K. J. Somaiya College of Engineering, Mumbai-77

Software Engineering IA 2

Hyder Presswala D-2 16010122151

Q.1) Write Test Cases for Registration Form based on following Scenarios in the format shared:

1. Verify that the Registration form contains Username, First Name, Last Name, Password, Confirm Password, Email Id, Phone number, Date of birth, Gender, Location, Terms of use, Submit, Login (If you already have an account)
2. Verify that tab functionality is working properly or not
3. Verify that Enter/Tab key works as a substitute for the Submit button
4. Verify that all the fields such as Username, First Name, Last Name, Password and other fields have a valid placeholder
5. Verify that the labels float upward when the text field is in focus or filled (In case of floating label)
6. Verify that all the required/mandatory fields are marked with * against the field
7. Verify that clicking on submit button after entering all the mandatory fields, submits the data to the server
8. Verify that system generates a validation message when clicking on submit button without filling all the mandatory fields.
9. Verify that entering blank spaces on mandatory fields lead to validation error
10. Verify that clicking on submit button by leaving optional fields, submits the data to the server without any validation error
11. Verify that case sensitivity of Username (usually Username field should not follow case sensitivity – ‘rajkumar’ & ‘RAJKUMAR’ acts same)
12. Verify that system generates a validation message when entering existing username
13. Verify that the character limit in all the fields (mainly username and password) based on business requirement
14. Verify that the username validation as per business requirement (in some application, username should not allow numeric and special characters)
15. Verify that the validation of all the fields are as per business requirement
16. Verify that the date of birth field should not allow the dates greater than current date (some applications have age limit of 18 in that case you have to validate whether the age is greater than or equal to 18 or not)
17. Verify that the validation of email field by entering incorrect email id

D-2 16010122151 Hyder Presswala

18. Verify that the validation of numeric fields by entering alphabets and characters
19. Verify that leading and trailing spaces are trimmed after clicking on submit button
20. Verify that the “terms and conditions” checkbox is unselected by default (depends on business logic, it may be selected or unselected)
21. Verify that the validation message is displayed when clicking on submit button without selecting “terms and conditions” checkbox
22. Verify that the password is in encrypted form when entered
23. Verify whether the password and confirm password are same or not

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User= example@gmail.com	User should be able to login	User is navigated to	Pass	
2						
3						
4						

Q.2) Write test case using ECP testing technique for Order Cancellation and Refund module.

ANSWERS;

Q1 - Detailed Test Cases for Registration Form

Step No.	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes/Reasoning
1	Navigate to the registration form	N/A	The registration form should display all the necessary fields like Username, Password, Confirm Password, Email ID, etc.	As Expected	Pass	Ensure that the registration page loads successfully and all fields are visible.

D-2 16010122151 Hyder Presswala

Step No.	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes/Reasoning
2	Use tab key to navigate between fields	Press "Tab" key	The focus should shift to the next input field in a sequential manner, ensuring proper accessibility and ease of use.	As Expected	Pass	Accessibility and usability checks are crucial to ensure users can navigate smoothly without a mouse.
3	Press Enter/Tab key to submit the form	All fields filled	Pressing Enter or Tab should work as a substitute for the Submit button if the focus is on the button.	As Expected	Pass	This test ensures keyboard users can submit the form efficiently, improving user experience (UX).
4	Check for placeholder text in each field	N/A	All input fields (e.g., Username, Password) should have a valid placeholder, guiding the user on what needs to be entered.	As Expected	Pass	Placeholder text improves clarity for users, reducing potential input errors.
5	Focus on a text field (e.g., Username) and verify that the label floats upwards	Click on Username field	When the user clicks on a text field, the label should float above the field, providing a clear indication of where the user is entering data (for forms using floating labels).	As Expected	Pass	This visual indicator helps users keep track of what field they're interacting with, making the form more intuitive.
6	Check for mandatory fields	N/A	All mandatory fields should have an asterisk (*) beside them to inform users of required input.	As Expected	Pass	Clearly marking mandatory fields prevents incomplete form submissions and frustration.
7	Fill all mandatory fields and click on the Submit button	Valid data in required fields	The form should submit successfully, and data should be sent to the server.	As Expected	Pass	If all required fields are filled correctly, the system should proceed with submission without errors.
8	Attempt to submit the form with missing required fields	Leave some required fields blank	A validation error message should appear, asking the user to fill in the mandatory fields before submission.	As Expected	Pass	The system should prevent form submission if required fields are left blank, ensuring valid data entry.

