HW#3

1 – SOAP vs Restful ?

- SOAP is a protocol, REST is an architectural style.
- SOAP allows XML data format only.
  REST allows different data format such as Plain text, HTML, XML, JSON etc.
- REST requires less bandwidth than SOAP.
- Providing security is faster and easier on SOAP than REST.
- REST works by using HTTP methods. GET, POST, PUT, DELETE etc.
  SOAP services use the RPC (Remote Process Call) working method, contain security protocols such as WS-*
- REST is more preferred than SOAP because it is easier to use for the most part and is more flexible than SOAP. Also the amount of data moved is less than SOAP.

2 - Difference between acceptance test and functional test ?

If your aim is to test only one function regardless of the environment and side effects; use functional testing.
If your aim is to check whether the expected operations from the software can be performed; use acceptance test.

3 - What is Mocking ?

We use mocking in unit testing. Mocking is creating fake objects that can replace an object we want. We can make these objects behave as we want.

Since unit test tests a unit, it ensures that the dependencies connected to this flow do not break the test flow while testing the flow there. While performing unit testing, it allows us to direct the test in the scenario we want.

Actual transactions can take a very long time. For example, a database operation can take a very long time. Considering that we have tens of hundreds of tests, we may have to wait for serious periods each time we test the software. Mock objects are very fast because they don't actually perform operations.

4 - What is a reasonable code coverage % for unit tests (and why) ?

Code coverage of 70-80% is a reasonable goal for system test of most projects with most coverage metrics. Use a higher goal for projects specifically organized for high testability or that have high failure costs.

Experimental studies on real projects have found that increasing code coverage above 70-80% is time consuming and therefore leads to a relatively slow bug detection rate. Your goal should depend on the risk assessment and economics of the project.

5 – HTTP/POST vs HTTP/PUT ?

POST and PUT are both HTTP methods used to send data to the server. POST is only used to send data to a specific resource and what to do with the data is up to the server. With PUT, the same source is accessed with the same address and if there is content, it is replaced with the incoming data, if there is no content, new content is created.

POSTing the same data twice means creating two identical data with different ids. PUTing the same data twice creates the user the first time and updates it to the same state the second time (no change).

6 - What are the Safe and Unsafe methods of HTTP ?

Safe HTTP methods: GET - HEAD - OPTIONS : Safe methods are "read-only" and they don't change the state of the server.

Unsafe HTTP methods: PUT, DELETE,POST : Unsafe methods change the state of an application.For example; to update a user's profile on a web application.

7 - How does HTTP Basic Authentication work ?

1. The user (client) sends a request to the server
2. The server sends the HTTP status code as 401 and the basic value WWW-Authenticate variable in the HTTP Header in the HTTP header.
3. When the user receives this request, an input field appears.
4. User enters username and password values here. The values here are sent with a colon between them and encrypted with base64.
5. The user sends this under the Authorization header under the HTTP protocol, as a basic value at the beginning.
   Authorization: Basic YWRtaW46cGFzc3dvcmQxMjMK

6. The server compares the base64-encrypted user name and password information in the Authorization header from the user with the base64-encrypted. If this comparison is true, it returns 200 HTTP status code. But if false it returns 401 status code.

8 - Define RestTemplate in Spring ?

RestTemplate is the default class in Spring library to handle synchronous HTTP requests on client side. It is used to create applications that consume RESTful Web Services.

9 – What is idempotant and which HTTP methods are idempotent ?

Idempotent means that you can safely repeat an operation. Idempotent methods do not have any side effects in the server state structure. Safe methods are idempotent.

GET – HEAD - OPTIONS – DELETE – PUT – TRACE methods are idempotent.

10 – What is DNS Spoofing ? How to prevent ?

DNS spoofing is a type of cyber attack that aims to trick internet users into sharing sensitive information by allowing them to reach a different site that looks like the original site instead of the one they target.

In DNS spoofing attacks, other than redirecting the traffic to the illegitimate server, different techniques can be applied, such as installing a virus that will cause immediate damage to the visitors' computers.

Any user that accesses the internet from public Wi-Fi is vulnerable to DNS spoofing. To protect from DNS spoofing, internet providers can use DNSSEC (DNS security). DNSSEC relies on public key encryption to make it possible to validate DNS data, which is not standard in existing internet protocols.

Safety precautions for users:

1. Never click on a link you don't know
2. Scan your computer regularly for malware
3. Clear your DNS cache to resolve the poisoning

11 – What is content negotiation ?

Content agreement between client and server. It is the customization of what type of data the user sends and what type of data the server accepts. The content type to be accepted is determined by the @Consumes notation, and the content to be produced is determined by the @Produces notation.

For example, a Header information specified as content-type:application/json indicates that the user requests JSON data.

12 – What is statelessness in RESTful Web Services ?

The server does not store any state about the client session on the server-side. This restriction is called **Statelessness**. Only the client holds such information. So, the server does not keep information such as how many requests the requesting client has made before or which requests. The client gives all the information the server needs in its request.

13 - What is CSRF attack? How to prevent ?

A CSRF attack involves a malicious link that sends a request to a web application through another previously authenticated website. With the identity information obtained, the identity of the victim is impersonated and the authentication information is bypassed in malicious activities.

How to prevent?

1. Token Usage
   The user is given random and unique "token" information for each session.
2. Using Post Method Instead of Get Method
3. Using CAPTCHA

Precautions To Be Taken By The User

1. Web application data and cookie information should be cleaned regularly.
2. Session information of web applications that contain personal information should not be stored on the computer.
3. Pay attention to e-mails and links of unknown origin.

14 - What are the core components of the HTTP request and HTTP response ?

Core components of the HTTP Request:

- HTTP Version – Indicates http version
- Request Body – Represents message content
- Request Header – Contains metadata, such as cache settings and client type
- URI – Identifies the resource on the server
- Verb – Indicates HTTP methods(GET, POST, and PUT)

Core components of the HTTP Response:

- HTTP Version – Indicates the version of HTTP
- Response Body – Represents the response message content
- Response Header – Contains metadata, like content length and server length, for the HTTP response message
- Status/Response Code – Indicates the server status for the requested resource.