Question No 1:

You are testing a form that allows users to schedule appointments with a doctor. The form has the following fields: first name, last name, email, phone number, and appointment date/time. Some more inputs:

If any field is blank, we should display "All fields are required".

If email is invalid, we should display “Please enter valid email”

If phone number is invalid , we should display “Please enter valid phone number”

If appointment date/time is not available, it should display “Please choose another date/time”

If all fields are correct and the appointment is available, we should schedule the appointment successfully.

For these requirements write down the correct list of conditions in form of decision table.

Answer:

Here's a decision table that captures the conditions and corresponding actions based on the given requirements:

| Condition | Action |

|-----------------------------------------------|---------------------------------------------------|

| First Name is blank | Display "All fields are required" |

| Last Name is blank | Display "All fields are required" |

| Email is blank | Display "All fields are required" |

| Phone Number is blank | Display "All fields are required" |

| Appointment Date/Time is blank | Display "All fields are required" |

| Email is invalid | Display "Please enter valid email" |

| Phone Number is invalid | Display "Please enter valid phone number" |

| Appointment Date/Time is not available | Display "Please choose another date/time" |

| All fields are correct and appointment is available | Schedule the appointment successfully |

| All other combinations | No specific action (default or validation message) |

In the decision table, each condition represents a possible state or input value, and the corresponding action describes what should happen based on that condition. The last row represents the successful scenario where all fields are correct and the appointment is available. The last row's action indicates that the appointment should be scheduled successfully.

Question No 2:

Suppose you are a software tester and you are assigned to test a new mobile app that has just been developed. The app has three different user roles: basic user, premium user and admin user. You have been given the following three test scenarios to execute:

Basic users can only access limited features of the app, while premium users can access all features.

Admin users have access to advanced settings and can perform all actions.

Users should be able to create and delete their accounts successfully. Passwords must meet the required complexity standards and users should receive a confirmation email upon successful account creation.

Users should be able to navigate through the app seamlessly without any crashes or performance issues. The app should also display appropriate error messages when user attempts to perform an action they are not authorized to perform.

For each of the test scenarios, describe the steps you would take to test them and the expected results.

Answer:

**Test Scenario 1: User Roles and Access Permissions**

Steps to Test:

1. Log in as a basic user and verify that only limited features are accessible. Attempt to access premium features and advanced settings to ensure they are restricted.

2. Log in as a premium user and verify that all features are accessible. Confirm that advanced settings and admin actions are restricted.

3. Log in as an admin user and verify that advanced settings and admin actions are accessible. Ensure basic and premium features are also accessible.

Expected Results:

- For the basic user, the limited features should be accessible, while premium features and admin actions should be restricted.

- For the premium user, all features except advanced settings and admin actions should be accessible.

- For the admin user, all features, including advanced settings and admin actions, should be accessible.

**Test Scenario 2: Account Creation and Deletion**

Steps to Test:

1. Fill in the account creation form with valid data, ensuring the password meets the complexity standards.

2. Submit the form and check if the account is successfully created.

3. Verify that a confirmation email is received upon successful account creation.

4. Attempt to delete the account and confirm if the account is deleted successfully.

Expected Results:

- The account creation form should accept valid data and create the account successfully.

- The password complexity standards should be enforced, rejecting weak passwords.

- Upon successful account creation, a confirmation email should be received by the user.

- The account should be deletable, and after deletion, the account should no longer exist in the system.

**Test Scenario 3: Seamless Navigation and Error Handling**

Steps to Test:

1. Navigate through different sections of the app, accessing various features and actions.

2. Check for any crashes or performance issues during navigation and usage.

3. Attempt to perform actions that the user is not authorized to perform, such as accessing premium features as a basic user or performing admin actions as a basic or premium user.

4. Verify if appropriate error messages are displayed when unauthorized actions are attempted.

Expected Results:

- The app should allow seamless navigation without any crashes or performance issues.

- Users should receive clear error messages when attempting unauthorized actions, indicating that they do not have the necessary permissions to perform those actions.

By following these steps and validating the expected results, the test scenarios can be effectively executed, helping ensure that the mobile app meets the specified requirements and functions as intended for different user roles.