SQACIRCLE

QA Interview Handbook

84 interview questions, with short answers

- **Fundamentals and types of testing**
- **Test design and bug triage**
- **Real-time scenarios (fund transfer, crashes, location)**
- **API, SQL and backend questions**
- **Agile, tools and HR questions**

Created By
IQRA ILYAS
iqrailyas.com

SQACircle
Let's Connect on Trust

QA Interview Handbook

ABOUT THIS HANDBOOK

Who this is for: Freshers and QA engineers preparing for interviews. These are questions that interviewers ask. Every answer is short on purpose, so you can say it out loud in under a minute.

How to use this: Go through one section a day. Answer each question out loud first, then read the answer. Use the reference links for topics you are less sure about.

Contents

1. Fundamentals	13 questions
2. Types of testing	14 questions
3. Test design techniques	6 questions
4. Bugs, severity and triage	8 questions
5. Real-time scenarios	12 questions
6. API and backend	9 questions
7. Mobile, security and performance	9 questions
8. Agile, tools and HR questions	13 questions

1. Fundamentals

The base every interviewer starts from. **13 questions**

Q1. What is SDLC?

SDLC (Software Development Life Cycle) is a process including phases: Requirement → Design → Development → Testing → Deployment → Maintenance.

Reference: geeksforgeeks.org/software-development-life-cycle-sdlc

Q2. On which stages of SDLC, STLC is involved?

STLC (Software Testing Life Cycle) runs in parallel with SDLC, starting from requirement analysis, test planning, design, execution, defect reporting, and closure. It aligns with SDLC phases from requirements through maintenance.

Reference: geeksforgeeks.org/software-testing-life-cycle-stlc

Q3. What is the difference between QA & QC?

- **QA (Quality Assurance):** Process-oriented, focuses on preventing defects.
- **QC (Quality Control):** Product-oriented, detects defects in actual deliverables.

Reference: guru99.com/quality-assurance-vs-quality-control.html

Q4. What is the difference between Verification and Validation?

- **Verification:** Ensures product is built correctly (reviews, inspections).
- **Validation:** Ensures correct product is built (executing test cases).

Reference: guru99.com/verification-vs-validation.html

Q5. Suppose you have a sign-up page. What is the verification and validation process?

- **Verification:** Review design docs, field requirements (email format, password rules).
- **Validation:** Execute test cases to ensure actual sign-up works as intended.

Reference: guru99.com/verification-vs-validation.html

Q6. What are Principles of QA?

Key QA principles: defect prevention over defect detection, early testing, customer focus, continuous improvement, and risk-based prioritization.

Reference: guru99.com/software-quality-assurance.html

Q7. What are different testing techniques?

Techniques include Black Box, White Box, Boundary Value Analysis, Equivalence Partitioning, Regression, Smoke, Sanity, Performance, and Security testing.

Reference: guru99.com/types-of-software-testing.html

Q8. What is the difference between Test Case and Test Suite?

- **Test Case:** A single scenario with steps, inputs, and expected result.
- **Test Suite:** A collection/group of related test cases executed together.

Reference: guru99.com/test-case.html

Q9. What is RTM?

RTM (Requirements Traceability Matrix) maps requirements to test cases to ensure full coverage and no missing functionality.

Reference: guru99.com/rtm-requirement-traceability-matrix.html

Q10. What is the difference between Build vs Release?

- **Build:** Compiled software given to QA for testing (may still have bugs).
- **Release:** Stable build delivered to end-users after QA approval.

Reference: sqa.fyicenter.com/FAQ/Testing-Techniques/What_is_the_difference_between_build_and_r

Q11. What is Hotfix?

A hotfix is an urgent patch released directly to production to fix a critical bug without waiting for the next release cycle.

Reference: guru99.com/what-is-hotfix.html

Q12. What is Latent Defect?

A hidden defect not detected during testing, only appearing under rare conditions in production. Example: memory leak after months of use.

