Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Engineering Academic Year 2024-2025

SOFTWARE TESTING & QUALITY ASSURANCE (STQA)
EXPERIMENT 01

NAME: Varenya Uchil
SAPID: 60004210121
BRANCH: Computer Engineering (C2-2)

AIM: To make a test case verification document for a smart hotel management system.

THEORY:
To develop a Test-Driven Development (TDD) Plan for a Smart Hotel Management System with a focus on payment processing, the following steps can be adapted and tailored to address the specific needs of the hotel industry. Here's a detailed explanation and application of the general TDD steps for this use case:

1. Identify Payment Requirements

Payment Process Definition for Smart Hotel Management System:

- Integration with Payment Gateways: Identify the specific payment providers the hotel system will use (e.g., Stripe, PayPal, or local payment solutions).
- Payment Methods: Support for various payment methods such as credit/debit cards, digital wallets, online bank transfers, and possibly cryptocurrency.
- Booking and Payment Flow: Determine if payments happen during booking, check-in, or checkout and whether they support partial payments, full payments, or deposits.
- Error Handling: Define how the system will handle payment failures, including network issues, declined transactions, or fraud detection.

2. Clearly Define Payment Requirements

- The payment gateway integration should be designed to accept payments securely.
- Ensure compatibility with different payment methods (e.g., cards, wallets).
- Error Handling: How to handle declined payments, failed transactions, timeouts, and invalid input (wrong card number, expired cards, etc.).

3. Write Test Cases

Create specific test cases that address various payment-related scenarios. These test cases should include both typical and edge cases:

- Positive Test Cases:
  - Successful payment for a hotel booking using a valid credit card. ○ Payment through PayPal for booking a premium room.

- Negative Test Cases:
  - Declined payment due to insufficient funds or incorrect credit card details. ○ Invalid credit card number or expiration date input.
- Edge Case Test Cases:
  - Payment handling for a booking with a very large amount (e.g., booking for an extended stay or multiple rooms).
  - Timeout scenario where payment process takes too long.
  - Failure handling when the payment gateway is temporarily unavailable.

4. Run Initial Tests (Red Phase)
Run the payment-related test suite. Since you haven't implemented the payment processing yet, all tests should fail, indicated by a "red" status. This shows the gap between the desired functionality and current implementation.

5. Write Code (Green Phase)
Implement the payment functionality:

- Integrate with the chosen payment gateways (Stripe, PayPal, etc.).
- Implement handling of different payment methods (cards, wallets).
- Ensure secure payment transaction handling according to PCI-DSS (Payment Card Industry Data Security Standard) requirements.
- Handle user inputs for booking amounts, billing info, etc., securely.

Once the code is written, run the tests again. The goal is for all tests to pass, indicated by a "green" status.

6. Run Tests Again (Refactor Phase)

- After implementing the payment system, run the test suite again to check that all tests pass. If any test fails, go back to the code and modify it until all tests pass successfully.
- Refactor the code to improve performance or make it more maintainable, ensuring that no test cases break in the process.

7. Error Handling and Edge Cases

- Write additional test cases to handle specific error scenarios: ○ Payment gateway is temporarily unavailable.
  - The user enters incorrect payment information (e.g., expired card).
  - Transaction amount exceeds the available balance (e.g., user tries to book more than available credit).
  - Handle large booking amounts (e.g., multiple rooms or extended stays).

- Ensure the system behaves correctly when network issues occur (timeout, connection failures, etc.).

8. Automate Testing

- Automate the payment test suite to be continuously executed as part of the development process. Automated tests help to catch regressions and ensure that the payment functionality remains reliable.
- Set up continuous integration/continuous delivery (CI/CD) pipelines that include these automated tests, ensuring that payment functionalities are verified on every code update.

9. Integration Testing

Perform integration testing with other parts of the hotel management system to ensure seamless operation. This includes:

- ○ Testing the booking process (selecting room, entering details, and making a payment).
- ○ Verifying that the system properly updates the room availability and guest records after a successful payment.
- ○ Ensuring that the system properly manages inventory, including booking and cancelling rooms.
- ○ Validate if the payment process is integrated well with order processing and check-out operations.

10. Security Testing

Conduct security testing to ensure that payment transactions are secure, and sensitive information like credit card numbers, user details, and payment data is protected.

- ○ Validate encryption during data transmission between the hotel system and payment gateway. ○ Ensure compliance with security standards such as PCI-DSS.
- ○ Perform vulnerability scans to detect and resolve any security weaknesses (e.g., SQL injection, cross-site scripting).
- ○ Test handling of fraudulent transactions and establish fraud prevention mechanisms.

