**THE CE SHOP PRACTICAL INTERVIEW ASSIGNMENTS – Roman Jazmin**

**Practical Interview Project – Senior Full Stack Software Engineer**

**Project #1.**

The CE Shop offers online courses in Pre-Licensing, Post-Licensing, and Continuing Education (CE) for real estate agents. We currently offer CE courses in all 51 jurisdictions in the U.S and Pre-Licensing courses in 13 states.

We are currently adding a new feature in our Learning Management Software (LMS) to show students what their progress is during their course. For example, as a student I want to see that I have 40% left of my course I am taking. Essentially 4 out of 10 questions

Please describe how you would implement the new feature of showing a user their progress throughout the course from an end to end solution. This includes adding a quick design (notes, diagrams, or thoughts of how you would approach it), JavaScript, Java code, and SQL to get the data from the database. Include how each layer talks to each other via API’s, REST, but does not have to be detailed. Please provide high-level code as part of this exercise.

***Answer:***

Initially create a StudentProgress table. Let us say that each program a student is enrolled in has 5 chapters for them to complete.

For this example, the table will have the following columns:

* StudentID – Student’s unique id for the program.
* ChapterTitle(1-5) – There will be 5 columns that reflects all the chapters students must complete. These field will store the date that students completed a particular section.
* TotalNumChapters – This column indicate how many chapters the students have to finish for the program.

The next thing to do is run 1 SQL query to find the number of columns with null values per record. This means the student has (null)# of chapters to finish up. The query could be looking like this:

SELECT PercentLeftBeforeCompletion = (5 - (COUNT(Chapter1CompleteDate) + COUNT(Chapter2CompleteDate) + COUNT(Chapter3CompleteDate) + COUNT(Chapter4CompleteDate) + COUNT(Chapter5CompleteDate)))\*(.1)

FROM [dbo].[StudentProgressTable]

where StudentID = 1

In order to receive a return percentage from the query, it needs to return a value where:

(5 - (COUNT(Chapter1CompleteDate) + COUNT(Chapter2CompleteDate) + COUNT(Chapter3CompleteDate) + COUNT(Chapter4CompleteDate) + COUNT(Chapter5CompleteDate)))\*(.1)

Example: 2 chapters a student needs to still complete. We have 5 chapters per program. So, the percentage still left to complete is value = (2/5)(0.1). There you have it.

**Project #2** - Write a Java Program that determines if a string is a palindrome.

**Project #3** - Write a function that takes in an array of numbers and prints the combinations that add up to 10.

**Project #4** – You are given a configuration of a chessboard with a rook and a king. Write a function that returns true if a king and rook are in the correct positions for castling.

**Additional Practical Interview Questions:**

*Please provide concise, yet specific and detailed answers to the following:*

1. How would you make an application testable, what do you do to ensure code quality?

***Answer:***

When writing test automation, one of the most important factors for determining the amount of time and resources you will consume (and ultimately the success or failure of the endeavor) is the testability of your application. By testability, I'm referring to how the app interacts with UI (and other) automation frameworks, the ease by which a test script can setup the scenarios you wish to test, and how you make your tests safe for concurrency.

Making elements accessible. Let's address the matter of controlling your application with automation frameworks first. Given that you are reading this blog, it is likely you are using either Selenium or Appium as your automation framework, so this blog post will only address these frameworks.

For web contents, let's start with Selenium. In order for your web app to be easily controlled by Selenium, you need to think ahead as to how you will identify important pieces of the DOM when constructing tests. Selenium provides many means by which you can do this, called "Locator Strategies". Some are better than others. You should consider which of these you will be using in your tests as you develop the user interface. Ideally, each element would have an ID attribute applied directly to the tag for any element that the test will exercise.

Sometimes, for one reason or another, you may not be able to use an ID. If this is the case, the next recommended technique is to use a CSS selector. If your web app is developed with good principles, such as BEM (Block Element Modifier), it is likely easier to automate as it should have a relatively short globally unique CSS selector. If it does not, I would not recommend adding a CSS class just for automation, as it isn't the purpose of the styling language to do automation. Rather, I would suggest that you use an HTML  data-\* attribute. You can use the CSS selector locator to grab an element by one of these attributes, but the good news is you aren’t adding unnecessary classes to your CSS, and the purpose of the HTML attribute will be much clearer.

As for some of the other locator strategies out there, it is my opinion they should be used sparingly. If you have an Angular app, where tag names are often unique, perhaps you would use the tag name locator strategy. I would strongly discourage using XPath, as its implementation across different browsers is inconsistent, which can result in flaky tests. In particular anything using indexes, or that is heavily dependent on the organization of the DOM, is likely to be brittle, as you can expect the DOM to change as you develop your application. I would also strongly discourage the use of locating by text as this will inhibit the ability of your tests to work in multiple languages. (e.g. "Hello" on American content, would be "Bienvenidos" on Spanish content)

Making mobile apps easier to automate follows similar principles to the web. In general, if you follow the accessibility guidelines set forth by Apple and Google, especially when constructing custom UI elements, you should be in pretty good shape. However, it is my experience that most apps do not adhere closely to these guidelines. No worries, if this is the case, you should still be able to make it work.

