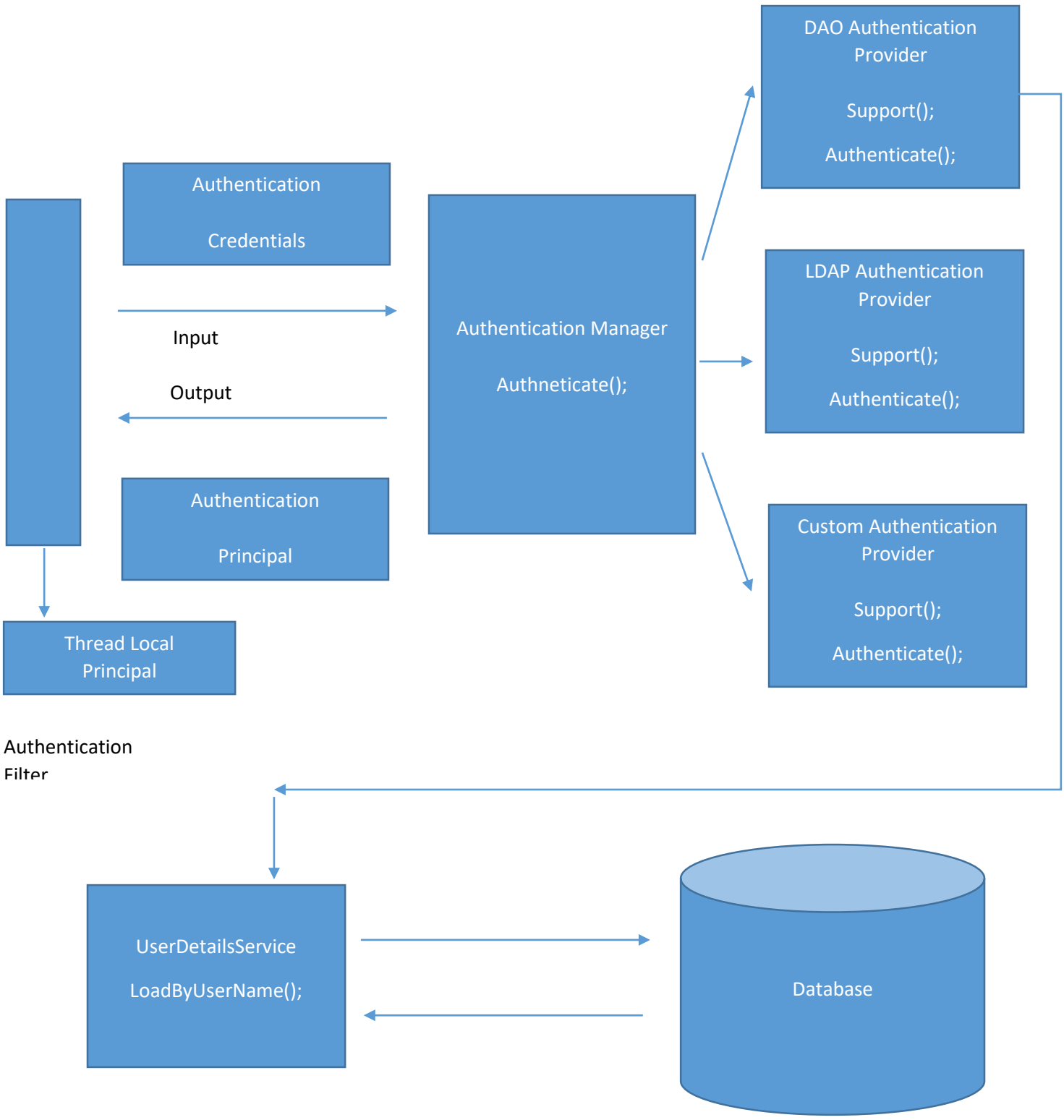
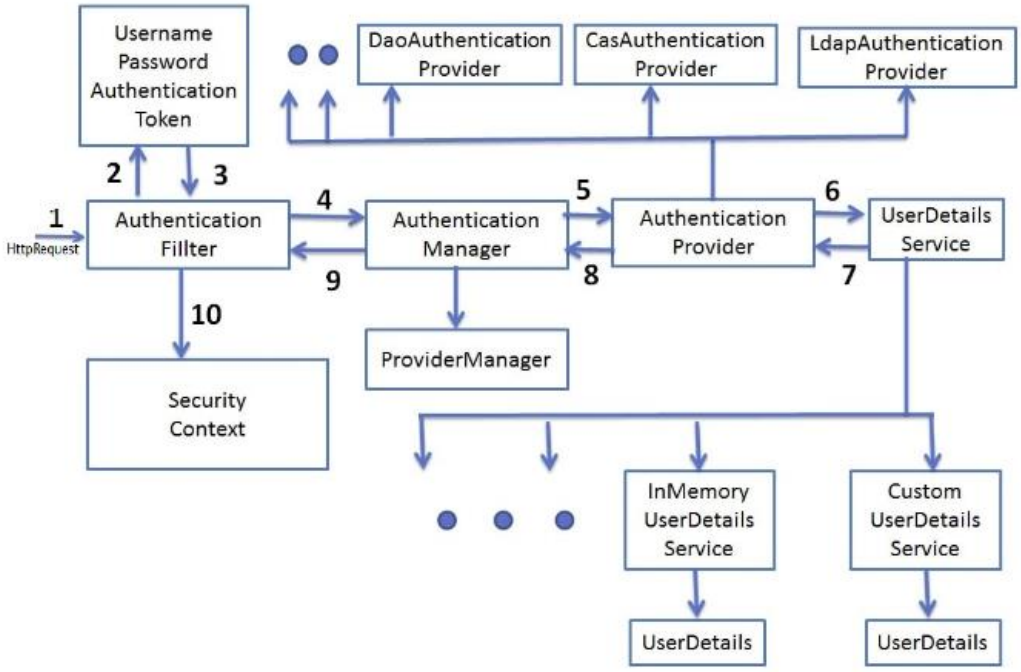
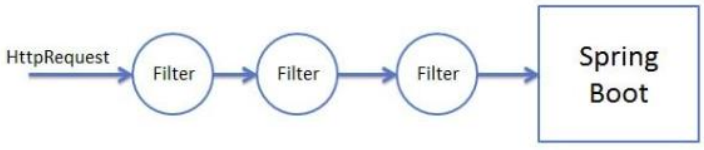

