On the basis of required validations map the different test cases to handle all possible critical cases which may arise during the life cycle of the software.

Table of Contents

[User Authentication 2](#_Toc152097399)

[1. Valid Login 2](#_Toc152097400)

[2. Invalid Login - Incorrect Password 2](#_Toc152097401)

[3. Invalid Login - Non-Existent User 2](#_Toc152097402)

[4. Account Lockout 2](#_Toc152097403)

[5. Successful Signup 2](#_Toc152097404)

[6. Duplicate Signup Prevention 2](#_Toc152097405)

[7. Email Verification 2](#_Toc152097406)

[8. Forgot Password - Reset Process 2](#_Toc152097407)

[Expense Tracking 2](#_Toc152097408)

[9. Add Expense 2](#_Toc152097409)

[10. Expense Validation - Negative Amount 2](#_Toc152097410)

[11. Expense Validation - Category and Description 3](#_Toc152097411)

[12. Budget Limit Check 3](#_Toc152097412)

[Dashboard and Reporting 3](#_Toc152097413)

[13. View Expense Report 3](#_Toc152097414)

[14. Data Privacy 3](#_Toc152097415)

[Settings and Preferences 3](#_Toc152097416)

[15. Change Theme 3](#_Toc152097417)

[16. Update Profile Information 3](#_Toc152097418)

[17. Test with Large Dataset 3](#_Toc152097419)

[18. Offline Mode 3](#_Toc152097420)

[19. Cross-Browser and Cross-Device Compatibility 3](#_Toc152097421)

[Security 4](#_Toc152097422)

[20. SQL Injection and XSS Testing 4](#_Toc152097423)

[21. Data Encryption 4](#_Toc152097424)

Certainly! Here's a comprehensive list of test cases, focusing on various required validations to handle critical cases throughout the life cycle of your expense-sharing app:

# User Authentication

## 1. Valid Login

- Input: Correct username and password.

- Expected Result: Successful login.

2. Invalid Login - Incorrect Password

- Input: Correct username and incorrect password.

- Expected Result: Login failure with appropriate error message.

## 3. Invalid Login - Non-Existent User

- Input: Non-existent username and any password.

- Expected Result: Login failure with appropriate error message.

## 4. Account Lockout

- Input: Multiple consecutive failed login attempts.

- Expected Result: Account lockout and appropriate notification.

## 5. Successful Signup

- Input: Valid user details during signup.

- Expected Result: Successful account creation.

## 6. Duplicate Signup Prevention

- Input: Attempt to sign up with an already existing username or email.

- Expected Result: Prevent duplicate accounts with a clear message.

## 7. Email Verification

- Input: Successful signup; check email verification process.

- Expected Result: Account activation upon successful email verification.

## 8. Forgot Password - Reset Process

- Input: User initiates the forgot password process.

- Expected Result: Secure and user-friendly password reset process.

# Expense Tracking

## 9. Add Expense

- Input: Valid details for adding an expense.

- Expected Result: Successful addition of expense to the user's account.

10. Expense Validation - Negative Amount

- Input: Attempt to add an expense with a negative amount.

- Expected Result: Prevent adding expenses with negative amounts.

## 11. Expense Validation - Category and Description

- Input: Test various combinations of valid and invalid expense categories and descriptions.

- Expected Result: Proper validation messages and correct storage of valid data.

## 12. Budget Limit Check

- Input: Adding an expense that exceeds the set budget limit.

- Expected Result: Notification and prevention of exceeding budget limit.

# Dashboard and Reporting

## 13. View Expense Report

- Input: Request to view the expense report.

- Expected Result: Display a comprehensive report with accurate expense data.

## 14. Data Privacy

- Input: Verify that user data is only accessible to the authenticated user.

- Expected Result: Ensure data privacy and security measures are in place.

# Settings and Preferences

## 15. Change Theme

- Input: Request to change the app theme.

- Expected Result: Successful theme change without affecting other functionalities.

## 16. Update Profile Information

- Input: Modify user profile information.

- Expected Result: Ensure accurate updating of user profile details.

Edge Cases and Performance:

## 17. Test with Large Dataset

- Input: Simulate performance with a large number of users and transactions.

- Expected Result: Verify system stability and responsiveness.

## 18. Offline Mode

- Input: Test app functionality in offline mode.

- Expected Result: Ensure graceful handling of offline scenarios and data synchronization when back online.

## 19. Cross-Browser and Cross-Device Compatibility

- Input: Test the app on different browsers and devices.

- Expected Result: Ensure consistent functionality and appearance.

# Security

## 20. SQL Injection and XSS Testing

- Input: Attempt to inject malicious code through input fields.

- Expected Result: Implement measures to prevent SQL injection and cross-site scripting (XSS).

## 21. Data Encryption

- Input: Verify that sensitive data is encrypted during transmission and storage.

- Expected Result: Ensure robust data security through encryption protocols.

These test cases cover various aspects of user authentication, expense tracking, dashboard and reporting, settings and preferences, edge cases, and security considerations. Regular testing throughout the software life cycle helps identify and address issues early, ensuring the overall reliability and quality of your expense-sharing app.