

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 | 07/15/2023 | Dylan Cavazos | Software Design document for Gaming application, 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)

The software design problem is that the game application needs to support multiple teams, where each team has a multiple number of players. The game and the team names need to be unique where the players can check the availability of the names. The game’s software should be designed so that only one instance of the game can exist in memory at any time. A proposed solution to this problem is to ensure that the application allows multiple teams each with their own unique name. The application will also check to make sure that only one instance of a name can exist at a time to prevent naming conflicts. To make sure that only one instance of the game will exist in memory at a time, the application will assign unique identifiers to each instance of the game, team, and players.

Critical information that the client must know is that they should emphasize the importance of unique names to the players to prevent conflicts. The client should also be aware that the unique identifiers will also prevent the use of multiple instances of a game at a single time.

## Requirements

Some of the business requirements for the gaming application include the use of multi-team support, team-based gameplay, and unique game and team names. The technical requirements focus solely on the importance of unique identifiers, so that only one instance of a game or team can exist in memory at a given time.

## [Design Constraints](#_2et92p0)

In a web-based distributed environment, typical design constraints for a game application include the use of scalability; so that the game can handle multiple users, and games in a single session. Another design constraint includes the ability for proper networking, as the game supports real time team-based gameplay, multiple users will need to access the same session of a game. For online gaming applications, security is also a required constraint as making sure users are properly authenticated when entering a session so that secure personal data is not compromised.

The implications of design constraints on application development include the game’s architecture design, so that the application can handle things like scalability and networking efficiently. The design should also support error handling, that help to prevent things like network failures and synchronization issues. Communication in real-time is also an important implication, as the application will need to support the ability for players to communicate or interact simultaneously throughout a single session.

## [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 Gaming Room UML diagram shows the class structure and relationships between the different classes. Each class has their respective methods and attributes that contribute to the overall design of the gaming application. The Game, Team and Player class all inherit from the Entity class. While the GameService class creates an instance of the Game class, and the Game class creates an instance of the Team class, and Team class creates an instance of the Player class.

The Entity class is the base class for the other classes in the diagram, so that they may inherit the public variables for use in the application. The GameService class utilizes the Singleton design so that it only creates one instance in the game.

The object-oriented principles that are exemplified in this application include inheritance, encapsulation, association, and the singleton design pattern. The inheritance principle is illustrated by the fact that the Game, Team, and Player classes all inherit the attributes and behaviors of the Entity class. The encapsulation principle is illustrated using each class having both public and private attributes. Association is portrayed when the GameService class associates with the Game, Team, and Player classes as can interact with each of them. The Singleton design pattern is used by the GameService class to ensure that only one instance of the game exists at a single time.

