**Question 1:** *Identify and list the summary of the test cases you would run to test this new (date*

*of birth) functionality?*

Please refer to the springTC.xlsx document for the test cases

**Question 2:** *List anything else you think that needs to be tested before this change could be*

*released to live?*

**Essential Tests Before Live Deployment**

Before the new Date of Birth field goes live, it's crucial to conduct comprehensive testing to ensure everything works smoothly:

**Integration Testing:**

Ensure the new Date of Birth field integrates seamlessly with existing form elements and backend systems. This includes testing data flow between the Date of Birth and other mandatory fields (First Name, Last Name, and Email).

**End-to-End Testing:**

Conduct a thorough end-to-end test covering a complete user journey from form initiation to submission of the form. This makes sure that adding the Date of Birth field hasn’t caused any unexpected problems.

**Regression Testing:**

Perform regression tests to ensure that existing functionalities, such as form reset and submission notifications, remain unaffected by the new changes. Re-test the whole form to confirm that everything still works as it did previously.

**Validation Messages:**

Test the clarity and correctness of validation messages for the Date of Birth field. Ensure the error messages are clear, user-friendly and match the style of other messages on the form. They should correctly point out if the field is empty or if the entered age is below 19.

**Cross-Browser and Device Testing:**

Test the form on different browsers(e.g., Chrome, Firefox, Safari, Internet Explorer) and devices. It’s important that the form looks good and works well whether someone is using a computer, tablet, or mobile phone.

**Usability Testing:**

Have real users try out the updated form. This helps find out if the Date of Birth field is easy to use and understand.

**Performance Testing:**

Test the impact of the new field on the form's loading time and overall responsiveness. Make sure the form still loads quickly and responds well even after adding the new field.

**Security Testing:**

Check that adding the Date of Birth field doesn’t create any new security risks, especially since it’s dealing with personal information.

**Conclusion:**

Such thorough testing will facilitate a smooth transition to the live environment without any issues, keeping the form easy to use and secure.

**Question 3:** *Which of the tests that you have provided above would you automate?*

**My Test Automation and Manual Testing Strategy for the Form**

In updating our form with a new Date of Birth field, I’ve selected which tests to automate and which to perform manually by focusing on efficiency, accuracy, and the nature of the tests. Here's how I would approach it:

**Tests I Would Automate:**

1. **Regression Tests (TC001-TC013, TC017-TC021):**
   1. **Category:** Data Validation and Integration Testing.
   2. **Reason:** These tests are essential to ensure that new updates, like the Date of Birth field, integrate seamlessly with existing form functionalities without causing any disruptions.
2. **Cross-Browser Testing (TC013):**
   1. **Category:** Navigation Testing and Integration Testing.
   2. **Reason:** This is important to confirm that the form, including the Date of Birth field, works correctly across various web browsers, ensuring a consistent user experience.
3. **Validation Message Testing (TC005, TC006, TC007, TC017-TC021):**
   1. **Category:** Error Handling and Data Validation.
   2. **Reason:** Automating these tests is key for checking the form's response to various inputs, particularly in verifying that error and validation messages are correctly triggered, providing the right feedback to the users.

**Tests I Would Prefer to Do Manually:**

1. **Usability Testing:** This is key to understanding how users interact with the form, especially with the addition of the Date of Birth field. It's best done manually to get genuine user feedback and falls under Exploratory Testing, where we learn how user-friendly our form is.
2. **Security Testing:** It is essential to ensure that the Date of Birth field, which contains sensitive information, is secure. This requires a detailed manual review.
3. **Certain End-to-End Tests:** These tests help us see how the entire process, from entering information to submission, works from a user’s perspective. They are crucial for **Exploratory Testing**.
4. **Cross-Device Testing (TC014):** Checking how the form functions on different devices ensures a good user experience regardless of the platform. Some aspects of this are best evaluated manually.
5. **Backend Data Validation (TC016):** This involves checking data storage and retrieval in the backend, crucial for maintaining data integrity and privacy. This requires manual checks for complete assurance.

By automating tests that are repetitive and data-intensive, we save time and increase accuracy. Manual testing, especially in areas that require human insight, helps us understand the user experience better and ensures that our form is not only functional but also user-friendly and secure.

**Question 4:** *Write an automation for the contact us form for the following site*

http://webdriveruniversity.com/Contact-Us/contactus.html *(preference in JAVA but you can use any language). Create some valid and invalid test cases for email using automation code.*

**Git:** <https://github.com/ShivaRap/SpringFinancial.git>