

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | <03/19/23> | Corvinna Curtis | Initial information related to the software design recommendations for the Gaming Room Software |
| 1.1 | 04/01/23 | Corvinna Curtis | Information added in the evaluation section |
| 1.2 | 04/16/2023 | Corvinna Curtis | Information added to the Recommendations section |

**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 servers multiple platforms based on their current game, Draw it or Lose It, which is currently only available on the Android app only.

The Gaming Room needs assistance in setting up the environment for the game where they will have the ability to have one more team with multiple players. All the teams and players must have unique names, and there must be a way to check if a name is already used.

All parts of the application will need to be implemented in a way that is consistent with making a profitable application.

## Requirements

* A game will have the ability to have one more team involved.
* Each team will have multiple players assigned to it.
* Game and team names much 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)

* Game must run on multiple platforms.
* Multiple teams with multiple people
* Check that each team and each user have unique names.
* Only one instance of the game is allowed at a time.

Some of the other considerations to account for while working on this project are making sure to stay within the time frame and budget set. The client is relying on a working and profitable application in a reasonable amount of time. We will have to develop a game based on their current game, Draw It or Lose it.

## [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 UML Diagram describes the principles of Object-Oriented Programming, which we will be using during this project. During this project we will be using 7 classes, Game, Team, Player, GameServices, SingletonTester, ProgramDriver, and Entity to implement the game design. Through the Entity class, the team, game, and player classes will inherit directly from the entity class. Team, game, and player reference each other, which is done through aggregation. We will also include a SingletonTester and a ProgramDriver class. The ProgramDriver is where we will access all the classes we created, and where we will execute the main method for the program. The SingletonTester is set up and allows the project to run according to the design constraints. The SingletonTester will allow for more than one team with more than one player, while having only one game session.

**"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** | Can be expensive since you would have to buy the Mac products with the licensing and web hosting services, has an excellent GUI allowing for ease of use for development. Mac has a pliable workstation for easy accessibility and sever configuration.  Does Offer server-based environments, which does offer some major advantages for mac clients on the network – generally has easy administration and a great graphic interface. However still can be expensive to maintain. | Cost effective, however, can be difficult to navigate. Linus has a command shell for a simple server configuration. Less prone to cyber threats. Since Linux is an open source, this makes it fully customizable and the more preferred choices. It can be harder to find applications and drivers that support the hosting needs. Still cost effective and most desired with open source, however it has a huge learning curve and little/lack of support. | More costly, however a more user-friendly platform. Available software options are tremendous. However, more susceptible to getting viruses, more updates, can had an expensive start up with up front costs. Most well-rounded OS. Licensing can be expensive. Can support many applications and many third parties and fully supported. | The inexpensive of the bunch. Mobiles devices varies from user to user. However, there are many challenges that arise when creating games that are compatible with most mobile platforms. Hardware is not upgradeable, little support. Oracle is one of the companies that offer mobile server-side implementation. The advantage of Oracle is the iOS dev tools support android and the synchronization NoSQL. |
| **Client Side** | More labor intensive. Need someone who knows the OS and who could ensure the coding is compatible.  High expertise is required to develop software. There are monthly charges. A lot of time is required to access the software. | Time needed to support is extensive, would also need someone who knows/expert. Linux is the hardest to set up. Open sources still make it the most cost effective, however since it is free there is a lack of debugging, and lack of tech support. While the most desired, it does require a high level of expertise for a few applications. | Should take less time since it is a more widely used system, and has more support. There are many applications that share compatibility which helps streamline this process. However, inconsistent functionality and off brand makers, malware, and ransomware poses and issue. Does require a higher level of knowledge since it has a higher-level resource requirement. Less loading time. | Most cost effective and user-friendly. Takes less time. Needed to support diverse and multiple mobiles devices. Disadvantage between Android and iPhone since they used their own languages, this could take longer and require more expertise. More common, takes less time to load a page. Does have more technical support for the clients. |
| **Development Tools** | More common programming languages are JavaScript, CSS, Python, HTML, and more. Some of the tools on the MAC include PyCharm, eclipse, visual studios, notepad++ | Linux can work similar to an IDE, however all the software is not supported.  PHP programming language. | Windows is the easiest to use, can run multiple IDEs. It is also possible to run both Linux and windows at the same time. Some of the languages: Ruby, Java, python, Java, C++, JavaScript, HTML, and many more. Some of the Development tools visual studio, eclipse, command prompt, | Some of the more common languages, Java, python, C++. The development tools included, GitHub, Visual studio, command prompt, PowerShell. Nodejs, Android studio, Android programming, Oracle. |

## 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**:

To expand the Gaming Room’s Draw It or Lose It application, I would recommend to the start the project with the Windows OS. Out of the four options for the beginning operation systems, Windows provides users with access to a variety of software, a wide range of integrated development environments for writing and developing code, and software engineering abilities for developers. There are license costs to consider, however the Windows platform is more flexible.

1. **Operating Systems Architectures**:

The Windows OS has a layered design that consists of two primary components, the user mode, and the kernel mode. Windows has servers and applications including file management, graphical user interface (GUI), command prompt, and PowerShell.

1. **Storage Management**:

Windows can handle storage through a cloud storage solution, or through the ability to modify through the library quickly and easily without using physical hardware. The windows server allows for memory management that allows you to manage your files on your hard drive with how much space you have.

In addition, I would suggest Database management. This will work effectively and efficiently with Windows. Due to its high compatibility, the storage system is suitable for Windows. It is easy and simple use and run on multiple operating platforms.

1. **Memory Management**:

To accommodate the use from the game Draw It or Lose It, On the Windows OS you have two options for memory management.

You can select from a physical or a virtual memory.

I would recommend virtual memory because virtual memory allows for larger program handling, memory protection, and virtual memory can be extended. Windows also applies a memory compression technique that will increase the responsiveness of the operating system.

1. **Distributed Systems and Networks**:

A distributed system is a system that operates over multiple machines. Multiple machines should be able to handle the request from different platforms.

When operating this distributed system over multiple networks, we should be able to deliver a quality product that can handle high traffic and the give application with more uptime.

I do recommend the Unity game engine because it is more cost effective and supports Windows, Android, and iOS.

1. **Security**:

Windows has several safety features, including windows defender, which also includes and VPN that will add a layer of protection for the user. Windows also offers user account control settings to help protect what data is coming and going. There are features for memory allocation, virtual address that will stay private for processes and cannot be accessed by other processes unless shared. Windows also comes with anti-spyware built in, however other programs such as McFee or Norton can be purchased for an additional layer of protection.