2018 June - DS

Q01

a) Students need to check exam results
Share time tables
If it is a stand alone - all crash
(distributed system have a backup)
reliability of the data by creating multiple versions.
To improve the capacity
(Accessibility is very limited in stand alone)

b) challenges

briefly explain. one or two sentense how to address.

Security → Password / encrypt
(Improve the security)

how to address ?

concurrency → Multithreading in server side
(Multiple students marking attendence same time)

how to address ?

reliability → can have multiple servers.
(In the rush hours there may be lot of crowd)

c) i) Justify why use the style?

i) layered / object based

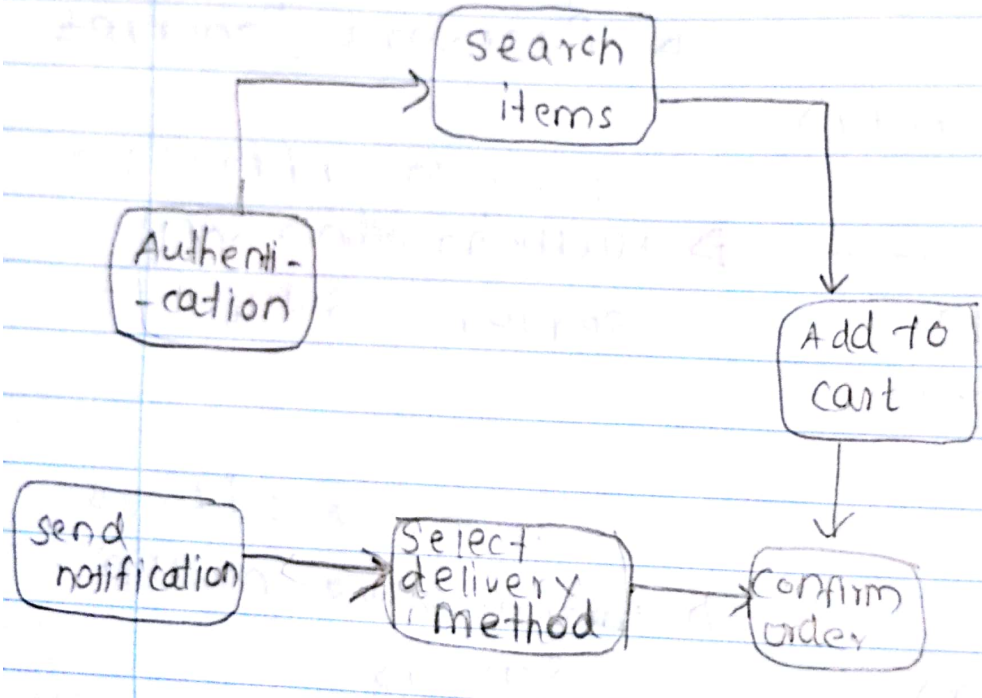
same one

ii) component based / object based.

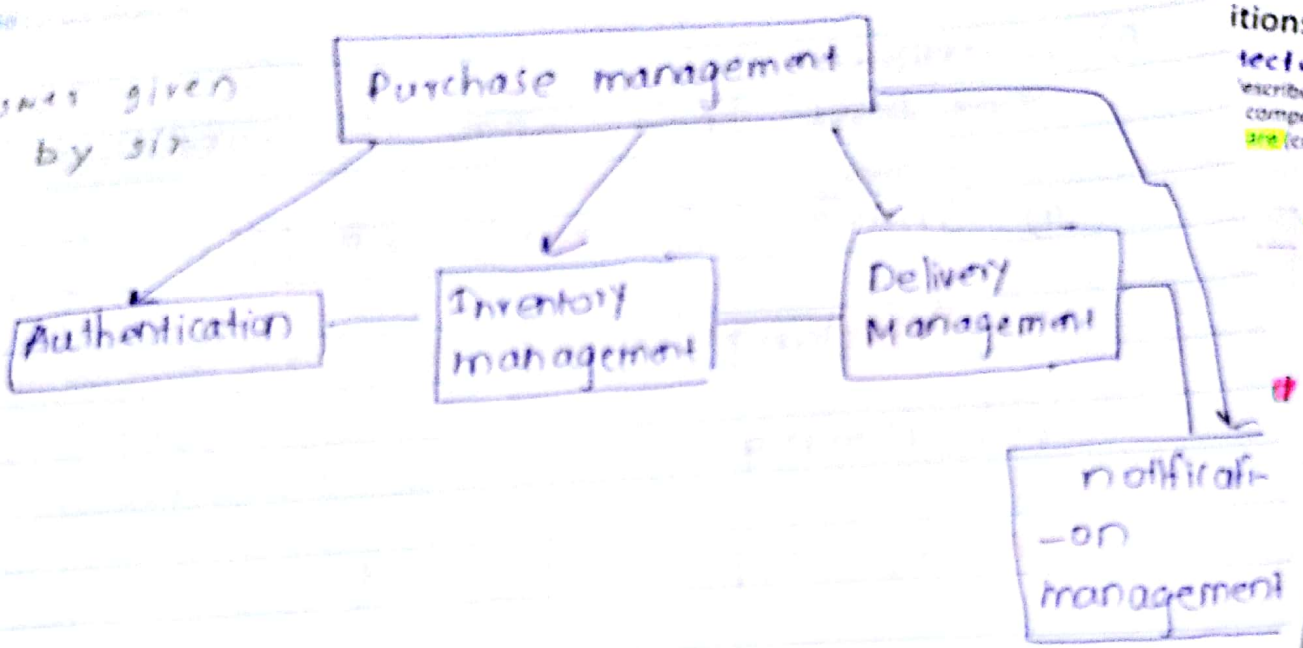
sensor layer
processing layer
} → so layered also correct

