**System Requirements**

**P01**

**<team member names & ids>**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **24100175** | **ABDUL MUIZ** |
| **24100127** | **ABDUR RAFAE HAROON** |
| **24100277** | **BISMA NAWAZ** |
| **24100308** | **HAFSA AHMED** |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction | 5 | 5 |
| System actors | 10 | 8 |
| Functional Requirements | 35 | 30 |
| Non-functional requirements | 10 | 9 |
| Security requirements | 10 | 10 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template | 5 | 5 |
| Github folder structure | 15 | 0 |
| Late submission penalty | -20 |  |
| **Grand Total** | **100** | **77** |

**Table of Contents**

[1.](#_gjdgxs) Introduction 3

[2.](#_30j0zll) System Actors 4

[3.](#_1fob9te) Functional Requirements 5

[4.](#_3znysh7) Non-functional Requirements / Quality Attributes 6

[5.](#_tyjcwt) Security Requirements 7

[6.](#_3dy6vkm) Who Did What? 8

[7.](#_1t3h5sf) Review checklist 8

# Introduction

The proposed project is a user-hosted trivia web application. This application aims to create an interactive quiz-based learning experience, taking its inspiration from the popular game-based learning platform, ‘Kahoot!’. Designed for students, teachers, trainers or simply anyone who wants to challenge their friends to a quiz, the extent of the web application’s features allow anyone to quickly create and host a quiz-based session and let others join in and participate. Thus, within a session, there will exist a host user who creates the trivia while the participants joining voluntarily act as players. The players will use the quiz ID given by the host to join a particular session.

# System Actors

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Host | A user with Host status who can create quizzes and host games for other Players to join. Hosts can comprise teachers, students or any other experts looking to test students on a specific subject/topic. |
| Player | A user who can join in any game/quiz made by a Host user. It can include students, teachers or anyone looking to participate in interactive learning activities. |
| Developer | A developer can be either a Player or a Host in developer mode. The developer has access to the advanced tools for the sake of debugging/ development. |
| Server | A server is the central point of communication between hosts and players, by sending their IDs to one another. It is responsible for establishing the connection between hosts and players, and ensuring the synchronization of the game state among all the connected players. [Why do you think that this server is an actor? Is it external to your system and accesses services from the system that you are going to develop?] |
| Database Manager [Admin?] | Overlooks all the users registered from the app. |

# Functional Requirements

|  |  |
| --- | --- |
| **Requirements** | |
| **Sr#** | **Requirement** |
|  | A Host should be given a unique room ID whenever they create room for the quiz. |
|  | A Host should be able to share the room ID online across social media platforms so the players can use it to access/join a quiz. |
|  | A Host should have editing access to all of their own trivia questions. |
|  | A Host should be able to create a quiz. |
|  | A Host should have the option to upload a quiz in CSV format from their locally stored library to initiate a session. |
|  | A Host should be able to start a new quiz/game session with the set parameters. |
|  | A Host should be able to set the total duration for a quiz prior to starting it. |
|  | A Host should be able to define the maximum number of participants/ players for a quiz/ game. |
|  | A Host should be able to define the minimum number of participants/ players to start a quiz/game. |
|  | A Host should be able to monitor when enough participants have joined the quiz/game so they can start the quiz. |
|  | A Host should be able to choose whether they want to allow players to skip questions or not. |
|  | A Host should be able to pause, resume or end quiz sessions as per their discretion. |
|  | A Host should be able to export quizzes to their device in excel. |
|  | A Host should be able to deny certain players from joining the quiz. |
|  | A Host should be able to see the names and IDs of all the players who have joined the quiz. |
|  | A Host should be able to remove certain players that have joined the quiz at any time. |
|  | A Host should be able to set the time limit for each question in the quiz (time limit for each question can vary). |
|  | A Host should be able to specify the date and time at which the quiz would be available to players. |
|  | A Host should be able to start/broadcast a quiz any time they want. |
|  | A Host should be able to view the real-time statistics of participants after each question including the number of players still present, their scores and rankings. |
|  | A Host should have the ability to download the quiz results in excel format (the marks will be displayed with the original name of the players so that it is easier for the host to keep track of the records). |
|  | A Host should be able to specify whether the answer for a question should be displayed once its timer ends or to display them with total scores at the end. |
|  | A Host should be able to display final scores and rankings (or winners) at the end of each quiz to all the players. |
|  | A Host should be able to declare winner based on the least amount of time taken to complete the quiz if there is a draw between two players. (if that is identical as well then it will be a draw). |
|  | A Host should be able to access previous quizzes and their results stored locally. |
|  | A Player should be able to join a quiz by entering the unique ID shared by the host. |
|  | A Player should be able to answer quiz questions broadcasted by the host within the set time limit. |
|  | A Player should have the ability to change their names which will be displayed to all other players and the Host. |
|  | A Player should have the option to undo his/her response to a particular question within the timed limit. |
|  | A Player should be able to see a timer, displaying the time left for answering a particular question as well as the time left for the entire quiz. |
|  | A Player should be able to see the serial number for the current question and the total number of questions in the quiz. |
|  | A Player should be directed to a waiting lobby if the minimum number of participants for a quiz has not been reached. |
|  | A player should be able to see the marks of the past quizzes he has taken (questions should not be accessible/editable by players). |
|  | A player should have the option to leave the quiz anytime. |
|  | The player should first register himself with his original name (can choose to use nicknames when joining a quiz session). |
|  | A player should not have the option to go back to the previous question once the timer is done for that specific question (i.e. player cannot undo the answer once the timer is over for the specific question). |
|  | Once the timer is done for a specific question, the player should be able to see the correct answer for the question (only if the host enables that option while making the quiz). |
|  | A Player should be able to receive immediate feedback after completing a quiz, including their relative standing, time taken on each question, and score. |
|  | A Developer should be able to see detailed outgoing and incoming response messages in the console as a Player/Host. |
|  | A Developer should be able to toggle different types of logs that can be seen in the console |
|  | A Developer should be able to send any type of POST/GET request to the server |
|  | A Developer as a Host should be able to freeze the timer during a quiz question |
|  | A Developer as Host/Player should be able to receive logs that inform details about the users joining the session |
|  | A Server should assign unique IDs to each player wanting to join a session.  [Server does not appear to be an actor here and also not in the requirements mentioned below. What you have mentioned here are some features supported by the server which gets executed whenever an actor such as a host needs them.] |
|  | A Server should allow players to join a quiz session by sending their IDs to the quiz host. |
|  | A Server should support simultaneous quiz sessions and handle multiple hosts and their respective players concurrently. |
|  | A Server should have the option to limit the number of players per session as per the cap set by the quiz host. |
|  | A Database Manager should have access to all the users who have signed up |
|  | A Database Manager should have the ability to add or delete any record of any user |
|  | A Database Manager should have the ability to make and give developer access accounts. |

