



QUIZICALLY

FUNCTIONAL TESTING TEST CASE

Version 1.0
11/07/2025

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Palmer Riley	11/07/2025	Khadija	11/07/2025	Test Case draft
2.0	Rana Abdellatif	11/21/2025	Khadija	11/21/2025	PM Approval - Revision

UP Template Version: 12/31/07

TABLE OF CONTENTS

1 INTRODUCTION	5
1.1 Purpose of The Test Case Document	5
2 TEST CASE SPECIFICATION	5
2.1 Description	5
2.2 Resources	5
2.3 Preconditions	5
2.4 Post Conditions	5
2.5 Flow of Events	5
2.6 Inclusion/Exclusion Points	5
2.7 Special Requirements	5
APPENDIX A: REFERENCES	6

1 INTRODUCTION

1.1 PURPOSE OF THE FUNCTIONAL TESTING TEST CASE DOCUMENT

The purpose of this document is to outline the Functional Testing Test Case for the Quizically application. Functional testing verifies that each feature of the Quizically system performs in accordance with the requirements defined in the Test Plan and Requirements Documentation. This test case ensures that all functional components, such as user authentication, quiz creation, question management, scoring, and leaderboard display, operate correctly and deliver the expected outcomes.

2 TEST CASE SPECIFICATION

This test ensures that all login inputs behave correctly and validate user data as designed. It confirms that users can only log in with valid credentials and that incorrect inputs show clear error messages.

2.1 DESCRIPTION

The functional test validates that the system behaves as expected when performing its intended operation. This test case checks that:

- The email field accepts valid email formats only
- The password field enforces minimum and maximum character limits
- Proper error messages appear when fields are blank or invalid
- Successful login redirects to the quiz dashboard

Participants Include:

- Tester(Palmer): Executes and documents test steps
- Team Lead(Khadija): Reviews results and signs off
- Backend Engineer: Monitors logs if authentication errors occur

2.2 RESOURCES

Role	Responsibility
Functional Tester (Palmer)	Runs test cases and records pass/fail results
Backend Engineer	Monitors system and confirms login behavior
Team Lead (Khadija)	Reviews and signs off on final results

2.3 PRECONDITIONS

1. Test environment and login page are accessible
2. Valid and invalid test user accounts exist
3. Internet connection is stable

2.4 POST CONDITIONS

1. Login validation results recorded
2. Any defects are logged for review

3. System returns to a stable state

2.5 FLOW OF EVENTS

Requirement Key:

UI + Authentication Requirements

- BR-017 — System shall provide intuitive, user-friendly UI
- UR-002 — System shall allow social login (OAuth, email login)
- FS-001 — System shall enforce authentication and identity validation
- S-001 — All communication shall be secure (affects login safety)

Player / Host Operation Requirements

- FIR-002 — Player dashboard supports login, join game/team, view score
- FIR-001 — Host dashboard supports creating and managing gameplay
- BR-015 — Host shall be able to create trivia games
- BR-020 — Host can search and filter trivia games (applies to finding created quiz)
- UR-001 — System stores and retrieves game history (score display)
- HR-004 — Trivia questions and leaderboard must be visible to users

Step	Action	Expected Result	Requirement ID
1	Launch the quizically app	App opens successfully	BR-017
2	Navigate to login and enter valid credentials	User is logged in and redirected to dashboard	UR-002, FS-001, FIR-002
3	Select “create quiz” and enter valid quiz info	Quiz is created and listed under user’s quizzes	BR-015, FIR-001
4	Add questions and correct answer to quiz	Questions are saved successfully and linked to the quiz	FIR-001, HR-004
5	Log in as a different user and attempt the created quiz	Quiz loads properly; user can select answers and submit	FIR-002, HR-004
6	Submit the quiz	System calculates the score and displays results accurately	UR-001, FIR-002
7	Navigate to leaderboard	Leaderboard shows updated results	HR-004
8	Log out from the system	User is safely logged out and redirected to the login screen	FS-001, UR-002

Validation Steps:

Step	Action	Expected Result	Requirement IDs
V1	Leave both login fields blank and tap “Login”	System shows required-field error messages	BR-017, FS-001
V2	Enter invalid email format	“Invalid email format” error appears	BR-017, UR-002
V3	Enter password below min length	“Password too short” error appears	FS-001
V4	Enter password above max length	“Password too long” error appears	FS-001
V5	Enter incorrect login credentials	“Incorrect login credentials” message displayed	FS-001, UR-002

Alternate Flows:

AF1 - Incorrect Login Credentials

- User enters wrong password
- System displays “Incorrect login credentials”
- Requirements: FS-001, UR-002

AF2 - Invalid Field Formatting

- Email missing “@” or domain
- UI blocks attempt and shows error
- Requirements: BR-017, UR-002

AF3 - Password Length Violation

- Too short or too long password
- UI prevents submission and displays limits
- Requirements: FS-001

2.6 INCLUSION/EXCLUSION POINTS

N/A - This test case is standalone and does not include or depend on any other test cases.

2.7 SPECIAL REQUIREMENTS

- Testing should be performed in a controlled staging environment
- Use consistent test data across all executions to ensure repeatability
- Document screenshots and system logs for each test run
- Any defect discovered must include reproduction steps and environment details

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 requirements for quiz access and scoring	Requirements Document
Quizically Requirements Traceability Matrix v1.0	Requirement IDs for mapping	Traceability Matrix