iii) Doesnt say anything about sendi notifications → Data centered.

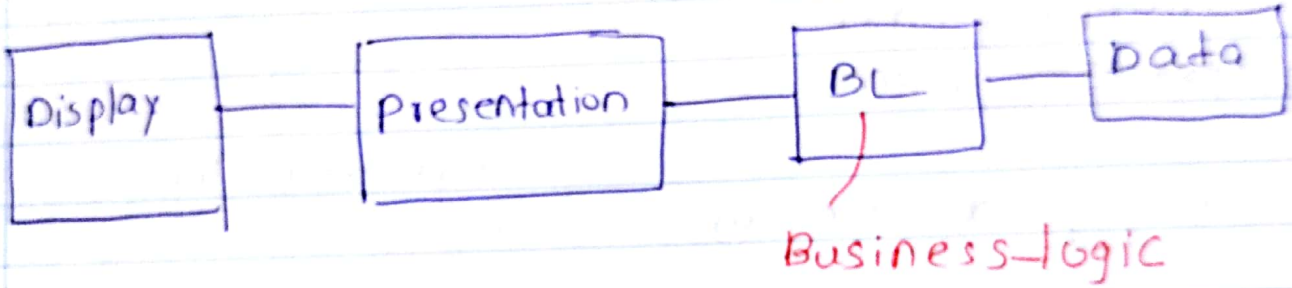
d)



Answer given by sir




ii)




4 tier architecture.

Q2

a) Remote interface defines the methods that are exposed to the client
Proxies → marshelling.
Stubs.

Refer lecture slides.

b) i) callback
You dont know when the rainfall occurs.

ii) ignore (socket) we didnt cover it

iii) Blocking

iv) Polling.

c) Identify types of EJBS

entity Bean

customer
shopping cart
grocery items

stateful Bean

session bean

message drive bean

JMS message.

Just write the different types of EJBs

Q3)

a) continuosly send the message until t
listen.
It supports Asynchonous communication
using call back.
Producer and consumer decoupling.
Act as a bridge to integrate differe
component

security question will be there

XML

b) Namespace → directly from slides
Schemas
Supported by JS

c) <xs:schema xmlns:xs="http://www
<xs:element name="name">
<xs:complexType>
<x

c) <xs:element name="customerlist">
<xs:complexType>
<xs:sequence>
<xs:element name="customer"
minOccurences="0" maxOccurrences="unbound"/>
</xs:sequence>
</xs:completeType>

d) Sample JSON object to represent multiple customer.

```
{
  "customer" : {
  "name" : {
   "firstName" : "Saman"
    "last name" : "Silva"
},
"email" : "saman@gmail.com",
"Phone : 07 7 1929771,
}
}
```

e) Each service we try to give one function then you can reuse it in many client application
Isolating functions

Q4) a) http://www.supermarket.com/customers- GET

http://www.supermarket.com/customer/0001 - GET

http://www.supermarket.com/customer/002 Delete

b) It can be use Orchestration

c) Auto scalling.
~~seasonal~~ load balancing - when the demand goes up.

No need to pay for many servers

d) i) IA
Infa structure.
Ans: (we have to have a virtual machine rented)

ii) Blackboard
cloud based system./software

iii) Private clouds - within the organization they can use virtual machine

iv) dropbox / google drive - cloud based storage.

