

Draw It or Lose It

# **CS 230 Project Software Design Template**

Austin Sonka

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 | 10/01/2022 | Austin Sonka | Completed thoughts on the table of contents. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants us to develop a website game that can run on Linux, Mac, and Windows. Currently they have an Android app for the game. The game will be called “Draw It or Lose It.” The game has multiple teams of multiple people playing four rounds of a minute each round. The game has a picture card where if it gets taken from the pile of cards then the team can guess what it is until the minute is up. After this then the opposite team can take a chance of guessing it correctly within 15 seconds.

## [Design Constraints](#_2et92p0)

-Needs lots of people for teams and team size

-Running the app on different platforms

-Cross platform

-App colors need to look identical to website

Designing the app from Android to work on multiple devices can be difficult as we would need to find a way to redesign the code or make new code. Perhaps using Java to use multiple languages can be a solution.

## [Domain Model](#_8h2ehzxfam4o)

The Game, Team, and Player all get information from the Entity box. The entity box is also the base class is introduced to hold the common behaviors and attributes. The GameService also has a call to Game, while Game calls to Team and Entity and Team calls Player.

**"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** | Quickly becoming popular.  Some pros are that it is upgradable, popular in networking and an easy terminal to make any changes to the server. | Not as popular. Pros are easy use of for web hosting and some security issues are detected early on. Cons are some applications may not be compatible with some web hosting. | Most popular. Been the original OS out of them all. Is the most compatible with any software. Very comfortable and easy to use. Cons is other OS getting more popular in software. | Pros are more comfortable and easier to use. Also, very popular as everyone has mobile device. Cons are limited in certain areas and can have not so good security. |
| **Client Side** | Like windows on the cost side as they were closely made together. The expertise is little to none. | A good amount of expertise to use this OS. The cost is not too much. | The expertise is little to known. The cost is like Mac. | The amount of expertise is little to none also. The cost can vary. |
| **Development Tools** | Mac can run many languages smoothly, these can include Java, CSS, HTML, Python, C++ and may others. Tools can include download apps like eclipse and PyCharm. | Linux works well with eclipse, visual studio, and PyCharm. Can run many languages like the Mac. | One of the easiest platforms to use and can run many apps and languages. These include Eclipse, PyCharm, Visual Studio, HTML, Python, Java, C++. | Uses android or IOS. The languages are usable in all devices and the have the same type of languages as the other platforms. |

## 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 would think starting on Windows would allow a easier transfer to other platforms as Windows is simple to use and is the most versatile with software and apps.
2. **Operating Systems Architectures**: The operating services on Windows can provide software and applications that allow the graphical interface to be shown while looking into the resources, messages, and the web services.
3. **Storage Management**: The storage is incredibly in windows as it shows you what is in storage and how much room it is taking. Also tells you what you can remove if you would like. Can also create folders for easier navigation for putting documents, pictures of the game into a file to view easily.
4. **Memory Management**: Windows is great for creating either a database or library in a server for documents and pictures. Doing this creates an easier time to navigate to find certain things relating to the project or game.
5. **Distributed Systems and Networks**: I believe once the game is created and finished, we can then export the game into web then to android then to windows then to IOS to eventually get cross play with other people on different devices. This will require other servers and upkeep with hem just to make sure none of them go offline and allow users to use them.
6. **Security**: This is a must have for any client. Using Windows automatically has security measures in their software. This security goes from canning for virus, malware, and other numerous security threats. With the constant updates allows the system to update with the latest and newest security as time goes on.