

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 | <11/12/23> | Fernando Lomeli | The executive summary, design constraints, and Domain model were updated. |
| 1.1 | <11/26/23> | Fernando Lomeli | Updated the Evaluation table. |
| 1.2 | <12/8/23> | Fernando Lomeli | Updated the recommendations sections operating platform, operating system architectures, storage management, memory management, distributed system and networks, and security. |

**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)

My client The Gaming Room would like to turn their current game, Draw It or Lose It, which is currently only available on Android, into a web-based game that can be played on multiple platforms. The game consists of the application rendering an image slowly until it is fully complete. The team must guess the image before the clock runs out and if they do not, other teams will have the opportunity to steal. The problem is that The Gaming Room does not know how to set up the environment. The Gaming Room also has some software requirements for the project. This web-based game must be made available to all web browsers. To set up the environment for this game, we must use iterator patterns to ensure that game names, team names, and game ids are unique. We must also use the singleton pattern to have only one instance of the game in memory at a time.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

There are a few different design constraints associated with this project. One of the technical design constraints is that we need to develop this game in a web-based environment that allows the game to function on multiple platforms. Another design constraint would be having to test the product on different operating systems to ensure it functions smoothly and displays data consistently through the different systems. Another design constraint would be to only have one instance of a game running in memory at a given time. These design constraints impact the development of the application because they limit the team to work within a web-based framework that runs data smoothly on different operating systems. More design constraints are storage for the images and how quickly to render the images. Security is another design constraint to look at.

## [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 UML class diagram below illustrates how the classes relate to each other and applies some object-oriented programming principles. By looking at the line with the unfilled arrow we can see that the Game, Team, and Player classes inherit from the Entity class. The Game, Team, and Player classes all inherit the attributes and methods from the Entity or parent class. Entity is also an example of an abstract class as it cannot be initialized or instantiated. There is also an association relationship between GameService and Game, Game and Team, and Team and Player with numerical constraints known as multiplicity. The multiplicity between these classes is zero to many. Encapsulation is also depicted in the diagram below with the private attributes as this hides and limits information. The ProgramDriver class is also depicted as using the SingletonTester class.

**"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 performs well when hosting web-based software applications. Mac’s web hosting servers are more stable because of the BSD system. Mac does offer a server-based deployment method. Mac is more expensive when looking at potential licensing costs to the client. | Linux uses some features of Unix and their web hosting servers are more stable than windows. Linux is the most cost effective in terms of licensing costs to the client. A disadvantage of Linux is that it is more difficult to use than Mac or Windows. However, Linux does have great security. Linux does offer a server-based deployment method. | Windows has less stable web hosting servers. Windows is more expensive than Linux but still relatively well priced in terms of the licensing costs to the client. An advantage of using Windows hosting is its simplicity making it easier for users to use. Windows does offer a server-based deployment method. | Mobile devices are not appropriate for hosting the server side of web-based software applications. |
| **Client Side** | The cost for software development on the Mac would be a bit higher than Windows. One advantage of using a Mac is the convenience of using a system based on Unix. Testing for cross-browser compatibility should be done for Safari and Google Chrome. | The cost considerations for Linux for software development would also be on the lower side. Linux has the advantage of being based on Unix which is good considering Unix can handle higher server loads better. Linux is the most customizable with a large variety of software. | The cost for software development on Windows would be relatively low. I would say windows is user friendly as it can run a large variety of programs. A possible text editor by Microsoft that can be used is VS Code. Testing should be done for the popular sites like Microsoft Edge and Google Chrome. | Software development on mobile devices is not ideal. Although it can be done, it is very inconvenient, and you are better off choosing from one of the other three operating systems. The cost is relatively high. When developing and testing for mobile devices, you must keep in mind the touch screen interface and different screen sizes. |
| **Development Tools** | -The IDEs VS Code and Xcode are popular on Mac.  -Mac also uses programming languages like C, C++, and Swift.  -The developers should not have a problem using these development tools. | -Linux can use C and JavaScript as programming languages.  -Some IDEs that can be used are Atom, Eclipse, and PHPStorm.  -The developers have plenty of options and the costs are kept minimal. | -VS Code is by Microsoft and is an IDE for Windows.  -Other IDEs include PyCharm and Eclipse.  -Windows can use a variety of program languages like C++, Java, and Python.  -The costs are low and there are a lot of options for the developers. | -The mobile devices would not be 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 operating platform I would recommend, that would allow The Gaming Room to expand Draw It or Lose It to other computing environments is Windows. Windows licensing costs are priced relatively well considering what they offer. Windows provides ease of use and requires low maintenance which makes it easier for the administrators. You also have a vast number of choices when it comes to programming languages and IDEs.
2. **Operating Systems Architectures**: The operating platform architecture for Windows consists of a user mode and kernel mode. Programs and subsystems are limited in user mode as they have limited access. User mode implements system APIs to access hardware or memory. Kernel mode on the other hand has full access to hardware and system resources as you can access memory management and thread prioritization. One approach I would take for the operating platform architecture/pattern is a microservices architecture to develop the web application. This would allow the services to be programmed independently and allow us to scale the function for rendering and displaying images as needed much more easily. This is because the team can update the service for rendering and displaying images without creating a second instance of the entire application.
3. **Storage Management**: The appropriate storage management system to be used with Windows is storage sense. Storage sense frees up drive space by getting rid of items that are not needed. Disk clean up can also be helpful for cleaning up temporary files and internet cache files. The best storage for the application itself would be an SSD drive because it is much faster than an HDD and acts like memory.
4. **Memory Management**: The operating platform Windows uses kernel mode to manage physical memory or random-access memory (RAM). The kernel mode memory manager allocates and deallocates memory virtually and dynamically and specifies if the allocated memory supports demand paging, data caching, and instruction execution. Windows separates memory into system and user address spaces. Windows routines used to allocate and manage memory usually start with the prefix “Mm” such as MmAllocateContiguousMemory.
5. **Distributed Systems and Networks**: Draw It or Lose It must communicate between various operating platforms through distributed software and a network that connects the devices. Distributed systems and networks provide communication and storage between a number of clients and the server for the game. We must consider the network connections in terms of latency or round-trip time (RTT/ping). A strong server network is needed to reduce connection problems. We used a REST API for scalability and communication between the clients and server.
6. **Security**: The client must also have their information secured while playing the game Draw It or Lose It. We are using REST API and securing it with authentication and authorization. Authentication verifies the user's username and password credentials. The user roles are then set into a principal object. Authorization is used by permitting users to perform certain actions based off if they have the verified roles. We can also use TLS for privacy and data security as it encrypts communication between web applications and servers. We can also secure the server by using firewalls.