

“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 | 05/23/21 | Talha Shafiq | Evaluated different operating systems for program, “Draw It or Lose It” |

**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 would like to make a browser game that can run on multiple platforms. The game is only available on Android and is named “Draw It or Lose It”. The goal of this game includes teams having several people going four rounds of play lasting one minute each. When an image is pulled from a library of pictures, one team will guess until the time runs out. If the team doesn’t guess in time, then the other teams will have a chance to guess once within the 15-second time limit.

## [Design Constraints](#_2et92p0)

These are all the requirements that must be followed while developing the software and code. Although this is the game aspect, we must still focus on application development. The Gaming Room wants to run on multiple platforms besides Android such as Windows, macOS, and Linux. In order to do this, we must find a way to use existing code in order to run on other devices by inheriting other languages or we could revise the code using Swift (for macOS). Just like how we use various programming languages to make the code more stronger.

* Needs to run on multiple platforms
* Each team consists of several people
* Must have one or more teams included
* Only one instance of the game can remain in memory at a given time.
* To be able to allow users to see if a name is being used when selecting a team name, team and game names should be unique.

## [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 relationship between Player, Team, and Game class is created by Entity. They all get information or inherit from Entity. The UML shows this with inheritance. Each class shares a common reference such as id and name, which means Entity is a Superclass. If we view their relationship, we can see that Player and Team is a “has a” type, while GameService has Game and Game has Team. Using the UML, we can call it aggregation (HAS A). When I say ”has a”, I mean that it is an instance of one class and has a reference to an instance of another class. If we take a look at the diagram, we notice that Team has a reference of Player, Game has a reference of Team, and GameService has a reference of Game.

****

## [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** | Versatile terminal commands for gaining access, configuring, and making changes to the server. | Similar to Mac and is more cost friendly. | When compared to other operating systems, there is much more software available on Windows. | It is preferable if the server is able to be tracked at a single place and that its immobile. Other devices have better specifications. |
| **Client Side** | Moderate amount of expertise and time are both necessary. The cost is comparable to Windows. | High expertise and time are both necessary. The cost is low. | Low expertise and time are both necessary. The cost is comparable to Mac. | Gives clients and even developers the flexibility to view updates from any location. Implementation is slightly more difficult than with other devices. |
| **Development Tools** | When it comes to running languages on macOS, the most popular option is Swift, while also incorporating useful tools such as Notepad++. Macs are able to run all languages. Languages include, but are not limited to, JavaScript/CSS/HTML, as well as libraries for supporting the general-purpose and front end languages. These include Python, Java, Ruby, and PHP. | Linux can be used with IDEs such as  Eclipse and Visual Studio, along with Notepad++ as a nice and simple tool. Linux also has access to many other tools and languages. Languages include, but are not limited to, JavaScript/CSS/HTML, as well as libraries for supporting the general-purpose and front end languages. These include Python, Java, Ruby, and PHP. | Much easier to work with compared to Linux but can run similarly. Eclipse and Visual Studio also work with Windows, including Notepad++. Languages include, but are not limited to, JavaScript/CSS/HTML, as well as libraries for supporting the general-purpose and front end languages. These include Python, Java, Ruby, and PHP. | You can make numerous apps using Swift and Android. Both software and languages are able to run on all three operating systems. Languages include, but are not limited to, JavaScript/CSS/HTML, as well as libraries for supporting the general-purpose and front end languages. These include Python, Java, Ruby, and PHP. |

## 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 definitely recommend Windows to expand Draw It or Lose It to other computing environments. This is because Windows has much more software and also low expertise and cost to get projects started. In addition, there are plenty of different IDEs to choose from.
2. **Operating Systems Architectures:** Windows gives services used by Windows-based programs that allow programs to display a GUI while gaining access to system resources and much more. These programs refer to web services, messaging, and multimedia and graphics. These services can be accessed through a user account or a server.
3. **Storage Management**: Windows 10 has a great feature called Storage Sense. This helpful feature helps free up space and manage files on your hard drive. Some other features are having the ability to select safe app locations to make it easier to find. You can also safely store data using the cloud. Big projects can be easily created and placed in the built-in storage system, ensuring that they don't get misplaced or accidentally deleted.
4. **Memory Management**: When making this game, we will need to make a library or database with plenty of pictures. Pictures can be easily stored outside of the default picture folder thanks to the memory allocation. This enables you to keep your entire project in a more safer location on your computer. This includes using your IDE to develop the game and open files from it.
5. **Distributed Systems and Networks**: In distributed systems, using networking support is an excellent approach to implement and use software. Distributed systems and networks have the advantage of allowing multiple workstations to communicate with each other and with different processors. Another significant feature of distributed systems and networks is the ability for users to communicate with various servers, such as data servers, web servers, etc. When computers are connected to a network, tasks are distributed and processed between the sever and the user, resulting in increased efficiency.
6. **Security**: Security software is included with Windows. However, using another source to secure information and user data is encouraged. But if we're talking about what's already included, Windows comes pre-installed with protection. Security threats, malware, and viruses are all scanned by this system. All of this occurs in real time, and as threats develop, the system automatically changes in order to keep user data and the system safe.