Abstract

The design document presents a rough draft of the overall product design and architecture. It includes Project Goals, Outcomes, Objectives & Deliverables.

multiplayer quiz project design document

By Team Code Keepers

Contents

[Project Goals 2](#_Toc7283096)

[Architecture Overview 3](#_Toc7283097)

[Front End Design 4](#_Toc7283098)

[Authentication 4](#_Toc7283099)

[Service / Back end Design 4](#_Toc7283100)

[Analytics and Logging 4](#_Toc7283101)

[User Flow Diagram: 5](#_Toc7283102)

[Admin Flow 6](#_Toc7283103)

[ Quiz creation / updating 6](#_Toc7283104)

[ Quiz Duration 6](#_Toc7283105)

[ Total Points 6](#_Toc7283106)

[ Quiz Rating 6](#_Toc7283107)

[ Questions Linking 6](#_Toc7283108)

[User Flow 6](#_Toc7283109)

[Entity–Relationship Diagram 7](#_Toc7283110)

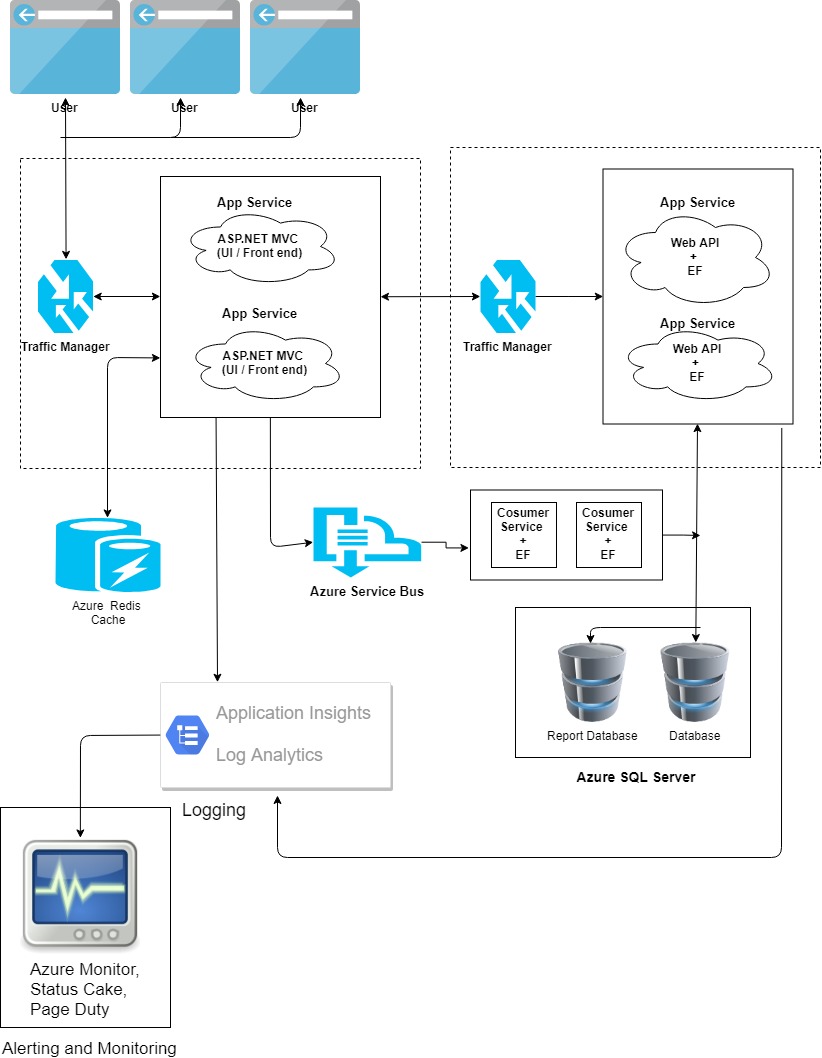
[Site Tree Diagram 7](#_Toc7283111)

[Configuration and Deployment 7](#_Toc7283112)

# Project Goals

* To develop a web application which can be used by participants/users to take quizzes and answer questions.
* Multiple users should be able to participate concurrently.
* It should be highly available and should allow 99.99% uptime.
* It should be able to handle massive amount of data coming in from a large user base at the same time.
* It should be fault tolerant and should provide redundancy in case of a service disaster.
* The web application should allow for horizontal as well as well as automatic vertical scaling.
* It should support two types of roles:
  + Administrator
  + User
* Administrator should be able to do the following:
  + Add Questions
  + Add Quiz
  + Link Questions to Quiz
* User should be able to do the following:
  + Should be able to register using email id, password, Name, etc.
  + Should be able to login to the web application.
  + Should be able to view the list of quizzes.
  + Register for one or more quiz.
  + Attend and answer questions presented in the quiz.
  + See the correct answer once the duration of the current question has ended and only then the next question should be displayed.
  + Should be able to view the leader board.
  + Should be able to view/edit profile.
* Assumptions
  + Admin cannot update / delete a completed quiz or live quiz.
  + Quiz start time should be future date time.

