

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 3.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 04/11/2023 | Jeffrey Bowman | Project recommendations |
| 2.0 | 03/31/2023 | Jeffrey Bowman | Evaluate the characteristics, advantages, and weaknesses of each operating platform |
| 1.0 | 03/18/2023 | Jeffrey Bowman | Named Design Constraints, detailed evaluation, and provided recommendations |

**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 is planning to create a web-based game called "Draw It or Lose It" that can be played across different platforms. Currently, the game is only available on Android. The objective of the game is to have multiple teams, each consisting of several players, compete over four rounds, each lasting a minute. A picture is randomly selected from a library of images, and one team must guess what it is before time runs out. If the team fails to guess correctly, each opposing team member gets a chance to answer within a 15-second timeframe.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

* Needs one or more teams involved
* Each team has multiple people
* Game and Team names must be unique to allow users to check whether the name is in use or free
* Only one instance of the game can exist at any time.
* Must run on multiple platforms

These are the requirements needed to follow while writing the code and software. While this is only the game aspect, we still need to look at application development. The Gaming Room would like this to run on all devices. This means we already have it on android but need to work it into another mobile device. Along with machines like Windows, Linux, and Apple. To do this we will need to find a way to either re-write the code in swift for (Apple devices) or come up with a way to use existing code to be run on other devices by inheriting other languages. Like when we use multiple computer languages together to make stronger code.

## [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)

By creating an Entity class and establishing relationships between it, the Game, Team, and Player classes can all inherit or access information from the Entity class. To illustrate this relationship, we can use UML diagrams and indicate inheritance, where each class shares common attributes such as "name" and "id", with Entity acting as a superclass. When examining the relationship between these classes, we see that Team and Player have a "has a" relationship, while Game has a Team, and GameService has Games. In UML, this is known as aggregation, which means that an instance of one class has a reference to an instance of another class. Looking at the UML diagram, we can see that GameService has a reference to Games, Games has a reference to Team, and Team has a reference to Player.

## 

## [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 terminal commands to configure the server, access, or make changes.  Characteristics It is popular in web hosting  Advantages It is upgradeable, it has various options for different web hosting requirements  Disadvantages It is less preferred for web hosting services | The same goes for mac plus more cost-friendly  Characteristics Secured, most preferred.  Advantages Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  Disadvantages It is more difficult to find applications to support the web hosting required needs. | More software available compared to other OS.  Characteristics It is dominant to the other platforms. Close platform  Advantages High resource requirements, less loading time, high comfortability  Disadvantages easy virus susceptibility, poor tech support | It's better if the server is immobile and can be tracked in a single place. Specifications are better in other devices.  Characteristics More popular, high portability.  Advantages Have a wider reach, better compatibility, cost-effective  Disadvantages It is highly selective to various smart mobile devices Poor security |
| **Client Side** | Moderate expertise and time required. Cost similar to windows. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Maximum expertise and time required. Minimum cost. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Minimum expertise and time required. Cost similar to mac. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | 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 we can run swift the more popular option. While mixing in nice tools like notepad++. Though Macs can run all languages. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Linux can work with visual studio, eclipse, along with notepad++ for a nice and easy-to-use tool. Along with many more languages and tools. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Easier to use than Linux but can run the same as it. So visual studio, eclipse to name a few of the many languages. And with multiple tools notepad++ is a simple to use the tool. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | You can create countless apps using android and swift. Both languages and software can be run on all three machines. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend 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**: My suggestion would be to begin The Gaming Room project on Windows devices due to the wider range of available software, the lower level of required expertise, and the reduced cost to get started. Additionally, there is an abundance of integrated development environments (IDEs) that can be utilized without encountering any scarcity.
2. **Operating Systems Architectures**: Windows offers essential services utilized by all applications running on the platform, which allow them to display a graphical user interface (GUI) and access various system resources, among other functionalities. These services cover graphics and multimedia capabilities, messaging features, and web services. They can be accessed through either a user account or a dedicated server.
3. **Storage Management**: One of the useful features that Windows 10 offers is called "Storage Sense," which enables users to examine and manage their files on the hard drive and monitor the amount of space they occupy. Additionally, the OS provides the ability to select preferred locations for apps, making them more accessible. Furthermore, users can use cloud storage to backup their data like with other storage devices. The in-built storage system offers a convenient way to create and store files for large projects, ensuring they are not lost or unintentionally erased.
4. **Memory Management**: During the process of creating this game, you will need to assemble a database or library of numerous pictures. Windows allows for simple memory allocation, which enables the storage of these pictures outside of the default picture folder. This means that you can maintain your entire project in a more secure area on your computer. This includes opening files from your integrated development environment (IDE) to create the game.
5. **Distributed Systems and Networks**: As each operating system varies, I conducted research on methods to publish the game to run on all devices. During my investigation, I discovered Develop 4, which facilitates the creation of cross-platform games. It's an integrated development environment that can run on any device. Once the game is developed, it can be easily exported into various formats, such as web, iOS, Android, and more, allowing for cross-play. This would assist in addressing dependencies issues. To avoid other challenges such as server outages or connectivity problems, the company should ensure that their servers are robust enough to accommodate a high volume of players, along with backup power in case of power outages.
6. **Security**: While Windows does offer built-in security protection software, it's advisable to consider using additional security measures to safeguard user data and information. Nevertheless, the pre-installed security features of Windows do provide some level of protection. The system automatically scans for malware, viruses, and other security threats in real-time, and it receives regular updates to adapt to evolving threats, thus ensuring the safety of both the system and user information.