Office Hours 12/12

```
var olddigest = convert.tobase64string(user.passwordhas);
var newdigest = convert.tobase64string(salteddigest.digest);

If (olddigest == newdigest)
{
        // create identity
        // create principal

        Thread.CurrentPrincipal =

        return true;
}



Public bool Logout()
{
        Thread.CurrentPrincipal = null();

}
```

Set up databases for unit/integration tests
- Anytime you need extra steps, provide instructions
    - If the steps are wrong then we get 0 points

Authentication and authorization implemented without IIdentity and IPrincipal
- Custom auth ticket used, similarly to IPrincipal, but stored in database
- Severely hampering architecture
    - Data store is session that is supposed to be universal, now all microservices need to point to the database
- Single source of failure as a result of the auth ticket
- Stateful implementation for a stateless architecture
    - Identity and Principal should be stateless, migrations won't affect how implementation works
    - Current implementation doesn't work for microservice architecture, handicapped scalability
- User Identity is stored on heap during runtime
    - Vong's example is all single-threaded now
    - Next semester will have multi-threaded web application will have the same concept scalable for the web
        - Stored as piece of data which will be passed around
        - Still stateless, passing data around
- Main issue with our implementation is the data store is being used
    - Scalability issue- every time something has to return back to a database it is stateful
    - Federated identity, has to be unique, cryptographically secure, identifiable
- Throughout course of entire request, will have the same principal
    - Next course has a completely different principal object
    - Only issue if a static variable is storing the principal to, then there is a conflict
- Horizontal Scaling
    - Horizontally scaling the web server will increase the thread pool
    - Horizontally scaling the data base only allows for more available slots
- Multiple threads
    - Async await jumps to original thread if you don't configure it correctly, if you do then it'll bring the ambient data/context
- Modifying the principal object

- - - Only way to modify the principal object is authentication where it's made, or user management when dealing with permissions
        - Force user to log in when changing permissions? Easiest way
        - Otherwise make the principal object expired using a data store (not scalable)
  - Identity Provider is system which deals with logging in and authorization
    - Facilitates expiration through this system
    - Microservices only sends principal to the identity provider to determine if it's accurate/expired
      - Typically passed through the header, generally secure enough
        - Only issue is size, role-based systems generally don't have a problem
    - Abstraction of authentication since it's such a major system

Four code review question
- Core components count for final grade
- Randomly pick one of the 6 that need to be finished by that date
  - Pool is chosen from project plans which are supposed to be completed

Scrum question
- 0 work due to being sick, denote 0
  - Note that you are sick in the burndown so it's easier to remember for sprint retrospective
- Project Backlog has 0 priorities
  - Hasn't gone through backlog grooming so it's valid to have no priority at this time
  - Parking lot

DAR Report Metrics

Each metric has a different weight
(x 0.5)

Calendar Integration
1. How many events can you have during a single day
   a. Example, minimum of 25 events per day
2. How many events can you create?
3. How many you can see or load up at a single time?
4. How many different types of views? List view? Calendar view?
5. What are the behaviors that we want? Add events directly? Can it swipe, can it add, can it pinch?

Chart and Graph
1. Remove export chart as PNG
2. Need trend chart
3. Need bar chart
4. Need time series
   a. How much data are you going to put into a time period

Email Provider
1. Decide a quota from BRD
2. Email reputation and verifier
   a. MX records on DNS
   b. Use a tool or service to perform it
3. Syngrid, mailchimp, can verify if the email is real before sending it
   a. Can also send emails
   Metrics
   Sending quota of email, text based content is for all emails, if you need pictures that's a metric to check, email validation

Front-end JS
1. Can you use front-end to build single-page application?
   a. Business use-case
2. Can you pass data across views?
3. How composable are components, can you combined them together?

- Some systems you need to download HTML templates files, some pass them as strings
- Single page application only allows strings
- Some frameworks you develop as separate pages html and then are combined together onto the same page