📄 Test Strategy Document for WhatsApp

1. Introduction  
   This Test Strategy outlines the testing approach for WhatsApp, covering its core functionalities including messaging, media sharing, voice/video calling, group chat, status updates, privacy controls, and notifications. This strategy ensures comprehensive testing, quality assurance, and risk mitigation prior to release.
2. Objectives

* Ensure WhatsApp core features work as intended across supported devices and platforms.
* Validate functional, non-functional, and regression scenarios with traceability to requirements.
* Deliver a defect-free, secure, and performance-optimized product.
* Maintain high-quality standards through systematic planning, execution, and automation.

1. Scope

In Scope:

* Functional Testing: Chat, group chat, status, calls, media sharing, settings, notifications.
* Non-Functional Testing: Performance, Security, Usability, Accessibility.
* Regression Testing: Existing functionalities post new feature integrations.
* Cross-Platform Testing: Android, iOS, Web/Desktop.
* Internationalization (i18n) & Localization (l10n).

Out of Scope:

* Backend database testing (unless explicitly required)
* Third-party integration (e.g., payment gateways, unless owned by WhatsApp)

1. Test Approach

4.1 Test Levels:

* Unit Testing: Owned by developers
* Integration Testing: Communication between modules (e.g., media + chat delivery)
* System Testing: End-to-end flows in production-like environment
* UAT (User Acceptance Testing): End-user validation (internal stakeholders)

4.2 Test Types:

* Manual Testing: For exploratory, UI/UX, and ad-hoc scenarios.
* Automation Testing: Regression suite using Appium/Selenium/TestNG/JUnit for mobile/web.
* API Testing: Postman/RestAssured for backend services.
* Performance Testing: Using JMeter or Gatling for stress/load tests.
* Security Testing: Vulnerability assessment using OWASP standards.

1. Test Deliverables

* Test Plan
* Test Strategy Document
* Test Cases (Functional, Non-functional)
* Test Data
* Daily/Weekly QA Status Reports
* Defect Reports with root cause analysis
* Traceability Matrix
* Final Test Summary Report

1. Test Environments

* Mobile: Android 10+ (various OEMs), iOS 13+
* Web/Desktop: Latest Chrome, Firefox, Safari, Edge
* Simulators & Emulators: Limited use (real devices preferred for core flows)
* Backend: Staging servers mirroring production setup

1. Entry & Exit Criteria

Entry Criteria:

* Feature requirements are finalized and signed off
* Development is completed and unit testing is passed
* QA environment is stable and accessible
* Test data is available

Exit Criteria:

* All critical and high severity bugs are closed or deferred with business sign-off
* Regression and smoke test suites are passed
* Test coverage is ≥ 90% of defined test cases
* Test summary report is approved by QA lead

1. Defect Management

* Defect Tracking Tool: JIRA
* Defect Lifecycle: New → Assigned → In Progress → Fixed → Retested → Closed
* Priority and Severity to be triaged daily with dev/product team

1. Roles & Responsibilities

| **Role** | **Responsibility** |
| --- | --- |
| QA Lead | Define strategy, manage risks, review deliverables |
| QA Engineer | Design test cases, execute tests, report defects |
| Automation Engineer | Build & maintain automation framework |
| Developer | Unit testing, defect fixes, support during testing |
| Product Manager | Requirements sign-off, UAT & acceptance |

1. Tools & Automation Framework

* Test Management: TestRail / Zephyr
* Bug Tracking: JIRA
* Automation: Appium, Selenium, Java/TestNG
* Performance: JMeter, BlazeMeter
* CI/CD Integration: Jenkins, GitHub Actions
* Device Cloud: BrowserStack / Firebase Test Lab

1. Risk Management

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Changing requirements | Agile ceremonies and early involvement of QA |
| Device fragmentation | Use of cloud device farms & prioritized testing |
| Delays in test data/setup | Automation of environment provisioning |
| Test coverage gaps | Review cycles with QA lead and developers |

1. Communication Plan

* Daily Stand-ups: Progress update & blocker discussion
* Weekly QA Status Report: Sent to all stakeholders
* Sprint Review: QA demo and feedback incorporation
* Post-Mortem: Analysis of release bugs and improvements

1. Metrics & KPIs

* Defect Density
* Test Coverage %
* Automation Coverage
* Defect Reopen Rate
* Time to Resolution
* Test Execution Rate

1. Approval

| **Name** | **Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
| QA Lead | QA Reviewer | [Sign here] | [YYYY-MM-DD] |
| Test Manager | Approver | [Sign here] | [YYYY-MM-DD] |