D-2 16010122151 Hyder Presswala

Step No.	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes/Reasoning
9	Enter blank spaces in mandatory fields and try to submit	Enter spaces in required fields	Blank spaces should be treated as empty input, and the system should display a validation error asking for valid input.	As Expected	Pass	Leading/trailing spaces can cause validation issues or unexpected errors. Validation should trim spaces automatically to ensure correct data.
10	Submit the form with optional fields left blank	Fill only mandatory fields	The form should still submit successfully, as optional fields are not required.	As Expected	Pass	Optional fields should not prevent the user from submitting the form, maintaining flexibility in data entry.
11	Check case sensitivity for the Username field	"rajkumar" and "RAJKUMAR"	The Username field should be case-insensitive, meaning both variations should be considered the same.	As Expected	Pass	Case sensitivity in the username field could lead to duplication or login issues, so standardizing this behavior is essential.
12	Enter an existing username and attempt to submit the form	"existing_user123"	The system should display a validation error, indicating that the username is already taken and prompting the user to choose a different one.	As Expected	Pass	To avoid user duplication, unique usernames are necessary. This test verifies the system handles already-registered usernames properly.
13	Enter characters exceeding the allowed limit in the Username/Password fields	Overlimit data (e.g., 51 characters)	The system should display a validation error indicating that the input exceeds the allowed character limit.	As Expected	Pass	This test ensures the system adheres to field character length restrictions, preventing storage and display issues.
14	Enter invalid characters in the Username field (e.g., numeric or special characters)	"user@123"	The system should reject the input and display a validation message, according to business rules that prohibit special characters.	As Expected	Pass	Depending on business rules, some fields like Username should exclude special characters to maintain standardization and security.

