

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 4.0

## 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)3

[**Recommendations**](#_heading=h.3rdcrjn)5

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

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0  2.0  3.0  4.0 | 1/22/2022  1/28/2022  2/6/2022  2/20/2022 | Matthew Muller | Executive summary and design constraints completed.  Rest of document completed.  Revisions made  Revisions made to 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](#_heading=h.2et92p0)

The client, The Gaming Room, seeks to develop a web-based game that serves multiple platforms based on their current game, Draw It or Lose It, which is currently available in an Android app only. The applicatication will render images from a large library of stock drawings at a steady rate until they are fully complete at the 30-second mark. Since the game currently only works on android, we need to develop the ability to serve Windows, Linux, and Mac OS.

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

* The game must 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
* Only one instance of the game can exist in memory at any given time
* Program will be written using Java language

We must ensure that the game is available to users on all three of the major operating platforms: Windows, Mac OS, and Linux. We will achieve this by making the game web-based so that users simply need to visit a website.

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

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](#_heading=h.1t3h5sf)

* The Game, Team and Player classes have an Inheritance relationship with the Entity class, which helps to prevent repetitive code being written in every class
* ProgramDriver class and SingletonTester class have an Association relationship as well as Navigability, making it possible for the main class to do the testing
* The GameService, Game, Team, and Player classes can be associated with zero to many of each other

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

**"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** | One advantage is flexible terminal commands to configure the server, access, and to make alterations. Also has diverse implementation technologies, imposing fewer technical restraints. One drawback is expensive scalability and licensing. | Linux shares the flexible terminal benefit with Mac while being a more cost-effective option. These savings come from cheaper scalability as well as the lack of licensing costs due to Linux being an open source OS. Minimal technical restraints. | One big advantage of Windows is the massive amount of software available for it. Since it is so widespread, it has more existing software than the other two operating systems. Windows licensing is less expensive than Mac’s and it | High scalability for web-hosting but requires an internet-connected device to support DNS conversion, adding extra relay steps. Another advantage is inexpensive licensing. |
| **Client Side** | Requires moderate time and expertise and a high Cost. Supports modern browsers, secure authentication, and transmission protocols that prevent session hijacking. However, like Linux, it supports a minimal number of file structures/media types. | Requires lots of time and expertise. Boasts low costs as the OS is accessible to the public for free. Supports a minimal number of file structures, limiting media type options that can be delivered to the client without additional programming. While not all devices run well with Linux, it does run on about 67% of all web servers as of 2016. | Requires the least time and expertise. Cost is higher than Linux but lower than Mac. Imposes technical constraints due to targeting of Microsoft web framework technologies. | Provides flexibility to clients as well as developers to see updates at any place. However, they are slightly more difficult to implement than other devices. To face these challenges, frameworks have been developed (based on HTML5 and CSS3) that are specifically designed to work as flawlessly as possible on a wide array of smartphones and tablets. |
| **Development Tools** | Much of the development done on the Mac OS is done using the Xcode IDE utilizing the language swift. Xcode is free to download. | Java is the most common programming language used on Linux. The OS supports Eclipse for development as well as other cross-platform IDEs like KDevelop. Both Eclipse and KDevelop are free. | C or C++ are the most commonly used languages on Windows. One popular IDE used on the OS is Visual Studio. Visual Studio is free to download. | iOS and Android favor different programming languages. Android favors Java, which is not natively supported on iOS. Most Android apps are built using the Android Studio or Eclipse IDE. Most Apple apps are written in either Objective-C or Swift and built using the Xcode IDE. It costs $99 to publish an app on the Apple app store and $25 for the Google Play store. |

## 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 Gaming Room use the Windows operating system when developing the “Draw It or Lose It” application. It requires minimal expertise and has a vast library of accessible software at a reasonable cost. It also has a large number of IDEs available for use. Another benefit of Windows is that it offers less expensive licensing than other operating systems like Mac.
2. **Operating Systems Architectures**: The Windows application programming interface (API) provides services used by all Windows-based applications to enable applications to provide a Graphical User Interface (GUI), access system resources, incorporate audio and much more. The architecture of Windows is a layered design that consists of two main components, user mode and kernel mode. User mode provides the interface between applications and kernel functions, known as the environment subsystem. Windows can have more than one of these, each implementing a different API set. This mechanism was designed to support applications written for many different types of operating systems. Kernel mode has full access to the hardware and system resources of the computer. Because of this, it controls access to critical areas of the operating system as user mode processes must ask the kernel mode to perform such operations on their behalf.
3. **Storage Management**: The game does not have massive storage requirements, as the main consideration is just the image library. Because of this, I recommend that The Gaming Room stores this library, as well as all user information on a cloud based server. Many of these servers are priced based on usage so this is probably the best decision for a game with low storage requirements like Draw It or Lose It. This will provide the company with the flexibility and scalability that it needs to continuously adapt to the needs of the game.
4. **Memory Management**: Each process on 32-bit Microsoft Windows has its own virtual address space that enables addressing up to 4 gigabytes of memory. Each process on 64-bit Windows has a virtual address space of 8 terabytes. Windows utilizes paging, a memory management scheme by which a computer stores and retrieves data from secondary storage for use in main memory. The mapping between virtual and physical addresses normally is done transparently to the running applications by the memory management unit (MMU), which is a dedicated part of the CPU. While Draw It or Lose It does not have massive storage requirements, managing memory is very important to ensure smooth gameplay. I recommend that The Gaming Room provides the server with sufficient RAM to make sure that images can be quickly rendered and avoid excess latency.
5. **Distributed Systems and Networks**: The Windows operating system provides mechanisms for facilitating communications and data sharing between applications. Collectively, the activities enabled by these mechanisms are called interprocess communications (IPC). Some forms of IPC facilitate the division of labor among several specialized processes. Other forms of IPC facilitate the division of labor among computers on a network. In a client/server environment, for clients to communicate with servers and servers to communicate with other servers, a mechanism must exist that can bridge multiple protocol and vendor issues. This mechanism is a layer of software called middleware. Middleware is software that provides common services and capabilities to applications outside of what’s offered by the operating system. Data management, application services, messaging, authentication, and API management are all commonly handled by middleware.
6. **Security**: Windows 10 and 11 include Windows Security, which provides the OS’s latest antivirus protection. Windows Security continually scans for malware, viruses, and security threats. Windows 11 is the most secure Windows OS yet, implementing new measures such as built-in advanced encryption and data protection, robust network and system security, and intelligent safeguards against ever-evolving threats. Microsoft also recently announced an important update to Microsoft Defender, which adds new cross platform mobile protections. It now supports vulnerability management for both Android and iOS, which the company says, “continuously monitors and identifies impacted devices, assesses associated risks in the environment, and provides intelligent prioritization and integrated workflows to seamlessly remediate vulnerabilities."