Workshop No. 1 Requirements, User Stories, and Story Mapping

Trivia

Ballesteros Ballesteros David Mateo Bejarano Ariza Santiago

Suaza Montalvo Santiago Gordillo Medina Juan Sebastian

November 5, 2025

1. Project Overview

The team's project consists of building a real-time trivia game similar to Kahoot! in which a host (e.g., a teacher or event organizer) creates a trivia session and players join using a game code. After joining, players answer general-knowledge questions displayed on their devices, competing for points based on accuracy and response time. A free, community-driven question bank API such as Open Trivia Database can provide questions – the API offers a JSON interface with no API key required, allowing developers to retrieve trivia questions from multiple categories. This report documents the requirements, user stories and planning artifacts necessary to begin the software engineering process.

2. General System Description

The proposed system is a **real-time multiplayer trivia game platform** designed to deliver engaging and interactive question-based competitions for players connected simultaneously. The system allows a **host** (such as a teacher, presenter, or event organizer) to create a trivia session by selecting a set of questions from an external trivia API and sharing a unique game code with participants. **Players** join the session using this code and compete by answering multiple-choice questions within a time limit, earning points based on speed and accuracy.

The platform provides a dynamic and responsive interface that updates in real time through WebSocket or similar technologies, ensuring synchronized question delivery and score updates across all connected devices. A **scoreboard** is displayed after each question, allowing users to track their performance and ranking. Additionally, the system incorporates administrative tools to manage users, control content categories, and monitor session activity.

This system is intended for educational, recreational, and organizational environments

where gamified learning or entertainment is desired. It integrates modern web technologies and scalable backend infrastructure to support concurrent users, data security, and seamless real-time interaction.

3. Project Objectives

The main objective of the project is to **design and implement a real-time trivia game platform** that promotes interaction, learning, and entertainment through gamified question-based activities.

The specific objectives are as follows:

- **To develop** a responsive web-based application that allows players to join trivia sessions, answer questions, and receive real-time feedback.
- **To implement** a host control module capable of creating, managing, and supervising trivia sessions dynamically.
- **To integrate** an external trivia API that provides a wide and diverse question bank, ensuring variety and content quality.
- **To provide** an intuitive user interface that enhances engagement and simplifies participation for all types of users.
- **To ensure** the platform supports real-time communication and synchronization using modern technologies such as WebSocket.
- **To establish** a robust architecture capable of handling multiple concurrent users with high availability and performance.
- **To include** administrative functions for user management, question moderation, and monitoring of active sessions.
- **To encourage** collaboration, competitiveness, and active participation in educational or recreational contexts.

4. Stakeholders and Roles

Stakeholders

The following stakeholders are individuals or entities who have a direct or indirect interest in the development, deployment, and operation of the real-time trivia game system.

- **Players:** End users who participate in trivia sessions and interact with the system in real time.
- **Hosts (Teachers or Event Organizers):** Users who create and manage game sessions, moderate players, and control game flow.
- **Administrators:** Responsible for maintaining the system, managing content (question categories), and supervising user activity.
- **Development Team:** Software engineers, designers, and testers responsible for building, deploying, and maintaining the system.

- **System Architect / Project Manager:** Oversees the design, scheduling, and implementation process, ensuring that requirements are met on time.
- **API Provider:** External trivia database or third-party API (e.g., Open Trivia DB) that supplies questions and categories.
- **Educational Institutions / Organizations:** Entities interested in using the system as an educational or entertainment tool to promote engagement and learning.
- **Investors or Sponsors:** Stakeholders who fund or support the platform's development and deployment.
- **Technical Support Team:** Provides maintenance, updates, and assistance to users after deployment.
- **End Users' Community:** The wider group of users who contribute feedback and suggestions for improving gameplay and user experience.

Roles

The following roles define the primary interactions that different types of users have within the system.

- **Player:** Joins existing game sessions, answers trivia questions, and competes with other users in real time.
- **Host:** Creates trivia sessions, selects categories and difficulty levels, starts and controls the flow of the game, and monitors player performance.
- **Administrator:** Manages question categories, oversees user accounts, monitors active sessions, and enforces system policies.
- **Developer:** Designs, implements, and tests software components to ensure functionality, performance, and scalability.
- **System Architect:** Defines system architecture, communication protocols, and integration with external services (e.g., APIs).
- **End User (Guest):** Accesses the platform without registration, possibly as a spectator or temporary participant.
- **Technical Support Agent:** Provides assistance to users, resolves issues, and maintains service uptime.

