

# Draw It or Lose It

# **CS 230 Project Software Design Document**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/24/2022 | Cory Remick | Initial version |
| 1.1 | 03/30/2022 | Cory Remick | Revisions to Executive Summary, Design Constraints, and Domain Model based upon feedback |
| 1.2 | 04/2/2022 | Cory Remick | Added Evaluation: Server Side, Client Side, Development Tools for MacOS Server, Linux, Windows, and Mobile |
| 1.3 | 04/16/2022 | Cory Remick | Revisions to Evaluation, Server, Client, Development Tools based upon feedback |
| 1.4 | 04/17/2022 | Cory Remick | Added Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

Our client, The Gaming Room, desires to create a Draw It or Lose It game. They want a web game based on their existing Android application. The following are client specified requirements:

* The application will render images from an extensive stock library.
* The game will consist of four 1-minute rounds of play.
* If time expires, the remaining teams will have an opportunity to guess within 15-seconds.
* A game will have one or more teams involved.
* Each team will have multiple players.
* The game and team names must be unique.
* Each game, team, and player objects need to have unique identifiers.

## [Design Constraints](#_2et92p0)

* Object-oriented design with Abstraction, Encapsulation, and Inheritance see Domain Model.
* Only one instance of the game can exist at any time.
* We need a client-server architecture using web standards for interoperability.
* Need to support multiple client browsers; Chrome, Firefox, Safari, and Edge will give 96% desktop market share (Vailshery, 2022) across Windows, Mac, and Linux operating systems.
* The server must be able to access an image library, which is to be determined.
* We must allow players to check if a name is already used for unique names.

## 

## [System Architecture View](#_ilbxbyevv6b6)

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](#_8h2ehzxfam4o)

Abstraction defines logical representations and operations for the GameService, Games, Teams, and Players. The data for each abstraction uses Encapsulation to hide the data in private fields that can only be operated on by public methods. The implementation details of each method are unknown to the calling objects.

The GameService is responsible for managing Games, Teams, and Players. The static method "getInstance" combined with the private constructor of the GameService ensures that only one GameService instance can be acquired. This follows the singleton pattern.

The GameService offers the mechanism to ensure identifiers are uniquely generated and used through static variables and methods that increment when called.

The GameService references zero or more Game objects through the private field "games." New games are added through the public "addGame" method.

Game objects represent the state of a single game. They hold references to zero or more Team objects through the private field "teams." New teams are added through the Game object's public "addTeam" method.

Team objects represent the state of a team. They have a reference to zero or more Player objects through the private field "players." New players are added through the team's public "addPlayer" method.

A player object represents the state of a single game user.

Game, Team, and Player classes inherit from the Entity class. The Entity class provides common properties, an identifier, and a name. The name property offers a way to know if names are unique by looking up entities by name in the list to see if they already exist. The identifier property provides a way to compare and ensure only one id exists, and therefore, there are no duplicates of an entity instance.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side**  **Jetty 11+**  Provides Webserver Platform for dropwizard.io REST services.  Requires JVM 11+  (Jetty, n.d.)  **Java Virtual Machine 11+**  Provides the Java Interpreter  (Jetty, n.d.) | Apple's macOS server is based on the UNIX operating system.  Apple's licensing requires the OS to run only on Apple hardware, and most cloud providers do not run Apple hardware.  Java VM Can run on macOS. (Java, n.d.) | Linux is ubiquitous with cloud service providers.  Its open-source licensing model is cost-effective.  Lean resource usage means being able to host more games with fewer servers.  Java VM Can run on Linux. (Java, n.d.) | Windows is also widely used as a server platform.  It has good support and compatibility.  It is more of a resource hog, meaning fewer games hosted per server.  Higher licensing costs.  Java VM Can run on Windows. (Java, n.d.) | Network connectivity could be intermittent due to their portable nature.  No known web server provided as part of the operating system.  Mobiles devices require Java-ME which is not known to be compatible with Jetty 11. |
| **Client Side**  All browsers must support HTML5 and ECMAScript (Javascript) | **Safari 15.4**  Introduced Mar. 2022  (Safari, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2017+  (kangax, n.d.) | **Firefox 99**  Introduced Apr. 2022  (Firefox, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.)  **Chrome 100**  Introduced Mar 2022  (Chrome, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.)  **Opera 85**  Introduced Feb 2022  (opera, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.) | **Firefox 99**  Introduced Apr. 2022  (Firefox, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.)  **Chrome 100**  Introduced Mar 2022  (Chrome, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.)  **Edge 100**  Introduced Apr 2022  (Edge, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.) | **Chrome 100**  Introduced Mar 2022  (Chrome, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2022  (kangax, n.d.)  **Safari 15.4**  Introduced Mar. 2022  (Safari, n.d.)  **HTML5**  Supported  (HTML5Test, n.d.)  **ECMAScript**  Limited support for 2017+  (kangax, n.d.) |
| **Development Tools** | Macintosh – OS Development/Testing  Safari – Browser Testing  Eclipse – Integrated Development Environment  Github – Source Code Versioning and Control  Jenkins – Continuous Integration and Continuous Delivery for managing releases | Ubuntu – OS Development/Testing  Chrome – Browser Testing  Firefox – Browser Testing  Eclipse – Integrated Development Environment  Github – Source Code Versioning and Control  Jenkins – Continuous Integration and Continuous Delivery for managing releases | Windows – OS Development/Testing  Eclipse  Chrome – Browser Testing  Firefox – Browser Testing  Edge – Browser Testing  Eclipse – Integrated Development Environment  Github – Source Code Versioning and Control  Jenkins – Continuous Integration and Continuous Delivery for managing releases | Android Device – OS Testing  iPhone Device – OS Testing  Chrome – Browser Testing  Safari – Browser Testing  Not suitable as a development environment. |

