

Consolidated QA Observations – Functional, Validation, and UX Issues

The application exhibits multiple gaps across input validation, error handling, authorization flow, UI responsiveness, and state management. These issues collectively indicate insufficient validation sequencing, weak transactional control, and inconsistent user feedback across critical workflows.

1. Input Field Validation and Constraints

Text input fields such as First Name and Last Name do not enforce any maximum character limit, allowing users to enter excessively long values without validation or error messaging. In contrast, the Middle Name field restricts input to a single character without communicating this constraint. Field limits and validation rules are neither consistent nor disclosed to the user, resulting in unclear and unreliable input behavior.

2. Incorrect Error Placement and Delayed Validation

Validation errors are not surfaced at the appropriate field level or step. For example, an invalid phone number format error is displayed only after the user proceeds to the password creation step, instead of being validated immediately when the phone number is entered. This reflects poor validation sequencing and negatively impacts user experience.

3. Broken Registration and Authorization Flow

The registration process lacks proper pre-validation and uniqueness checks. When attempting to register with an already existing email and a different mobile number, the system allows progression to password creation and only later throws an “email already exists” error. Despite this, the newly entered mobile number appears to be saved against the existing email.

Similarly, when registering with a new email but a previously used mobile number, the system throws a database uniqueness error, indicating missing or improperly enforced unique key constraints. The registration API does not follow a transactional approach—users are allowed to proceed through steps and partial data is persisted even when core validation rules fail. This is a fundamental flaw in authorization and data integrity handling.

4. UI Responsiveness Issue – Message Button

The message button requires multiple taps to trigger its action, suggesting issues with event handling, UI responsiveness, or debounce logic. This degrades usability and gives the impression of an unresponsive interface.

5. Missing Validation in Job Creation – Amount Field

During job creation, the Amount field allows unrealistically large or invalid values to be entered. However, a \$99 limit is enforced later during the payment step. Validation rules are inconsistent across stages and should be applied at the point of data entry to prevent invalid data from progressing further in the workflow.

6. Missing Mandatory Validation and Loss of State in Job Creation

When the Time field is not provided during job creation, the system allows the user to proceed through all remaining steps without showing any validation error—even at the final submission stage.

Additionally, when navigating back to edit the specific step where the data is missing, previously entered information is not retained, forcing the user to re-enter subsequent steps. This highlights both missing mandatory validation and poor state persistence in the multi-step job creation flow.

7. Incorrect Billing Amount Display and Conflicting Validation

When an amount of \$20 is entered in the billing section, the system displays a modified value (e.g., \$16.xx). Following this, an error is shown instructing the user to increase the amount, despite the originally entered value meeting the expected minimum. This points to

inconsistencies between UI display, backend calculations, and validation logic—particularly concerning for a payment-related flow.

8. Booking Status Not Visible

After completing the booking process, the user is not shown any booking status such as pending, confirmed, failed, or cancelled. The absence of clear status feedback leaves users uncertain about the outcome of their action and undermines trust in the system.

Consolidated QA Observations – Functional, Validation, and UX Issues

The application exhibits multiple gaps across input validation, error handling, authorization flow, UI responsiveness, and state management. These issues collectively indicate insufficient validation sequencing, weak transactional control, and inconsistent user feedback across critical workflows.-----

1. Input Field Validation and Constraints
 - **Issue:** Text input fields such as **First Name** and **Last Name** do not enforce any maximum character limit, allowing users to enter excessively long values without validation or error messaging.
 - **Contrast:** The **Middle Name** field restricts input to a single character without communicating this constraint.
 - **Conclusion:** Field limits and validation rules are neither consistent nor disclosed to the user, resulting in unclear and unreliable input behavior.

2. Incorrect Error Placement and Delayed Validation

- **Issue:** Validation errors are not surfaced at the appropriate field level or step.
- **Example:** An invalid phone number format error is displayed only after the user proceeds to the password creation step, instead of being validated immediately when the phone number is entered.
- **Conclusion:** This reflects poor validation sequencing and negatively impacts user experience.

3. Broken Registration and Authorization Flow

- **Email Already Exists:** When attempting to register with an already existing email and a different mobile number, the system allows progression to password creation and only later throws an “email already exists” error. Despite this, the newly entered mobile number appears to be saved against the existing email.
- **Mobile Number Uniqueness:** When registering with a new email but a previously used mobile number, the system throws a database uniqueness error, indicating missing or improperly enforced unique key constraints.
- **Transactional Flaw:** The registration API does not follow a transactional approach—users are allowed to proceed through steps and partial data is persisted even when core validation rules fail. This is a fundamental flaw in authorization and data integrity handling.

4. UI Responsiveness Issue – Message Button

- **Issue:** The message button requires multiple taps to trigger its action.
- **Conclusion:** This suggests issues with event handling, UI responsiveness, or debounce logic, which degrades usability and gives the impression of an

unresponsive interface.

5. Missing Validation in Job Creation – Amount Field

- **Issue:** During job creation, the **Amount** field allows unrealistically large or invalid values to be entered.
- **Inconsistency:** A **\$99 limit** is enforced only later during the payment step.
- **Conclusion:** Validation rules are inconsistent across stages and should be applied at the point of data entry to prevent invalid data from progressing further in the workflow.

6. Missing Mandatory Validation and Loss of State in Job Creation

- **Missing Validation:** When the **Time** field is not provided during job creation, the system allows the user to proceed through all remaining steps without showing any validation error—even at the final submission stage.
- **Loss of State:** When navigating back to edit the specific step where the data is missing, previously entered information is not retained, forcing the user to re-enter subsequent steps.
- **Conclusion:** This highlights both missing mandatory validation and poor state persistence in the multi-step job creation flow.

7. Incorrect Billing Amount Display and Conflicting Validation

- **Display Issue:** When an amount of **\$20** is entered in the billing section, the system displays a modified value (e.g., \$16.xx).
- **Conflicting Error:** Following this, an error is shown instructing the user to increase the amount, despite the originally entered value meeting the expected minimum.
- **Conclusion:** This points to inconsistencies between UI display, backend calculations, and validation logic—particularly concerning for a payment-related flow.

8. Booking Status Not Visible

- **Issue:** After completing the booking process, the user is not shown any booking status such as pending, confirmed, failed, or cancelled.
- **Conclusion:** The absence of clear status feedback leaves users uncertain about the outcome of their action and undermines trust in the system.