For iOS, the best technique for identifying elements in the app is the accessibilityIdentifier property on UI elements. There are 3 ways of setting this. You can add them in your XIB or storyboard using Interface Builder. They will appear on the "Identity Inspector" tab in the accessibility section. However, some UI does not use XIBs or storyboards and is generated programmatically. This is not a problem as you can call the [UIElement setAccessibilityIdentifier] on most UI elements. You can also add a user defined runtime attribute for the key “accessibilityIdentifier” in either Interface Builder or via code. This is the preferred method, however using XPath for more complicated pieces of UI is OK, especially dynamic content such as tables.

Note that you may have heard of the accessibilityLabel and accessibilityHint fields in the past. The problem with using these fields is that users who are using VoiceOver or other assistive devices will hear the values you place here spoken to them, so it is strongly encouraged that you use accessibilityIdentifier instead. You may also need to set the "isAccessibilityElement" property to 'true' as some UIElements are excluded from the accessibility layer by default. Lastly, you may want to explore the "groupAccessibilityChildren" and "accessibilityElementsHidden" attributes for parent elements if you notice your element is not accessible. The easiest way to verify this is to use the Appium inspector, or the built in accessibility inspector that can be enabled in your phone’s settings.

For Android, the story is a bit simpler. The resource-id is the most reliable way to identify an element. You can find information on how to set the resource id [here](https://developer.android.com/guide/topics/resources/accessing-resources.html). If a resource-id will not do it, then I’d recommend building an XPath expression that heavily leverages resource-ids. You may have seen previous guidance to use the content-desc attribute for Android, but the problem is the same as with accessibilyLabel and accessibilityHint in iOS in that users with assistive devices will hear/see the values you use here.

As part of your tests you will inevitably need to get the app into certain states to test certain things. There are many techniques to go about this, and you should consider how you will set up UI tests for different features in your application as they are created.

Technique #1, UI automate your way to the scenario. Imagine a scenario where you need to create two user accounts and then perform some action in your application involving those two accounts (e.g., send a message from one to the other) Without any tools, your test will have to complete your product's registration flow twice before attempting to send the message between the two accounts. This is lengthy and slow.

When writing UI Automation such as with Selenium or Appium tests, a good rule of thumb is not to use the UI unless you are verifying it. The test for the registration flow ideally should be the one test that automates the UI of the registration flow rigorously. Automating this flow in other tests will make your tests take longer and will inject more brittleness into your test suite. However, I understand, that sometimes this is the only way, especially when using 3rd party components.

Technique #2, build a test API. Good applications are well factored into many layers. A common layer is an API. A good API will handle all transactions that modify the business data, i.e. the front-end is responsible for display and the back-end makes changes to the data on your servers and performs business logic and calculations.

Having an API just for testing can greatly simplify your scripts and will increase the reliability with which they run. Imagine the previously mentioned scenario. Instead of running the lengthy registration UI automation two times, you could simply create an API endpoint that will generate the two accounts in the required states. Perhaps your API might even already have endpoints that your web front-end already calls to do the same thing. Be sure to test the UI flow once in a specific test, but for follow on tests, such as the example of sending a message, you can just call the API to set up state and only test the piece you care about, in this case, sending messages.

One of the biggest ways I see developers shoot themselves in the foot is by not considering that their tests will run in parallel. You should, whenever possible, avoid writing any tests that depend on shared state in your application, and should always take appropriate measures to isolate your tests from one another. It can be extremely time-consuming to unravel this mess later on.

Consider the above example concerning sending a message between two accounts. If we share the same accounts with other tests we may have a message show up that is detected by one test, but was actually sent as part of another test, leading to a false result. This is why testing in parallel requires advance planning, such as creating separate accounts for different tests.

Take another example wherein we need to test blocking a user from our service. If we create an account for the test and then block the account, we may think we have isolated the side effects of our test. However, imagine that if -- at some point -- we block enough accounts from the same IP address, the server will eventually blacklist the IP address for the account. Now we have introduced consequences that will cause errors in other tests that will be difficult to track down.

Bottlenecks in automation amount to resources that tests are required to share. They can be anything from environments (e.g., servers), accounts, and IP addresses (as mentioned in the example above), or simply devices on which to execute the tests. It can be quite time consuming to completely isolate every aspect for all of your tests. To do so may require a lot of time or resources. Imagine having to reformat a device and spin up an entire environment and private network for each test; that is quite time consuming.

You should try to intelligently choose where your bottlenecks will be and how to manage them. For example, if you have a limit on the number of accounts for your tests, you may have to write code to lease a test account to a test and then, when it is returned, clean and reset it for the next test. You may have to run certain tests in series rather than in parallel to avoid them stepping on each other's toes.

In a perfect world, a well-written application can be spun up instantly in the cloud. Software from cloud providers, such as Cloud Formations from Amazon, make this possible. For devices and browsers, services such as Sauce Labs can dispense any configuration you desire nearly instantly. It's noble to strive for complete isolation and independence, but be prepared to make calculated compromises around this as you develop your application.

