



Draw It or Lose It.

**CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

<b>CS 230 Project Software Design Template</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>Document Revision History</b>	<b>2</b>
<b>Executive Summary</b>	<b>3</b>
<b>Design Constraints</b>	<b>Error! Bookmark not defined.</b>
<b>System Architecture View</b>	<b>Error! Bookmark not defined.</b>
<b>Domain Model</b>	<b>Error! Bookmark not defined.</b>
<b>Evaluation</b>	<b>5</b>
<b>Recommendations</b>	<b>7</b>

## Document Revision History

Version	Date	Author	Comments
1.0	10/10/2022	Joseph Farrish	Added recommendations.

## **Executive Summary**

Creative Technology Solutions, creators of *Draw It or Lose it*, is a game loosely based on the 1980's television game, Win lose or Draw. Hardware requirements will come later as a result of software application decisions. Gaming Room wants to expand *Draw It or Lose it* from the current version available for Android into a web-based game. The Game will need to serve multiple clients across several platforms.

## **Requirements**

- The Game will have one or more teams involved.
- Each Team can have one or more players assigned to it.
- Uniqueness checks for Game and Team names allow the user to see if a name is in use.
- Only one game instance can exist in memory at any given time.

## **Design Constraints**

Because of increased support across multiple architectures, storage and memory constraints must be re-considered and adapted accordingly. Security is also a concern while the current system supports Authentication via Google, which is well suited for Android, Windows, and Linux access. Sign-in with Apple should be implemented to allow authentication of Apple clients. In addition, email and password authentication that is referenceable to the operating platform should be implemented to allow for login independent of Google or Apple services.

Testing will require three environments with the appropriate hardware and operating system to serve the Game. Solutions can be on-premises or cloud-based. Servers will require

increased storage, memory, and processing power to render images at the required speed to meet or beat mobile performance.

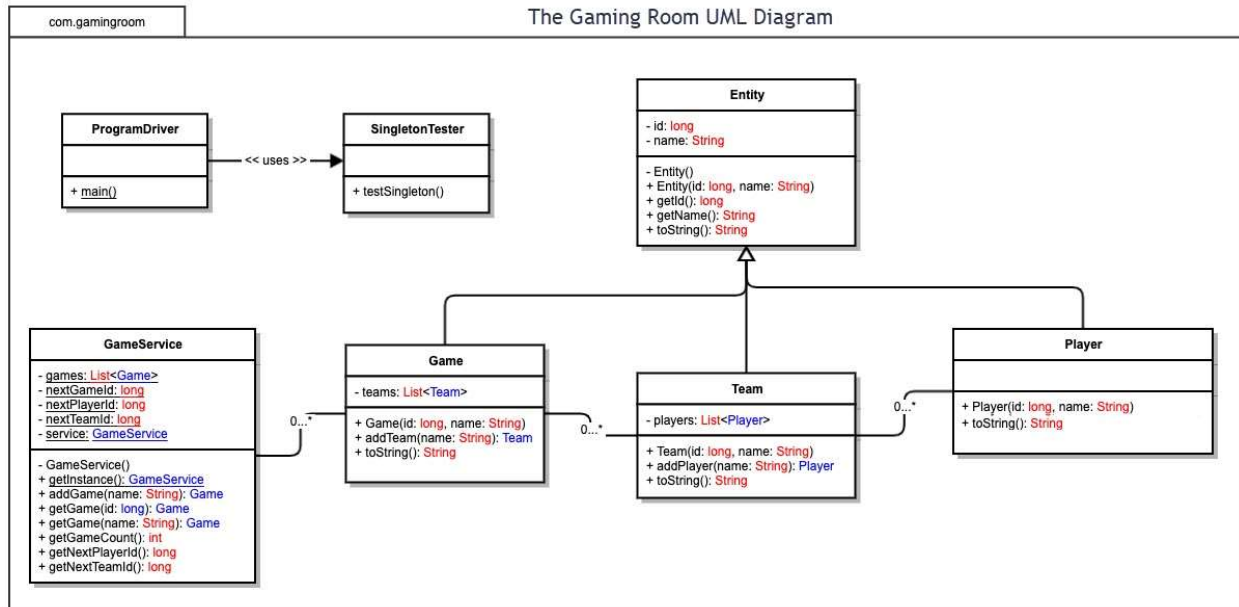
The Game service will also need to impose login restrictions to ensure a User isn't logged into multiple systems. In addition, administrative interfaces will need to be implemented to manage users and teams.

### **System Architecture View**

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

### **Domain Model**

The Domain Model satisfies the client's requirements first by establishing the entity class as a base class for Game, Team, and Player classes. By inheriting from the Entity class, identification and name functionality is efficiently implemented and encapsulated via the appropriate get and set methods. The game and team class use an iterator pattern for addTeam() in Game and addPlayer() in Team. These functions ensure that team and player names are unique by first checking for a duplicate entry. If one is found that instance is returned, else the name is created for the Team or Player. The toString() method is polymorphic, exists in all classes, and will print the respective names and identification for a Game, Team, or player. Finally, the Game Service class is where a singleton pattern is employed to limit the application to a single instance, this is achieved via the getInstance() function.



