UNIVERSITY OF COLOMBO, SRI LANKA

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY

Academic Year 2021 – 3rd Year Examination – Semester 6

IT6505 – Middleware Architecture

Structured Question Paper

(TWO HOURS)

To be completed by the candidate

BIT Examination Index No: .....

Important Instructions:

- The duration of the paper is **2 (Two) hours**.
- The medium of instruction and questions is English.
- This paper has **4 questions** on 9 **pages**.
- **Answer all questions.** Questions carry **equal** marks.
- **Write your answers** in English using the space provided **in this question paper**.
- Do not tear off any part of this answer book.
- Under no circumstances may this book, used or unused, be removed from the Examination Hall by a candidate.
- Note that questions appear on both sides of the paper.
  If a page is not printed, please inform the supervisor immediately.
- Calculators are **not** allowed.
- *All Rights Reserved.*

Questions Answered

Indicate by a cross (×), (e.g. [X]) the numbers of the questions answered.

	Question numbers				
To be completed by the candidate by marking a cross (×).	1	2	3	4	
To be completed by the examiners:					

Index No …….

1) (a) Define a *Remote Procedure Call (RPC) and briefly explain its relevance in distributed programming.* (5 Marks)

ANSWER IN THIS BOX

RPCs allow client programs to transparently issue calls to procedures defined by remote server programs. The complexity of communication is completely hidden from the application by stubs that mimic the interface of the procedure calls.

(b) Explain the difference between the concepts of *syntactic transparency* and *semantic transparency* in relation to Remote Procedure Calls (RPC). (6 Marks)

Syntactic transparency: A remote procedure and a local procedure call should have the same syntax.
Semantic transparency: The semantics of a remote procedure call and local procedure call are identical.

(c) Define each of the following *non-functional Requirements* of Distributed Systems. (8 Marks)

i. *Scalability*: Accommodate a growing load by adding hosts.

ii. *Openness*: Easily extended and modified due to well defined interfaces.

iii. *Heterogeneity* : Supports various H/W and S/W platforms, languages, network protocols, etc.

iv. *Fault-Tolerance* : Ability to function correctly even if faults occur

(d) Explain the difference between failure types~~,~~ *fail-silent failures* and *fail-stop failures.*

(6 Marks)

fail-silent: in the halting failures model, a process either works correctly, or simply stops and crashes without performing incorrect actions.
Fail-stop failures: in the fail -stop failures model, in addition to the process crashing without performing incorrect actions, processes that are interacting with the faulty process have an accurate way to detect such failures.

2)

(a) What is meant by the term *data access middleware*?

(3 Marks)

Data Access Middleware: This type of middleware is characterized by the interaction of the application with local and/or remote databases (legacy, relational and non-relational), data warehouses, or other data source files.

(b) State *three (03)* practical disadvantages of the concept of *Remote Procedure Calls (RPC)* in distributed application design.

(6 Marks)

- Remote Procedure Call Passes Parameters by values only and pointer values, which is not allowed.
- Remote procedure calling (and return) time (i.e., overheads) can be significantly lower than that for a local procedure.
- This mechanism is highly vulnerable to failure as it involves a communication system, another machine, and another process.
- RPC concept can be implemented in different ways, which is can't standard.

- Not offers any flexibility in RPC for hardware architecture as It is mostly interaction-based.
- The cost of the process is increased because of a remote procedure call

(c) Briefly explain each of the following eight *components* of the *middleware* paradigm.

(12 Marks)

i. *Communication Link*: Physical communication link for Local or Wide Area.

ii. *Protocols*: Two categories of protocols; Network and Middleware. Together provide reliability, performance. Different middleware protocols provide a wide variety of connection types such as Response time and Guarantee of delivery.

iii. *Programmatic interface*: This specifies the way the data are presented to the middleware

iv. *Common format for data*: Describes how the data should be structured so that both end will understand it

v. *Server Process Control:* This dictates how the Middleware, O/S and other S/W manage the scheduling and execution of applications

vi. *Naming and Directory Services*: Provides means for locating the communication elements

vii. *Security*: Ensures the communication between both ends is safe enough to meet requirements

viii. *Systems management*: Concerns with configuration, operation, fault management, performance of environment

(d) List *two (02)* reasons for you to use *object oriented middleware* instead of RPC in distributed application development.

(4 Marks)

- Exposing an object interface through middleware is more natural and simpler to them than exposing a non-object interface.
- The interface can be delinked from the server program
- Can use exactly the same interface even though the underlying implementation is completely different.
- If there is a change to the interface, this can be handled incrementally by adding an interface to an object rather than changing the existing one.
- Having both old and new interfaces allows the clients to be moved gradually rather than all at once.

