

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 | 08/1/2021 | Izzy DeRocco | Overview on the program, in addition to recommendations on operating systems. |

**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)

Draw It or Lose It is a game where teams compete to guess on what is being drawn. The game is like the 1980’s television show Win, Lose or Draw. The game is 4 rounds at 1 minute each, where the picture is slowly rendered till its completed at 30 seconds. If the teams do not answer in time, the remaining teams that were not in the round have a chance to guess within 15 seconds.

There is currently an android version for the game but a web-based edition will be developed for the application so it can be used on multiple different platforms. The game is required to have the ability to have one or more teams involved with multiple platers assigned to each team. All players and team names should be unique to the respective team or player so, a system must be created to check the data base to see if the name is already taken. A single instance of the game must exist in the memory at any given time.

## [Design Constraints](#_2et92p0)

* + The game should have the ability to have one or more teams involved.
  + Each team will have multiple players assigned to it.
  + Game and team names must be unique to all users to check whether a name is in use when choosing a team name.
  + Only one instance of the can exists in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team or player.

The client wants these constraints to the application itself. There are also constraints that are for the development side. The client wants to have the program work on multiple platforms. The code for the android version needs to be ported to other platforms. The constraint is that not all operating systems use the same languages or file executables. The code will need to be written and compiled in the operating systems language.

## [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 Entity class is the super class, and the classes Game, Team and Player all inherit the traits from it. The Gameservices class references the game class, which references the team class. The Team class then references the Player class. There are also the ProgramDriver and SingletonTester classes. The ProgramDriver class executes the program files, but also runs the singletonTester. These classes allow for one game to go on with multiple teams with multiple players all in a single instance. All players and teams are in lists that are done in the classes themselves.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac does support a server-based deployment method. It costs money monthly but is easy to use and to set up. The license for the server would be MacOS server starting at $19.99 a month. | Linux has a server-based option to run the program that is free and open-source. The data files are available online. It is a secure option and the least costly. | Windows server is the secure but also has a monthly charge like MacOS. The server licenses start at $20.00 a month. | Most any computer can be turned into a server, including mobile devices. A server program is Servers Ultimate which starts at $8.59. This being a web-based game, it would be the least recommended choice to run the server. A stationary computer would be the best idea. |
| **Client Side** | This being a web-based server, a client can be created using xCode or any other IDE. Simplest alternative would be using the web- browser as well. The cost could be minimal to free due to clients being easy to develop and web browsers are freeware. If the game is written in HTML5, that would allow for the highest chance of compatibility with all the other web browsers. | Like MacOS a client can be created using an IDE, like Eclipse. Being an open-source operating system. Most web browsers have been ported over so they can be used as well to run the game. Depending on the version of Linux a web browser must be installed that supports HTML5 or Flash player depending on the format of the web-based game. | Like MacOS windows comes out of the box with web browsers that support HTML5 and doesn’t need much to support flash. Most web-based programs run off HTML5 so that shouldn’t hinder the user from playing the game. Like the other operating system a client can be made that will emulate the game in a private browser. | For mobile gaming it would be a better choice to go with a client over the web browser. The ability is still there but with using a client, the user just needs to install the game from their respective market/store. Many of the web browsers for mobile can view HTML5 but since mobile technology is more based on doing one main task at a time, unlike computers, it would be easier and more user friendly to design a client and app install. |
| **Development Tools** | With the server, XCode can be used to develop the server and its updates. The initial software is free, but a developer license will be required that costs $99. This will also come with beta operating system data so features and the applications can be up to date prior to the release of the new OS update. Being a web-based game, only a small team will need to be used. The servers are easy to maintain, and the cost of paying to many staff members can be avoided. | There will be cost to the licensing when developing with Linux. Being an open-source operating system, everything is free and open to the developers and the user. Right now, the most common version of Linux is Ubuntu, and they have their own install store that an application can be submitted at no cost. With the cost of the server and client being free, this is the most cost-effective choice. The client can also be written in eclipse which is also a free IDE that works on most operating systems. This option would require a medium sized team, due to everything being open-sources and the amount of different Linux versions that each have their own update patterns. | To develop for Windows, the developer fee is $19.00 for a individual, but the company account would cost $99 dollars. Software for the design of a client and server would be free when using a free IDE, but a more professional stance would be using Visual studio, where the professional licenses cost 99 dollars a license. The team would be also small due to all windows computers having very malleable drivers to and constant support. | The development fee for android is the cheapest at $25 dollars. But if thinking about mobile as a whole, this, would be the most expensive to develop for due to the vast number of operating systems. To develop for most operating systems, you would need an Android, MacOS and Windows developer license to post the client on each different app store. This would then require the most teams due to it not just being a single OS system. This would be the costliest, but it would be seen by the most viewers due to most people have a mobile device (mobile phone) with them at all times. |

## 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**: To maximize the exposure to the masses without increasing the budget of the project to high the best choice for operating system would be Windows. It’s the easiest to develop for, needing minimal expertise which will also bring the cost of the project down. Windows also provides numerous APIs with its SDK to allow for easy software creation. There is also support for multiple coding languages to allow for many different options for the games main coding language.
2. **Operating Systems Architectures**: Windows provides great support for the user graphic interface of the software. Windows has the best services when gaming is involved. Windows as an operating system can utilize strong components and have the graphics of the product rendered fast and with extreme details. Windows also has an updated command prompt, power shell, to allow easy configurations of a server.
3. **Storage Management**: Windows has a disk manager and storage can be cleaned or added with ease. The disk manager allows for adding more storage if necessary. Windows also supports a cloud-based server that allows for plenty of storage expansion.
4. **Memory Management**: Windows has its own memory manager that will be done automatically. No need for the software to need to do anything involving that. In addition to having physical memory, there is also virtual memory that can be expanded to allow for a smoother experience.
5. **Distributed Systems and Networks**: To distribute the software, it costs $49 dollars for an individual account, or $99 dollars for a company account yearly. After, the developers’ license is bought, the application can then be published to the Microsoft store, for users to download and play. By using Microsoft Azure as a cloud server, this would prevent the servers going down due to outages or any other reason. Azure supports multiple languages, is free to start, but when more storage or higher speeds is needed, more can be bought.
6. **Security**: Windows systems all come standard with Windows Defender for its protection. But windows do have many other virus protectors that also protect the network connections. Windows as the most software compatible with the operating system, so security is easy protected. To protect the network, a VPN service can be used to help protect from online threats. Windows also has BitLocker to encrypt the storage device in case the machine is stolen or lost to help protect the sensitive data.