

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 | 07/16/2023 | Daniel Escobedo | Access which Operating System to use for development. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wishes to create a web-based game that can operate on a variety of platforms. The game will be called "Draw It or Lose It" and will be accessible solely on Android for the time being. The goal of this game is for numerous teams of several people to go four rounds at a minute each. When a photo is chosen from a library, one team guesses till time runs out. If no one responds, each opposing team member has 15 seconds to respond.

## Requirements

The Gaming Room should work on all devices. This implies we have it on Android but need to incorporate it into another mobile device. Along with operating systems such as Windows, Linux, and Apple.

## [Design Constraints](#_2et92p0)

* Must be able to operate on many platforms.
* Game and team names must be distinctive for users to determine if the name is in use or available.
* One or more teams must be involved.
* There are several members on each team.

## [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)

Entity establishes a connection between the Game, Team, and Player classes. This implies that they all inherit or get information from the Entity. Entity becomes a superclass. When we look at their relationship, we can observe that Team and Player are of the "has a" variety. Game has a Team, whereas GameService has Games. GameService has a reference to Games, Games has a reference to Tea, and Team has a reference to Player.

**"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 must 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** | Characteristics: Unix-based operating system  Advantages: Stable, secure, and easy to use  Weaknesses: Not as widely supported as Windows or Linux  Server-based Deployment: Yes  Licensing Costs: Free for personal use, but paid for commercial use | Characteristics:  Unix-based operating system  Advantages: Stable, secure, and customizable  Weaknesses: Can be complex to set up and manage  Server-based Deployment: Yes  Licensing Costs: Free for personal use, but paid for commercial use | Characteristics: Most widely used operating system  Advantages: Easy to use and supported by a wide range of software.  Weaknesses: Not as secure as Linux or Mac.  Server-based Deployment: Yes  Licensing Costs: Paid | Characteristics: Designed for mobile devices.  Advantages: Portable and easy to use.  Weaknesses: Not as powerful as desktop operating systems.  Server-based Deployment: Yes, but limited to cloud-based hosting.  Licensing Costs: Varies by device and carrier. |
| **Client Side** | Cost: Medium  Time: Medium  Expertise: Medium  Considerations: Standard web technologies, responsive design, cross-browser testing. HTML, CSS, and JavaScript | Cost: Free  Time: Low  Expertise: Low  Considerations: Standard web technologies, responsive design, cross-browser testing. HTML, CSS, and JavaScript | Cost: Medium  Time: Medium  Expertise: Medium  Considerations: Standard web technologies, responsive design, cross-browser testing. HTML, CSS, and JavaScript | Cost: High  Time: High  Expertise: High  Considerations: Standard web technologies, responsive design, cross-browser testing, mobile-specific features. HTML, CSS, and JavaScript |
| **Development Tools** | MacOS is home to its proprietary high level programming language called swift.  Programming Languages: Swift, Objective-C  Tools: Xcode, VScode. There are licensing costs associated with some of the development tools that are used to build this type of software.  Multiple development teams required: Yes | Linux is versatile and can run programs such as eclipse and visual studio.  Programming Languages: Java, Python, C/C++  Tools: Eclipse, IntelliJ IDEA. There are licensing costs associated with some of the development tools that are used to build this type of software.  Multiple development teams required: Varies | This operating system is by far the OS with the most compatible IDE’s.  Programming Languages: C#, Visual Basic, Java  Tools: Visual Studio, Eclipse, VScode. There are licensing costs associated with some of the development tools that are used to build this type of software.  Multiple development teams required: Yes | Programming Languages: Java, Kotlin, Swift, Objective-C  Tools: Android Studio for android and Xcode for iOS. There are licensing costs associated with some of the development tools that are used to build this type of software.  Multiple development teams required: Yes |

## 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 recommend that The Gaming Room uses Windows as their operating platform for Draw It or Lose It. Windows is a powerful and versatile operating system that is well-suited for gaming applications. It offers a wide range of features and capabilities that are essential for running a successful gaming service, including:

* A high-performance kernel that can handle the demands of multiple concurrent users and games.
* A robust security infrastructure that protects user data and prevents unauthorized access.
* A comprehensive set of tools and utilities for managing and monitoring gaming servers.
* Support for a wide range of hardware and software platforms

1. **Operating Systems Architectures**: Windows uses a layered architecture, which means that the operating system is divided into a number of layers, each of which provides a specific set of services. This architecture makes Windows modular and scalable, and it allows The Gaming Room to easily add or remove features as needed. Windows offers services that are utilized by all Windows-based applications. These services allow applications to display a Graphical User Interface (GUI) while accessing system resources. Graphics and multimedia, communications, and online services are additional examples of applications. These services can be accessed via a user account or a server.
2. **Storage Management**: The Gaming Room should use a cloud-based storage solution for storing their Draw It or Lose It game data. A cloud-based storage solution will allow them to store their data on remote servers, which will make it accessible from anywhere. An alternative option would be if we use Windows Storage Management, which is a built-in storage management tool that is available in Windows Server and Windows 10. WSM provides a graphical user interface (GUI) for managing storage devices, volumes, and file shares.
3. **Memory Management**: The Gaming Room should use a memory management system that is designed for cross-platform applications. This will ensure that the game can run smoothly on all platforms. Windows can use paging which will allow Draw It or Lose It to access memory that is not currently in the physical memory of the system. When Draw It or Lose It needs to access memory that is not currently in physical memory, it will page this memory into physical memory.
4. **Distributed Systems and Networks**: Draw It or Lose It is a distributed application, which means that it runs on multiple computers that are connected to a network. This allows users to play the game from anywhere in the world. The network that connects the computers to a distribution system must be reliable and secure. It must also be able to handle the traffic generated by the game. We can use remote procedure calls which are a way for two processes running on different machines to communicate with each other. They allow one process to call a function on another process as if the function was running on the same machine. This can be used to make Draw It or Lose It communicate between different platforms.
5. **Security**: A security solution that includes authentication, authorization, and encryption is a good choice for The Gaming Room because it will protect user data. Authentication ensures that only authorized users can access the game, authorization determines what actions authorized users can take, and encryption protects user data from unauthorized access. We can also incorporate controls such as user account controls which help to prevent unauthorized changes to your system by requiring users to confirm certain actions before they can be performed. We can also use BitLocker drive encryption which can be used to encrypt the hard drive on your device, making it more difficult for unauthorized users to access your data. Lastly, we can use antivirus software to do consecutive scans within the OS to check for any malware or spyware.