00210

Sri Lanka Institute of Information Technology

B. Sc. Special Honours Degree
in
Information Technology
Specialized in Software Engineering

Final Examination
Year 3, Semester I (2023)

SE3030 – Software Architecture

Duration: 02 Hours and 10 minutes

Instructions to Candidates:

- This paper is preceded by **10 minutes reading period**. The supervisor will indicate when answering may commence.
- This paper contains **Four** questions. **Answer All** Questions.
- Marks for each question are given in the paper.
- Total Marks: 100.
- This paper contains **6** pages with the Cover Page.

Question 01 (20 marks)

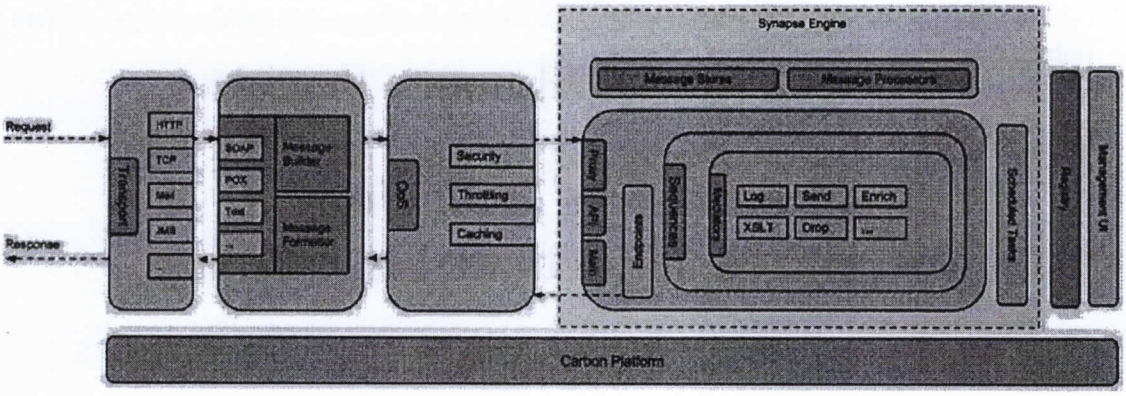
a) This question is based on the **Enterprise Application Integration (EAI)**.

i). Apply Enterprise Service Bus (ESB) with a practical scenario and explain it by designing a simple diagram.

(06 marks)

ii). Explain the following architecture diagram and discuss how **mediator pattern played a vital role** in here.

(06 marks)



b) Compare ***Micro-kernel Architecture*** with ***Micro-services Architecture*** and discuss the advantages and disadvantages by applying 04 examples.

(08 marks)

Question 02 (30 marks)

a) This Question is based on the implementation of **Presentation Layer** and **Business Layer patterns** that can be applied for **HealthCare Application** for accessing **Doctor** and **Hospital Services** through a common controller class called **FrontController.** This works as per the **Command pattern.** All Controllers follow **Common Life Cycles implementation** called **LifeCycleController** work according to the **Template method** and refer the **Test class** and the output of the console given in the below diagrams.
[You will be given the project Structure and sample code segments of Request, Response, and IService classes. You should attempt each sub section separately]

FrontControl... IService.java Request.java

```
1  package com.sa.reg.controller;
2
3  public class Request {
4
5      private String ID;
6
7      private String name;
8
9      public Request(String iD, String name)
10         super();
11         ID = iD;
12         this.name = name;
13     }
14
15     public String getID() {
16         return ID;
17     }
18
19     public String getName() {
20         return name;
21     }
22 }
```

Response.java

```
1  package com.sa.reg.controller;
2
3  public class Response {
4
5      private String result;
6
7      public Response(String result) {
8          super();
9          this.result = result;
10     }
11
12     public String getResult() {
13         return result;
14     }
15 }
16
```

```
9  public class Test {
10
11     public static void main(String[] args) {
12
13         LifeCycleController doctorController = new DoctorController();
14         LifeCycleController hospitalController = new HospitalController();
15
16         FrontController frontController = new FrontController();
17         frontController.setControllers(doctorController, hospitalController);
18
19         Request doctorReq = new Request("D1001", "Dr.Tom");
20         doctorController.template(doctorReq, "doctor");
21         frontController.getDoctorController(doctorReq);
22
23         Request hospitalReq = new Request("H2002", "Colombo General Hospital");
24         hospitalController.template(hospitalReq, "hospital");
25         frontController.getHospitalController(hospitalReq);
26     }
27 }
```

Console | Problems | Javadoc | Declaration

```
<terminated> Test (4) [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Apr 13, 2023, 10:31:35 AM)
Get Doctor ID = D1001, Doctor Name = Dr.Tom
Get Hospital ID = H2002, Hospital Name = Colombo General Hospital
```

i). Construct the code for **DoctorService** class according to the **Singleton design pattern** and you can implement **getBusinessService** method as follows.