Q13. What is a Freaky Bug?

A freaky bug is an intermittent or rare defect that occurs under unpredictable conditions, often hard to reproduce consistently.

Reference: softwaretestinghelp.com/software-tester-interview-questions

2. Types of testing

What each type is, and when you use it. **14 questions**

Q14. What is the difference between Black Box Testing and White Box Testing?

- **Black Box:** Testers check functionality without knowing internal code.
- **White Box:** Testers verify internal code structure, logic, and paths.

Reference: guru99.com/black-box-testing.html

Q15. What is Regression Testing?

Regression testing ensures that new code changes don't break existing functionality. It is executed after bug fixes, enhancements, or new features.

Reference: atlassian.com/continuous-delivery/software-testing/regression-testing

Q16. What is Smoke Testing?

Smoke testing checks critical functionalities of a new build (like login, navigation) to ensure it's stable enough for deeper testing.

Reference: guru99.com/smoke-testing.html

Q17. What is Sanity Testing?

Sanity testing is a quick check after small code changes to ensure the bug fix or minor feature works without testing the whole application.

Reference: guru99.com/sanity-testing.html

Q18. What is Usability Testing?

Usability testing evaluates how easy and user-friendly an application is. Real users perform tasks while observers note difficulties in navigation, design, or clarity.

Reference: nngroup.com/articles/usability-testing-101

Q19. What is Beta Testing?

Beta testing is performed by real users in a live environment before product release. It provides feedback on usability, performance, and unexpected bugs.

Reference: guru99.com/beta-testing.html

Q20. Difference between Regression Testing and Integration Testing?

- **Regression:** Ensures old functionality works after code changes.
- **Integration:** Validates correct interaction between combined modules.

Reference: guru99.com/regression-vs-integration-testing.html

Q21. What are Stubs and Drivers?

- **Stubs:** Dummy modules that simulate low-level components (used in top-down testing).
- **Drivers:** Dummy programs that simulate high-level modules (used in bottom-up testing).

Reference: geeksforgeeks.org/stubs-and-drivers-in-software-testing

Q22. What is Responsive Testing?

Responsive testing ensures the app/website works correctly on different devices, screen sizes, and orientations.

Reference: browserstack.com/guide/what-is-responsive-testing

Q23. What is Cross Browser Testing?

Verifying application functionality and UI consistency across multiple browsers (Chrome, Firefox, Edge, Safari).

Reference: browserstack.com/guide/what-is-cross-browser-testing

Q24. What is the difference between Monkey Testing and Gorilla Testing?

- **Monkey Testing:** Random, ad-hoc inputs to check system stability.
- **Gorilla Testing:** Repeatedly testing one module/function heavily to find defects.

Reference: geeksforgeeks.org/difference-between-monkey-testing-and-gorilla-testing

Q25. What is Testing with No Documentation?

This is Exploratory Testing, where testers explore the application, learn on the go, and design tests simultaneously.

Reference: guru99.com/exploratory-testing.html

Q26. Suppose a graphic designer just designed a login page and you have to test only the design. Which type of testing did you use?

For design-only checks, we use UI Testing or Visual Testing to verify layout, alignment, fonts, colors, and responsiveness match design specifications.

Reference: browserstack.com/guide/ui-testing

Q27. What is Confidential Testing?

Confidential testing is limited-scope testing done in a production/live environment with real data but minimal exposure, mainly to validate critical flows like payments.

Reference: guru99.com/types-of-software-testing.html

3. Test design techniques

How to choose test values and cover risk. **6 questions**

Q28. What is Boundary Value Analysis (BVA) and Equivalence Partitioning (EP)?

- **BVA:** Test values at boundaries (e.g., min/max).
- **EP:** Group inputs into valid/invalid partitions and test one from each group.

Reference: guru99.com/equivalence-partitioning-boundary-value-analysis.

Q29. Which technique do you follow for testing username and password?

