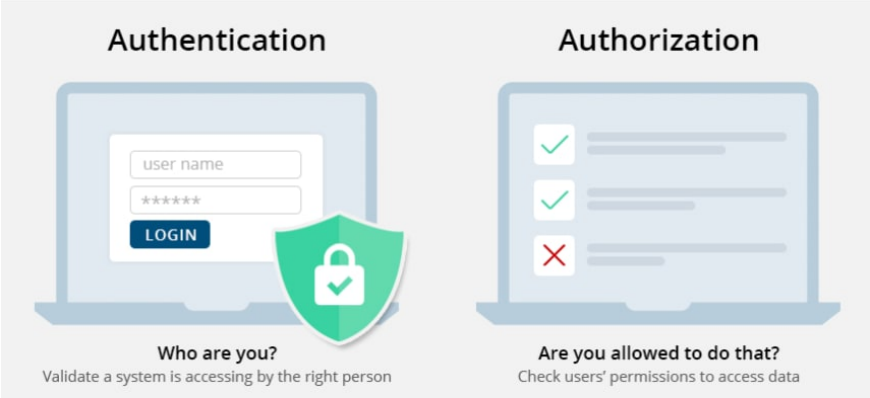
HOMEWORK4
FATMA BETÜL UYAR

1 – What are Authentication and Authorization ?

Authentication means an entity **proves its identity.** It validates a system is accessed by the right person. Usually used by entering username and password.

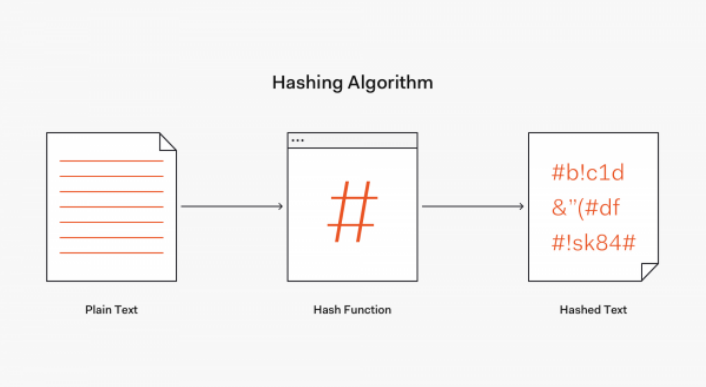
Authorization defines the **permissions** of the **authenticated** user on the resource they want to access.



2 - What is Hashing in Spring Security ?

It is the process of **taking an input** of any length and **converting it into an encrypted output** with a set of mathematical algorithms.

Hashing allows to greatly increase data security.



- **Different inputs** should **not give the same hash**
- The **same input** always returns the **same hash**
- It should be **impossible to identify** the **input by looking** at the hash
- The hash **value should change** when there is a **change in the input**

3 - What is Salting and why do we use the process of Salting ?

Salting is a unique value that can be added to the **end of the password** to create a different hash value.This adds a layer of security to the hashing process, specifically against brute force attacks.

In summary, we use salting because it increases security.

7X57CKG72JVNSSS9

7X57CKG72JVNSSS9SALT

4 - What is “intercept-url” pattern ?

- **hasRole, hasAnyRole:**They are responsible for defining the access control or authorization to specific URLs or methods in our application.
- **hasAuthority, hasAnyAuthority:**Roles have special semantics
- **permitAll, denyAll:** We either may permit access to some URL in our service or we may deny access.
- **isAnonymous, isRememberMe, isAuthenticated, isFullyAuthenticated**: We focus on expressions related to login status of the user.
- principal, authentication
- hasPermission

5 - What do you mean by session management in Spring Security ?

- **Detect Session Timeout**

If someone tries to access it when the session times out it may be necessary to redirect to a url like login.On the other hand,Spring security detects an **invalid session** ID and **redirects to another url.**

```
<session-management invalid-session-url="/login" >
```

If someone logs out and then tries to log in again, it will still be considered an invalid session because cookies are present in the browser.So, we can delete.

```
<logout delete-cookies="JSESSIONID" />
```

- **Concurrent Session Control**

It means a user can have multiple sessions at the same time.

```
<concurrency-control max-sessions="1" error-if-maximum-exceeded="true" />
```

6 – Why we need Exception Handling ?

Exception handling **maintenance a program’s flow**.
I mean,when an error occurs the compiler will skip only one line of the code where the exception occurs, and the rest of the code will be executed in the natural flow of the program.

Another benefit is that it provides a **user-friendly** program.
For example, if programmers do not use exception handling in code, when an error occurs, the user can not understand anything.
So, if we use exception handling, we can show a special message about exception and the user can understand the cause of exception.

7 - Explain what is AuthenticationManager in Spring security ?

