

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 | 07/13/2023 | Eric Breznen | Added executive summary, requirements, design constraints, and UML diagram description |
| 2.0 | 07/27/2023 | Eric Breznen | Added evaluation for multiple platforms |
| 3.0 | 08/10/2023 | Eric Breznen | Added recommendations |

**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)

The client wishes to develop a web browser version of their game application. The web browser version of their game must enforce all rules of the game and contain objects for the player, the teams.

## Requirements

*Client requires that the game will be able to include multiple teams with multiple members in each team along with proper input validation. The game can only be ran one instance at a time.*

## [Design Constraints](#_2et92p0)

Design constraints so far are that hardware requirements are unknown at this time and incremental approval by the client and the technical manager.

## [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 objects Team, Game, and Player all inherent from the entity class and are connected to eachother. GamesService runs in conjunction with Game and the ProgramDriver uses the SingletonTester class.  
  
Encapsulation is shown in the various classes and their private attributes which keep them safe from manipulation by other classes. Game, Team, and Player all inherit from Entity showing security.

**"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** | Pricier, however strong communications with Linux clients as well as their own, though it requires a virtual machine to work well with Windows clients and with Windows dominating the market, this makes it difficult for the majority of users. | Open sourced, no licensing fees, and strong security are among the benefits of Linux servers. Most back end servers are on Linux machines making it widely available. Downsides include that it is limiting with Windows, however it works well with MacOs. | Windows has it’s ups and downs, it’s source code is secret and therefore troubleshooting issues becomes difficult and even though non-Windows clients will need virtual machines to connect it is still powerful and secure. | Top benefit of using a mobile device server is it’s connectivity through carrier satellites and towers so that a hard cable or WiFi connection is required to connect to native clients though functionality may be limited for personal computer clients as functionality on the clients part is often limited to a touch screen and limited computing power. |
| **Client Side** | On the client side, MacOS is great for it’s computing power, creative tools. The downside is the price of the devices which tends to be on the higher side. Another benefit is that it works very well with Linux servers. | Linux client services are going to be the cheapest (free) since it is an open-sourced OS. Since the OS is open sourced, it means that the user base has an active role in improving its processes making it more user friendly. Pain points include working with Windows. | Windows dominates the market which means its OS is widely available and decently affordable with little expertise needed to run. Pain points include integrating with other operating systems as well as issues with troubleshooting issues since its source code is well protected by Microsoft. | Mobile devices are readily available and provide stronger connectivity every year and are increasingly user friendly. However their processing power, while improving with each new model is not near as powerful as larger machines. Benefits do include their portability and that they are often connected to a network wirelessly through their carriers not requiring a WiFi or wired connection. |
| **Development Tools** | Using the eclipse IDE would provide powerful compiling and debugging capabilities for Java and C++, both OOP languages which would be ideal to fulfill the requirements of the program. Swift could also be used on Xcode however this would be unideal for cross platform compatibility as the Apple developed open source language still has limitations on other platforms such as Linux. | Eclipse and visual studios are both powerful IDEs that can be used on Linux for Java and C++ languages. Visual studios and eclipse are both available for all four platforms which limits the number of teams with specified knowledge and decreases the amount of polyglot programming required. | In addition to Java and C++ in Eclipse and Visual Studios, Python in an IDE such as PyCharm can also be used in Windows to effectively create the programing. For all platforms, since it is a web-based program, HTML would also be useful in designing the structure of the webpage and embedding the animations and images that are being drawn for the game. | For Android devices, using Android studios is the obvious choice for an IDE using either Java or C++. For the sake of cross platform integration including iPhone, another IDE such as Eclipse may be preferable. |

## 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**: A clear choice in server is a Linux server. A Linux server is free, which cuts costs, and its widespread compatibility allows for connectivity between a wide array of client operating systems. It is also an open-sourced server which means access to its source code is widely available for customization and optimization for a particular task or process by the company. While Windows is popular it lacks ease of compatibility with other systems such as MacOS and while Linux servers do not innately connect with Windows, it is simpler to adapt the server to connect. MacOS is also more expensive than Linux which drives up costs and both Windows OS and MacOS operating systems source code is protected, meaning that the company has no say or ability to alter the systems to better support the game.
2. **Operating Systems Architectures**: A layered architecture would provide high modularity and the ability to make updates without having to reboot the entire server. It would also allow for an extra layer of protection between the users and the hardware by making any request go through many layers of operating system processes before reaching it which can filter out attacks by unscrupulous users. At the cost of some response time, a layered architecture keeps distance while allowing multiple users to interact with the system processes.
3. **Storage Management**: Given the importance of maintaining the set of 200 images, solid state drives (multiple for the purpose of having backups should one fail) should be used. Each drive would have the images stored in various files. Rapid access memory would also need to be powerful as it is where a given game would also be centered. User information should be stored on client devices to protect their personal information.
4. **Memory Management**: Memory management needs to be modular, ideally rapid access memory would be able to create a temporary file which reads a predetermined number of randomly picked images from the library as well as the interpret public information from the user’s profile to display to the other players in the same game. The game would then execute and upon completion of a game be able to transmit statistics to the user’s profile for personal statistic tracking before deleting the temporary file freeing up space. When it comes time to pay the subscription fee the user’s card number can be transmitted to the system from the client device which also maintains other personal information that can be sent to the server when appropriate such as logging in.
5. **Distributed Systems and Networks**: Using a cloud server for Draw it or Lose it would provide high connectivity and reliability, it would also assist with connecting to multiple different platforms and client devices and can be distributed worldwide which would be highly beneficial. Utilizing a third-party cloud-based server could drive up costs, however the exchange of functionality can be worth the cost. Utilizing a cloud-based server would also reduce repair costs during an outage as it is not on the company to repair a hard server, the repairs would be on the third-party company running the cloud. Using virtual machines also adds an extra layer of protection for the clients and the company.
6. **Security**: In any system where multiple clients connect to a central server security is vital. Some aspects of protection have been touched on in previous aspects of recommendations. Another benefit of using a cloud-based server is that virtual machines can act as firewalls, which means cost, material, and creation of multiple layered firewalls is easier than if material machines needed to be used. The application would also be able to utilize other built-in security protocols in the third-party’s server. Physical security is also important, in the case of using a cloud-based system, the importance lies on the drives where images and user profile information is stored. Storing credit card numbers and personally identifiable information on the client device rather than the main server also means that any attacks on the server would not necessarily compromise user data that can be used in malicious ways, or at the very least make it so that the attacker needs to work harder to connect to each individual user’s devices to gain that data which takes longer allowing more time to catch and defeat the attack mitigating the damage that can be done in an attack. Of course, utilizing a third-party server opens up the potential for “inside access” from a compromised network once connectivity is established which cannot always be controlled by the application’s security protocols and therefore needs to be monitored.