

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 |
| --- | --- | --- | --- |
| 1.3 | 10/12/2023 | Jose Perla | Updates to the Executive Summary, Design Constraints and Domain Model and Evaluation  Updates to Recommendation |

**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 based on their current game Draw it or Lose and have it work on multiple platforms. To make this game work on multiple platforms we will work on memory management for the program. This will be solved by allowing only one instance a game to run at a time using a singleton pattern. This will serve in giving us control object creation reducing memory usage, by using only one instance of gameService we can avoid allocating and deallocating memory .

## Requirements

1. Game will have one or more teams involved
2. Each team has multiple players assigned to it
3. Game name and team names are unique to allow users to check whether a name is in use, when choosing team name
4. One instance of game can exist in memory at any given time. Can be accomplished using unique identifiers for each instance of game, team or players.

## [Design Constraints](#_2et92p0)

The game must be designed to be web-based and work on multiple platforms. The game must allow for more than one team to participate, this will be done by using iterators to go through an array to check for team and game names to see if they are available. Basically, checking if the names are unique and if they are then they will be added to the array. The teams have multiple players assigned to them so each player will have a unique identifier, so they are not assigned to multiple teams. There can only be one instance game existing in memory at any time, so there must be unique identifiers for game, teams, or players an iterator will loop to do this check as well.

## [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 newly created entity class is used to abstract the id and name attributes that are going to be inherited by the other classes. The Game, Team and Player classes all inherit the attributes initiated in the Entity class. There also exists a zero to many relationships between the classes, Player has one with Team meaning there can be zero to many players in a team, Teams has a zero to many relationship with Game, and Game has a zero to many relationship with GameService. Finally for the Program driver class this class uses the SingletonTester for its method to test the singleton class in the program. By having inheritance and abstraction in our code we can reduce redundant code of having to create id and name constructors in each class and allows for calling of these methods instead.

**"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 is very strong in terms of security. Advantages include a user-friendly UI and compatibility with other apple products, development can be easy going between macOs and IOS.  The disadvantage of mac is that the cost is very high and while its compatible between mac its not compatible outside its ecosystem | Linux is also very secure and is open source.  An advantage for Linux is the cheapest option for web hosting and has a lot of stability.  The downside to Linux however is that there is not much support for it and is not as simple utilize | Windows is widely used for web hosting.  The advantage of windows is that it offers a lot of support for application and 3rd party software.  It’s also well-maintained, offering frequent updates.  The disadvantage for windows is also a high cost for hosting, | Using mobile devices as a server/ web hosting is not really an ideal situation, it’s better left to other operating systems. If it was to be hosted it would probably have to be small and simple since mobile devices are not meant to be server hosts. |
| **Client Side** | Will support the html interface as modern web languages are used like html, CSS or javaScript  Medium amount of time for development and deployment  Only compatible with macOs devices | Since Linux is a more complex OS to use it requires a higher level of expertise to utilize, but its open source so it can work with multiple web browsers. It also has a fast development and deployment. | The expertise required to use Windows is low and time required is also low. The cost is higher but will still support use of HTML interface using languages like HTML CSS or javascript | The advantage that we have is that we have already a set environment to android devices so we just need to continuously update with through the application  The cons to mobile devices is that there is difficulty in testing in other environments and browsers since sites are not often optimized for mobile and they also require longer development time. |
| **Development Tools** | For this program we are currently using the java language and use Eclipse as our IDE since it is compatible with this OS or even Xcode if we want to create code that runs on IOS. If we want to run this on webpages, we can also use HTLM,CSS or JavaScript. | For this program we are currently using the java language and use Eclipse as our IDE since it is compatible with this OS. If we want to run this on webpages, we can also use HTLM,CSS or JavaScript. | For this program we are currently using the java language and use Eclipse as our IDE since it is compatible with this OS. If we want to run this on webpages, we can also use HTLM,CSS or JavaScript. | For mobile applications that run on Android devices we can still use java as our programming language and use Android Studio as our IDE, but eclipse can also create mobile apps using google ADT extensions, For IOS however we can consider using swift as the programming language and Xcode for the ide. |

## 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**: Based on the development requirements we have the operating platform I think would be best for our development is Linux. The primary reason for this is that is one of the most cost-effective options due to its free-source nature. However, this does not mean that we are sacrificing performance because utilizing Linux still gives us strong performance, offers scalability and fast development and deployment.
2. **Operating Systems Architectures**: The architecture of Linux can be broken down into different elements such as the Kernel, system library, hardware, and shell. The Kernel is the main part of this architecture since it manages the operating systems resources to ensure there is enough memory available for application to run, its also in charge of optimizing processors usage delegating what applications should utilize the most resources. Its because of how the Kernel in Linux manages that leads to its performance stability.
3. **Storage Management**: When considering storage, we want something that will be cost effective for our use and will still be compatible with the different platforms we will be servicing. With this in mind the cloud offers effective storage and fast speeds. We can use cloud services like Amazon Web Services to run our program. While there are costs associated with this option we can pay for as little or as much storage as we need so it allows us to scale up if we see an increase in usage for our game.
4. **Memory Management**: With memory management we can leverage the fact that our language is Java and how it utilizes garbage collection to clear up memory. The other benefit of using Linux is that it can run multiple processes at the same time without interference due to virtual memory that map memory address from a program to physical memory locations, It does this by paging which is breaking memory into chunks that are allocated or deallocated when needed. When the program needs more memory than what is available in the physical disk the kernel uses the hard drive as an extension this is called swap space which is how Linux prevents memory related crashes.
5. **Distributed Systems and Networks**: With distribution we can utilize Ubuntu which is a Linux distribution that is also open source and free that has multiple editions that lets it work for Desktop, internet of things devices and is also compatible with cloud computing. This means that it will also be compatible with our AWS services. There is also a fair amount of support for it and it also user-friendly making it simpler and less daunting to set up servers. Additionally, something else we have to consider is the user of an API such as a REST so we can support multiple users sessions and have them each be unique so there are no duplicate sessions of the same game occurring.
6. **Security**: When it comes to security there are features in our program that we can use to ensure that users’ data are protected. One feature such as login credentials allows us to authenticate users before allowing them access to the game. Also ensuring we are utilizing role-based access control will also ensure that users of the game are not given unauthorized access to the program and the ability to alter our code, which would be actions reserved for admins or the dev team. Another thing we can consider is creating secure connections to our network by using HTTPS while there are costs to obtaining and maintaining a SSL or TLS certificate it will give users and our client the added layer of security by having an encrypted connection to our game.