## Evaluation

<b>Development Requirements</b>	<b>Mac</b>	<b>Linux</b>	<b>Windows</b>	<b>Mobile Devices</b>
<b>Server Side</b>	While Apple has a Mac OS server X at one point, it's now discontinued. Mac on the server side is nonexistent. Serves Apache HTTP Server, Nginx, Lighttpd. Serve Based Deployment available through third party software. Mac is not a good option for the customer.	Ubuntu, RedHat, SUSE Linux all have open-source, stable and secure server platforms. All can run Apache HTTP Server. With Cloud platforms offering these operating systems. Server Based Deployment and low cost.	Windows Server is the de facto choice for many enterprises, and smaller business IT infrastructure. Coupled with MS Azure Microsoft is the top player in server technology. Offers HTTP via IIS and MS Web Platform. Server Based development is available. Cost will likely be higher.	While android has a huge potential to act as a server it's not a go to solution in a professional setting. No server editions of the software exist and no solutions for Server based deployment exist.
<b>Client Side</b>	Mac OS X Version 10.6 and below supports Java 6. Mac OS X 10.7.3 or newer supports Java 7 and later versions. Fully Supports HTML 5 and Java in multiple browsers.	Supports Java Versions of 7.0, 8.0. Supports Chrome, Firefox, both can run the latest versions of HTML 5 and support Java in the browser.	Supports Java Versions of 8.0. Supports Chrome, Firefox, Internet explorer can run the latest versions of HTML 5 and support Java in the browser.	Does not support Java applets via the browser. Support for Java Applications is supported.
<b>Development Tools</b>	Java - Eclipse, VS Code with Java SDK, and NetBeans. HTML – Atom, VS Code, and browser for testing. May require a front and back-end team. Testing environment from the client and server end required.	Java - Eclipse, VS Code with Java SDK, Blue J and NetBeans, more. HTML – Atom, VS Code browsers for testing. May require a front and backend team. Testing environment from the client and server end required.	Java – Eclipse, VS Code with Java SDK, NetBeans, more. HTML – Atom, VS Code browsers for testing. May require a front and backend team. Testing environment from the client and server end required.	While there is a lot of development tools for android you would typically develop for android on Linux or Windows. Kotlin, Android Studio, Android SDK. A separate team would be required for developing and testing on android.

## Recommendations

**1. Operating Platform:** To develop Draw It or Lose so that Gaming Room can reach users on Linux, Mac, and Windows and to meet the four development environments - development, testing, production, and quality assurance. CTS Recommends Microsoft Azure. Azure is a cloud computing platform offering several solutions to meet The Gaming Rooms Cross Platform requirements. Azure is a specialized operating system that runs on top of Windows Server, heavily utilizes virtualization, and allows for Platform independent operation on both the server and Client end. With Azure, a Server Solution can be configured to run on several different Linux distributions such as Red Hat, Ubuntu, and Cent OS as well as a Windows Server Operating system, all in a Virtual Machine housed within Microsofts optimized infrastructure (Pricing - Windows Virtual Machines, n.d.).

**2. Operating Systems Architectures:** CTS recommends a Red Hat Linux Virtual Machine with 8 Cores and 32 gigabytes of RAM to target a specific configuration. With an Optimized REST API, the Draw It Or Lose It application will operate via multithreaded functionality. Thus, Game Players will experience responsive gameplay through shared memory and message-passing systems that support concurrent operation, prompt process scheduling, and high-speed i/o operation for reading and writing to temporary and permanent storage. All are provided by the Azure storage infrastructure, such as blob storage. A REST API running on the recommended architecture is favorable for client-end operation, as it is platform-independent, thus clients will be able to access the Draw It Or Lose it App from the Mac, Linux, or Windows using standard browsers (Azure VM Sizes - Memory - Azure Virtual Machines, 2022).

**3. Storage Management** In order to store images and user chats generated in Draw It Or Lose It, Azure Blob storage is a suitable storage solution for storing massive amounts of unstructured data. It allows serving images or documents directly to a browser, storing files for distributed access, and writing log files. Users or client applications can access objects in Blob storage via HTTP and HTTPS with support for Java. End-user information can be stored and recalled by the users via storage accounts by providing end users with a unique namespace in Azure. This operates using an address and a unique account name; below is a typical blob storage address. A container organizes blob sets in a fashion similar to a directory file structure, and accounts can include unlimited numbers of containers. Containers can store an unlimited number of blobs—all of which tie into a Domain Name Space. There are three types of Blobs in Azure - Block blobs - are used for text and binary data. Append blobs are ideal for logging data, and Page Blobs are used as virtual hard drives (Introduction to Azure Storage - Cloud Storage on Azure, 2022).