5. Functional Requirements

The functional requirements describe what the system must do and the main features that enable the trivia game to operate effectively.

- **FR1: User registration and authentication.** The system shall allow users to register, log in, and manage their accounts securely (players, hosts, and administrators).
- **FR2: Join a trivia game.** The system shall allow players to join a game session using a unique game code and nickname.

- **FR3: Create and configure a game session.** Hosts shall be able to create new trivia sessions by selecting question categories, difficulty levels, and number of questions.
- **FR4: Retrieve questions from external API.** The server shall retrieve questions dynamically from an external trivia API to ensure variety and freshness of content.
- **FR5: Display questions and collect answers.** The system shall show each question to players with multiple-choice answers and collect their responses within a defined time limit.
- **FR6: Scoring and feedback.** The system shall automatically evaluate answers, calculate points based on correctness and response time, and display feedback to players.
- **FR7: Real-time updates.** The system shall provide real-time synchronization of questions, timers, and scoreboards between the server and all connected clients.
- **FR8: Scoreboard management.** After each question, the system shall update and broadcast the scoreboard, showing rankings and total scores for all players.
- **FR9: Session control by host.** Hosts shall be able to start, pause, resume, or end a trivia session at any time through a control panel.
- **FR10: Player management.** Hosts and administrators shall be able to remove players who violate rules or disrupt the session.
- **FR11: Question category management.** Administrators shall manage question categories and optionally add, modify, or delete custom questions in the database.
- **FR12: Session and user monitoring.** Administrators shall monitor active sessions, number of connected users, and system performance metrics in real time.
- **FR13: Export results.** Hosts shall be able to export final results (e.g., in CSV or PDF format) for analysis or record-keeping.
- **FR14: Password recovery.** The system shall allow users to reset their password through a secure email link when forgotten.
- **FR15: Role management.** Administrators shall assign or update user roles (player, host, admin) to define permissions and access levels.

Non-Functional Requirements

The non-functional requirements define how the system should behave and the quality standards it must meet.

- **NFR1: Performance.** The system should handle real-time updates with response times under two seconds for standard operations.
- **NFR2: Scalability.** The architecture should support hundreds of concurrent players and be capable of horizontal scaling.

- **NFR3: Reliability and availability.** The server must be available at least 99.9% of the time. Failover mechanisms shall ensure that sessions are not lost in case of a crash.
- **NFR4: Security.** All user data and credentials must be encrypted. Sensitive operations such as authentication and password reset shall use HTTPS and token-based security.
- **NFR5: Usability.** The user interface shall be intuitive and accessible, allowing new players to join and play without prior training.
- **NFR6: Maintainability.** The system shall follow modular design and coding standards to facilitate updates and bug fixing.
- **NFR7: Portability.** The client-side application must work across modern web browsers and be adaptable to mobile devices.
- **NFR8: Compatibility.** The game shall support communication via standard web protocols (HTTP and WebSocket) to integrate with common API services.
- **NFR9: Data integrity.** All gameplay data, scores, and user information must be stored consistently in the database, ensuring no loss or duplication.
- **NFR10: Localization and accessibility.** The system should support multilingual interfaces and accessibility features for users with visual or motor impairments.

6. Historias de Usuario

Se presentan dos historias de usuario con la plantilla solicitada.

User Story 1

Title: Join a trivia game

Priority (1–25): 25

Time estimation: 4 hours

As a player

I need to join a trivia game using a game code

So that I can participate with my friends.

Acceptance criteria:

Given I have a valid game code,

When I enter my nickname and the code,

Then the system should add me to the lobby and display a waiting screen.

User Story 2

Title: Answer trivia questions

Priority (1–25): 25

Time estimation: 6 hours

As a player

I need to answer trivia questions within a time limit

So that I can earn points based on my performance.

Acceptance criteria:

Given a question is displayed,

When I select an answer before the timer expires,

Then the system should record my response and show feedback indicating whether it is correct or incorrect.

User Story 3

Title: View the scoreboard

Priority (1–25): 22

Time estimation: 3 hours

As a player

I need to see the scoreboard after each question

So that I can track my position compared to other players.

Acceptance criteria:

Given all players have answered the question,

When the round ends,

Then the system should display the updated scoreboard with rankings and total points.

