Acko.com

**Test Strategy**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| 12/3/25 | 1.0.0 | XYZ | Made some changes |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Scope 3](#_Toc534636696)

[2. Test Approach 4](#_Toc534636697)

[3. Test Environment 5](#_Toc534636698)

[4. Testing Tools 6](#_Toc534636699)

[5. Release Control 7](#_Toc534636700)

[6. Risk Analysis 8](#_Toc534636701)

[7. Review and Approvals 9](#_Toc534636702)

# Scope

This section defines the boundaries of testing for the Acko.com platform.  
It outlines the functionalities that will be **included** and **excluded** during the testing phase. The test scope ensures that all critical journeys like **insurance purchase**, **policy renewal**, **claim filing**, and **user management** are covered. It also highlights areas such as **backend services** or **third-party systems** that are not within the scope of QA efforts.

# Test Approach

This section describes the **strategy and methodology** that will be followed for testing.  
Acko.com will follow the **Agile testing model**, where testing is conducted within sprints, ensuring continuous integration and early bug detection.  
Key points in the approach:

* Manual testing of new features
* Regression testing in each sprint
* Automation of stable and frequently used flows
* API testing for backend validations
* Collaboration with developers for early validation (shift-left testing)

# Test Environment

This section specifies the hardware and software configuration required for test execution.  
It includes details like:

* **QA, UAT, and staging environments** setup
* Supported **browsers** (Chrome, Firefox, Safari, Edge)
* Supported **devices** (Android, iOS, desktops)
* Simulated **network conditions** (4G, Wi-Fi, low-bandwidth)This section also ensures that the environment mirrors production settings as closely as possible for accurate testing.

# Testing Tools

A list of tools that will be used for test case management, defect tracking, automation, and performance monitoring.

| **Purpose** | **Tool** |
| --- | --- |
| Test Management | Jira / TestRail / Zephyr |
| UI Automation | Selenium / Cypress |
| Mobile Testing | Appium |
| API Testing | Postman |
| Cross-Browser Testing | BrowserStack |
| CI/CD Integration | Jenkins, GitHub Actions (if applicable) |

# Release Control

Define show releases will be **planned, managed, and approved**.  
Key practices include:

* **Build versioning and tagging**
* All changes must be tested and passed in UAT before production deployment
* QA must validate the final release candidate and sign off
* Any blocker or unresolved high-severity issue halts the release
* Release notes, rollback plans, and post-release validations are included

This ensures high confidence in each production release.

# Risk Analysis

Identifies potential risks that may impact the success of testing or application performance.

| **Risk** | **Mitigation** |
| --- | --- |
| * Unstable environment | * Allocate buffer time for setup and monitoring |
| * Incomplete requirements | * Involve QA in sprint planning and grooming |
| * Browser/device compatibility | * Use BrowserStack for wide coverage |
| * Third-party failure (e.g., payment gateway) | * Use test stubs or mocks in lower environments |

# Review and Approvals

Outlines the **review and approval process** for the test plan, test cases, and test execution results.  
It ensures that:

* QA artifacts are reviewed by QA Leads and Business Analysts
* Product Owners review test coverage and outcomes
* Approvals are documented before UAT and production sign-off
* Post-execution reviews (retrospectives) help improve future test cycle