# 

# [ I suggest that you build mechanisms to store the historical data for analytics and reporting.]Non-functional Requirements / Quality Attributes

| **Sr#** | **Requirements** |
| --- | --- |
| 1 | A Player's response to a question should not take more than 2 seconds to be visible to the Host. |
| 2 | It should not take more than 10 seconds for a Player to join a quiz hosted by a particular host. |
| 3 | The latency should be below 500 ms for question loading and scoring. |
| 4 | The application must support upto 10 simultaneous groups on its initial build. |
| 5 | The web application should work on the latest versions of chrome and safari. |
| 6 | The service should be available 90% of the time (0.10 \* 8760 = 876 hours of allowable downtime per year). |
| 7 | The data (quiz questions for host and marks/history for the players) should be retained for at least 30 days on each users’ devices. |
| 8 | The web application should automatically resize to fit onto any small, medium, large screens (i.e. phones, tablets, or laptops) |
| 9 | The application should be scalable such that it should be able to support at least 100 users at the same time (this includes all the hosts and players) without significant degradation of performance. [Isn’t this number (100 users) too small for a system that intends to support multiple quizzes simultaneously?] |
| 10 | It should not take more than 5 seconds for the landing page of the website to be visible to a user. |
| 11 | Server should send out responses to Hosts and Players every 1-2 seconds when the question is being answered to ensure the timer for the question for each User is synced |
| 12 | The system can not let more than the MAXIMUM and less than the MINIMUM players in a session. The number can be modified by the Host but a fixed MINIMUM of 2 and a MAXIMUM of 20 is set. |
| 13 | Permanent storage for quizzes and scores is done by registering as a user. The MAXIMUM limit of quizzes is set to 5 per user and scores are recorded to last 10. |
| 14 | Each quiz must have 4 options to the question, one of them being correct. |
| 15 | Quizzes are formatted in CSV form for exporting and importing purposes, with a column for question, columns for all 4 options and the correct answer. |
| 16 | The web application should have fault tolerance such that error in one component should not cause the application to completely crash. [Which system features, in your opinion, can work independent of each other so that failure in one does not affect other system functionality.] |
| 17 | No Player should be able to enter a quiz session once it starts |
| 18 | A fallback central server should always be present to allow immediate accessibility |

# [How would you ensure authentication of students?]Security Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr#** | **Security Risks** | **Potential Losses** | **Controls** |
| 1 | Broken access control (A01: 2021) | Leakage of user data, unauthorized access to the quiz sessions. | Implement a strong role-based access control system such that only the host should be able to control the current quiz session. |
| 2 | Security misconfiguration (A05: 2021) | Exposure of user sensitive data to others. | Regular security audits and disable unnecessary features based on roles. |
| 3 | Identification and authentication failures (A07: 2021) | Identity theft and unauthorized access. | Multi-factor authentication and strong password policies. |

# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Hafsa Ahmed | Introduction, Actors, Functional requirements, Non-functional requirements, Review |
| Abdul Muiz | Introduction, Actors, Functional Requirements, Non Functional Requirements, Review |
| Bisma Nawaz | Actors, security requirements, functional requirements, Non-functional requirements, review |
| Abdur Rafae Haroon | Actors, Non-functional requirements, functional requirements |

# Review checklist

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Introduction | Bisma, Hafsa |
| Actors | Bisma, Hafsa |
| Functional Requirements | Muiz, Hafsa, Bisma |
| Non-functional requirements | Muiz, Hafsa |
| Security Requirements | Muiz, Rafae |