User Story 4

Title: Create a trivia session

Priority (1–25): 24

Time estimation: 5 hours

As a host

I need to create a trivia session and choose question categories and difficulty

So that the game matches the audience preferences.

Acceptance criteria:

Given I am authenticated as a host,

When I select a category, number of questions, and difficulty,

Then the system should create a new trivia session with a unique game code.

User Story 5

Title: Start and control the game

Priority (1–25): 20

Time estimation: 4 hours

As a host

I need to start, pause, or resume the game

So that I can ensure all players are ready and manage the flow of the session.

Acceptance criteria:

Given players have joined the game,

When I click "Start Game,"

Then the first question should be displayed with a countdown timer visible to all players.

User Story 6

Title: Monitor player participation

Priority (1–25): 17

Time estimation: 4 hours

As a host

I need to view player participation statistics during the game

So that I can ensure fair play and manage inactive players.

Acceptance criteria:

Given a game is in progress,

When I open the host dashboard,

Then I should see the number of connected players, their status (active/inactive), and response times.

User Story 7

Title: Manage player behavior

Priority (1–25): 16

Time estimation: 3 hours

As a host

I need to remove players who violate rules or disrupt the game

So that the session remains fair and enjoyable for others.

Acceptance criteria:

Given I am the session host,

When I select a player from the list and click "Kick,"

Then the system should immediately remove the player from the session and update the player list.

User Story 8

Title: View final results

Priority (1–25): 21

Time estimation: 3 hours

As a player

I need to see the final results after all questions are completed

So that I can know my final ranking and performance.

Acceptance criteria:

Given all questions in the session have been answered,

When the host ends the game,

Then the system should display the final scoreboard showing ranks, points, and correct answers per player.

User Story 9

Title: Manage question categories and custom content

Priority (1–25): 19

Time estimation: 5 hours

As an administrator

I need to manage question categories and add or remove custom questions

So that the game content remains accurate and appropriate.

Acceptance criteria:

Given I am logged in as an administrator,

When I access the content management panel,

Then I should be able to add new categories, modify questions, or delete outdated ones.

User Story 10

Title: Monitor active sessions and users

Priority (1–25): 18

Time estimation: 4 hours

As an administrator

I need to monitor active sessions and users in real time

So that I can enforce platform policies and detect irregular activity.

Acceptance criteria:

Given the game platform is running multiple sessions,

When I access the admin dashboard,

Then I should see all active sessions with their game codes, number of players, and host information.

User Story 11

Title: Register and log in

Priority (1–25): 23

Time estimation: 4 hours

As a user

I need to register and log in to the platform

So that my profile, scores, and achievements can be saved.

Acceptance criteria:

Given I am on the registration page,

When I provide valid credentials,

Then the system should create my account and redirect me to the home screen.

User Story 12

Title: Reset password

Priority (1–25): 15

Time estimation: 3 hours

As a user

I need to reset my password when I forget it

So that I can regain access to my account.

Acceptance criteria:

Given I am on the login page,

When I click "Forgot Password" and enter my email,

Then the system should send a reset link to my email address.

User Story 13

Title: Manage user roles

Priority (1–25): 14

Time estimation: 4 hours

As an administrator

I need to assign roles (player, host, admin) to users

So that they have access only to the functions allowed by their role.

Acceptance criteria:

Given I am in the admin panel,

When I select a user and change their role,

Then the system should update their permissions immediately.

User Story 14

Title: Retrieve questions from API

Priority (1–25): 22

Time estimation: 4 hours

As a system

I need to retrieve trivia questions from an external API

So that the game always has a dynamic and varied set of questions.

Acceptance criteria:

Given the system is connected to the trivia API,

When the game starts,

Then the server should request questions in JSON format and store them temporarily for the session.

User Story 15

Title: Export game results

Priority (1–25): 10

Time estimation: 3 hours

As a host

I need to export the game results after a session ends

So that I can analyze performance or share it with players.

Acceptance criteria:

Given the game has ended,

When I click "Export Results,"

Then the system should generate a downloadable report (CSV or PDF) with player names, points, and accuracy.

User Story 16

Title: Real-time game updates

Priority (1–25): 25

Time estimation: 6 hours

As a player

I need to receive real-time updates during the game

So that I can see new questions, countdowns, and scores instantly.

Acceptance criteria:

Given I am connected to an ongoing game,

When the host triggers a new question or scoreboard update,

Then the system should immediately push the update to all connected clients via WebSocket or similar technology.