Architecture Overview**:**



Front End Design**:**

Front end will be designed using asp.net MVC along with HTML, CSS, bootstrap, JavaScript and jQuery. Which will be hosted as a web app in azure. The traffic to the site is controlled and load balanced with the help of Azure Traffic Manager.

We use Azure Redis caching in the asp.net MVC to cache the relevant data and serve the same on the subsequent request.

Authentication**:** Authentication will be a simple forms authentication with the user name and password stored in the databased. Password will be hashed and stored in the database. Hashing will be done with the help of AES Encryption.

Authentication at the web API will be handled using JSON Web tokens.

Service / Back end Design**:** Services will be designed using asp.net web API, and Entity framework will be used to connect to the SQL Database. Azure traffic manager is used to manage the instances and load balancing of the web API services

In User critical scenarios like attending the quiz will be handled through azure service bus for concurrency management and reliability. The question answer will be pushed to azure service bus, the listener will pick up the answer details from the azure service bus and uses Entity framework to save the data into the SQL server

We are having a separate database for showing the reports/ leader board results for the user, so that the load will be reduced on the online database.

Azure database backup strategy for backup of the database in case of any failure in the database server.

Analytics and Logging**:**

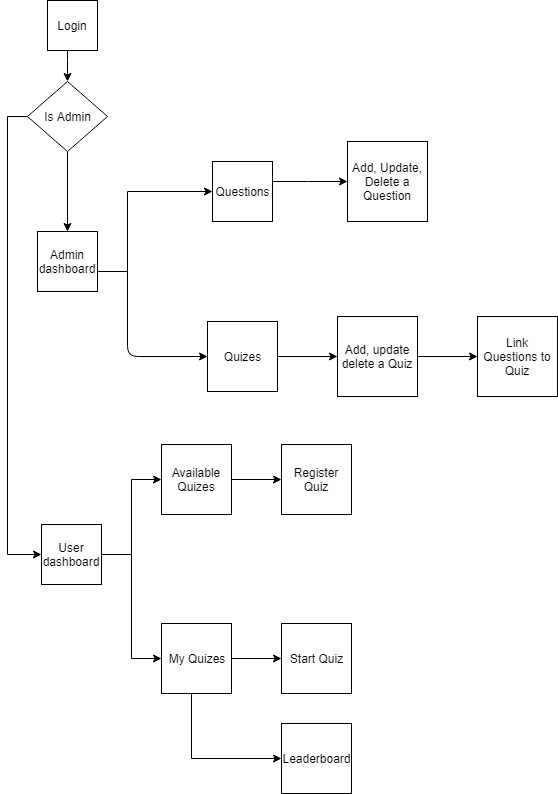
We will be using Azure Application Insights for Trace Level logging and Azure Log Analytics for persistent logging.

Both will allow us to send alerts based on error conditions like HTTP 500, HTTP 403, etc. error codes, exceptions occurred in the application, etc.

We will also be able to monitor the health of our web application, services and APIs through Agentless monitoring system like StatusCake and we would be able to receive alerts on PagerDuty.

The DevOps/SRE team can the work in mitigating the issues and perform escalations as and when required.

# User Flow Diagram:

****

Admin Flow: Once the admin successfully logged into the website, the admin is presented with the dashboard where admin can view all the quizzes, create a new quiz, select an existing quiz and update/delete it. Admin can view the leader board of the completed quizzes.

* Quiz creation / updating: Admin creates a quiz by giving the quiz title, description, start time and visibility i.e. public or private. And then admin links the available question to the quiz.
* Quiz Duration: The quiz over all duration will be sum of the durations of the individual questions.
* Total Points: The total available points of the quiz will be sum of the points of the all the questions
* Quiz Rating: The rating will be calculated based on any of the specified algorithms in the requirement
* Questions Linking: User can link any number of questions to the quiz.