**"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 utilizes a Unix-based system that provides a stable environment for hosting web applications. Mac is also developer friendly because Unix offers a CLI and built-in development tools. Some of the advantages of using Mac include its security, user experience, and reliability. Its weaknesses however are that Mac only supports Apple’s proprietary hardware. Also, Mac tends to be more expensive than their Windows or Linux counterparts.  MacOS web-based servers used to be utilized but are no longer recommended for use in web-based applications that host thousands of players. | Characteristics for the Linux operating system are that it’s open source, which allows users to freely customize the system based on their needs. Linux is also known for its stability because it can handle heavy workloads. It is also very secure because it’s open source, so the software is openly scrutinized for vulnerabilities.  The advantages of Linux are its cost-effectiveness since it is free to use. It’s also customizable compared to other operating systems. Some of the weaknesses of Linux is that it can be difficult to learn at first since it utilizes a command-line interface. Linux web-based servers are the most cost effective choice due to their open-sourced software, there might be some subscription costs associated but platforms like Apache or Ubuntu are good choices for a Linux based platform. | Windows is perhaps the most widely used operating system. Partly due to its user-friendly interface. Windows is also compatible with most devices and applications. Some of the advantages of Windows include its variety of development tools like Visual Studio. Windows is also used by most companies, making it the most popular and beginner friendly operating system. Some of the weaknesses associated with Windows are its security vulnerabilities. Being the most popular OS also means it’s heavily targeted by attackers, requiring constant updates. When considering the licensing costs associated with hosting a web-based application using a Windows server the licensing costs tend to be more costly than other platforms in comparison. It also depends on the edition of the Windows server being used | Mobile devices are both compact and portable, which allows users to access web-based applications on the go. Some of the advantages of mobile devices include the fact that it has the broadest user-base apart from other OS. This allows hosting a web-based application to reach a larger audience. The app store is also unique because it has the potential to host a web-based app for distribution and use. The weaknesses with mobile devices, however, are their limited screen size and performance. Due to its small size, the performance tends to be lesser than that of its computer OS counterparts. They also rely on a network connection which can be difficult to maintain depending on the situation. |
| **Client Side** | Mac hardware tends to be more expensive compared to other systems. Macs also require the use of primarily Apple software, which will need to be taken into consideration. The Mac development environment utilizes its own tools and frameworks which will need to be learned by the developers working on the system. This will impact any time constraints necessary for development production. The expertise involved will require that teams be familiar with the tools and frameworks supported by Mac and ensure that development is coherent across multiple platforms. | Compared to other operating systems the cost associated with Linux programs and hardware is less, because it’s open source. But there may be specific tools or IDEs that require costs associated with Linux. The time associated with using Linux may be increased since developers need to ensure that Linux is compatible with software or programs which requires manual configuration sometimes. The expertise with Linux will need to be addressed since it uses a CLI, so developers will need to be familiar with the language when building applications. | When supporting multiple types of clients using Windows, the cost associated needs to be considered. Licensing can be a major factor, IDEs, third-party software, and libraries may require purchase for use. Windows also requires the use of different types of hardware like servers, workstations and any other devices required. The time required also needs to be considered as supporting multiple clients on Windows will require the users to ensure compatibility across interfaces. Clients will also need to develop expertise in Windows for things like APIs, frameworks, and environments. | Using Mobile devices when supporting multiple types of clients bears its own associated costs, time, expertise. The costs associated can depend on the various manufacturers, models, and operating systems. This can lead to increased costs for the large variety of mobile devices that are on the market. The time for development may be increased due to a variety of factors; such as, the user interface used, performance, and compatibility with each device. The expertise required will need to be fluent in different platforms like iOS, and Android. Mobile devices also use their own frameworks like Swift and React, that developers will need to be familiar with. |
| **Development Tools** | When building the game application on Mac, programming languages like Java are used which uses familiar IDEs like Eclipse or IntelliJ IDEA. Tools commonly used are Maven, and Git. Git helps promote collaboration between developers, while Maven is a build automation tool.  When comparing the costs associated with each IDE, Eclipse is the cheapest because it’s open source, so it doesn’t require a subscription for use. There are some plugins that could be added which may require a fee for use. In contrast, IntelliJ IDEA does not offer a community version so a license will be required for use. | The gaming application utilizes the Java programming language, which is supported by Linux using different IDEs. Such as IntelliJ IDEA, and Eclipse. Required tools may include the terminal which uses the CLI on Linux based systems. Git, and Maven are also used for software development. As previously mentioned, Eclipse offers a community free edition for use and IntelliJ is subscription based. However, for a Linux system, Maven can be used for development for free since it operates under an open-source license through Apache. | Building a gaming application in Windows will require the use of programming languages such as the one used in this application, specifically Java. Different IDEs windows support include Microsoft Visual Studio, and Eclipse. Windows also utilizes tools like the command prompt. Git and Maven will also be used in support of software development. When considering the licensing costs associated with the use of IDEs like Visual Studio, there is either a community edition which is free or Visual Studio editions that require a subscription for access but provide better tools and benefits | For the gaming application outlined in this design template, Java is the primary programming language. Mobile devices, however, support the use of Kotlin as a language for android app development. Some of the IDEs associated include Android Studio, and Eclipse which also supports android development. Tools include the android software development kit (SDK). When it comes to costs associated with the SDK for mobile applications like Android, there exists an Android SDK licensed by Google which is free for use. There is also Android Studio which is also freely available for 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**:

A web-based platform would be most ideal for the gaming application ‘Draw It or Lose It’. Web-based platforms provide the benefit of cross-platform compatibility, including different OS like Windows, macOS, and Linux. A Web-based platform also provides the benefit of being easily deployable and providing updates when necessary. The Web-based platform also reaches a large audience since most users are familiar with web-based applications through Windows and other OS. Taking into consideration the dependencies required for the gaming application, Linux would be an ideal operating platform to host a gaming server using software like Apache. Linux offers the benefit of being open-sourced software which provides the benefit of being more secure and compatible with most software applications. This makes managing the gaming applications’ server easier in terms of support and development.

