**Assignment 3: Secure Session Management**

**Objectives:**

This homework makes you familiar with design and implementations of architectural tactics. An important goal of this assignment is to emphasize that designing architecture is important but implementing the design, testing and maintaining the design decisions are critical.

**Objectives:**

You are supposed to implement “Secure Session Management” Tactic. The implementation is minimum prototyping of the tactic than full implementation of a system.

**Description:**

Web applications can create “sessions” to keep track of anonymous users after the very first user request. An example would be maintaining the user language preference. Additionally, web applications will make use of sessions once the user has authenticated. This ensures the ability to identify the user on any subsequent requests as well as being able to apply security access controls, authorized access to the user private data, and to increase the usability of the application.

HTTP is a stateless protocol, where each request and response pair is independent of other web interactions. Therefore, in order to introduce the concept of a session, it is required to implement session management capabilities that link both the authentication and access control (or authorization) modules commonly available in web applications.

Please consider the following items:

1. Create a light web application.
2. First give the client an identifier.
3. Have a mechanism to maintain identifier/client association. For example, a database table on the server side that associates identifier 3742346562000004320aabbccddeeff with client/user #22152.
4. Have the client resend the identifier sent to it in all subsequent requests, so the server can know who the request comes from.
5. The web application looks up the received identifier, and check if the user is already authenticated and is authorized to do requested operation, and then proceed with the operation on behalf on the user.
6. Have at least three users, and one task. One user is authorized to perform the task, one is not, and the last one is not authenticated.

**Note:** You can use the session management APIs of any existing library. This assignment requires you to implement authentication and authorization as well. You are allowed to use existing security frameworks.

**Constraints:** Languages limited to Java, C#. You can use the existing frameworks or implement from scratch.

**Deliverables:**

* Runnable Source Code
* Read me file including guidance on how to run the code, list of frameworks used.
* Documentations if necessary