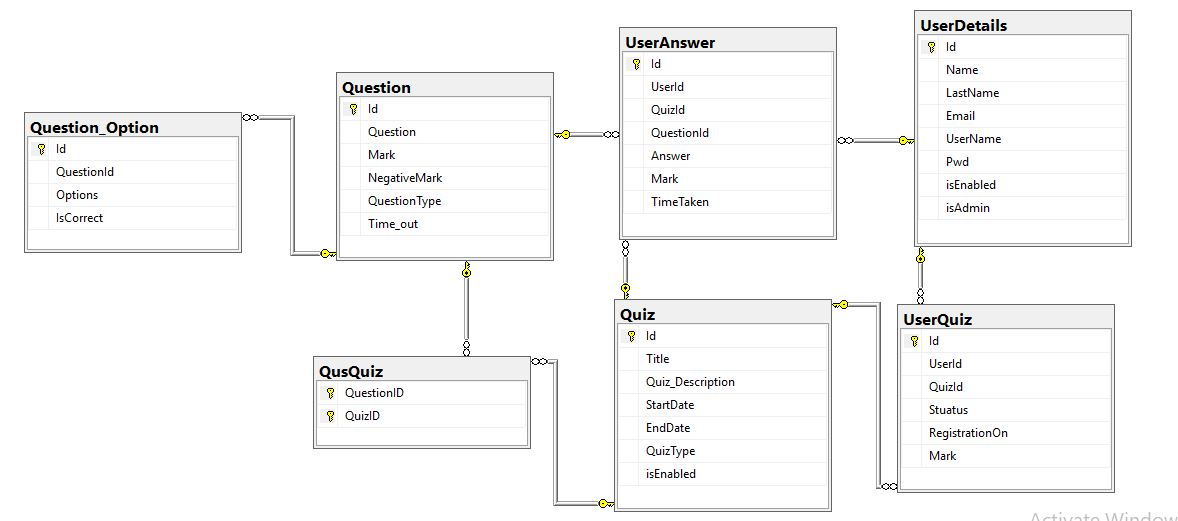
User Flow: Once the user successfully logged into the website, user is presented with the dashboard where user can view all the available quizzes (All quizzes), quizzes which user has registered (My quizzes).

Once the user navigates to All quizzes, list of all the quizzes will be displayed to the user and user can register to a quiz by selecting a quiz. Only quizzes which start in future time will be displayed to the user in this page.

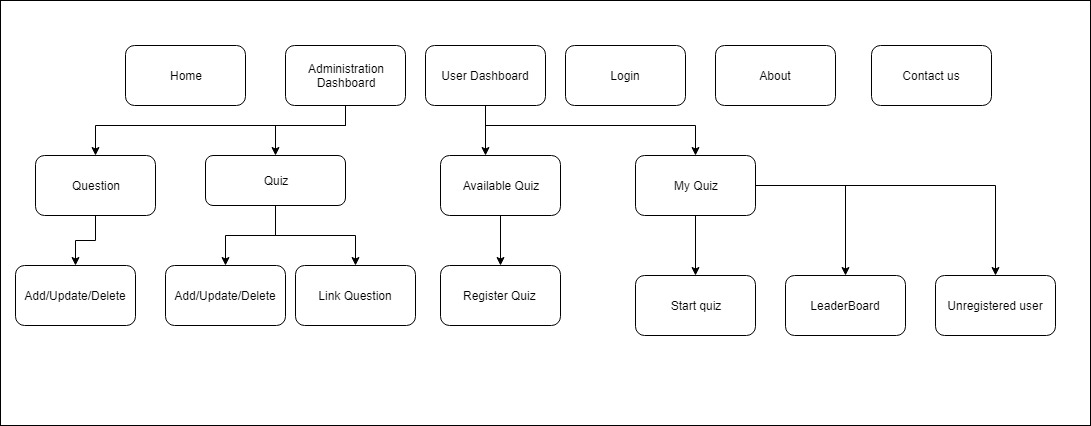
Once the user register to a quiz, the quiz will be moved to My quizzes. Once the user navigates to My quizzes, user will get all the completed quizzes and the all the registered quizzes. User can view leader board of the registered quizzes and user can unregister from a registered question.

User can start a test when only during the test duration i.e. between the start time and end time of the test set by the admin. Once the user starts a quiz, the user will be presented with questions in a sequence, only one question will be displayed at a time. The question will be displayed for a duration of time set by the admin. Once the question has timed out, correct answer of the question will be displayed for the user for 2 seconds and user will be navigated to the next question.

Entity–Relationship Diagram:



Site Tree Diagram:



# Configuration and Deployment

* Need to create azure SQL Server.
* Need access to azure portal for deploying the asp.net MVC and asp.net web API to azure web apps, redis cache, traffic manager, Log Analytics, app service, etc.
* Need to setup azure Service bus as a messaging queue.