

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 | 03/18/21 | Zac McLaurin | First draft of what is needed for this application |

**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 web-based game that can run on multiple platforms. The purpose of this game is to have multiple teams consisting of several people complete four rounds at a minute each. When a picture is rendered from a library of images one team guesses what they think the images is. If they do not guess correctly before time expires then the remaining teams have an opportunity to guess what they think the image is.

## [Design Constraints](#_2et92p0)

* A game will 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 allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

These are the requirements that need to be met when writing the code for this application. The biggest challenge is that it needs to be able to work across multiple platforms. To do this the code will need to be written in multiple languages or find a way to use existing code on different devices.

## [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)

All three of Game, Team, and Player are an Entity because they all inherit from Entity. All three have common attributes like name and id. This makes Entity a super class. The Team and Player class have a aggregation relationship.

****

## [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** | Flexible commands for the terminal to configure, access, or make changes to the server. | It’s the same as Mac but more cost friendly. | There is more software available compared to other operating systems. | It is better if the server is immobile. Specifications are better in other devices. |
| **Client Side** | Moderate time and expertise required. The cost is similar to windows. | Maximum expertise and time are required. The cost is minimum. | Minimum expertise and time are required. The cost is similar to Mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | When running languages on Macs you can run swift the more popular option. While mixing in tools like notepad++. Mac can run most languages. Languages consist of HTML/CSS/JavaScript while supporting libraries to support the front end and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Linux can work with visual studio, eclipse, and notepad++. Linux can also work with many other tools and languages. Languages consist of HTML/CSS/JavaScript while supporting libraries to support the front end and general-purpose languages. These can be Java, Python, and PHP. | Can run same as Linux so visual studio, eclipse, and many others. The same goes for tools as well. Languages consist of HTML/CSS/JavaScript while supporting libraries to support the front end and general-purpose languages. These can be Java, Python, and PHP. | You can create many apps using android and swift. Both of these languages can be used on all three machines. Languages consist of HTML/CSS/JavaScript while supporting libraries to support the front end and general-purpose languages. These can be Java, Python, PHP, and Ruby. |

## 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 recommend that The Game Room start on windows devices since it has more software available as well as minimal expertise and cost to get the project going. You also will not run into a shortage of IDE’s.
2. **Operating Systems Architectures**: Windows provide services used by all of the windows-based applications that enable applications to show a GUI, while accessing system resources and much more. These applications also refer to Graphics and Multimedia, messaging, and web services. These services can be used by using a user account or a server.
3. **Storage Management**: Windows 10 comes with a feature called storage sense. This allows you to manage files on your hard drive as well as how much storage they take up. Other features include being able to choose the save locations for apps which makes them easier to find. With windows 10 you also have a cloud that you can save data on. The built-in storage system allows for easy file management for larger projects.
4. **Memory Management**: While creating this game you will need to create a database or library with a lot of pictures. The memory allocation allows for easy storage of pictures outside of the default picture folder. This allows you to keep your entire project together in a more secure area. This includes while working with your IDE and opening files from it to create the game.
5. **Distributed Systems and Networks**: Because of each operating system being different I had to find ways to publish the game so that it could run on all drives. I found Develop 4 which enables cross platform game creation. It is an IDE that can be ran on any device. Once the game is created you can simply export the game into web, iOS, Android, and many other options that will allow for cross platform play. In order to prevent other problems like outages or connectivity, the company will need to make sure that the servers are strong enough to have a large volume of players as well as backup power incase of an outage.
6. **Security**: Windows comes with security protection software. This system will scan for malware, viruses, and security threats. This happens in real time and they system is always updating to notice new threats and to keep the users information safe.