

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 | 03/19/2023 | Martin Richardson | Showing the company how to develop the web-based version of the game. |

**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 company The Gaming Room has asked to help them design a game called Draw it or Lose it. They want our help to help them making a web-based version of their game that allows for one or more teams to compete in this game. They also want our help in making sure that there are multiple people assigned to every team and with that they want everyone to have a unique name so we have to make sure when people are signing up for the game each individual has a different name. Finally they only want one game to exist in an instance at a time so we will help with that as well.

## Requirements

* One or more teams involved in each game.
* Each team will have multiple players on it.
* The game name and player’s names all have to be unique and different. (No same names anywhere)
* The game must only have one instance of existence at any given time.

## [Design Constraints](#_2et92p0)

* If they want the game to be a web-based version of the game they will have to use a web-based IDE like Java or python are the two easiest to achieve this.
* In order for the game to have multiple teams and players on each team the program has to be written with player and team objects to support this function.
* The company want everyone to have and each game to have different names so we would use a singleton pattern here to achieve this function. Allowing the uniqueness to everyone’s name and game.
* Finally the company only wants one game instance in existence at a time so each team, player, and game would get its own unique identifier which we can add in with the singleton pattern.

## [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)

This UML diagram shows the entire project we are working on and how the actual project should be set up and written. The Entity class is the superclass in this project and the Game, Team, and Player class all inherit once from this class allowing them to only write once from this superclass. The Game, Team, and Player classes all have a zero to many inheritance from the GameService class allowing the game to draw any number of players sorting them into any number of teams and putting them into any number of games. Then finally the Program driver class actually drives the class and uses/inherits from the singleton tester class.

**"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 OS X server is a very inexpensive way to run a server on a Mac. It is easy to implement but it is not as popular of a hosting platform as windows or Linux. | Linux is a very inexpensive and open source for hosting. With the open source nature of Linux it comes with a lot of features and resources for the programmer. The only problem is that Linux is not widely used and would require someone with great expertise. | Windows server is a much more expensive option then the other two forms though it is the most widely used hosting platform that finding someone with the skills to use this server would be fairly easy and inexpensive. | When it comes to hosting a web browser based game mobile devices are not at all the option to go here. They do not have the processing capabilities that computers do and they are more focused towards apps then any web based stuff. |
| **Client Side** | The cost to implement this type of software would depend on the expertise you find in the staff. As it is not as widely used as like a windows server or as easy as a Linux the cost to find staff trained in this server would be more expensive. | Cost for a Linux server would be relatively inexpensive. But the time and experience that would need to go into it would be the expensive part because yes it is easy, but someone adept with the knowledge on Linux would make the process faster and more efficient. | This is the same as the Mac OS because it is the most widely used server but the time and expertise to make the server run would be expensive plus the fact that it is mostly used the actual implementation would also be expensive. | The cost for the mobile device would be relatively inexpensive and the experience needed would not be as much because mobile devices are designed to be easy to use. The time aspect would be much greater because there are multiple different devices and makers of the devices to set-up across the board. |
| **Development Tools** | XCode would be the preferred IDE for any Mac OS since it’s the formal IDE and then Swift would be the appropriate language for Mac. | I found Eclipse voted to be the best IDE for Linux programming and then the appropriate language for Linux is C. | The best IDE’s for windows is either Eclipse or Visual Studios. I think Visual Studios is the formal IDE for windows. The programming language would then be C, C#, C++. | Depending on your device it would vary. For and IPhone it would be the same as mac. Android can be Java, C++, or Kotlin, which then use Eclipse. |

## 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 research and my work on this project leads me to conclude that the best operating platform for the Draw It or Lose It project would be windows. Windows is the most widely used and most practical when it comes to platforms. The variety of IDE’s that can be used on windows is endless the cost of using windows would be relatively low because it is far easier to use and would be less demanding on an employee that was hired to use it.
2. **Operating Systems Architectures**: With Windows the application can have access to the most basic processes of windows without affecting any of the processes that make windows run. It has access to the memory systems of a windows platform that would not slow the platform down. With windows the application could gain access to a windowed or a GUI form of windows to not inadvertently affect the process that make the platform function.
3. **Storage Management**: With the use of windows the actual platform has a few storage management techniques built into the platform. One being Disk management and the other being Storage Sense. Disk management is windows advanced disk storage device that allows you to store as much as you can fit on storage device. While the Storage sense in conjunction with the disk cleanup helps keep the storage device clean and deletes files that are no longer necessary to the device.
4. **Memory Management**: Much like the storage management windows has a built in memory management systems that helps balance the use of memory. But it would need a database designed by us to store all the images for the game which would not be too difficult on the windows platform.
5. **Distributed Systems and Networks**: The best distributed systems and network to use for this type of game would be a client-server distributing system. This will be set up so that each client application will be built off a single server application and that each client application can be developed to the client’s systems strengths and help mitigate their weaknesses when it comes to their application. Along with the addition of a very strong server network addition because this game relays on the use of a single server with multiple players to run the game properly.
6. **Security**: The windows operation platform has many security functions. Everything from a windows defender to a firewall protocol that won’t allow any suspicious activity to go on within the windows platform. So for security we are well protected by the platform itself. We may have to help with the encryption of the data so that we can send it back and forth between players and the server without any complications.