Question Sheet

1. What is a situation where you had to solve a tough engineering problem?

What was the problem?

Needed to provide an automated mechanism to poll a third-party website/service to retrieve patient lab report documents securely, store them on an internal network share, and send events indicating that new documents had been retrieved.

What did you do to solve it?

Turns out, the third-party website was using a non-standard SAML2 configuration for authentication and authorization and did not have actual API endpoints. I had to leverage a .Net Framework component library for SAML2 for authenticating (the out-of-the-box .Net Framework SAML implementation I first tried worked fine for our internal testbed, but was inconsistent against the production site) and extended the WebClient to make it ‘cookie-aware’ for web requests. These were wrapped up in a windows service to do scheduled web scraping sessions to pull down new documents (HTML, PDF, and HL7 files), including a lot of error correcting code to handle inconsistencies in the web site/service behaviors and availability.

What was the outcome?

This service has been running well for over a year and half now. I am considering revisiting it during the next IP sprint to determine if a newer, DotNet Core approach could be found that would allow for running in a Docker pod and deploy it to the Azure cloud and get it off the legacy on-premise Windows box.

1. An enterprise company is interested in building a web service complete with front-end website. You are hired to do the job. What are the considerations and tradeoffs that you consider to do the job?

Considerations:

1. Infrastructure:
   1. Existing server farm / cloud services in place for hosting, DNS, etc., or must it all be provisioned and configured from scratch?
   2. Existing authentication providers (AD, Okta, etc.) or need to setup?
   3. Existing security measures (domains, SSL certificates, etc.) or need to purchase and provision?
2. Build + Deployment Pipelines:
   1. Source Control (BitBucket, GitHub, AzDevOps , etc.)
   2. Build/Pipeline (Jenkins, TeamCity, AzDevOps, etc.)
   3. Deployment/Pipeline (Jenkins, Octopus Deploy, AzDevOps, etc.)
3. Development Stack, Components & Toolsets
   1. .NET Core? .NET Framework? Other?
   2. Angular/Vue/React? MVC? Other?
   3. Other Components/Packages Needed or Expected?
      1. Examples: Telerik, ComponentOne, Sharepoint, various CMS platforms…
4. In-House Company Support Personnel
   1. Is there a dedicated business owner / liaison?
   2. Does the company have UI/UX or marketing to provide branding specifics, or must develop from scratch?
   3. Are there any other support personnel: business analysts, quality assurance, etc., or are all functions to be filled/provided?
5. Timeframe / Milestones
   1. Is the company familiar with Agile methodologies, or used to waterfall?
   2. Is there already an established completion date?
   3. Is the MVP (minimum viable product) definition clear and detailed or require further development and clarification?

Tradeoffs:

1. Infrastructure:
   1. Existing infrastructure will need assessment of any necessary upgrades / patching / reconfiguration to support new sites and services.
   2. New infrastructure will require some time to purchase, configure, harden, before ready for deployments.
2. Pipelines:
   1. Existing source control and build/deploy pipelines will require assessment to see if they support the required development stack.
   2. New source control and pipelines will require some time to purchase and setup.
3. Development Stack:
   1. New technologies will have greater compatibility and better support when deployed into cloud services, and provide a richer user experience in the browser.
   2. Older technologies may be needed if integrating with existing legacy systems and services that are not being upgraded, or if on premise infrastructure requires it.
4. Support Personnel
   1. Company personnel can provide deeper insight into the nuances of the business requirements, branding considerations, etc.
   2. I am capable of fulfilling all the roles/needs of building an entire enterprise application on my own, but a small, lean, knowledgeable team is preferred.
5. Timeframe / Milestones
   1. If waterfall is expected, would recommend Agile approach instead. Depending on a number of factors, this may be a non-starter.
   2. If there is an established completion date without MVP, initial analysis needs to be done prior to determine if feasible given all constraints. Discussion on deliverables may need to be done.
   3. If there is a known MVP and completion date, some assessment may still be needed. Discussion on deliverables may need to be done.

3.

# Coding

The following 2 questions are expected to be answered without the help of the web.

**Please submit original code.**

Also, for each coding question, please submit test cases to ensure your code is working correctly.

Consideration for coding sections:

1. Functionality – does it work?
2. Algorithmic complexity – does it work efficiently?
3. Testability – how easy is it to test?
4. Test harness – how well do you think through corner cases?
5. Readability – how clean is your code?

## Coding part 1 - Min Stack

Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

Implement the MinStack class:

* MinStack() initializes the stack object.
* void push(val) pushes the element val onto the stack.
* void pop() removes the element on the top of the stack.
* int top() gets the top element of the stack.
* int getMin() retrieves the minimum element in the stack.

**Example 1:**

Input

["MinStack","push","push","push","getMin","pop","top","getMin"]

[[],[-2],[0],[-3],[],[],[],[]]

Output

[null,null,null,null,-3,null,0,-2]

Explanation

MinStack minStack = new MinStack();

minStack.push(-2);

minStack.push(0);

minStack.push(-3);

minStack.getMin(); // return -3

minStack.pop();

minStack.top(); // return 0

minStack.getMin(); // return -2

**Constraints:**

* -2^31 <= val <= 2^31 - 1
* Methods pop, top and getMin operations will always be called on **non-empty** stacks.
* At most 3 \* 10^4 calls will be made to push, pop, top, and getMin.

## Coding part 2 - Valid Parentheses

Given a string s containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

**Example 1:**

Input: s = "()"

Output: true

**Example 2:**

Input: s = "()[]{}"

Output: true

**Example 3:**

Input: s = "(]"

Output: false

**Example 4:**

Input: s = "([)]"

Output: false

**Example 5:**

Input: s = "{[]}"

Output: true

**Constraints:**

* 1 <= s.length <= 10^4
* s consists of parentheses only '()[]{}'.

**Notes On Example Source Code (Full Source in Zip File):**

I have provided three (3) different example implementations of the ‘MinStack’ class described above in coding part one.

I have provided an injectable ParentheseValidationService that validates an input string of only parentheses, curly braces, and brackets for consistency as described above in coding part two.

Based on the specification details, I have made a few assumptions:

1. The ‘pop’ function is defined as returning ‘void’ so I have made it internally remove the top element from off the stack but does not return it.
2. The ‘top’ function should ‘get’ the top element, so I have made it retrieve and provide the top element’s value but does not remove it from the stack.
3. I understood that the ‘getMin’ function should operate in ‘constant time’. This level of consistency is very difficult to achieve in a higher-level language such as C#, as the compiler tends to do various optimizations that do not allow for it, but I have made my best attempt at it anyways. It is possible to achieve via lower languages, particularly assembly language, but I have not used that since college. Having a function perform operations in ‘constant time’ is often important to use in cryptography to minimize side channels attacks, but I prefer to rely on third-party libraries (ex: Bouncy Castle) when the built-in cryptographic capabilities in dot net are insufficient.
4. The ParentheseValidationService leverages the generic Stack class in its implementation, since the requested MinStack implementation was for integers and the generic one supports chars.
5. I have included a number of XUnit tests for all three MinStack implementations and the ParenthesesValidationService. These are by no means all inclusive, but I trust are sufficient to demonstrate understanding.

Please find included an archived VS2019 solution for the above coding challenges.