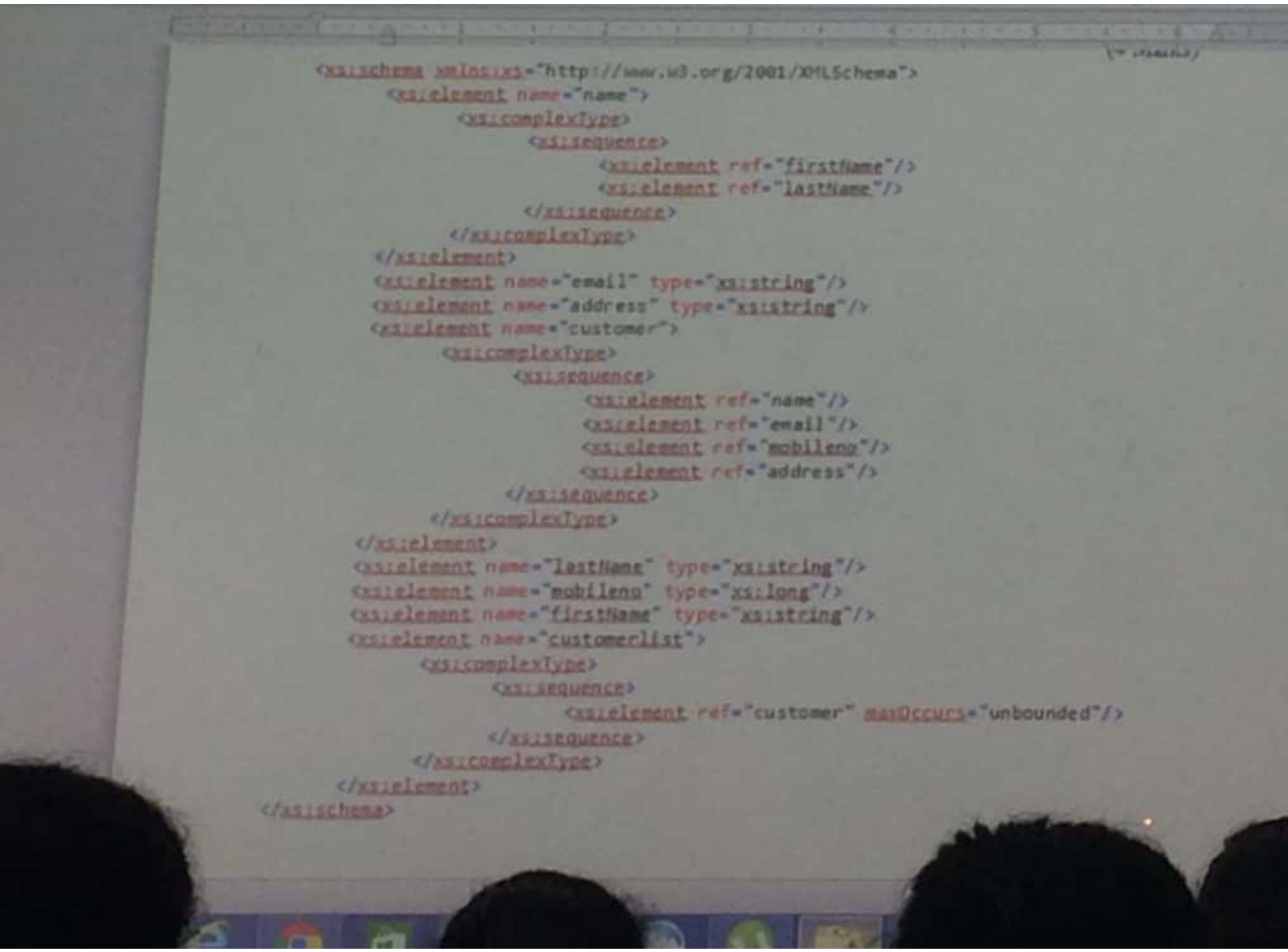
~~Git~~

```
        <xs:element name="lastName" type="xs:string"/>
        <xs:element name="mobileno" type="xs:long"/>
        <xs:element name="firstName" type="xs:string"/>
        <xs:element name="customerlist">
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="customer" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    </xs:schema>
```

d) For the above Customer data structure, write a sample JSON object to represent a customer.

(2 Marks)

```
{
 "customer": {
  "name": {
   "firstName": "Saman",
   "lastName": " Silva"
  },
  "email": "saman@saman.lk",
  "mobileno": "0777111111",
  "address": "20, Nugegoda, Colo
```


<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="name">
<xs:complexType>
<xs:sequence>
<xs:element ref="firstName"/>
<xs:element ref="lastName"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="email" type="xs:string"/>
<xs:element name="address" type="xs:string"/>
<xs:element name="customer">
<xs:complexType>
<xs:sequence>
<xs:element ref="name"/>
<xs:element ref="email"/>
<xs:element ref="mobileno"/>
<xs:element ref="address"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="lastName" type="xs:string"/>
<xs:element name="mobileno" type="xs:long"/>
<xs:element name="firstName" type="xs:string"/>
<xs:element name="customerlist">
<xs:complexType>
<xs:sequence>
<xs:element ref="customer" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

these operations. Indicate the appropriate HTTP method to use with the URL.

(5 Marks)

http://www.supermarket.com/customers - GET
http://www.supermarket.com/customer/0001 - GET
http://www.supermarket.com/customer/0002/Saman/dsd@gmail.com/07776723/223, Nugegoda,Colombo-3 – PUT
http://www.supermarket.com/customer/0002/Saman/eer@gmail.com/07776443/223, Kirulapane,Colombo-5 – POST
http://www.supermarket.com/customer/0002 - DELETE

b) Briefly explain two reasons for having Service Orchestration in Service Oriented distributed systems.

- The services themselves do not contain the business logic. Therefore, need an additional layer to define the business logic