(03 marks)

```
@Override
public Response getBusinessService(Request request) {
    return new Response("Get Doctor ID = " + request.getID()
    + ", Doctor Name = " + request.getName());
}
```

ii). Design **HospitalService** class according to the **Singleton design pattern** and there should be a method to reserve rooms as follows.

(03 marks)

```
@Override
public Response getBusinessService(Request request) {
    return new Response("Get Hospital ID = " + request.getID()
    + ", Hospital Name = " + request.getName());
}
```

iii). In the **ServiceDelegator class** you should **design a simple Factory** to get the services (**doctor or hospital**) by name using a method called getServiceByName(String service)

(03 marks)

iv). You should **design** the **FrontController** class and it has a method to set doctor and hospital controllers using **setController()** method and design **getDoctorController(Request req)** and **getHospitalController(Request req)** methods. Assume this **FrontController** class works according to the **Presentation Layer FrontController Pattern.**

(06 marks)

```
public void getDoctorController(Request req) {
    String result = this.doctorController.process(req).getResult();
    System.out.println(result);
}
```

v). Now assume you have to **construct** a **LifeCycleController** class according to the **Template method pattern** and assume you let **DoctorController** and **HospitalController** classes to implement the **two life cycle methods** init() and process() both expects **void init(String service), Response process(Request request),** and there should be a template method **void template(Request request, String service)** to call init method as first priority and the process method as the second priority.

(05 marks)

vi). **Construct DoctorController**, and **HospitalController** classes then in each class should extends the **LifeCycleController class** and override **init(service)** and **process(req)** methods with respective code segments. **In the init(service) method** you should access the Backend services through the **ServiceDelegator** class and access the service using **getServiceByName()** method and in the process(request) method access the business service directly.

[05 marks for DoctorController and 05 marks for HospitalController]

(10 marks)

Question 03 (25 marks)

a) Explain the **role** of a Software Architect.

(02 marks)

b) Describe what **Non-Functional Requirements** are and outline the **3 main categories** of Non-Functional Requirements.

(03 marks)

c) What is **4+1 View Model** and explain the focus areas of that model.

(05 marks)

d) Analyse **Cloud Architecture** style and describe the **different service offerings** of Cloud Architecture together with where those different offerings can be used.

(04 marks)

e) Explain **ATAM** (Architecture Trade-Off Analysis Method) and outline its main objectives and benefits.

(05 marks)

f) Analyse **Component-based Architecture** style and describe its challenges in build time and propose a method to mitigate the same.

(03 marks)

g) What are the tactics that would generally apply to improve the **Security** of online Banking Application.

(03 marks)

Question 04 (25 marks)

a) If you are asked to design a high-performance trading platform with multiple modules where modules would need to scale independently, explain what considerations you will prioritize when designing the inter system (module) communication. Propose a **design** that can help above and **explain** how it will improve the systems quality.

(05 marks)

b) Apply Quality Attribute scenarios for a pacemaker (medical device that helps to normalize patient's heart beat) embedded software for below Quality Attributes considering 06 factors.

i). Reliability

(03 marks)

ii). Security

(03 marks)

c) You are asked to re-architect a large system which consists of **multiple software sub-systems** which are directly connected to each other at present. As the business plans to expand into new areas; it is expected to integrate the newly build software with old/existing software. The system(s) need to have the ability

of **upgrading/integrating/replacing** each sub-system independently while other system(s) have the least impact on their operations.

i). Identify 2 key Quality Attributes in above.

(02 marks)

ii). Draw and explain how you would re-architect the above solution.

(03 marks)

iii). What are the Trade-offs in above proposal and discuss mitigation plans.

(03 marks)

d) A **cross-platform** mobile gaming application needs to be developed that requires ultra-high **responsiveness**. If you are to architect this system, how would you plan to implement the key architectural considerations of this system.

(06 marks)

----------End of the Paper----------