Apply Boundary Value Analysis (min/max length), Equivalence Partitioning (valid/invalid ranges), and Negative Testing (empty, special chars).

Reference: geeksforgeeks.org/software-testing-boundary-value-analysis

Q30. Boundary Value Analysis with 1 lakh value but only 6 tested values—how will you ignore remaining?

By applying Equivalence Partitioning & Boundary Value Analysis, only representative test values are selected, covering ranges without testing every input.

Reference: guru99.com/equivalence-partitioning-boundary-value-analysis.html

Q31. How can you do risk analysis in regression testing?

Identify modules most affected by changes, analyze business impact if they fail, assign risk levels (high/medium/low), and prioritize regression tests accordingly.

Reference: softwaretestinghelp.com/regression-testing-tools

Q32. How will you make a document of regression testing and smoke testing when the 1st sprint is delivered with changes?

Prepare a test document including: Update execution sheet with results after retesting.

- **Regression:** Full/partial test cases from sprint 1 + changed areas.
- **Smoke:** High-level core functionalities from sprint 1 for quick health check.

Reference: softwaretestinghelp.com/regression-testing-tools

Q33. Suppose you have a second sprint delivered with some functionalities changing which testing technique you used?

Regression Testing is used to ensure existing functionality still works. Risk-based or impact-based testing may also apply.

4. Bugs, severity and triage

How you report, rank and defend a bug. **8 questions**

Q34. What is Severity and Priority (with examples)?

- **Severity:** Technical impact of defect (e.g., login crash = High severity).
- **Priority:** Business urgency to fix (e.g., typo in homepage = Low severity but High priority if in production).

Reference: geeksforgeeks.org/difference-between-severity-and-priority

Q35. What is Bug Triage?

Bug triage is the process of prioritizing, assigning, and reviewing reported defects based on severity, priority, and project timelines.

Reference: geeksforgeeks.org/what-is-bug-triage-in-software-testing

Q36. Footer Facebook link not working — what's the priority?

Since it's a non-critical external link, priority is Low (unless business requirement marks it as critical for user engagement).

Reference: guru99.com/severity-and-priority.html

Q37. How to identify if a bug is frontend or backend?

Reproduce bugs, check UI behavior, then verify API/network calls and DB logs. If UI displays wrong data → frontend; if API/DB fails → backend.

Reference: softwaretestinghelp.com/difference-between-frontend-and-backend

Q38. Suppose you found a bug which the developer said it's a feature, how did you convince the developer it's not a feature?

Collect evidence by showing requirement docs, acceptance criteria, or user stories to prove expected behavior. Compare actual vs expected results and highlight business/user impact to explain why it's a defect, not a feature.

Reference: guru99.com/defect-life-cycle-in-software-testing.html

Q39. After production a bug is found. Who is responsible (QA or Developer)?

Responsibility lies with both QA and Developer since QA missed detection and Developer introduced the defect. Ultimately, the team shares accountability.

Reference: softwaretestinghelp.com/software-tester-interview-questions

Q40. Suppose a customer complains about an application that has a bug so what will be your approach?

Steps: Acknowledge → Reproduce → Log bug with details → Communicate timeline → Fix & verify → Close.

Reference: guru99.com/defect-life-cycle.html

Q41. Suppose you have a login page one module and it has a bug so what will be your test closure activity?

Log and verify defect fix, update test cases/results, prepare closure report with executed tests, defects, and lessons learned.

Reference: geeksforgeeks.org/test-closure-in-software-testing

5. Real-time scenarios

The "what would you do" questions. **12 questions**

Q42. Suppose your application works on other browsers but not on Edge (10% users), what will be your priority?

Since 10% is significant, fixing Edge compatibility is High Priority. Cross-browser testing ensures equal performance across browsers.

Q43. Suppose your app crashes on Play Store with 11–12 user comments, how will you identify the crash phase?

