QUIZICALLY

Compatibility test case

Version 1.0
11/07/2025

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Team 10	11/06/2025	Khadija	11/07/2025	Initial Compatibitliy Test

UP Template Version: 12/31/07

TABLE OF CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF THE COMPATIBILITY TEST CASE DOCUMENT

This document defines the compatibility testing required to ensure Quizically functions consistently across supported mobile devices, operating systems, screen sizes, and network types. It verifies that core game functionality login, create game, join game, gameplay interactions, and results rendering, operates correctly on both iOS and Android platforms, in alignment with Quizically's performance and usability requirements.

2 TEST CASE SPECIFICATION

The compatibility test validates Quizically's behavior across multiple devices and OS versions. It ensures consistent UI layout, feature parity, gameplay responsiveness, and backend synchronization for cross-platform, cross-device users.

2.1 DESCRIPTION

This test case verifies that Quizically:

- Installs and runs on iOS and Android devices, including older models and tablets.
- Maintains identical gameplay and UI rendering across device types.
- Supports OAuth 2.0 login uniformly across devices.
- Processes game actions (joins, submissions, scoring) consistently across all devices.
- Syncs with the backend in real time across different network types (Wi-Fi, LTE, 5G).
- Scales UI appropriately for different screen sizes, pixel densities, light/dark modes.

Participants:

- **QA Lead** → Executes compatibility tests, documents findings
- **Frontend Engineer** → Debugs UI rendering or device-specific issues
- **Backend Engineer** → Verifies API and data synchronization
- **DevOPS Support**→ Ensures stable testing environment

2.2 RESOURCES

Role	Responsibility
QA Lead	Executes compatibility test cases across all devices, records results, and identifies UI/OS issues.
Frontend Engineer	Reviews UI inconsistencies across platforms, fixes

	rendering/layout issues, and verifies scaling on different screen sizes.
Team Lead (Khadija)	Oversees entire compatibility testing process, reviews documentation for completeness, and approves final results.
DevOps Engineer	Ensures backend environment stability, monitors network variability tests (Wi-Fi/ LTE/ 5G), and captures server/device logs.
Backend Engineer	Monitors API responses during cross-device gameplay, ensures consistent behavior across iOS/ Android, and checks data synchronization.

2.3 PRECONDITIONS

1. Latest staging build deployed to TestFlight& Android internal testing.
2. Backend services and databases operational.
3. Test accounts for host and players are active.
4. Devices have stable internet or configurable network switching.

2.4 POST CONDITIONS

1. All UI, OS, and device compatibility issues logged and documented.
2. Verified records of gameplay parity across devices.
3. Confirmed ability to join and play cross platform games.
4. Defects assigned to engineering teams for resolution.

2.5 FLOW OF EVENTS

Normal Flow:

Steps	**Description**	**Expected Results**
1	Install Quizically on iOS device (latest OS).	App installs, opens, and loads successfully.
2	Install Quizically on Android device (latest OS).	App installs, opens, and loads successfully.
3	Log in using OAuth on both devices.	Login succeeds on iOS and Android with consistent flow.
4	Host an iOS creates a new game session.	Game session code is generated and visible.
5	Player on Android joins using session code/QR.	Player enters host’s lobby with no errors.

6	Host starts the game.	First game prompt/action appears simultaneously on both devices.
7	Submit game action from Android.	Response accepted, backend logs action, host sees update.
8	Submit game action from iOS.	Android receives updated leaderboard in ≤ 2 seconds.
9	Compare UI components on both devices (buttons, font size, spacing).	UI elements properly rendered, no distortions or clipped text.
10	Switch Android device to dark mode.	Colors remain readable; UI adapts correctly.
11	Switch iOS device to dark mode	UI remains consistent with platform guideline.
12	Test app on older iOS hardware.	App responds smoothly, no layout breakage.
13	Test app on older Android hardware.	Gameplay remains functional, responses delivered within acceptable limits.
14	Test app on tablets.	Layout scales properly, no oversized/stretch UI.

Alternate Flow A-Specific UI Rendering Issue

Step	**Description**	**Expected Results**
1	UI element fails on a specific OS version (e.g., Android button overlaps).	Issue logged with screenshots.
2	Compare rendering to platform guidelines.	Confirms deviation from expected UI behavior.
3	Mark as compatibility	Assigned to frontend

	defect.	developer.

Alternate Flow B- Device Performance limitations

Step	Description	Expected Result
1	Test game on older/low-memory device.	Lag or slow response identified.
2	Record performance degradation logs.	Issue filed for optimization.
3	Retry gameplay after clearing device cache.	Device performance improves or stabilizes.

2.6 INCLUSION/EXCLUSION POINTS

Included

- FT-LOGIN-001 → OAuth Login
- FT-GAMECREATE-002 → Create Game Session
- FT-JOIN-003 → Player Join Flow
- UT-API-004 → Prompt/Action Delivery
- UT-PLAY-005 → Player Action Submission
- RT-LB-004 → Leaderboard Synchronization
- UT-UI-007 → UI Rendering Across Platforms
- LT-REJOIN-001 → Network Rejoin Behavior

Excluded

- AI question generation engine
- Game set creation/editing workflows
- Social media marketing & event posting
- Analytics/event store behavior

2.7 SPECIAL REQUIREMENTS

- Multiple devices required
- Must test Wi-Fi, LTE, 5G scenarios
- Screen recording required for UI consistency validation
- All tests must be run on both light mode and dark mode

Appendix A: References

The following table summarizes the documents referenced in this document.

Document Name and Version	Description	Location
Quizically Requirements Definition v1.0	Defines functional, non-functional and business requirements used to determine and validate the compatibility testing criteria.	Requirements Document

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

Term	Definition
Compatibility Testing	Verifies system performance across multiple OS, devices, screen sizes, and network types.
Game Session	Hosted game instance players join.
OAuth 2.0	Authentication protocol for social/email login.
UI Rendering	How interface components display across devices.