## 

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: The server application will be built on DropWizard.IO to provide REST endpoints to service requests from Browser clients. DropWizard.IO requires a Jetty web server. The Jetty web server requires a Java Virtual Machine to run. Therefore, the server host must be capable of running a Java Virtual Machine. The web server may also be responsible for delivering HTML, ECMAScript (Javascript), images, and CSS files to the clients.
2. **Operating Systems Architectures**: Aside from the development costs, the next significant cost to the client will be operating expenses. To ensure our client can achieve profitability, I recommend using the Linux operating system to host the server architecture. Linux is an open-source operating system with a multiprocessing architecture, memory and storage management, and process and job scheduling capabilities (Silberschatz, Galvin, & Gagne, 2009). Zero licensing costs and being able to host more games per server will result in lower operating costs for the client.
3. **Storage Management**: Recommend servers with Solid-State drives (SSD) for improved read performance. Each server will need at least 1.6 gigabytes of storage space to handle the image library plus extra for the server code, Java Virtual Machine (JVM), and client code.
4. **Memory Management**: Recommend loading the image library from disk into physical RAM to improve performance in serving images to the clients. This will require at least an extra 1.6 gigabytes of RAM on top of whatever the operating system, JVM, Jetty, and the server code need.
5. **Distributed Systems and Networks**: The use of the following standards, HTTP, HTML, ECMAScript, REST, and JSON, will allow exchanges of data across the globe with other operating systems and devices. To maximize our client's customer base while streamlining support costs, I recommend supporting the following web browsers; Firefox, Chrome, Edge, and Safari. This will give our customer a 90% browser market base worldwide across desktop and mobile operating systems (Browsers, 2022).
6. **Security**: The Jetty web server can perform user authentication and authorization to ensure that the server system is accessible to authorized users. It is anticipated that our client will want to run multiple game servers to handle customer demand. Therefore it does not make sense to store user account information on each Jetty web server but to have it hosted in a separate server accessible to all web servers. I recommend storing account information on a different Linux server. While it is possible to use identity services provided by cloud providers, they will lock our client into a specific cloud provider, preventing shopping for the best hosting price. Using an additional Linux server, our client only has to move the servers to another provider, and they should run with minimal to no modifications. Each Jetty server will forward identity requests to the Linux server for account creation, deletion, update and reading.

# **References**

Browsers. (2022, March). *Browser Market Share Worldwide*. Retrieved April 17, 2022, from statcounter.com: https://gs.statcounter.com/browser-market-share#monthly-202201-202203

Chrome. (n.d.). *Google Chrome version history*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Google\_Chrome\_version\_history#:~:text=Versions%20%20%20%20Major%20version%20%20,First%20stable%20release%20%2018%20more%20rows%20

Edge. (n.d.). *Microsoft Edge release schedule*. Retrieved from microsoft.com: https://docs.microsoft.com/en-us/DeployEdge/microsoft-edge-release-schedule

Firefox. (n.d.). *Firefox*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Firefox#:~:text=Version%201.0%20of%20Firefox%20was%20released%20on%20November,January%202010%2C%20and%20version%204.0%20in%20March%202011.

HTML5Test. (n.d.). *HTML 5 Test*. Retrieved from html5test.com: https://html5test.com/results/desktop.html

Java. (n.d.). *Java Downloads*. Retrieved from oracle.com: https://www.oracle.com/java/technologies/downloads/#jdk18-linux

Jetty. (n.d.). *Jetty Downloads*. Retrieved from eclipse.org: https://www.eclipse.org/jetty/download.php

kangax, w. z. (n.d.). *ECMAScript compatibility table*. Retrieved from kangax.github.io: http://kangax.github.io/compat-table/es2016plus/

opera. (n.d.). *Opera (web browser)*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Opera\_(web\_browser)#:~:text=Opera%20was%20initially%20released%20on%20April%2010%2C%201995,first%20browser%20for%20mobile%20device%20platforms%20in%201998.

Safari. (n.d.). *Safari version history*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/Safari\_version\_history#Safari\_15

Silberschatz, Galvin, & Gagne. (2009). *Operating System Concepts.* John Wiley & Sons, Inc.

Statcounter. (2022, Feb). *Operating System Market Share Worldwide*. Retrieved 03 25, 2022, from gs.statcounter.com: https://gs.statcounter.com/os-market-share

Vailshery, L. S. (2022, Feb 21). *U.S. desktop internet browsers market share 2015-2021*. Retrieved 3 30, 2022, from statista.com: https://www.statista.com/statistics/272697/market-share-desktop-internet-browser-usa/