Collect crash reports/logs (Firebase, Crashlytics), analyze stack trace, and reproduce issues with the same device/OS reported by users to locate phase.

Reference: firebase.google.com/docs/crashlytics

Q44. Suppose your application crashes. What techniques do you follow and whose fault is it?

Techniques: reproduce issues, analyze logs, stress/load testing, and debug. Fault usually lies with development (code issue), but QA ensures detection before release.

Reference: softwaretestinghelp.com/stress-testing

Q45. In the crash process which testing type we use to identify the bug?

Stress Testing and Load Testing are used to identify crash points by pushing the application beyond normal limits.

Reference: guru99.com/stress-testing.html

Q46. Suppose the food menu is shown on your side but not on the user's app, how will you reproduce?

Reproduce using the same user account, device, and browser/app version as the customer. Compare logs, network calls, and role-based access permissions.

Reference: browserstack.com/guide/how-to-reproduce-bugs

Q47. Suppose you have 3 large projects and 6 tasks, all same priority, all deadlines today. How do you manage?

Break tasks into smaller units, check dependencies, communicate with stakeholders, prioritize by risk/impact, and manage time across parallel efforts.

Reference: asana.com/resources/how-to-prioritize

Q48. Suppose a person does something unethical but doesn't violate company policies. What will you do?

Raise concern with manager/HR to review, as ethics affect team culture even if not a direct policy breach. Document the case neutrally.

Reference: shrm.org/resourcesandtools/tools-and-samples/toolkits/pages/managingethics.aspx

---

Q49. Test cases for social media "Delete Post" feature.

- **Positive:** Delete own post successfully, confirmation shown.
- **Negative:** Try deleting another user's post (should fail).
- **Edge:** Delete post with large media attached.

Reference: guru99.com/how-to-write-test-cases.html

Q50. Test cases for fund transfer module.

- **Positive:** Transfer valid amount between accounts.
- **Negative:** Transfer without balance.
- **Boundary:** Transfer minimum/maximum allowed.
- **Security:** Verify authorization & confirmation.

Reference: guru99.com/test-case-template.html

Q51. Which technique for testing location in an app?

Use GPS/location-based testing with simulators, mock locations, and field testing across devices.

Reference: browserstack.com/guide/location-testing-for-mobile-apps

Q52. Test cases for location in application.

- **Positive:** App fetches correct location via GPS.
- **Negative:** Location denied by user.
- **Edge:** Location with weak network.

Reference: guru99.com/how-to-write-test-cases.html

Q53. Demo website review from functional & non-functional perspective, with positive/negative cases.

- **Functional:** Verify login, navigation, forms, and links work correctly.
- **Non-Functional:** Check usability, performance, compatibility, and security.
- **Positive Case:** Valid login with correct credentials.
- **Negative Case:** Wrong password shows error message. Reviews: Improve load speed, mobile responsiveness, and accessibility.

Reference: guru99.com/functional-vs-non-functional-testing.html

6. API and backend

Requests, tokens, SQL and data checks. **9 questions**

Q54. What are Headers in API Testing?

Headers are key-value pairs in API requests carrying metadata (e.g., Content-Type, Authorization, Accept). They define how client and server exchange data.

Reference: postman.com/api-documentation-templates/api-headers

Q55. What is API Testing?

API testing validates request/response, functionality, reliability, and security of APIs. It ensures data is exchanged correctly between services without UI.

Reference: guru99.com/api-testing.html

Q56. What is Authorization and Authentication?

- **Authentication:** Verifying user identity (e.g., login with username/password).
- **Authorization:** Granting permissions after authentication (e.g., access roles).

Reference: auth0.com/intro-to-iam/what-is-authentication-authorization.

Q57. What is Token in API and how to generate it?

A token is a string used to authenticate API requests. Generated by logging in with valid credentials and receiving it from the server (often JWT or OAuth). Used in headers for subsequent requests.

Reference: jwt.io/introduction

Q58. What is API Chaining?

