

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## 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 |
| --- | --- | --- | --- |
| 1.0 | 09/17/2022 | Maximillian Wolfe | Added the Entity class as well as the Team and Player class which inherited from the Entity class. Refactored the Game class to inherit from the Entity class and added Unique identifiers to Teams and Player to remain using the singleton method. |

**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 application based off of their existing game Draw It or Lose It which is currently an android application. They wish to provide this game to multiple clients as well as multiple platforms. The game must have multiple teams with multiple players for each team. Each of these games and teams must have unique names which the program checks to ensure uniqueness. The final request is that only one instance of the game can exist at a single time.

## [Design Constraints](#_2et92p0)

* Memory allocation to ensure the storage of data for multiple games, teams and players.
* Processing speed to support the amount of players at a single time.
* Stable and reliable network access to ensure smooth playing throughout the game.
* Authentication to ensure that player accounts aren’t hacked or stollen.

The Gaming Room also want this to be able to be used on multiple platforms. This will require that the code either be rewritten in the languages used by each platform or we need to utilize a software that can accomplish this for us.

## [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 all have an inheritance relationship with Entity. This means they inherit specific information from entity such as the id and name variables allowing each entity to have its own id to distinguish between players as well as a name that we can search through to ensure uniqueness. These variables are private so the classes also inherit the getid and getname functions to access these variables. This allows them to access information like id and name from the Entity class. Then there is the relationship between GameService, Game, Team, and Player. They Each contain instances of the one to the left in the order I stated them. GameService contains multiple instances of Game which contains multiple instances of Team which contains multiple instances of Player. This allows us to meet the requirement of having multiple teams per game as well as multiple players per team.The Program driver is the main function which calls the individual classes and uses the SingletonTester.

**"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** | Characteristics: popular for web hosting, Unix based  Advantages: Top of the line hardware, secure software, and no fear of data hacking, closed source making it more secure  Disadvantages: Mac servers often cost more and are hard to find, licensed | Characteristics: Simple and cheap, Unix based  Advantages: Cheap and easy to get a server, open sourced, great security options available  Disadvantages: Not secure and simpler software, open source making the os less secure | Characteristics: versatile but slower  Advantages: Able to utilize multiple different applications to integrate, readily available plans for servers  Disadvantages: Can run slower or choppier than the others, Mid-tier cost margins, licensed, requires knowledge and ability for security | Characteristics: Similar to the respective web types associated with the mobile device, Unix based  Advantages: Portable, cost effective, better compatibility, open sourced for android, many security options available for android  Disadvantages: Not as secure and selective to the device being used, licensed for ios, open source for android makes it less secure |
| **Client Side** | The second most popular used OS at about 15% of desktops utilizing it. The default browser is Safari but it supports google chrome and firefox as well | The most popular used OS at about 75% of desktops utilizing it. The default browser is Microsoft Edge but supports google chrome as well as firefox | The least popular used OS at about 2% of desktops utilizing it. The default browser is firefox but supports goggle chrome and Microsoft edge | IOS is the most popular mobile OS followed by android. These are utilized in almost every home with multiple devices usually being the case even if no desktop is present. The IOS default browser is safari but it supports chrome and firefox, the android default browser is google chrome but it supports firefox as well |
| **Development Tools** | There are many tools that can be used to program is macOS such as Homebrew and iTerm2. The most relevant language would be swift as well as HTML, CSS, and Javascript. | Using Visual studio and or eclipse we can use a variety of languages such as python, c++, .net and more. That will be utilized with HTML, CSS, and Javascript. | Using Visual studio and or eclipse we can use a variety of languages such as python, c++, .net and more. That will be utilized with HTML, CSS, and Javascript. | Can be made using both android and swift along with the devices supporting app development software. This will primarily be done on one of the other three operating systems to be rolled out to the mobile devices. |

## 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**: I recommend Windows as the starting development platform. With how diverse the platform is as well as the number of applications available to be implemented it is the best option to start. That along with the lower cost makes it the best to start with. The ability to custom install individual parts to meet the specifications as well as the open-source resources that enable easy modification to any needs makes this the go to operating system.
2. **Operating Systems Architectures**: Windows allows applications to use GUIs while accessing system resources. There are two modes within the windows operating system which are the user made and the kernel mode. User mode is made up of system defined processes and DLLs while the kernel mode has access to all system resources and hardware.
3. **Storage Management**: Windows allows the use of hardware or cloud-based storage with easy management through storage sense. This allows for easy file or project creation and management. The hardware is easily interchangeable and cheaper than other options. The ability to upgrade and install SSD and HDD allows for the management of storage to be a fairly simple process.
4. **Memory Management**: Windows has a very intuitive memory management system with the ability to interchange memory hardware to fit the needs of the user. There are built in systems that can visualize the memory and what is using it and how much is being used at any given time. The windows operating system is the go-to system if you want to customize the sizes of your cache as well as the amount of ram for the system.
5. **Distributed Systems and Networks**: Using Visual studios with SDL2 libraries will allow the development of the game through the C++ programming language with an easy compatibility with all platforms. To help with the consistency of reliability in network the company will have to ensure that their servers are strong enough to support large volumes of players.
6. **Security**: Windows has some built in security for what is on the machine but for securing user data an outside software will be needed.