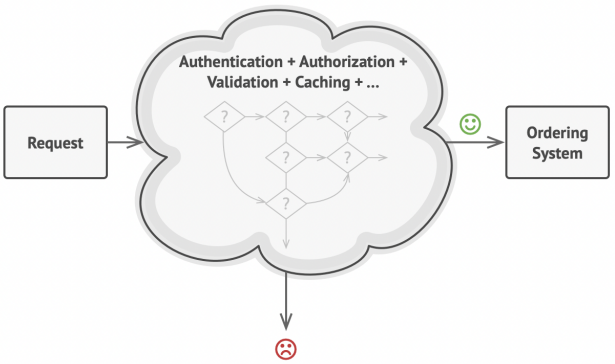
AuthenticationManager the main strategy interface for authentication.
It can do 3 things;

- If it can verify that is a valid user, return an Authentication
- If it believes that an invalid principal, throw an AuthenticationException
- If it cannot decide, return null.

If we want , we can customize that.

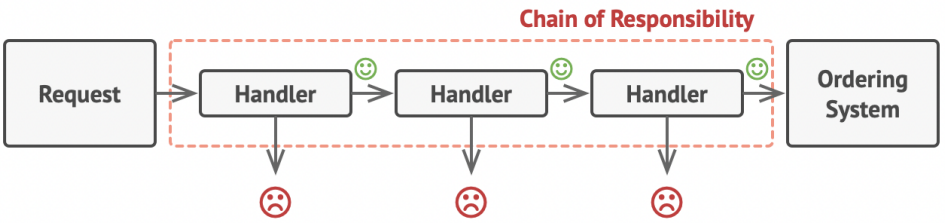
8 - What is Spring Security Filter Chain ?

Servlet filters that allow us to manage Spring security works by intercepting the request before it reaches the actual resource.
For example; I only want those logged into the app to see the ordering page, when the request is sent to the ordering page.



The application will become quite complex when you want to add new controls like new authentication rules, validation or caching after a certain time has passed.

So we can use the Chain of responsibility pattern. It suggests that you link these handlers into a chain. In addition to processing a request, handlers pass the request further along the chain. The request travels along the chain until all handlers have had a chance to process it.



A handler can decide not to pass the request further down the chain and effectively stop any further processing.

9 – What are the differences between OAuth2 and JWT ?

- **JWT** is a JSON based security **token** for API Authentication.
- **OAauth** is an **open standard** for Authorization. Client software can be authorized to access the resources on-behalf of the end user using a token.
- Security protocols like OAuth2 use JWT tokens to secure API's.
- **OAuth2** is very **flexible**.
- **JWT** implementation is very easy and does **not take long to implement**.
- **JWT** is excellent for **server-to-server authorization**,
- **OAuth** 2.0 is excellent for **session management**.
- **JWT** tokens are **stateless**; hence the information is not stored on the server site

10 - What is method security and why do we need it ?

The method security level Spring security allows us to add security to individual methods within our service layer.

Using method security we can configure which roles are allowed to access which method within the same class of service.

We can define some annotations like ***@Secured, @RoleAllowed, @PreAuthorize,@PostAuthorize* .**

11 – What Proxy means and how and where can be used ?

A **proxy server** is a system or router that provides a **gateway** between **users** and the **internet**.On other words, A proxy server is essentially a computer on the internet that has an IP address of its own.

Proxies provide a valuable layer of security for our computer.We can use proxies for personal purposes or for a company. I mean, when I want to hide my location, I can use it. Or we can save bandwidth by caching files or compressing incoming traffic

12 – What is Wrapper Class and where can be used ?

We can wrap a **primitive value** like **int** into a wrapper class object like **Integer** using wrapper classes.

- The classes **in java.util package** handles only objects
- Data structures **in the Collection framework** store only objects An object is needed to support synchronization in multithreading.
- Objects are needed if we wish to modify the arguments passed into a method
- An object is needed to support synchronization **in multithreading**.

So, we can **use** the Wrapper class in **these states.**

Autoboxing : int => Integer convert

Unboxing: Integer => **int** convert

13 – What is SSL ? What is TLS ? What is the difference ? How can we use them ?

Both are the protocols used to **provide the security between web browser** and **web server.**

SSL(Secure Socket Layer) :

- **Message digest is used** to create master secret and It provides the basic security services which are **Authentication** and **confidentiality**
- Supports Fortezza algorithm.
- In SSL, **Message Authentication Code protocol** is used.
- complex than TLS, less secure

TSL(Transport Layer Security):

- **Pseudo-random function is used** to create a master secret.
- Not supports Fortezza.
- In TLS, **Hashed Message Authentication Code protocol** is used.
- simple, high secure

14 - Why do you need the intercept-url ?

The intercept-urls of the security namespace use to define what URL are to secure.

For example;

- /accounts/edit*
- /accounts/account*
- /accounts/**
- /customers/**

If we want to define the admin role and only admins can edit. We can do something like **access="ROLE_ADMIN".**

We need them because it allows permission access set for each role as you like.