

Task 1: Write Test Cases for UPI Payments (Any App)

Objective: Develop comprehensive test cases for a UPI payments feature, ensuring the functionality meets requirements.

Analyze requirements

Write Test Cases

Suggest improvements in the UPI flow for better UX, security, or performance.

Test Case Categories

1. User Registration and Login

TC1: Verify if users can register successfully using their mobile number.

TC2: Verify if users can log in with valid credentials.

TC3: Verify if OTP authentication works as expected during registration/login.

TC4: Verify error messages for incorrect OTP or expired OTP.

2. Bank Account Linking

TC5: Verify if users can link their bank accounts through UPI.

TC6: Verify if the app fetches the bank details automatically using the registered mobile number.

TC7: Verify error handling for invalid bank account numbers or unregistered mobile numbers.

3. UPI PIN Setup

TC8: Verify if users can set a UPI PIN securely.

TC9: Verify if the app prompts for a UPI PIN reset in case of a forgotten PIN.

TC10: Verify error messages for invalid or mismatched PIN entries.

4. Transaction Flow

TC11: Verify if users can initiate a payment by entering the correct UPI ID or scanning a QR code.

TC12: Verify if the app prompts for UPI PIN during payment.

TC13: Verify if transactions are successful with valid UPI PIN and sufficient balance.

TC14: Verify error messages for:

Insufficient balance.

Incorrect UPI PIN.

Invalid UPI ID.

TC15: Verify if users receive a confirmation message after successful transactions.

TC16: Verify if failed transactions provide clear error details.

5. Payment Requests

TC17: Verify if users can request money by entering a UPI ID or sharing a QR code.

TC18: Verify if the sender receives a notification for the payment request.

TC19: Verify if users can cancel pending payment requests.

6. Transaction History

TC20: Verify if users can view transaction history, including date, time, and status.

TC21: Verify if users can filter transactions by type (sent/received/failed).

7. Notifications

TC22: Verify if users receive notifications for all transactions (successful/failed).

TC23: Verify if notifications include all relevant details like amount, recipient name, and tran

8. Security Features

TC24: Verify if the app logs out users after a period of inactivity.

TC25: Verify if users cannot access UPI features without unlocking the app using biometric a

TC26: Verify if sensitive data (UPI PIN) is not visible on the screen or logs.

Suggestions for Improvement

1. Enhanced Security:

Implement two-factor authentication for large transactions.

Notify users immediately in case of unusual transaction patterns.

2. Better UX:

Simplify error messages with actionable suggestions (e.g., "Please re-enter the correct UPI

Add a progress indicator during transactions to reduce user anxiety.

3. Performance Optimization:

Ensure quick transaction processing times, even during peak hours.

Optimize QR code scanning for faster recognition.

4. Additional Features:

Allow users to save frequent UPI IDs as favorites for quick access.

Enable dark mode for better usability in low-light environments.

Task 2: Write Test Cases and Report Bugs for Specific Applications

1. Online Sweet Shop (<https://sweetshop.netlify.app/>)

Test Cases

Positive Cases

TC1: Verify if users can browse products by categories.
TC2: Verify if users can add items to the cart.
TC3: Verify if users can place an order by filling in valid details.
TC4: Verify if users receive an order confirmation message.

Negative Cases

TC5: Verify if an error message is shown when trying to add items without logging in (if login is required).
TC6: Verify behavior when payment is canceled by the user.

Edge Cases

TC7: Verify if the app handles a large number of items in the cart without crashing.
TC8: Verify response when entering invalid delivery address details (e.g., special characters, missing fields).

Bug Report

Bug ID	Issue	Steps to Reproduce
BUG-001	Image of some sweets is not loading	Browse to “Cakes” section.
BUG-002	Discount code not applying to total price	Apply a valid discount code during checkout.

Suggestions for Improvement

1. Add a search bar to find sweets quickly.
2. Include customer reviews for each product.
3. Improve mobile responsiveness.

[2. Parking Cost Calculator \(https://www.shino.de/parkcalc/\)](https://www.shino.de/parkcalc/)

Test Cases

Positive Cases

TC1: Verify if users can calculate parking costs by selecting valid dates and times.
TC2: Verify if different parking types (e.g., Short-term, Long-term) show accurate pricing.

Negative Cases

TC3: Verify behavior when users enter invalid dates or times (e.g., end time earlier than start

TC4: Verify response when required fields are left empty.

Edge Cases

TC5: Verify calculations for maximum allowed parking duration.

TC6: Verify if the app handles overlapping dates correctly.

Bug Report

Bug ID	Issue	Steps to Reproduce
BUG-001	Error in cost calculation	Select "Valet Parking" for 5 days.
BUG-002	End time earlier than start time	Enter start time as "10:00 AM" and end as "9:00 AM".

Suggestions for Improvement

1. Add tooltips explaining parking types.
2. Highlight required fields to avoid confusion.
3. Improve error handling for invalid date/time inputs.

[3. Polymer E-Commerce \(https://shop.polymer-pr\)](https://shop.polymer-pr)

Test Cases

Positive Cases

TC1: Verify if users can search and browse products.

TC2: Verify if users can add items to the cart and proceed to checkout.

Negative Cases

TC3: Verify behavior when adding out-of-stock items to the cart.

TC4: Verify if the app handles payment failures gracefully.

Edge Cases

TC5: Verify performance when loading many products on the homepage.

TC6: Verify handling of large coupon codes.

Bug Report

Bug ID	Issue	Steps to Reproduce
BUG-001	Website freezes with 50+ products	Add 50+ products to the cart.
BUG-002	Incorrect stock status	Check the stock status for any random product.