**4. Memory Management** - Using a Virtual Machine on Azure requires decisions on the amount of memory a VM with the recommended 32GB RAM at \$313.22 a month (Azure VM Sizes - Memory - Azure Virtual Machines, 2022; Pricing - Windows Virtual Machines, n.d. ). When considering the game service, the stand-alone game service application consumes 25 megabytes of ram; if we take the stand-alone authentication server, that's an additional 108 megabytes of ram consumed. If we generate 50 Players and add them to a team, the game service will consume 25.3 megabytes of ram. The server will hold  $200 * \sim 8$  megabytes of ram =  $((200 * (1024 * \sim 8)) + (25 * 1024) + (108 * 1024))/1024 = \sim 1.73$  Gb of Ram needed for the game based on the current design. This configuration should be enough memory caching to cache all 200 images,



thus avoiding paging and fragmentation while maintaining high performance by storing image data in a contiguous manner. This configuration will allow memory segments for kernel code, application code, and application data.

**6. Distributed Systems and Networks** - Using Azure with a Java back end, REST API, and a Single Page Website allows for much flexibility where the Client and Server roles are separated. This design provides different layers of operations to handle security, perform load balancing, to cache the image set. Using HTTP headers, CTS and the Gaming Room can exploit origin servers that cache along a response path, thus improving the overall server quality by reducing bandwidth, latency, and load on servers. Using Azure enables CTS and The Gaming Room to write a back-end application in the language of choice. Supporting Java should not be a problem; supporting an operation that runs alongside the current Android solution should not be a problem; migrating from the current solution will require additional planning. The Rest API is favorable to clients as it will allow CTS to produce a uniform interface that is stateless, cacheable, and layered. Additionally, the Gaming room will be able to serve Draw It Or Lose It to the end via HTTPS and HTML(Gupta, 2022; What Is REST, 2022).

**7. Security** - Microsoft has an array of services to support a secure operation. Azure Active Directory services allow security features such as Single sign-on and multifactor Authentication. The Application Gateway service offers front-end security features such as a Web application security system that interfaces with all Azure services. These days in Gaming, DDOS attacks are a common threat that has hindered Top Gaming Companies. Azure offers Threat intelligence using traffic monitoring.

Additionally, support for Rest API styled Authentication is not a problem. Through these facilities, CTS and The Gaming Room can support customer Security requirements that establish a Domain of Protection where hardware and software objects have a well-defined operation and, in turn, only be allowed access to the necessary resources. Through this practice, Azure can provide a Domain Structure where objects operate within a protection domain that specifies software or hardware resources dictated through access rights. Using high-security measures that utilize Secure Socket Layer Certificates to establish encrypted communication. By implementing a unique RSA encryption algorithm that's on par with MD5 or SHA-1 standards, we can ensure the data safety of the end user and the uptime of the Game Service as a whole. Using a Singleton Design Pattern CTS will limit the Game Service to a single instance and ensure the user experience isn't hijacked by nefarious characters (What Is an SSL Certificate? | DigiCert, n.d.).

## References

- Azure VM sizes - Memory - Azure Virtual Machines. (2022, August 27). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/virtual-machines/sizes-memory>
- Choose an Azure compute service - Azure Architecture Center. (n.d.). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/compute-decision-tree>
- Cloud design patterns - Azure Architecture Center. (2022, August 29). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/architecture/patterns/>
- Gupta, L. (2022, January 1). Caching REST API Response. REST API Tutorial. Retrieved October 15, 2022, from <https://restfulapi.net/caching/>
- Introduction to Azure Storage - Cloud storage on Azure. (2022, October 12). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/storage/common/storage-introduction>
- Pricing - Windows Virtual Machines. (n.d.). Microsoft Azure. Retrieved October 15, 2022, from <https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows/>
- Serverless web application - Azure Architecture Center. (n.d.). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/serverless/web-app>
- Understand data store models - Azure Application Architecture Guide. (2022, October 13). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/data-store-overview>
- Web API design best practices - Azure Architecture Center. (2022, September 30). Microsoft Learn. Retrieved October 15, 2022, from <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>
- What is REST. (2022, April 7). REST API Tutorial. Retrieved October 15, 2022, from <https://restfulapi.net/>
- What is an SSL Certificate? | DigiCert. (n.d.). Retrieved October 15, 2022, from <https://www.digicert.com/what-is-an-ssl-certificate>

Wikipedia contributors. (2022, October 11). Microsoft Azure. Wikipedia. Retrieved October 15, 2022, from [https://en.wikipedia.org/wiki/Microsoft\\_Azure](https://en.wikipedia.org/wiki/Microsoft_Azure)