

“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> | Brady Goodwin | Detailed Outline of “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 Gaming Room wants to develop a web-based game that is available on multiple platforms, based on their current from their Android application, “Draw It or Lose It”.

## Requirements

* *A game will have the ability to have one or more teams involved.*
* *Each team will have multiple players assigned to it.*
* *Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.*
* *Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.*

## [Design Constraints](#_2et92p0)

1. *Data Storage – The application will need to have access to a database that can store teams and players during the game.*
2. *Compatibility – The application will need to be accessible, and functional from multiple browsers, and maintain a consistent behavior and appearance.*
3. *Design – The application will have to be able to fit multiple screen sizes in various aspect ratios on phones, desktops, tablets, etc.*
4. *Connection – Network speed must be accounted for, as lower speed internet users may have a more difficult time connecting and actively playing with others.*
5. *Budget – The project’s budget will dictate the allocation of resources, limit features and overall scope of the project.*

## [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)

*Game, Team and Player are child classes of the parent class, Entity. This means that Game, Team and Player are inherited from Entity, shown by the open arrow towards the parent class. Since Game, Team and Player all have the attributes “id” and “name”, that makes Entity a super class entity.*

*The classes Team and Player are associated. Team has Players, indicated by the line showing “0…”, or zero to many. GameService and Game have the same relationship, as well as Team and Game. GameService has Game, indicated by the line showing zero-to-many, and Game has Team, indicated by the line. This means that the classes GameService has a reference of Game, Game has a reference of Team, and Team has a reference of Player. Each GameService can have many games, each Game can have many teams, and each Team can have many players.*

**"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** | Mac has flexible terminal commands to configure the server, access, or make changes.  Although difficult in newer devices, they are upgradeable, and have various options for different web hosting requirements. Overall, it is less preferred for web hosting services | Flexible terminal commands, plus more cost friendly. Linux has a more highly preferred security. Any security flaws are caught before they become an issue, and it is the most preferred choice for web hosting services. But it is more difficult to find applications to support the web hosting required needs. | Windows has much more software available than compared to other OS. It is dominant over other platforms. Windows has high resource requirements, less loading time, and is user friendly. Window is a higher virus susceptibility, and high licensing cost. | Having a mobile server is not a great option. It’s better to have a stable server in a single location. Devices differ and specifications differ per device. They are in very high supply, highly portable, and can be very cost-effective. These devices can be highly selective about compatibility between various smart devices and have poor security comparative to other platforms. |
| **Client Side** | The intricacies of Mac-specific features and UI may increase development time. Expertise in macOS is crucial as developers need to understand Apple’s guidelines to optimize the user experience across a wide range of devices. | Will need to adapt software to various Linux distributions and ensuring compatibility. Increase in time and cost required for development for as developers need to account and address compatibility issues across Linux distributions. A Linux developer must be well versed in package management systems, libraries, and kernel variations. | The windows ecosystem is very diverse with many variations in user interfaces. A developer needs to be mostly familiar with navigating windows specific APIs to ensure optimal performance across windows devices since windows is mostly user-friendly. | Development costs are influenced by creating flexible, responsive designs for mobile devices. The time required for development lies in optimizing the application to be compatible with different browsers. Expertise in web-development, specifically with a responsive design and mobile-friendly coding practice, are critical for development of a web-based software application through a mobile device. |
| **Development Tools** | Swift is the most common option when it comes to running programming languages on Mac computers. Nevertheless, Macs support all languages. Languages include, but are not limited to, HTML, CSS, and JavaScript, along with supporting libraries for general-purpose and front-end languages. These include Ruby, PHP, Python, and Java. The simple IDEs like Eclipse, Basic Visual Studio, and Xcode are all free to use, but additional IDEs may bring in more licensing costs. | Linux can work with basic development tools like Visual Studio and Eclipse, alongside more languages and tools. Like the other platforms, these languages include, but are not limited to, HTML, CSS, and JavaScript, along with supporting libraries for general-purpose and front-end languages such as Ruby, PHP, Python, Java, etc. Multiple teams may be required for fostering compatibility between different kernel variations.  For basic programs such as Eclipse and Basic Visual Studio, there are no licensing costs, but more advanced IDEs can incur additional costs. | Windows is more user-friendly than Linux, but equally capable, also supporting tools like Visual Studio and Eclipse. Once again, these languages include but are not limited to HTML, CSS, JavaScript, and other supporting libraries for general-purpose and front-end languages, including Ruby, PHP, Python, Java etc. Other than the licensing cost of Windows, most IDEs do not require a license. But, to use additional features or use more advanced environments, Paid licenses are available. | Using Android and Swift (Apple), you can make numerous apps. On all three machines, the software and languages are compatible. These languages include but are not limited to, HTML, CSS, JavaScript, and other supporting libraries for general-purpose and front-end languages. There is support for Visual Studio and Eclipse, as well as Android Studio and Xcode for free on mobile devices, and other specific environments that require a license fee. |