DAO Authentication Provider
Support();
Authenticate();
Authentication
Credentials
Authentication Manager
Authneticate();
LDAP Authentication Provider
Support();
Authenticate();
Input
Output
Authentication
Principal
Custom Authentication Provider
Support();
Authenticate();
Thread Local Principal
Authentication Filter
UserDetailsService
LoadByUserName();
Database

- **Spring Security Architecture**



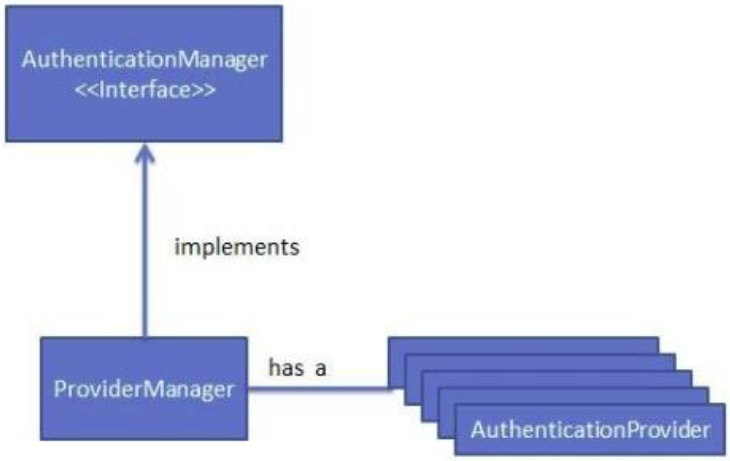
1. Request from client goes to Filter.
   Filters - : Filters Are Responsible for Authorization and Authentication and based on type of request there are different authentication filters.
   1) Basic Authentication Filter
   2) UserNamePassword Authentication Filter
   3) OncePerRequest Filter



2. Once request is intercepted by appropriate Authfilter it will retrieve username and password for request and create **UserNamePasswordAuthenticationToken** which is type of **Authentication**.

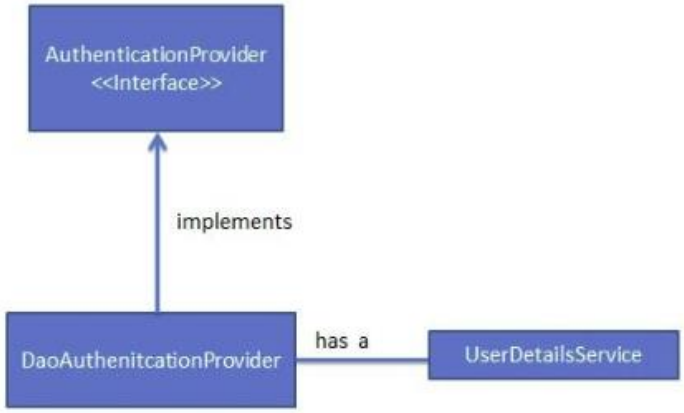
3. **Authentication Manager** – Using the Authentication type of object; Filter will call the **Authenticate** method of **Authentication Manager**.
   Authentication Manager is an interface whose implementation is provided by **Provider Manager**.
   Provider Manager has list of **Authentication Providers**.
   Provider Manager will call **Authentication Provider's** support method so on the basis of type of request appropriate **Authentication Provider** will be selected.
   From Authenticate method of **Authentication Manager** it will invoke Authenticate method of Appropriate selected **Authentication Provider**.
   Provider Manager implements **Authentication Manager**.

Field	Authentication(User Request before Authentication)	Authentication(After Authentication)
Principal	username	User Object
Granted Authorities	Not granted any authorities	ROLE_ADMIN
Authenticated	false	true



4. **AuthenticationProvider** – It has Authenticate and Support methods.
   As per Request and Support method appropriate Authentication Provider will be selected from various implementations.
   1) DAO Authentication Provider
   2) LDAP Authentication Provider
   3) Custom Authentication Provider

   And Authenticate method where all actual authentication takes place.
   Authentication Provider has **UserDetailService**.



5. **UserDetailService** – Using this **Authentication Provider** fetches user from either **Database** or **internal memory** .
   Spring Security uses **UserDetails** type object for user .
   Then **UserDetails** object credentials compared with incoming authentication object credentials.
   If successful Principal Authentication object is returned in response.
   Then Authentication object is then stored in SecurityContext using SecurityContextHolder.

6. Resouces –
   https://www.javainuse.com/webseries/spring-security-jwt/chap3