API chaining means using the output of one API call as the input for the next, creating a sequence of dependent requests.

Reference: postman.com/api-platform/api-chaining

Q59. What is Postman Cloud?

Postman Cloud lets teams store, share, and run API collections online. It supports collaboration, scheduled runs, and CI/CD integrations.

Q60. How can you test the backend (like a database) manually?

Use SQL queries to validate CRUD operations, data integrity, triggers, and stored procedures. Compare front-end actions with backend database updates.

Reference: softwaretestinghelp.com/database-testing

Q61. SQL Query – highest salary using joins.

To get the employee with the highest salary, join the two tables and compare each salary with the maximum.

```
SELECT e.name, s.salary
FROM employee e
JOIN salary s ON e.emp_id = s.emp_id
WHERE s.salary = (SELECT MAX(salary) FROM salary);
```

This also returns every employee who shares the top salary.

Reference: geeksforgeeks.org/sql-queries-for-practice

Q62. What is SQL Injection?

SQL Injection is a security vulnerability where attackers insert malicious SQL queries into input fields to access or modify database data.

Reference: owasp.org/www-community/attacks/SQL_Injection

7. Mobile, security and performance

Devices, builds, load and attacks. **9 questions**

Q63. Which type of devices are used during mobile application testing?

iPhones/iPads for iOS apps, Android phones/tablets for Android apps. Emulators (Android) and Simulators (iOS) are also used for virtual testing without physical devices.

Q64. What is Emulator & Simulator?

- **Emulator:** Mimics both hardware and software of a device, close to real performance (e.g., Android Emulator).
- **Simulator:** Mimics OS behavior only, good for UI/function testing (e.g., iOS Simulator).

Reference: guru99.com/real-device-vs-emulator-testing-ultimate-showdown.html

Q65. Which type of link is used in your build while uploading on Diawi?

Typically a signed build link (APK for Android, IPA for iOS) or direct download link is used for testers to install builds easily.

Q66. What are native and hybrid applications?

- **Native Apps:** Platform-specific, built with Swift (iOS) or Kotlin/Java (Android), offer high performance and full hardware access.
- **Hybrid Apps:** Built with web technologies (HTML, CSS, JS), run across platforms but slightly less performant.

Q67. How to do performance testing of mobile apps?

Use tools like JMeter/LoadRunner, simulate concurrent users, measure response time, CPU/memory, battery usage, and network latency.

Reference: guru99.com/performance-testing.html

Q68. What is a Thread Group in JMeter?

Thread Group defines number of virtual users, ramp-up time, and test duration to simulate load in performance testing.

Reference: jmeter.apache.org/usermanual/test_plan.html#thread_group

Q69. What is Stress and load Testing?

- **Load Testing:** Checks performance under expected workload.
- **Stress Testing:** Pushes system beyond normal limits to find crash points.

Reference: guru99.com/load-vs-stress-testing.html

Q70. What is Penetration Testing?

Pen testing simulates real-world attacks to identify vulnerabilities in systems, networks, or applications.

Reference: owasp.org/www-community/penetration-testing

Q71. How to verify that when we need security testing? How do we overcome security issues?

Security testing is needed when applications handle sensitive data (like payments or personal info). To overcome issues, apply penetration testing, encryption, secure coding, and strong authentication. For example: validating payment data transmission and preventing XSS/SQL injection.

Reference: owasp.org/www-project-web-security-testing-guide

8. Agile, tools and HR questions

Teams, process and the talking part. **13 questions**

Q72. What is Agile?

Agile is an iterative development approach emphasizing collaboration, flexibility, and incremental delivery through sprints.

Reference: atlassian.com/agile

Q73. What is Product Backlog?

Product backlog is an ordered list of features, enhancements, and bugs maintained by the product owner for agile development.

Reference: scrum.org/resources/what-is-a-product-backlog

Q74. What are User Stories?