11. Documentation

Document the entire payment testing process, including:

- ○ Detailed test cases, with expected outcomes for each scenario.
- ○ Description of payment methods and payment gateways used.
- ○ Any special configurations or environments required for testing (e.g., sandbox accounts, test servers).
- ○ Record the results of security tests and any vulnerabilities identified and resolved.

Department of Computer Engineering Academic Year 2024-2025

TEST CASE SCENARIO 1 - LOGIN SYSTEM

Test	Test Objective	Precondition	Steps	Test data	Expected result	Postcondition	Pass / Fail
TC_001	Ensure that a user cannot access the system using incorrect login details.	None	Open the login page.	Invalid username and incorrect password	Error message should appear stating that login credentials are incorrect.	Error message displayed indicating invalid credentials	Pass
			Enter an incorrect username and password.				
			Click the "Login" button.				
TC_002	Confirm that a user can successfully log in using correct credentials.	The user must have an active account in the system.	Open the login page.	Correct username and password	The system should successfully log in the user and navigate to the main dashboard or home screen	User successfully logged in and redirected to the home page.	Pass
			Input a valid username.				

			Input the correct password				
			Click the "Sign In" button.				
TC_003	Validate "Remember Password" option retains login.	Active user account required.	Open login page	Correct username & password	User stays logged in after reopening the browser.	User remained logged in.	Pass
			Select "Remember Me."				
			Enter valid credentials.				
			Click "Login."				
			Close and reopen the browser.				
TC_004	Ensure the user can log out successfully.	Users must be logged in.	Click the "Logout" button or link.	User is redirected to the login page.	Logged-in user session	User is redirected to the login page.	Pass

TEST CASE SCENARIO 2 - FEATURES

Test	Test Objective	Precondition	Steps	Test data	Expected result	Postcondition	Pass / Fail
TC_001	Verify that the user can check table availability.	User is on the table reservation page.	Navigate to the table reservation page. Enter date, time, and number of guests.	Date, Time, Guests	."Available" message for open tables.	Actual availability status.	Pass
TC_002	Verify that the user can reserve a table.	Tables are available for the selected time	Select the desired table. Confirm reservation details.	Customer Name, Contact Details	Table reservation confirmation displayed.	Actual confirmation message.	Pass
TC_003	Verify that the user can order food through the app.	User is logged in and has selected a table.	Navigate to the menu page. Select food items and add them to the cart. Place order.	Menu Items, Quantity	Order placed successfully with an order ID.	Actual order ID displayed.	Pass

TC_004	Verify that the system updates table status postreservation.	Table reservation is successful .	Check the table status for the selected time slot.	Database Table Info	Table status updated to "Reserved" in the database.	Actual table status in the database.	Pass
TC_005	Verify the customer who booked the table	The customer should be able to read properly	A reCAPTCHA window is displayed	Invalid reCAPTCHA	User Verification successful	The table is booked successfully	Pass
		The code must be entered before the window timeout	After entering valid reCAPTCHA the user is verified	System timeout			
TC_006	Verify Customization option is available for every order	Customer must select food item	Choose the "Customise" option	Items needed for customisation available	Order status updated to "Updated" in the database.	Customisation of selected food item successful	Pass
			Customise the food item according to his/her needs				
			Save the customisation and add to order				

TC_007	Return Food Item	The customer must have purchased the food item.	Select the food item to return and initiate the return process through the system.	Wrong Food order(s)	Request for Return Food Item Successful	The returned food item should be removed from the customer's order history.	Pass
		The food item must be in its original condition.	Provide a reason for the return (if required).	Valid Reason for return	Refund Received	The customer should be notified of the return status.	
		The return policy must be valid (e.g., within a certain time frame).	Complete the return process.				

TEST CASE SCENARIO 3 - PAYMENT SYSTEM

Test	Test Objective	Precondition	Steps	Test data	Expected result	Postcondition	Pass / Fail
TC_001	Payment Successful	Customer has selected items for purchase.	Proceed to the checkout/payment screen.	Valid payment methods	Payment Successful	The purchased items should be marked as paid.	Pass

		Payment method is selected and valid.	Enter payment details (e.g., card details, cash amount).		Receipt Displayed	The customer should receive a receipt (if applicable).	
			Confirm the payment.				
			Verify the payment confirmation message.				

Department of Computer Engineering Academic Year 2024-2025

TC_002	Payment Unsuccessful	Customer has selected items for purchase.	Proceed to the checkout/payment screen.	Invalid payment methods	Payment Unsuccessful	The customer should be able to retry the payment or choose a different payment method.	Pass
		Payment method is selected.	Enter payment details (e.g., card details, cash amount).				
			Simulate scenarios where the payment might fail, such as: Invalid card details. Insufficient funds. Network error during payment processing.				