

The Gaming Room

# **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 | 09/18/23 | Justin Arruda | Created initial design for The Gaming Room software design |
| 2.0 | 10/02/23 | Justin Arruda | Revised design for The Gaming Room software design and added evaluation. |
| 3.0 | 10/12/23 | Justin Arruda | Added ending evaluation and recommendation |

**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 Gaming Room wants to expand the possible audience for their existing Android app by providing support for multiple platforms. The Gaming Room company would like to create a web-based game that expands on their existing game they have developed for Android, Draw It or Lose It. The Gaming Room would like the web-based game to be able to be accessed across both mobile platforms.

## Requirements

The web-based game once developed must be able to accommodate both major mobile platforms, iOS and android, as well as a web-based service. For efficiency in the development process, creating the initial code in Java will help translate across all the various platforms as it has support for web, iOS and android development.

## [Design Constraints](#_2et92p0)

The program must be able to render images from a large library of stock drawings, and have multiple players compete in a timed web-based environment. The application is also intended to simply expand from their existing game, so using their game application as a template and adapting it for multiple platforms would be the most efficient process. The application must also utilize singleton pattern design structure to ensure unique players, games, and team instances exist and cannot be duplicated.

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

From the UML class diagram, we can see that an Entity class is inherited by the Game, Team, and Player classes. Depending on the request of the GameService class, it can create different Entities in the style of a Game, Team, or Player format. The ProgramDriver class is used to run the program and uses the SingletonTester class to ensure no duplication of the class exists.

**"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** | Using a Mac OS X Server could be useful for the extensive Java support and access to a large amount of server-side support, it does have a licensing fee and limited cross functionality. It will be very convenient when using other MacOS or iOS devices but will not provide the wide accessibility The Game Room is looking for. | Linux offers very affordable and lots of different distributions with server hosting features and would be a very strong candidate. Some downsides however are that there are some web application limitations with the Linux environment and could lead to future restrictions. | Windows offers a comprehensive Windows Server based OS, but it comes with an expensive licensing fee and is only more beneficial than other solutions when explicitly looking for .NET Framework support. | While Mobile Device server hosting is possible and offers the convenience of being able to deploy the web hosting service in virtually any location, it is very performance limited and would have some severe limitations. It is also not user friendly or intuitive to set up and can lead to development issues in the future. |
| **Client Side** | Thanks to the ability macOS has to use virtualization or virtual machines, it will be easy to develop cross compatible programs that can be utilized on macOS. However, designing the program natively on macOS will take extra development cost and time. | Thanks to the wide range of Linux builds available, there is a wide range of support for any number of programming languages and applications. This is useful in our scenario and will let us focus less on how to set up the project and more on a simple deployment method. | Thanks to the widespread popularity of windows OS, there are many features and lots of support for a wide range of applications. This will make development costs and time easier thanks to the large amount of support available. | Because of the performance limitations of mobile devices when compared to MacOS, Windows or Linux, development must be very careful with how much processing is done on the device locally, which will take extra development time. |
| **Development Tools** | While it takes a similar amount of time to set up an IDE for java development on macOS, it is even easier to deploy it to macOS X thanks to their close compatibility. It will be very easy to deploy and run the program on a MacOS X Server | There are a lot of useful development tools we can take advantage of when developing and deploying on a Linux based system. For example, using a version of Ubuntu will be helpful thanks to other apps that can used with Ubuntu to help with web deployment and development. | Out of all the operating systems, there is a very large amount of support and features in Windows that are useful for java development. While specifically for Java MacOS X Server is simpler to deploy and host, there are a very large number of tools and IDEs available for developing Java web-based applications. | Like the other systems, there is a wide range of support for developing and testing mobile device applications. A downside however is that like Linux there are many different versions of operating systems available and making sure the program is widely accessible is a concern to be addressed during development. |

## 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 that The Gaming Room should utilize a Linux based operating system for their server, specifically an Ubuntu server distribution for added server support and useful features. Thanks to the large community base there are lots of third-party solutions and applications that can help aid The Gaming Room in developing their program. As a final added benefit, the base version of the Ubuntu Server distribution is free so there is a very low cost involved.
2. **Operating Systems Architectures**: Since the Ubuntu distribution of Linux uses a monolithic operating system architecture, it gives The Gaming Room developers a lot of freedom and control over the server system. This will be useful for when the server is running and needs access to more resources from the computer, it will have the authority to allow it. This is possible because the server is run inside the Kernel which is directly able to control all the hardware in the computer and does not need any additional verification.
3. **Storage Management**: A popular method to managing and storing data on a Linux based system is with drive partitioning. Using this method, the server will be able to keep the data organized and easily accessible, creating different file paths with the Linux file system to clearly identify storage for things like the image files needed for The Gaming Room’s application.
4. **Memory Management**: The Linux Server can use virtual memory management to help with the multiple instances the live web service will be able to utilize. Doing this will allow the server to provide an appropriate amount of memory for each individual instance of the application. This will be important for the performance of the server and will help the users have a smoother experience when interacting with the web service.
5. **Distributed Systems and Networks**: Since the Gaming Room will be utilizing a client-server distribution architecture, it will be important to keep frequently accessed data readily available and easy to access from the clients. This also means that during the development of the project caution will need to be exercised to make sure that unnecessary data is not being stored on the client and vice versa. Keeping a separate server to frequently store backups for the server will be very beneficial as well as, if necessary, having the ability to roll the server back to a recent back-up if anything unexpected or catastrophic occurs.
6. **Security**: The most important thing to approach for security of the web application is to have good encryption. Making sure that the data being transmitted between the server and client is encrypted will be a very effective measure for security. This is especially important when handling user log-in information and information about the client devices. Making sure to use strong password requirements for the user information will also help keep the accounts more secure.