### Assignment 4-4: Journal: Software Application Requirements

### David A. DiPesa

### CS-230-11604: Operating Platforms

### **Alexander Dubinski, MS CS, MBA**

July 27, 2024

Recently, I’ve been thinking about our project at CTS and, more importantly, how to help The Gaming Room achieve their goals with Draw It or Lose It. We need to achieve running the game on multiple operating platforms per the Gaming Room’s requirements. Upon reflection, I realize that the client-server architectural pattern has yet to be defined.

The client-server pattern will be used to satisfy our software requirements, specifically, where a web-based game application *must* be able to run on multiple operating platforms. The architecture of the pattern is made up of a single server and its clients, with data storage and data access logic being situated solely on the server side. The server will be able to hold the base game and main attributes that will be inherited by each different client. Clients will have the presentation and application logic of the game to contend with as all different devices have their own unique way of managing data and performance. As we further understand the abilities of the server and clients, we can form a solid bridge between them to carry our game application. No user shall be left behind!

Developing the application on the server side has many advantages. Using the REST API style to transmit data from the server side to the client side definitely tightens up security. It’s never a bad thing to make a web-based game secure as much as it’s enjoyable – both parts should hold hands, I think. Also, a huge advantage of the REST API is that it’s stateless, meaning that calls can be made independently of one another and that each call contains all of the data necessary to complete itself successfully. In plain, unabridged English, this will increase speed and improve the game’s overall reliability. Our clients will feel the benefits while either loading up the game or playing for hours on end.

When any of us are creating an application for use on a client’s device, it’s crucial to consider their requirements based on the different types of devices they may use and the software that they employ. Things like screen size, user input methods and device performance are circumstantial but cannot be taken lightly. You need to account for all of it or you won’t be successful. Now, to ensure the application on all three clients is able to be used, our CTS developers will need to fully understand the scope of this project. They will need to create secure authentication and also make sure to limit the functions of the game’s “players” within the gaming environment. To do this, we can assign players a normal user role verses an admin one. After all, you don’t want the users to access everything and even change the game on you, right? These methods are successfully employed by PLM / PDM systems to ensure that not all end-users wield power, behind the scenes, that they shouldn’t. In my time maintaining proprietary data at different companies, I’ve learned the value of this practice.

We at CTS should compile a list of the intended actions a user can do and build around it. Lastly, we want to create a modular and maintainable codebase so that future updates, additional content and changes will be as easy as possible on our client. There are times when implementing a small change can crash a game. Our developers want to do their best to never enter this irritating and dangerous zone.

A way to add more users to the database is as simple as just creating a streamlined, registration process with user authentication shown on the client side that interacts with the server side through a cert.

I’ve been thinking about some features I would like to add to the Draw It or Lose It game application. One feature that is clear, in my mind, is the ability to view high scores based on time to solve the puzzles and the number of puzzles solved in a single play through. Most games call these leaderboards. I think it would be a great addition to show teams who’re the best of the best. It might provoke some fierce competition too!

Also, wouldn’t it be an overall “awesome” gaming experience if you received redeemable points to spend on extra special content, higher level puzzles or stakes? I play a lot of web-based games and these types of functions tend to give me more incentive to continue playing. Even the drabbest of games gets more users thanks to additional, added content.

An interesting expansion project would be if The Gaming Room asked us to host the application on a fourth and fifth client. For example, they may choose Xbox and / or PS5. This could be done, but the server would definitely need the proper bandwidth to support adding more clients – anticipating huge amounts of traffic on the client and server sides respectively.