

<Draw It or Lose It>

# **CS 230 Project Software Design Template**

Version 1.1

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.1 | 12/12/2021 | Samantha Bell | Final 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 Gaming Room is looking to expand their hit Android game *Draw It or Lose It* to a web-based version. This game is played using at least one team, each team is made up of multiple players. Games and teams will need to be set up using unique identifiers so that a user can check if a team name is already in use and to maintain continuity of the current game and avoid duplication.

## [Design Constraints](#_2et92p0)

* Programming language compatibility across devices and operating systems
* Memory and data usage efficiency for portability
* Ability to create one or more team – computer generated team to play against solo team
* Ability to create and store multiple players and assign to team
* Lookup for game/team/player id with validation to ensure uniqueness

## [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 application has an Entity base class with encapsulated data, a private constructor, and accessor and mutators. The Game, Team, and Player classes inherit from the Entity class. The Player class creates a player. The Team class creates a team, and associates with the Player class to add players to the team. The Game class creates a game and associates with the Team class to add teams to the game. The GameService class associates with the Game class to create instances of the game and store them in an array. This class allows a user to lookup a game instance, as well as check how many instances are stored in the array. A ProgramDriver class contains the main() method and uses a SingletonTester class to check that the Singleton design pattern is working to prevent duplication of instances in memory at the same 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 servers are secure and reliable. Mac only operates on Apple products, so compatibility is an issue. | Linux is an open-source server, which makes it highly secure, but is only a kernel operating system, requiring additional software to round it out. It is known for stability but is not familiar to many users comparatively. | Windows is familiar and popular among users. It also is known for compatibility. Security may be an issue with Windows, as well as speed. | Mobile device servers are open source, there are multiple options of operating systems. Due to the nature of mobile devices, storage space is limited, and may affect speed. |
| **Client Side** | Mac is the most expensive operating system choice. This OS requires some familiarity and expertise with Mac. A moderate amount of time will be required for Mac development. | Linux cost is free, as it is open source. This option will require expertise as it is more specialized and less common. Linux may require additional time. | Windows has a moderate cost to consider. Ease of use is a plus with Windows, as it is the most familiar among developers and users. Windows development will not require additional time. | The cost for developing applications for mobile devices can vary. This option will need the most expertise and could be the most time consuming. |
| **Development Tools** | Relevant languages to consider are JavaScript, CSS, Java, and HTML. The most popular IDE for iOS is Xcode. Homebrew and iTerm2 are tools that can aid development. | Relevant languages to consider are Python, C++, and Ruby. IDEs to consider include Visual Studio and Eclipse. Software development tools to consider include Seamonkey and Quanta. | Relevant languages to consider are C#, C++, Python, Java, JavaScript, and Swift. Visual Studio is the leading IDE for Windows. Other resources for software development are GitHub and Stack Overflow. | Android Studio and XCode are the leading mobile device IDEs. React Native is an open-source development tool for developing mobile applications for both Android and iOS systems. |

## 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**: It is recommended that Draw It or Lose It be developed using Windows Server 2022 to expand its reach. Windows meets the clients needs in cost efficiency, usability, Memory and Storage Management, and Security.
2. **Operating Systems Architectures**: The Windows server platform uses Windows Containers to modulate applications and make web applications that can be distributed across multiple interfaces. Security has been bolstered in the recent release of the 2022 server making it a quality choice.
3. **Storage Management**: Storage Migration Service included in Windows Server 2022 allows for ease of migration from other servers and networks. Compression allows files to be compressed as they are transferred over the network, saving time and valuable storage space easily. Windows uses a file system for convenient storage management.
4. **Memory Management**: Windows uses paging and virtualization. These two memory management structures combined allow for data to be stored and retrieved from a storage space quickly and easily. Virtualization keeps memory management organized and running smoothly.
5. **Distributed Systems and Networks**: Windows uses an RPC approach to distributed systems. Stubs are compiled and linked without containing the actual code. This allows for communication between multiple nodes over a cloud-like system.
6. **Security**: Windows Server 2022 has just been released with upgraded security. They use a multi-layered approach by verifying that hardware partners meet their security standards, IT and SecOps teams to monitor security, and state of the art encryption with server message block protocol. Users can be assured that through API and Windows security, their information is protected.