A user story is a short requirement from an end-user perspective, e.g., "As a user, I want to reset my password so that I can regain account access."

Reference: atlassian.com/agile/project-management/user-stories

Q75. What is a Sprint Retrospective Meeting?

Held after each sprint, the team discusses what went well, what needs improvement, and action items for the next sprint.

Reference: atlassian.com/agile/scrum/retrospectives

Q76. Do you have any idea about GitHub?

GitHub is a cloud-based platform for version control using Git. It allows code hosting, collaboration, pull requests, and issue tracking.

Reference: docs.github.com/en/get-started

Q77. What is Implicit and Explicit in Selenium?

- **Implicit Wait:** Global wait for element to load before throwing error.
- **Explicit Wait:** Conditional wait applied to specific elements until condition is met.

Reference: geeksforgeeks.org/waits-in-selenium

Q78. What is multilevel and multiple inheritance?

- **Multilevel Inheritance:** Class inherits from another, then further extended (A → B → C).
- **Multiple Inheritance:** One class inherits from two or more base classes (A & B → C).

Reference: geeksforgeeks.org/inheritance-in-java

Q79. What is Heap and Graph in Data Structure?

- **Heap:** A tree-based structure satisfying heap property (min/max).
- **Graph:** A set of nodes connected by edges, used to represent networks.

Reference: geeksforgeeks.org/fundamentals-of-algorithms

Q80. Which type of payment gateways are you currently using?

Common gateways include PayPal, Stripe, Braintree, Square, Authorize.Net, plus Apple Pay/Google Pay for mobile.

Reference: razorpay.com/learn/payment-gateways

Q81. What type of technology stacks currently your company is working on?

Example answer: “We’re working on MERN stack (MongoDB, Express, React, Node.js) for web apps and Flutter/Kotlin/Swift for mobile apps, with AWS/ Azure for cloud hosting.” (Answer varies by company).

Reference: geeksforgeeks.org/mern-stack

Q82. Any complex task for you during your testing?

Example: Testing a payment flow with multiple gateways and currencies, where edge cases like failed transactions or refunds required deep regression testing and coordination with devs.

Reference: softwaretestinghelp.com/software-tester-interview-questions

Q83. Which type of challenges do you face daily when testing websites in your company?

Common challenges: handling frequent requirement changes, cross-browser issues, test data setup, reproducing intermittent bugs, and tight deadlines.

Reference: lambdatest.com/blog/software-testing-challenges

Q84. Any difficult scenario you faced during testing?

Example: Reproducing intermittent bugs like session timeout or inconsistent payment failures that occur only under specific load or network conditions.

Reference: softwaretestinghelp.com/software-tester-interview-questions

About SQACircle: Connect With Us

SQACIRCLE

SQACircle is a growing community for QA and Software Testing professionals. Through our **WhatsApp group** and **LinkedIn page**, we share **remote and onsite QA job opportunities**, learning resources, interview preparation material, and practical guides like this one.

Our motto: Let's Connect on Trust.

CONNECT WITH SQACIRCLE

Follow our LinkedIn page for QA and AI testing content, learning resources, and regular remote and onsite QA job posts:

SQACircle on LinkedIn
https://lnkd.in/d-9bTHvf

Created by Iqra Ilyas
linkedin.com/in/iqra-ilyas-2aa665231 · iqrailyas.com

If this handbook helped you, please follow the SQACircle page, share it with your network, and tag us. It helps our community grow and reach more QA engineers.

CREDIT & USAGE

Credit & usage: This handbook was prepared by **Iqra Ilyas** for the SQACircle community. You are welcome to read, download, and share it freely in its original form. Please do **not** republish it as your own work or remove the SQACircle branding. When sharing, kindly credit **Iqra Ilyas** and **SQACircle**. Reference links go to other websites that belong to their owners.

© 2026 Iqra Ilyas / SQACircle. QA Interview Handbook, prepared for the QA community. | Let's Connect on Trust.