1. **Operating Systems Architectures**:

Since Linux is the recommended operating system to host the gaming application so when understanding the operating system, it’s also important to understand the system architecture. Linux supports both x32 and x64 bit architecture with x64 being more popular for most systems now. Its architecture also supports the use of multiuser and multitasking, which enables multiple users to use the platform simultaneously while also having the system run applications concurrently as well. This is an advantage for a system that’s required to host a gaming application that will have multiple users all required enough resources to run the game application properly. Linux also uses what’s called a Monolithic Kernel, which is the layer between the shell and the system’s hardware that includes all the necessary software for the system to run.

1. **Storage Management**:

When considering Linux as the primary operating system to host the game application a storage management system is a necessary component. Theres a plethora to choose from, especially when Linux is the primary platform, as Linux supports a multitude of file systems. A recommended storage management system using Linux for a gaming application is the Ext4 file system or (Fourth Extended Filesystem). Ext4 file system is especially good for hosting gaming applications as it’s the highest performing file system compared to Ext3 and Ext2. Ext4 also has the capability of supporting large volumes of data, like all the files associated with a game application. Plugins, textures, sound bites, sprites, etc. Ext4 also supports the use of journaling, which helps prevent the loss of data in the event of corruption or loss. Since Ext4 is popular in Linux systems, it’s guaranteed to be compatible with most programs and support, making it ideal to host a gaming application.

1. **Memory Management**:

Linux is the ideal operating platform to host a gaming application like Draw it or Lose it, especially because of the memory management systems it uses. Linux utilizes a virtual memory system that allows each process to have its own virtual address space. Since Linux uses a kernel, the kernel maps the virtual addresses to physical addresses in RAM. Which supports the use of memory allocation in terms of performance. Linux also supports memory protection by separating each process, making them unable to access the memory of other processes. It also utilizes paging and caching, which allows the system to manage the memory in fix-sized pages, so if the RAM becomes full, pages can be moved to disk to free up space. Caching helps to improve the speed at which data is being accessed, so when memory is being accessed by a program the kernel can access the data much faster than without it. In terms of the gaming application, Linux is ideal for memory management because it uses specific techniques that make memory storage and retrieval optimal and more secure in contrast to other platforms.

1. **Distributed Systems and Networks**:

To take into consideration the distributed software and network to connect the devices, some of the dependencies between the components include a multitude of aspects. Some of these considerations include emphasizing the importance of building APIs that support a standardized interface that supports communication between platforms. Internet connectivity is also vital so that each platform can communicate and maintain connection. For outages, error handling and circuit break patterns are recommended as they help to address potential problems regarding any errors that may occur or temporary network outages that could damage a system.

1. **Security**:

To ensure that security is a priority for the ‘Draw It or Lose It’ game, several methods can be implemented between various platforms. For instance, some secure communications methods like implementing TLS or Transport Layer Security can help to encrypt data when being transmitted over the network. Certificate management can also be utilized so that a server’s identity can be properly authenticated by different devices.

User authentication is also a ubiquitous security feature as it requires the user to verify his/her or identity before attempting to login to an application. This prevents unwarranted access from potential threat actors. Access control is also another method of ensuring security across platforms as it also inhibits a user’s ability to access facets of a program’s sensitive resources or functionalities.

Password security will also be one of the most efficient methods of maintaining security for user protection across platforms as each user will implement a strong password to prevent unauthorized access to their account.

Data encryption is also another helpful method as a user’s sensitive data can be protected while it is stored in a database system.