Suggestions for Improvement

1. Optimize product loading speed.
2. Add a filter for sorting products by price or category.
3. Include a wishlist feature.

[4. To-Do List \(https://todolist.james.am/#/\)](https://todolist.james.am/#/)

Test Cases

Positive Cases

TC1: Verify if users can add tasks with valid titles.

TC2: Verify if users can mark tasks as completed.

Negative Cases

TC3: Verify behavior when adding a blank task.

TC4: Verify if users can delete tasks.

Edge Cases

TC5: Verify app performance with 500+ tasks added.

TC6: Verify if special characters in task titles are accepted.

Bug Report

Bug ID	Issue	Steps to Reproduce
BUG-001	Tasks disappear after refresh	Add tasks and refresh the page.
BUG-002	Unable to delete completed tasks	Complete a task and try to delete it.

Suggestions for Improvement

1. Add a priority option for tasks (e.g., High, Medium, Low).
2. Enable syncing with cloud storage for task persistence.
3. Include a dark mode for better usability.

Task 3: Write User Stories for Online Banking.

Objective: Create user stories to describe key functionalities of the online banking demo

<https://www.digindemo.com>

Explore the demo application to understand user interactions.

User Stories

1. User Login

As a registered user,

I want to log in securely using my username and password,

So that I can access my banking account.

2. View Account Balance

As a user,

I want to view my account balance on the dashboard,

So that I can keep track of my finances easily.

3. Transfer Money

As a user,

I want to transfer money to another account by entering the recipient's details,

So that I can pay bills or send money quickly.

4. View Transaction History

As a user,

I want to see a list of my past transactions,

So that I can review my spending and income.

5. Pay Bills

As a user,

I want to pay utility bills (e.g., electricity, water) through the app,

So that I can manage my payments conveniently.

6. Add Beneficiary

As a user,

I want to add a new beneficiary to my account,

So that I can save their details for future transactions.

7. Reset Password

As a user,

I want to reset my password if I forget it,

So that I can regain access to my account securely.

8. Logout

As a user,

I want to log out of the application securely,

So that my account information remains safe.

9. Set Transaction Limits

As a user,

I want to set a daily transaction limit,

So that I can manage my spending securely and avoid unauthorized transactions.

10. Receive Notifications

As a user,

I want to receive notifications for transactions or changes in my account,
So that I stay informed about my account activities.

11. Update Profile Information

As a user,
I want to update my personal details, such as email and phone number,
So that my account information remains accurate.

Suggestions for Enhancements

1. Add a **dark mode** for better user experience in low-light conditions.
2. Include **chat support** for resolving user issues quickly.
3. Implement **multi-factor authentication** for enhanced account security.

Task 5: Explain Different Types of Testing with Examples

Testing Type	Description	Example
Manual Testing	Testing the software manually without using tools.	Opening a login page and entering a username and password to see if it works.
Automation Testing	Using tools or scripts to test the software automatically.	Using Selenium to test 100 login attempts with different usernames and passwords.
Functional Testing	Testing if the features of the software work as expected.	Checking if the "Add to Cart" button adds the selected product to the cart.
Non-Functional Testing	Testing performance, speed, and usability instead of features.	Checking if a website loads within 3 seconds with 1000 users accessing it simultaneously.
Unit Testing	Testing small parts (units) of the software, like individual functions.	Testing a function in a banking app that calculates interest on savings.
Integration Testing	Testing how different parts of the software work together.	Verifying if the payment gateway works correctly after entering payment details.
System Testing	Testing the entire application to ensure it meets requirements.	Testing the full e-commerce app, from login to payment and order confirmation.
User Acceptance Testing	Testing by actual users to confirm the software meets their needs.	A bank employee tests a new banking system before it is launched.

Performance Testing	Testing how fast or stable the software is under different conditions.	Checking if a social media app loads quickly with 1 million active users.
Load Testing	Testing software behavior with a large number of simultaneous users.	Simulating 10,000 users logging in to a website at the same time.
Stress Testing	Testing how the software behaves under extreme conditions.	Checking if a payment app crashes when processing 1 million transactions in a short time.
Regression Testing	Ensuring new updates or fixes don't break existing features.	Verifying login functionality after a bug fix for the signup process.

Task 4: Create a Bug Logging Template

Field Name	Example
Bug ID	BUG-001
Bug Title	Login button not working
Reported By	Dharmveersinh Jadeja
Date Reported	09-01-2025
Environment	Device: Desktop, OS: Windows 10, Browser: Chrome v114
Severity	High
Priority	High
Steps to Reproduce	1. Open the login page. 2. Enter valid credentials. 3. Click the login button.
Expected Result	The user should be logged in and redirected to the dashboard.
Actual Result	Clicking the login button does nothing.
Attachments	Screenshot of the login page after clicking the button.
Status	Open
Assigned To	Devesh Pant

s required).

).

Expected Result	Actual Result	Severity
All product images should load.	Images are missing for some items.	Medium
Discount should reduce the price.	No change in total price.	High

time).

Expected Result	Actual Result	Severity
Total should be accurate.	Incorrect total displayed.	High
Show an error message.	No error message shown.	Medium

Expected Result	Actual Result	Severity
App should work normally.	The app becomes unresponsive.	High
Display correct stock info.	Shows "In Stock" for out-of-stock items.	Medium

Expected Result	Actual Result	Severity
--------------------	------------------	----------

Tasks should persist.	Tasks are missing.	High
-----------------------------	-----------------------	------

Task should be deleted.	Deletion does not work.	Medium
-------------------------------	-------------------------------	--------