Answers:

**Q1:**

Trees represent an example of a data structure. A tree is a data structure where data is organized in hierarchically distributed nodes, with each child-node having one single parent node. A node that has no children is called leaf. At the top of the tree, there is a node with no parents, called root.

A good example of tree usage that I have in mind is in the case of file systems: a file system manager works with a tree, where nodes are folders and files (files can be only leaves).

Another example of a data structure I have in mind is the graph. In this case, data is organized in a set of nodes and one of arcs. Each arc is associated to two nodes, and defines a relationship between these. Generally, the maximum number of arcs between two nodes can be either one (defining a bidirectional relation) or two (each arc representing a direction).

Graphs can be used in modeling the behavior of an application. An example of graph used in this scope is the finite-state machine: a graph where each node represents a state (one of them being a beginning state), while each (unidirectional) arc has several inputs and/or outputs associated.

**Q2:**

The IP (V6) of tss-yonder.com is 2606:4700:20:0:0:0:681a:13e. This data is retrieved by the browser from DNS. The DNS is a system of servers that are associating IPs (and other data) with domain names. When you enter the domain name of the website on the browser, the browser requests from a DNS server the associated IP, and the DNS server provides the information to the browser (including the IP).

**Q3:**

Two examples of transport protocols are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

TCP does error checking and retransmits the data until it is correct. Thus, TCP is suitable for applications where the data integrity is important. One such application is a Key Management Server (KMS), which works with very sensible data (such as cryptographic keys).

On the other hand, UDP increases the speed, while reducing the checks made over data. Thus, it can be used in applications where speed is more important that the overall quality of data. Such applications are those used for video/audio meetings, such as Microsoft Teams.

**Q4:**

Supposing the security-related requirements are solved, one important thing that I’ll need for this application would be a device to run it on. Then, in order to run it, I would get a DNS domain name. Finally, I would find a way to promote it (such as a web site).

**Q5:**

My application is “already famous”. That means it is public, and any modification ether will involve stopping the application, or will result in a new, updated version.

One way to update the app is using an application where everyone can work, in parallel, on the same private code (such as GitHub). The original application is duplicated to a private repository and, we work on the copy, until the bugs are repaired. Then, the new version is made public.

Another way to update the app can be summarized so:

* each one takes a part of the program;
* each one works on his part;
* after everyone finishes, someone integrates the new parts in the application;
* finally, the updated app is made public.

**Q6:**

The way I see in solving this problem is by adding end-to-end encryption. This can be done by adding cryptographic functionalities to the endpoints, and/or by integrating a Key Management Service. Using a right KMS would help not only in encrypting the messages, but also in realizing key exchanges and implementing further security measures (such as key encryption).

**Q7:**

Cookies are data pieces, structured as *name-value* pairs, that are sent by web servers to browsers and stored locally by clients. They are resent to the servers when the client reconnects, and are used in various purposes, generally in relation with the client’s past activity with the server.

A cookie (*name* : *value*) from <http://www.tss-yonder.com> is:

***PHPSESSID*** : ***tkidfb3bmso2efl9dm2i9c2gfn***

It seems to store the session ID when communicating with the tss-yonder.com web server.

**Q8:**

In Java, I usually create child processes is by extending the *java.lang.Thread* class. The result inherits methods *void start()*, which triggers the execution of the process, and *void run()* that runs in the child process. For creating the child process, you should instantiate the class you created, and run its *start()* method.

**Q9:**

The PID of my app can be found in Task Manager, at the details section. The way I would debug it depends on the application and the error that it throws. However, any debugging process should be done based on the error message.

**Q10:**

Until now, I’ve usually used MySQL for storing data, as it is a satisfying DBMS and I could always resolve my DB needs with it. About storing passwords, I can say it depends a lot on the purpose of the user. For example, if I make a simple app for testing the connection to the DBMS, with an associated profile, I would store the password in-line; however, on more complex applications, I prefer to store the passwords in configuration files, maybe even encrypted (I haven’t tried this yet).