TEST CASE DEVELOPMENT REPORT – AGILE MODEL

Project: ACKO Insurance Website (https://www.acko.com/)  
Prepared by: Group 2  
Date: 6th August

## 1.0 INTRODUCTION

ACKO is a digital-first insurance platform offering motor, health, and travel insurance directly to consumers. The goal of this document is to define a structured and Agile-based test case development plan for ACKO’s web application, considering the core customer journeys like policy buying, login, claims, and renewals.

## 2.0 OBJECTIVE

- To thoroughly test ACKO’s features through well-documented test cases.  
- Ensure all test cases are derived from real user stories or scenarios.  
- Enable automation readiness by identifying repeatable test cases.  
- Maintain test quality through rigorous review and updates per sprint.

## 3.0 TEST CASE DEVELOPMENT PHASE

This phase involves:  
- Test Design: Identifying scenarios, positive/negative paths, boundary conditions.  
- Test Case Creation: Writing test steps with clear input, action, and expected results.  
- Test Case Review: Peer-review of each test case to validate completeness and accuracy.  
- Automation Identification: Marking test cases suitable for Selenium automation.

## 4.0 TEST METHODOLOGY (AGILE)

Testing follows the Agile Scrum model:  
- Test cases are created per sprint, based on user stories.  
- QA is involved from Sprint Planning to Retrospective.  
- Testing begins in parallel with development using early builds or mock APIs.  
- Automation scripts are developed for regression in parallel or following sprints.

## 6.0 Requirements Overview:

ACKO is a digital insurance provider with modules for user registration/login, insurance policy purchase, claim processing, policy renewal, and payments.

**Sprint Plan:**

**Sprint 1: User Login & Dashboard (1 week)**

- User Story 1: As a user, I want to log in using mobile number and OTP.  
- User Story 2: As a user, I want to view my policy dashboard.  
- Deliverables: Login with OTP, basic dashboard validations.

**Sprint 2: Policy Purchase Flow (1.5 weeks)**

- User Story 1: As a user, I want to get car/bike insurance quotes.  
- User Story 2: As a user, I want to enter vehicle & personal details and buy a policy.  
- Deliverables: Policy quote, details page, summary and mock purchase flow tested.

**Sprint 3: Claims & Renewals (1 week)**

- User Story 1: As a user, I want to raise a claim for my policy.  
- User Story 2: As a user, I want to renew my policy before expiry.  
- Deliverables: Claim form testing, document upload, renew flow with validation.

**Sprint 4: Payments & Offers (1 week)**

- User Story 1: As a user, I want to pay using UPI/card/net banking.  
- User Story 2: As a user, I want to apply promo codes for discounts.  
- Deliverables: Payment gateway (mock), coupon validation.

**Sprint 5: Regression & Automation (1 week)**

- Regression testing of login, purchase, claim, and payment flows.  
- Automation using Selenium for key flows.  
- Deliverables: Test report, automation scripts.

## 6.1 TEST CASE SCENARIO

**Sprint 1: User Login & Dashboard**

**Positive Scenarios:**

User can log in successfully with valid mobile number and OTP.

User is redirected to dashboard after successful login.

Dashboard correctly displays all active policies.

**Negative Scenarios:**

User enters invalid mobile number and receives an appropriate error message.

User enters incorrect OTP and is denied access.

Attempting login without entering mobile number or OTP triggers validation messages.

**Sprint 2: Policy Purchase Flow**

**Positive Scenarios:**

User can fetch car/bike insurance quotes by entering valid vehicle details.

Personal details form accepts valid inputs and allows proceeding to payment.

User can view policy summary before final confirmation.

**Negative Scenarios:**

User enters invalid vehicle number or incomplete details – form shows validation errors.

Policy quote is not generated for missing mandatory fields.

Invalid inputs in personal details section (e.g., alphabets in phone number) block submission.

**Sprint 3: Claims & Renewals**

**Positive Scenarios:**

Logged-in user can raise a claim by entering valid policy number and uploading documents.

System allows successful policy renewal before expiry with valid credentials.

User gets confirmation after submitting claim.

**Negative Scenarios:**

Claim fails when user submits form with missing documents.

Renewal fails for expired policies or invalid details.

Error shown when trying to claim a policy that is not associated with the user.

**Sprint 4: Payments & Offers**

**Positive Scenarios:**

User completes payment using valid UPI/Card/Net banking details.

User applies valid promo code and receives discount.

**Negative Scenarios:**

Invalid payment details (e.g., expired card) are rejected.

Applying invalid or expired promo code shows proper error message.

Attempt to pay without selecting a policy triggers error.

## 6.2 SAMPLE TEST CASE STRUCTURE

Test Case ID: TC\_ACKO\_LOGIN\_001  
Module: Login  
Test Scenario: Valid mobile OTP login  
Test Steps:  
 1. Go to https://www.acko.com  
 2. Click on "Login"  
 3. Enter valid mobile number  
 4. Enter correct OTP  
Expected Result:  
 - User is redirected to dashboard  
 - Policy summary is displayed  
Test Type: Functional / Smoke  
Priority: High  
Automation Candidate: Yes

## 7.0 TEST CASE REVIEW

- Peer testers or QA leads review the test cases using checklists.  
- Ensure all edge cases, validations, and UI elements are covered.  
- Update test cases based on changing user stories during sprint.  
- Approved test cases are pushed to Test Management Tool (e.g., TestRail).

## 8.0 AUTOMATION SELECTION

- Stable, repeatable test cases (e.g., login, policy purchase, payment flow) are selected.  
- Selenium with Java/Python used for writing automation scripts.  
- Scripts are tagged by module and run in CI/CD pipelines using Jenkins.

## 9.0 DELIVERABLES

- Test Scenarios Document  
- Test Case Document (Reviewed)  
- Automation Test Scripts (Sprint-wise)  
- Defect Reports (from JIRA)  
- Sprint-wise QA Sign-off Document

## 10.0 TOOLS USED

- Test Case Management: Excel / TestRail  
- Bug Tracking: JIRA  
- Automation: Selenium WebDriver (Java)  
- API Testing: Postman  
- CI/CD: Jenkins

## 11.0 CONCLUSION

The Agile test case development process ensures each sprint delivers test-ready and automation-friendly test cases. By testing continuously and iteratively, ACKO ensures product stability, faster releases, and enhanced user experience.