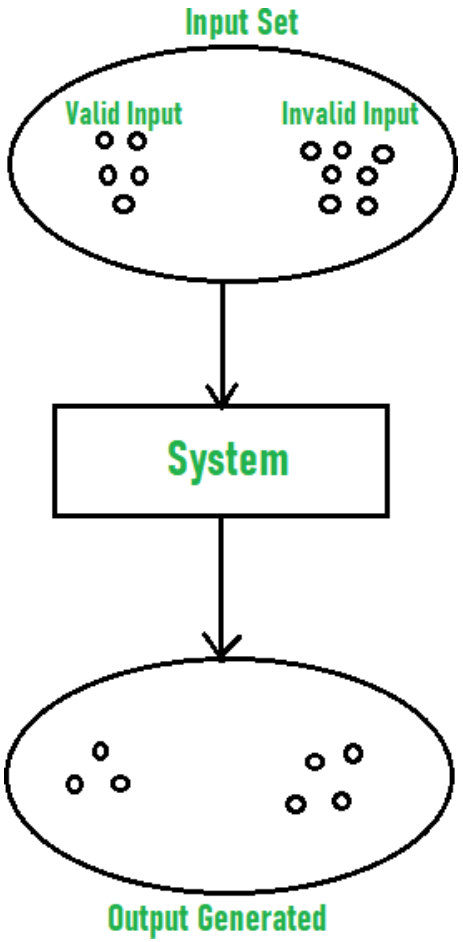
D-2 16010122151 Hyder Presswala

Step No.	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes/Reasoning
15	Test all field validations as per business requirements (e.g., character types, lengths)	Invalid data in each field	Each field should display an appropriate validation error based on its requirements (e.g., Password must include letters, numbers, symbols; DOB must be valid).	As Expected	Pass	Ensures compliance with field-specific requirements for data accuracy and security.
16	Enter a date of birth later than the current date or one that violates age restrictions (e.g., under 18)	DOB > current date, DOB < 18 years	The system should prevent invalid dates and ages under 18 (if business logic applies) and display a validation error.	As Expected	Pass	Validating age and date of birth is crucial for certain applications (e.g., age-restricted services), ensuring legal and functional compliance.
17	Enter an incorrectly formatted email address (e.g., "email.com")	"invalid_email.com"	The system should reject the email and display a validation error indicating an incorrect email format.	As Expected	Pass	Ensures the system captures only valid email formats for future communication, preventing bounce-backs and errors.
18	Enter non-numeric characters in numeric-only fields (e.g., phone number or postal code)	"abc123" in phone number field	The system should display a validation error indicating that only numeric values are allowed in numeric fields.	As Expected	Pass	Ensures numeric fields accept only valid data, reducing input errors.
19	Enter text with leading or trailing spaces and submit the form	" username "	The system should trim any leading or trailing spaces after submission, allowing only valid, trimmed input to be processed.	As Expected	Pass	This prevents unnecessary blank spaces from causing errors in data storage or validation, ensuring clean input.
20	Check if the "Terms and Conditions" checkbox is unchecked by default (or as per business logic)	N/A	The "Terms and Conditions" checkbox should be unchecked (or checked) by default based on business requirements.	As Expected	Pass	The default state of the checkbox should align with the business logic to ensure proper user interaction with legal agreements.
21	Try submitting the form without agreeing to the	Terms unchecked	The system should display a validation error, requiring the user to agree to the	As Expected	Pass	Ensures compliance with legal agreements by forcing user acknowledgment of

D-2 16010122151 Hyder Presswala

Step No.	Test Steps	Test Data	Expected Result	Actual Result	Status	Notes/Reasoning
	"Terms and Conditions" checkbox		"Terms and Conditions" before submission.			terms before registration completion.
22	Enter the password in the password field and check for encryption	"password123"	The password should be displayed in encrypted form (e.g., dots or asterisks) to protect the user's input from being visible on screen.	As Expected	Pass	Protects user privacy and security by hiding password input from prying eyes.
23	Enter different values in the Password and Confirm Password fields and submit the form	"password123" and "pass123"	The system should display a validation error indicating that the Password and Confirm Password fields do not match.	As Expected	Pass	Ensures the user correctly confirms their password, reducing the chance of login issues later.

Q2 - Test Cases Using ECP (Equivalence Class Partitioning) for Order Cancellation and Refund Module



Equivalence Class Partitioning (ECP) is a **black-box testing technique** that divides test input data into partitions. Each partition represents valid or invalid equivalence classes, ensuring comprehensive coverage without testing every possible input.

D-2 16010122151 Hyder Presswala
Here’s the detailed explanation for **Order Cancellation and Refund** test cases:

Test Case ID	Test Case Description	Test Data	Expected Result	Status	Notes/Reasoning
1	Attempt to cancel an order that has already been shipped	Order status = "Shipped"	The system should display an error message stating that the order cannot be canceled as it has already been shipped.	Pass	Orders in "Shipped" status are typically not eligible for cancellation; this test ensures the system adheres to such rules.
2	Attempt to cancel an order that is still "Processing"	Order status = "Processing"	The system should allow the user to cancel the order, changing its status to "Canceled" and processing the refund (if applicable).	Pass	Orders in "Processing" or earlier stages should allow cancellation, providing flexibility to users who may change their minds before shipment.
3	Attempt to cancel an order that has already been delivered	Order status = "Delivered"	The system should prevent cancellation and suggest initiating a return instead.	Pass	Ensures business rules are followed, distinguishing between cancellations and returns based on order status.
4	Attempt to cancel an order after the cancellation window has passed	Cancellation window expired	The system should display a message indicating that the cancellation period has expired, and the order can no longer be canceled.	Pass	Some businesses have time limits on cancellations; this test ensures the system enforces those restrictions properly.
5	Cancel an order that is eligible for free cancellation (within time window, not yet shipped)	Order status = "Processing," within time window	The system should successfully cancel the order, change its status to "Canceled," and issue a full refund.	Pass	This test verifies that eligible orders are canceled correctly, ensuring a smooth experience for users during the cancellation process.
6	Cancel an order paid via credit card and verify refund	Order paid via credit card	The system should issue a refund to the user’s credit card and display a confirmation message stating that the refund process has started.	Pass	Refund processes vary by payment method; this test ensures refunds for credit card payments are handled correctly.
7	Cancel an order paid via wallet balance and verify refund	Order paid via wallet	The system should credit the refund to the user's wallet, confirming that the refund has been issued.	Pass	Ensures proper handling of refunds when orders are paid through wallet balances, improving customer satisfaction.

Test Case ID	Test Case Description	Test Data	Expected Result	Status	Notes/Reasoning
8	Cancel an order paid via a combination of payment methods (e.g., credit card + wallet) and verify refund split	Order paid via credit card + wallet	The system should refund the correct amounts to both the credit card and the wallet in accordance with the original payment split.	Pass	This test ensures the system can handle complex refund scenarios where multiple payment methods are involved, maintaining financial accuracy.
9	Initiate a refund for an order that was canceled outside of business hours (e.g., weekend or holiday)	Cancellation request during non-working hours	The system should queue the refund for processing on the next business day, informing the user of any delay in processing.	Pass	Ensures refunds are properly queued during non-working hours, maintaining transparency and accuracy in refund processing.
10	Cancel an order that was part of a promotional deal (e.g., bundled offer, discount) and verify partial refund	Promotional discount applied	The system should calculate the refund correctly, excluding any non-refundable items and adjusting for promotional discounts.	Pass	Promotional deals may have special refund rules; this test ensures refunds are calculated correctly based on promotional pricing and exclusions.
11	Attempt to cancel an order where the cancellation button is not visible	No cancellation button shown	The system should not allow cancellation if the cancellation option is unavailable for the order (e.g., after shipment).	Pass	Ensures that the system disables or hides cancellation options when not allowed, reducing confusion for the user.
12	Cancel an order and ensure the inventory is updated accordingly	Order status = "Processing"	Upon cancellation, the system should return the ordered item to inventory and update the available stock count.	Pass	Verifies that inventory systems are synced with order cancellation, ensuring stock levels are accurate for future orders.
13	Verify refund email or notification sent after successful order cancellation	Order successfully canceled	The user should receive an email or notification confirming the order cancellation and the refund process.	Pass	This test checks that proper communication is sent to the user, confirming that the cancellation and refund process has been initiated.

Notes and Reasoning

1. **Test Case Design**:
    - Each test case is designed to ensure that the registration form behaves as expected under both valid and invalid conditions.

D-2 16010122151 Hyder Presswala

- The use of **ECP** for the Order Cancellation module ensures that various classes of inputs are tested without redundant tests, making the testing process efficient.

2. **Validation Testing**:
   - Testing for empty fields, incorrect input formats (email, numbers), and invalid entries (e.g., passwords that don't match) ensures robustness in the system.
   - The reasoning behind testing edge cases (like long usernames or blank spaces) is to guarantee that the system is equipped to handle unusual user input without breaking.
3. **Refund Logic**:
   - Different payment methods require specific handling. For instance, refunds to credit cards may take time, and the system must notify the user about the expected processing time.
   - For promotional deals or combined payment methods, ensuring accurate refunds is crucial to avoid disputes with users.
4. **Inventory Update**:
   - Ensuring the inventory is updated correctly upon cancellation helps prevent overselling and stock management issues.