Having a highly testable application will make writing automated tests for that application a much less time-consuming and a much more worthwhile experience. Creating a user interface that is automatable and conforms to vendor supplied accessibility standards, providing convenient and reasonable ways to access application states, and preparing your test suite to run concurrently will put you off on the right foot towards achieving sustainable application stability via test automation.

1. From a software development perspective, what steps might you take, on an ongoing basis, to continually improve your skillset?

***Answer***

The question is what I would do continually to improve my skillset. Here are five ways I keep my job skills and knowledge up-to-date:

## **1. Take Professional Development Courses**

Professional development courses can help you expand your professional skill set, learn something new, or even earn academic credit to put towards a degree. [Online training courses](https://www.mbopartners.com/blog/professional-development-how-to-pick-the-right-online-training-course) are particularly convenient because they are affordable and flexible. Just be careful to do your homework—evaluate instructor bios, read reviews, and check the syllabus carefully before putting down your credit card. You can also find professional development courses through vendor-taught classes, traditional universities, and training institutions.

## **2. Utilize Online Resources**

The Internet is a limitless source of free information and educational resources. Attend educational [webinars](https://www.mbopartners.com/webinars), follow the blogs or social media accounts of industry experts, or bookmark and regularly check industry news sites and online forums to stay current on the latest trends. If you haven’t already, sign up for news alerts for your inbox (Google Alerts works well) or set up an RSS feed like Feedly.com to easily put all of your industry news in one place.

## **3. Attend Professional Events**

Professional events are valuable ways to learn about growth and development in your industry. Local companies, business associations, and professional groups often host seminars, forums, or workshops that can give you direct access and insight to experts in your profession. Treat these events as constructive [networking opportunities](https://www.mbopartners.com/blog/category/networking) to brainstorm and share ideas with colleagues who can provide fresh insight and perspective.

## **4. Network Online**

As an independent consultant, you know the importance of building and maintaining a[list of contacts](https://www.mbopartners.com/blog/networking-tips-for-small-business-owners) to ensure a steady flow of work. Use LinkedIn to connect with high-ranking people at companies you’re interested in working with. Employ social media platforms to promote your own service or brand, network with industry experts, and keep in touch with former and current clients.

Start by finding which social site [works best for you](https://www.mbopartners.com/blog/dos-and-donts-networking-and-marketing-your-consultancy-on-social-media)—connect with fans and followers on Facebook, creatively network and share news on Twitter, or utilize blogging to boost your online credibility, and connect with potential clients.

## **5. Invest in Continuing Education and Certification**

While [continuing education and certification programs](https://www.mbopartners.com/blog/professional-certifications-for-independent-contractors) typically require a more intensive time and financial obligation, they can help boost reliability, and demonstrate a commitment to your profession. Becoming proficient in a new software platform before it becomes mainstream, committing to upholding industry standards through a certification program, or staying on top of market trends by taking a class can increase your income and position you competitively within your line of business.

No matter which tactic you choose, maintaining enhanced knowledge and skills in your field shows clients you are well informed and dedicated. Set yourself up for success by investing in your job skills and knowledge today.

1. Provide an example of a time when you just could not complete all of your work on time to meet a deadline. What did you do? What was the result?

***Answer:***

The possibility of not meeting your deadline is something all programmers face and must resolve quickly if we are to be successful. This situation happened to me when I was hires to revamp Nailor Industries Order Entry Program, OEP. The whole project took me a year and a half to finish with many department deadlines.

How did I resolve my deadline problems? I focused on being proactive in what I do, as best as I can. My solution is to implement a Scrum/Agile approach to software development.

From years of experience developing in this framework environment, I learn that we as a team are able to effectively communicate feedback or roadblocks to teammates and users. We can relay specifications that might have changed over time, and describe better methods that would improve on the performance of the applications we are working on. We can prioritize ever task or activities. Again, this will allow our stake holders and users to be update on the progress of the applications we are making for them.

Along with our Daily Stand Up meetings, implementing a bi-weekly Sprite meeting will allow everybody to see our application in action. We can inform our team and users how each part of the code was coded what it can do. People can also be told what restrictions or problems that each component might have and what still needs to be worked on or improved.

If we continue to approach our development in this way, no one will have any excuse of not knowing where we are on this project. Users, who are required to accept and sign off on a basic set of specifications needed to complete the given applications, no longer add or modify anything unless the team and everyone else that is involved, have the opportunity to investigate how complex the modifications might be and how it will affect our deadlines.

If everyone agrees on the added modifications, then users will be aware that timetable and deadline will change even pushed back. Another thing that we will experience is that it will be impossible to discover major issues late in our development. We would find and fix every conceivable, major roadblocks and problems early on.

In the end, we are left with minor problems or issues that we can eventually resolve or reprioritize with users. The only thing left to do to run a series of activities dealing with QA testing, documentation, and Users final signoffs, and then release the application to production.