

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/17/24 | Kyle Musclow | Initial draft of Draw It or Lose It |
| 2.0 | 08/02/24 | Kyle Musclow | Second draft of Draw It or Lose It |
| 3.0 | 08/13/24 | Kyle Musclow | Final draft of Draw It or Lose It |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

We at Creative Technology Solutions (CTS) were asked by a new client, The Gaming Room, to develop a web-based game that can serve multiple platforms . Currently Draw It or Lose IT is an android app only. The application will include multiplayer function, teams and a system where only one instance of the game will exist at a time. Two key design features will be singleton and iterator that will help with maintaining the game and performance. Below we will go over additional requirements and more in-depth information pertaining to the development of the additional game platforms.

## Requirements

* The game will need to be able to support multiple players and teams.
* Provide unique game play.
* Allow only one instance of the game to be in the memory.
* We must use the Singleton pattern to manage the game instance.
* We must use the Iterator pattern to help manage the game flow and team names / assigning players to a team.

## [Design Constraints](#_2et92p0)

Currently The Gaming Room has an Android app for the game Draw It or Lose IT. We at CTS have been tasked with expanding the hit game to a web-based game to work alongside its Android app. The web-based game will run off of Java which will allow for ease of maintenance and use. Ensure both the current Android app and the web-based game work seamlessly while providing high levels of security to protect the end users and The Gaming Room.

## [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)

The Entity class serves as the base for the whole game. This class provides almost all of the intel to the other classes and extends to Game, Team and player. Doing so allows these classes to inherit the properties of the entity class. To manage the game instance, the GameService class is used to act as a program manager and extends to the game, team and player classes. This would be used to provide the data for player name, team name and creating the GameService instance.

As for the ProgramDriver class, this is used to initialize an instance of GameService while using the SingletonTester to ensure that only one instance of GameService exists. The ProgramDriver class also holds the Main which is basically the starting point for the program and acts as a controller.

**"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)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac servers are known to be very stable using macOS. Strong security and ease of use for developers. Con is they are expense and less hardware options.  macOS Server  Very costly | Easy ability to customize and is open-source. Very const effective and lots of support.  LAMP Stack  Generally, the cheapest | Great for integration to all Microsoft technology. Ease of use as it is a familiar OS. Lower lcensing fees than macOS. | This is more of a user endpoint than a server side. However it is possible to run a server off an Android device. This does not typically happen. Much less scalable. |
| **Client Side** | Since the game will be running through a web browser hosted by the client’s devices, you would need to consider only that it is available to use across all browsers like Chrome, Safari and Edge for example. macOS is used far less than Windows  MAC SDK is SWIFT or Objective-c | Again, very similar to macOS. The biggest factor is to ensure the application runs across all browsers. Perks of Linux though are that its cheaper and much more customizable.  Being that Linux is open source it is free to use on the client side.  Linux can run just about any software. | Same as macOS. There is no special consideration due to it being a web based application. Just that it would need to support all browsers.  Windows runs on most computers in the market.  There are little to no limitations of Windows that couldn’t handle the application. | As for a mobile device it gets a bit more complex because you need to ensure the game is able to be played in a mobile view native to the phone and have the ability to request and display the desktop version. |
| **Development Tools** | The development tools available for Mac are Xcode, Visual Studio code, these allow languages like Swift, HTML and JavaScript .  Xcode is the most used IDE with a cost of $99 per year. | Linux uses GCC and eclipse as a few tools and languages like C++, Python and Java.  These are more well used languages making it easier to build on.  Eclipse IDE available and is free while also allowing several languages. | Windows is similar to Linux, using VS Code, Eclipse among other IDE’s.  Languages like C++ and JavaScript are common use.  Visual Studio is free for an individual but offer a paid full suites. | Mobile- Tools like Xcode for iOS and Android Studio for Android are used. These require languages like Objective-C for iOS and Kotlin for Android.  iOS uses Objective-C and SWIFT languages but is done in Xcode.  Xcode is priced at $99 per year.  Android Studio is free to use. |

## 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**: I would recommend using Windows as the operating platform for this project. Using Windows allows for better integration across windows devices along with lower licensing fees. Another major benefit is it is the most popular and has the most users which means that tools and parts are plentiful.
2. **Operating Systems Architectures**: Windows architecture would be great to use for the server to its high efficiency and ability to customize. However, on the client side we will need to ensure access across all operating systems. Windows allows all users to be developers and create on its platform. Providing a ton of support, forums and help from many resources.
3. **Storage Management**: Windows servers use Storage Spaces as the storage management system that will be used for the Windows server. Which is very effective at managing storage. We would also be able to capitalize on cloud-based storage which will help offload some storage to the cloud and only retrieve is as needed.
4. **Memory Management**: We will use effective memory management techniques for windows OS to ensure we are using the least memory needed. We need to store 200 photos that are 8MB each on our platform but one thing we are able to do is compress the images and use the RAM to preload the images we need instead of all images.
5. **Distributed Systems and Networks**: To ensure the game Draw It or Lose It can effectively communicate between several platforms we will need to use distributed systems along with a strong network. Since this will be a webserver, we are able to use REST API for communication between the application and the server. We will also be using cloud technology that will allow us to run the server which will help with storage and also allow access to the server from anywhere at anytime.
6. **Security**: To provide the highest level of security we will implement HTTPS for secure communication. Using HTTPS will use encryption through secure sockets layer (SSL) This will help protect our users login information, personal information and possible credit card info for purchases.