

PROGRAMME: B.SC (HONS) IN SOFTWARE DEVELOPMENT

YEAR/STAGE: 4
TIME:

MODULE: DISTRIBUTED SYSTEMS

INTERNAL EXAMINERS: Mr. John Healy

EXTERNAL EXAMINERS: Mr. Tom Davis
Dr. Michael Schukat

TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES:

ATTEMPT ANY 4 QUESTIONS

Attachments: Yes ☐ No ☒ **If yes, please list details:**

Special Requirements: Yes ☐ No ☒ **If yes, please list details:**

Calculators Permitted: Yes ☐ No ☐ **Not applicable** ☒

1. (a) Explain, using diagrams and examples where appropriate, the following terms as they apply to distributed systems:

- Heterogeneity (3 Marks)
- Transparency (3 Marks)
- Marshalling (3 Marks)

- (b) *“Inter-process communication mechanisms based on synchronous method invocations lack the scalability required from enterprise systems.”*

You are required to provide a critique of this statement. Your answer should address the limitations of distributed systems based on **synchronous** method calls and discuss how **message queues** can be employed to increase scalability. Illustrate your answer with diagrams where appropriate.

(16 Marks)

2. *“Next generation (NoSQL) databases provide a level of distribution and scalability not possible with the traditional relational model.”*

Discuss this statement. Your answer should provide a critique of the **Relational Model** as a distributed system and address how **Wide Column**, **Tuple Store** and **Graph Database** technologies can be exploited to create a scalable distributed system. Use diagrams, where appropriate, to illustrate your answer.

(25 Marks)

3. (a) *“Java Remote Method Invocation provides a robust framework for passing object parameters, both by value and by reference, between remote Java Virtual Machines.”*

Discuss this statement. Your answer should be accompanied by a diagram illustrating the component parts involved in the process.

(12 Marks)

- (b) Explain how **garbage collection** over remote Java Virtual Machines (JVMs) manages references to remote method arguments and return types that are passed by reference.

(8 Marks)

- (c) Briefly describe how Java RMI can be used to provide an **object façade** to a suite of server-side objects. Your answer should include a brief discussion of the rationale for applying such an approach.

(5 Marks)

4. “The choice between Web Services and REST is a vital architectural decision for integration projects”

Discuss this statement. Your answer should address the intent and the key components of both **RESTful** services and the **Service Oriented Architecture** (SOA). Illustrate your answer with diagrams where appropriate.

(25 Marks)

5. The system architecture depicted in **Figure 1** is used by a web-based messaging application. The application, deployed on Apache Tomcat, allows users to send messages to other users using an Applet-plugin rendered in a web browser.

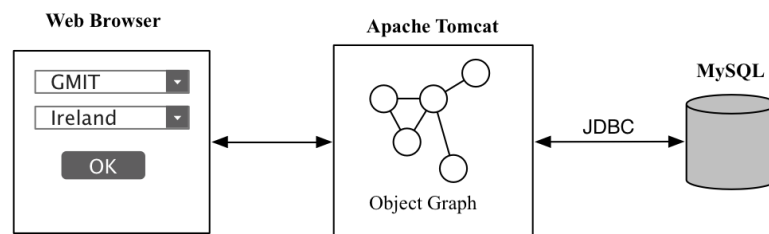


Figure 1

You are required to provide a re-design of the system that provides the following:

1. The system should be **highly scalable**. The performance of the current implementation degrades significantly as the number of concurrent users increases.
2. The system should **support a diverse range of Tier 1 client types**, including Android, iPhone, Windows Mobile and different desktop operating systems. The system should also be capable of supporting a variety of **different database models** at Tier 3.
3. The system should be flexible enough to enable peer-to-peer communication at Tier 2, by **exposing APIs** that are remotely accessible from applications written in different programming languages.
4. Where possible, **open or de facto standards** should be used to increase system flexibility and extensibility.

Your answer should include a fully labelled **diagram** of the new system architecture, along with a description of the role of each constituent technology.

(25 Marks)

6. (a) Using a fully labelled diagram, describe the main components of a **CORBA orb** and their role in the CORBA framework. Include in your answer a description of the services provided by the CORBA object adapter.

(12 Marks)

- (b) Figure 2 below describes two Java interfaces that provide a simple abstraction of a message and a messaging system:

```
public interface Message{
    public void setSender(String name, int id);
    public String getSender(int id);
    public void setReceiver(String name, int id);
    public String getReceiver(int id);
    public int getReceiverId(String name);
    public boolean deleteReceiver(int id);
}

import java.util.Vector;
public interface MessagingSystem{
    public void sendMessage(Message message) throws MessageException;
    public Vector getSentMessages(int senderId);
}
```

Figure 2

Show how **Interface Definition Language** (IDL) can be used to represent these interfaces in a CORBA application. You may assume that both interfaces are packaged using the namespace “gmit”.

(8 Marks)

- (c) Briefly describe the mechanism through which CORBA and Java RMI can be natively integrated together in a distributed system.

(5 Marks)