Technical Implementation Plan, CS-554 Final Project

Revisied May 16, 2021

Deletions shown with ~~line strike~~ .

Additions shown in red.

Group Tempest

Group members

Kyle Bernardes CWID 10412644

Mahima Chandan CWID 10467889

Dale Pippert CWID 10469880

~~Madeline Rys CWID 10416134~~ (dropped the course)

Project Overview

Falling under the broad classification of *gamification [1]*, our project is **Topic Tempest**.

Topic Tempest is a subset of a Jeopardy!-like game [2]. Unlike Jeopardy!, the board holds *questions*, which players compete to *answer* first, by typing into a text box on their browser screen [3]. Topics can be anything, but our intent is to target the game towards learning and professional training environments, as opposed to trivia topics characteristic of Jeopardy! [4].

Planned Course Technologies

This section lists technologies covered in course lectures that are planned for use by our group.

React. Client side will be built using Facebook React framework. React is an excellent choice for this project, because the notion of *state* is first-class and central to its existence, and in game play, understanding and modeling the *state*, *actions*, and *events* that cause state transitions, is just *everything* important.

~~***Redis***. We will use Redis for some of the non-game-stateful data. Possibilities for this include a leaderboard and/or holding the questions and answers for the various topics we create.~~

MongoDB. We will use Mongo for the topic database of questions and answers. As we gained experience with Redis during the course, it became apparent that Mongo was a better fit for traditional CRUD which is what we needed.

GraphQL. Unaware of GraphQL when our plan was first submitted, we will use the Apollo GraphQL framework for all queries and mutations between frontend and backend.

Firebase Authentication. Our app will authenticate users using Firebase Authentication.

Planned Independent Technologies

This section lists technologies that were not covered in course lectures, that we plan to investigate and make use of.

Firebase Realtime Database. Another separate and distinct product under the Google Firebase umbrella. We will use this to keep the game state (as represented by your browser screen) across players in sync.

Screen reader. We intend to explore use of screen reader technology (text-to-speech) within the client-side application. Our plan is to make use of this for speaking questions to the players, as well as providing guidance, status, and control information during game play. We believe leveraging this type of assistive technology within our platform, is both course-relevant and compelling in terms of application accessibility. Note this is not just automated screen reading under operating system control. This is a browser-only API driven programmatically by our application to only synthesize certain aspects of the screen to guide the player along.

Notes

[1] Wikipedia defines gamification as *the strategic attempt to enhance systems, services, organizations and activities in order to create similar experiences to those experienced when playing games, in order to motivate and engage users.*

[2] https://en.wikipedia.org/wiki/Jeopardy!. (N.B. And with special tribute to longtime-host Alex Trebek, who recently passed.)

[3] In Jeopardy!, the board holds *answers*, and players must respond with (e.g.) "What is.." as a lead-in to their proposed *question*.

~~[4] Creating topics, questions, and answers and loading them into the database is an administrative, offline exercise, and not included in the application user interface.~~

[4]. Creating topics, questions, and answers and loading them into the database will be included in the application user interface.