## 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**: My recommendation for an operating platform would be to use a Windows server to expand Draw It or Lose It to multiple platforms. Windows is the optimal choice for development as it can integrate with current Android build and cross-platform applications like React, Flutter, or Xamarin. The Microsoft suite already has many development tools for deployment on Windows PCs, with plenty of emulation software to test compatibility on any platform.
2. **Operating Systems Architectures**: Windows uses dual-mode operation, User mode, and Kernel Mode. User mode is where user applications are run, and Kernel Mode is used for OS operations, such as memory management, routines, and security. Using dual-mode operation adds a layer of protection to prevent unauthorized access by user applications and protects the system’s stability from user application crashes or errors. Windows systems also support scalability, and hardware can be customized to support the project as it gets bigger.
3. **Storage Management**: My recommendation for storage management would be cloud storage, specifically through Microsoft Azure. It works with Microsoft systems such as Windows servers or Active Directory, allowing seamless integration. Also, it will allow for scalability in the future by enabling the organization to scale its needs up or down, depending on demand. Azure also has data locations across the globe, which allows for lower latency worldwide. Azure has a strong security system using encryption and identity management. Azure is flexible with pricing, even with a pay-as-you-go, so only the space used is paid for, which is an excellent option for an expanding organization.
4. **Memory Management**: Although Windows 11 has been published, I recommend using Windows 10, specifically because Windows 11 has a prevalent issue with memory leaks that cause excessive memory use on a system. Windows 10 still employs virtual memory, which lets processes use more memory than is physically available but instead uses RAM and disk space to simulate additional memory. Each process is also given its own address space, preventing processes from accessing each other’s memory, which increases security and stability. Another feature of Windows 10’s memory management is process priority and resource allocation, which uses a priority-based system to consider the priority of each process and allocate more memory to higher-priority applications and processes. Windows also supports large amounts of RAM, up to 2TB, on Windows 10 Pro and Enterprise editions—the more memory available to processes, the higher the performance.
5. **Distributed Systems and Networks**: We recommend using Microsoft Azure as a cloud service provider because of its ease of access when handling distributed systems and networks. Azure boasts maximum uptime, email alerts, and their monitoring service, Azure Monitor. A monitoring service will be essential when there are large numbers of concurrent games with multiple players connected to each one. Azure has a global presence with data centers across the globe, which allows the distributed system and network to be deployed closer to users, improving the user experience. Since Azure handles the network load, it will enable more focus on the application and features.
6. **Security**: As stated earlier, Windows 10 and Microsoft Azure use Data encryption and secure authentication. Windows uses multi-factor authentication, and Azure uses identity management to prevent unauthorized data access. Microsoft Azure also uses IP configurations, allowing whitelisting of specific access to data, and gives the option of storage in a VPN within the cloud to add an extra layer of security. There are also options to hide user data to protect the information in the case of a breach.