Any correct answer will be given 02 marks

3)

(a) List *two (02) HTTP methods* you can use in *Restful applications* as well as in *SOAP applications*. Describe the purpose of using each of those methods.

(4 marks)

POST, GET methods.

GET is used to request data from a specified resource.
POST is used to send data to a server to create/update a resource.

(b) The following piece of code was taken from a *Controller* class of a Restful backend application.

```
Student student = studentService.findAdressById(studentId);
```

Briefly explain the expected functionality of the above *service call*.

(4 marks)

This service call will search for a particular student record in the database to a given ID and return a Student object. Based on the method name we can assume that address field must be included in the resulting object.

c. RESTful resources are identified by resource URIs which make REST extensible. Describe the *type of resources* represented by the three (03) *URI*s below.

(6 marks)

`/v1/coffees` : This is used to represent all the coffees that are sold by a coffee shop. V1 represents the first version of the backend.

`/v1/coffees/orders` :This is used to represent all the coffees that are ordered

`/v1/coffees/orders/123` :This is used to represent a single order of coffee identified by "

d. Briefly explain the functionality of the *SOAP message* given below.

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi=http://www.w
3.org/2001/XMLSchema-instance
xmlns:s="http://www.w3.org/2001/XMLSchema">
      <SOAP-ENV:Body>
            <DivideIntegerResponse xmlns="http://abc.org">
                  <DivideIntegerResult>56</DivideIntegerResult>
            </DivideIntegerResponse>
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

(5 marks)

This SOAP message is a response message. It delivers the value of "56" as the data. Most likely "56" is the result of a calculation where one integer was divided by another integer number.

e. Briefly explain *three (03)* benefits of *REST* over *SOAP* in developing distributed applications.

(6 marks)

–REST allows a greater variety of data formats, whereas SOAP only allows XML.
–Coupled with JSON (which typically works better with data and offers faster parsing), REST is generally considered easier to work with.
–Thanks to JSON, REST offers better support for browser clients.
–REST provides superior performance, particularly through caching for information that's not altered and not dynamic.
—REST is generally faster and uses less bandwidth. It's also easier to integrate with existing websites with no need to refactor site infrastructure.

4. a. Assume that your company is developing a software solution for an *automated garment showroom* which has *robot assistants* and automated *self-checkout machines* in which all the operations and processes are automated and there are no human staff.

The overall system is supposed to have the following features:

- Through the surveillance cameras, the proposed system will track customers (identify group/individual, body size and dimensions, age, gender, etc.) and learn their 'purchasing intentions' (i.e., what they want to buy).
- After learning the customer's 'purchasing intension', a robot assistant will approach the person to provide help.
- Previous purchasing records would also be used to deliver 'enhanced customer experiences'.
- Customers can check out the items and settle the payment using self-checkout machines.
- Video feeds, order processing system, machine learning algorithms, and data are hosted/stored either in company owned servers or in cloud services.

i. Draw a *high-level block diagram of the overall system (architecture diagram)* identifying and labeling key *hardware and software components*.

(8 marks)

Students should illustrate the key components of the system such as robots, surveillance cameras, cashier counters, key software processing elements (order processing application, pattern recognition, Machine Learning components), other sensor networks, and database. They should also illustrate how communication should take place between these elements.

ii. Identify appropriate *technologies* that could be used to implement each *software component* of this system. Briefly explain your *reasons* for the selections.

(6 marks)

Students should identify suitable technologies for critical components such as databases, order processing application, video analyzing and machine learning, build tools, etc.

b. Briefly discuss *three (03)* practical benefits of using *in-process servers (DLLs)* in DCOM.

(6 marks)

In-process servers (Dynamic Link Libreries or DLLs) can be dynamically load into the program when they are required.
Practical Usages:

- No additional requirement of memory at runtime or disk storage as a common binary file will be there,
- All the application or clients will be using one single version of the common code provided by the DLL,
- Operating system loads only one instance of the DLL when the first application/client loads it then for every subsequent application it shares the memory pages of the DLL

with their process address space. Thus, there is no unnecessary memory overload in the system.

c Briefly explain the functionality and use of the *Dynamic Invocation* feature in CORBA.

(5 marks)

When using dynamic invocation, the CORBA client application can dynamically build operation requests for a CORBA object interface that has been stored in the Interface Repository. This supports deferred synchronous communication, where the CORBA client application sends the request and it is free to do other work. Dynamic Invocation also enables a CORBA client application to invoke a method on a CORBA object whose type was unknown at the time the CORBA client application was written.

****