to use vector clocks to determine which booking occurred first, allowing clients to mark any subsequent bookings as in conflict and therefore cancelled. Describe when the vector clock algorithm cannot determine which booking is first, how this is detected, and propose a solution which resolves the ambiguity. [4 marks]