

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_heading=h.gjdgxs)1

[**Table of Contents**](#_heading=h.30j0zll)2

[**Document Revision History**](#_heading=h.3znysh7)2

[**Executive Summary**](#_heading=h.2et92p0)3

[**Design Constraints**](#_heading=h.tyjcwt)3

[**System Architecture View**](#_heading=h.3dy6vkm)3

[**Domain Model**](#_heading=h.1t3h5sf)3

[**Evaluation**](#_heading=h.2s8eyo1)4

[**Recommendations**](#_heading=h.3rdcrjn)6

## [Document Revision History](#_heading=h.3znysh7)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 09/14/2022 | Jackson Foster | Updated all sections of the design document to include the different design aspects of the application |

## [Executive Summary](#_heading=h.2et92p0)

The Gaming Room wants to develop a web-based game that can run on multiple platforms such as mobile application and web-based application. The game will be called “Draw It or Lose It” and it is currently only available on android. The game will consist of multiple teams and consist of multiple players going for 4 rounds of play for one min each. The game will bring out a picture and will be slowly revealed over 30 seconds and then a team tries to guess it til the time runs out. If no one can guess it each team gets 15 seconds to answer it.

## [Design Constraints](#_heading=h.tyjcwt)

* Needs to have 1 or more teams involved
* Each team must be able to consist of multiple players
* Game must be unique and only one instance at a time
* Team names must be unique and be able to check if their team name is in use
* Game must run on multiple platforms as IOS, Android, macOS, Windows

These are the requirements that are provided from the client. The gaming room wants this application to run on all devices which means we need to add all the other ones and not Andriod because that is already made. We will need to add Windows, Linux, and Apple. To do this we need to make sure we can write this on a program that makes it possible for other languages to inherit it.

## [Domain Model](#_heading=h.1t3h5sf)

The entity class creates the relationship between Game, Team, and Player. These classes inherit their information from the entity class. So these classes will share the references like “name” as well as “id” making Entity a superclass which is why it is at the top of the UML design. We can also see that GameService has an instance of Game in it. Also, the Game class had an instance of Team in it, and Team has an instance of players in it. This is to make it possible to verify that there is only one instance of each at a time. This also shows aggregation throughout the classes.

**"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](#_heading=h.2s8eyo1)

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 has flexible terminal commands to configure, access, or make changes to the server.  Mac is popular in web hosting  Mac is able to be upgraded and has options for web hosting requirements  Mac on the more expensive side | Cost friendly  This is the most preferred because it is so secure  Security flaws are caught before they can be exposed, This is the most preferred choice for web hosting  It is difficult to support the requirements of web hosting | Most software available compared to other OS  It is the most used compared to other platforms  very easy to use, more resources to use  bad tech support, and easy virus susceptibility | Most popular and has high portability  Mobile devices has a wider reach with it being the most cost-effective and most accessible  This has poor security and is very selective to various mobile devices |
| **Client Side** | The experience needed is moderate and so is the time required. The cost is similar to Windows. | The experience required is the highest out of all the other OS’s and so is the time required. But the cost is the lowest. | This is one of the least expertise and time required. The cost is similar to Mac. | This is harder to translate to multiple platforms with different mobile devices. But provides flexibility for people to see updates |
| **Development Tools** | The most popular language on mac you can run swift which is the most common option. Mac also has notepad++ built in which is helpful. But this can run most languages. | Linux can work with Visual Studio, Eclipse, aswell as notepad++. But this can run most languages. | Windows is easier to use than Linux but it will run the same. So it has all the same ides as Eclipse, Pycharm, and Visual Studio. But this can run most languages. | You can create so many apps with android and Swift which can be run on all OS and can run HTML, CSS, and Java. |

## 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 have to recommend our client The Gaming Room start with developing on Windows Platforms because it has more software availability as well as minimum experience required to learn. It is also a minimum cost to start projects.
2. **Operating Systems Architectures**: The Windows operating system provides services used by all Windows-based applications to enable an application to show a graphical user interface (GUI) while accessing resources and other services. The Windows operating system offers these services using a user account or a server.
3. **Storage Management**: Windows comes with a very useful function called storage sense which allows users to look over all their files and applications and decide if they are needed or not. They also offer cloud storage services which makes it possible for plenty of storage.
4. **Memory Management**: Windows makes it easy for files easy to find and store exactly where you want them. This could be useful to pull pictures for the Game Draw it or Lose it. This will make the game run smoother by being able to store the images in a certain place to pull from.
5. **Distributed Systems and Networks**: Due to each operating system being different there are ways to publish the game to run on all devices. There is a program called Develop 4 which enables cross-platform game creation. It is an IDE that can be run on all devices. Once the game is created all you have to do is export the game file onto the web, iOS, Android, and other platforms which will allow cross-play. Very common problems come up like outages or connectivity, so the Gaming Room will need to make sure their servers are strong enough to support high player volume as well as backup generators for power outages.
6. **Security**: The great thing is that Windows comes with built-in security protection software. Although to secure user data and information it is recommended to use another source of security. Another great thing about windows is that it is scanning for malware and virus which also adds some good security to the application.