HOMEWORK #3

1 – SOAP vs Restful?

SOAP is a protocol which was designed before REST. The main idea behind designing SOAP was to ensure that programs built on different platforms and programming languages could exchange data in an easy manner. SOAP stands for Simple Object Access Protocol.

REST was designed specifically for working with components such as media components, files, or even objects on a particular hardware device. REST stands for Representational State Transfer.

SOAP requires more bandwidth for its usage. REST does not need much bandwidth when requests are sent to the server. REST messages mostly just consist of JSON messages.

2 - Difference between acceptance test and functional test?

The functional test confirms the software performs a function within the boundaries of how you've solved the problem. A functional test verifies that the product actually works as you (the developer) think it does. Acceptance tests verify the product actually solves the problem it was made to solve. This can best be done by the user (customer), for instance performing his/her tasks that the software assists with.

3 - What is Mocking?

While testing, it is to create a setup and test the tested object and the services used by the class as if they were running. Mocks are useful if you have a dependency on an external system, file reading takes too long, the database connection is unreliable, or if you don't want to send an email after every test.

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

The rate at which the written unit tests give the expected outputs to the inputs. Code coverage of 70-80% is a reasonable goal for system test of most projects with most coverage metrics.

5 – HTTP/POST vs HTTP/PUT?

While sending a new data to a specific resource use POST, if we send it with save&update, it is created use PUT, if it is not created, it is changed if any. If it is getting the Id from the client, put is used, if it is from the server, post is used.

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

All safe methods are also idempotent, but not all idempotent methods are safe. Common safe HTTP methods GET, HEAD, OPTIONS. For example, PUT and DELETE are both idempotent but unsafe. In safe methods, there is no change in the server, it happens in unsafe methods

7 - How does HTTP Basic Authentication work?

Basically, an authentication based on username and password allows to be included in the application. It is based on base64-encoded credentials.

8 - Define RestTemplate in Spring ?

Rest Template is used to create applications that consume RESTful Web Services. The RestTemplate offers templates for common scenarios by HTTP method.

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

From a RESTful service standpoint, for an operation (or service call) to be idempotent, clients can make that same call repeatedly while producing the same result. In other words, making multiple identical requests has the same effect as making a single request. The following HTTP methods are idempotent: GET, HEAD, OPTIONS, TRACE, PUT and DELETE. All safe HTTP methods are idempotent but PUT and DELETE are idempotent but not safe.

10 – What is DNS Spoofing? How to prevent?

Domain Name Server (DNS) spoofing is an attack in which altered DNS records are used to redirect online traffic to a fraudulent website that resembles its intended destination. It can be prevented with using several methods; DNS Security Extensions (DNSSEC), Navigating the Internet safely is made easy by the secure connection symbol that indicates a page is authentic, Advanced DNS traffic filtering and Ensure that the DNS server you're using has been patched to the latest version.

11 – What is content negotiation?

In HTTP, ***content negotiation*** is the mechanism that is used for serving different representations of a resource to the same URI to help the user agent specify which representation is best suited for the user.

12 – What is statelessness in RESTful Web Services?

As per the REST architecture, a RESTful Web Service should not keep a client state on the server. This restriction is called Statelessness. It is the responsibility of the client to pass its context to the server and then the server can store this context to process the client's further request.

13 - What is CSRF attack? How to prevent?

Cross-site request forgery (also known as CSRF) is a web security vulnerability that allows an attacker to induce users to perform actions that they do not intend to perform. It allows an attacker to partly circumvent the same origin policy, which is designed to prevent different websites from interfering with each other. The most robust way to defend against CSRF attacks is to include a CSRF token within relevant requests. The token should be:

- Unpredictable with high entropy, as for session tokens in general.
- Tied to the user's session.
- Strictly validated in every case before the relevant action is executed.

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

Structure of HTTP Response:

- Status Line
- Headers
- Body (Optional)

Structure of HTTP Request:

- A request line.
- A series of HTTP headers, or header fields.
- A message body, if needed.