

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 2.0

## 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 |
| --- | --- | --- | --- |
| 1.0 | 05/18/23 | Lauren Weaver | Updated the executive summary, design constraints, and domain model. |
| 1.5 | 05/30/23 | Lauren Weaver | Updates the evaluation section. |
| 2.0 | 06/12/23 | Lauren Weaver | Updates the Recommendations section. |

**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 current version of the application was developed as an Android application. To convert the game to a web-based environment, a team of software developers may be needed to restructure the application. In this process, the application will be reconfigured to optimize program maintainability.

The Singleton pattern will be implemented to ensure only one instance of the Game class is in memory at a time, and the application will be tested to determine functionality on different web browsers.

## Requirements

* Teams must have more than one player.
* Application must handle single team and multi-team games.
* All game and team names must be unique.
* All games existing in the memory must be unique.
* Application must be web-based.

## [Design Constraints](#_2et92p0)

* Application structure must be compatible with a web-based environment.
* Ensure only one instance of Game exists in memory at any time.
* Confirm that application will function on all web browsers.

Since the game was originally developed as an Android application, a team of software developers may be needed to optimize the application’s functionality in a web-based environment. Within the application structure, the Singleton pattern can be used to control the number of Game class instances. As the application is converted to a web-based environment, the developers will need to determine which web browsers the application functions on. This may require additional testing to ensure functionality.

## [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)

Class Entity holds the attributes and functions common to the Game, Team, and Player classes. This allows them to use or overwrite Entity’s functions, reducing code and simplifying future maintenance. Within each class, attributes are declared private to encapsulate and protect information.

Between GameService, Game, Team, and Player there is a zero to many, represented by the “0…\*” symbol that builds through these classes. This results in necessary information being shared between classes without giving every class equal access to every other class, which protects unnecessary information from other classes while reducing the amount of code needed. The ProgramDriver class contains instantiates new Game and GameService objects. This class uses the SingletonTester class to ensure that only one instance of GameService can exist in memory at a time.

**"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](#_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** | Mac OS X offers server-based website hosting using the built-in Apache web server. Though this option comes with high licensing costs, it offers a stable website. | Linux offers a low-cost solution for hosting server-based websites. Though it is a stable platform for hosting sites, it is a difficult operating system to use. | While the licensing fees are high, Windows does offer server-based website hosting. Though this platform offers ease of use, the websites it hosts are often unstable and prone to errors. | Mobile devices can host web-based software through a browser app if the device has access to the internet. However, the software tends to perform poorly, especially when experiencing technical errors. |
| **Client Side** | Applications distributed on Mac must be built using Xcode 14.1. This includes SDKs compatible with Apple platforms. Applications built on Mac are also able to run on other operating systems. | Though Linux offers stable low-cost hosting, it is not compatible with clients developed on Windows operating system. Since Linux is open sourced, it is a low-cost option. | Web servers on Windows can have multiple clients. However, the licensing fees are upwards of $6,000, making it an expensive choice. | Mobile web servers are capable of hosting multiple clients with proper planning. Lightweight code and quick load time are essential to success on mobile operating systems. However, mobile web servers are often expensive to develop. |
| **Development Tools** | Developers with experience in Xcode 14.1 and the included SKDs compatible with Apple platforms. | Developers fluent in one of Linux’s scripting languages (such as PHP, Pearl, or MySQL). Developers will also need experience in Ubuntu Server or another one of Linux’s IDEs. | Developers with experience in some of Windows’ server hosting features such as ASP, .NET, Access, or MS SQL. Developers also need fluency in a Windows compatible language. | Developers with experience in writing lightweight code without compromising functionality. Developers will also need experience in a framework such as jQuery mobile or Sencha Touch. |

## 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**: For this project, using Linux as the operating platform would support the Draw It or Lose It application. Although it could be run on other operating platforms, Linux offers a highly secure network, and fast runtime that will aid Draw It or Lose It in expanding their application beyond mobile devices.
2. **Operating Systems Architectures**: The Linux operating platform is comprised of several layers. After the hardware layer is the kernel layer, which controls the hardware resources and software packages. Beyond this is the shell layer and the applications running on the operating system.
3. **Storage Management**: Linux uses a file directory to store non-volatile information. Components of this system include a root directory, which holds root directories in files. The file directory structure enables the system to be easily scaled up or down depending on the needs of the user application. This would aid Draw It or Lose It as they expand their picture directory.
4. **Memory Management**: The Linux operating platform uses a virtual memory management system. Addresses that are used by the application are given a physical address. This translation process is done by specific hardware in the lower levels on the system with the help of page tables. This management system will allow Draw It or Lose It to recall the images needed for their application without creating system overhead.
5. **Distributed Systems and Networks**: Distributed systems create an easily scalable base, and, since the systems run concurrently, system demands are completed efficiently. When using this system design with the Linux operating platform, the systems generally communicate through a link layer protocol. Though Linux can create secure networks between systems, there is a risk that some data will be lost during an unexpected power outage. Creating redundant network connections or using cloud-based storage are ways to prevent damage from these outages.
6. **Security**: Since Linux was designed with security in mind, its security systems are built into the kernel of the operating system, and, since Linux is open source, holes in the security are quickly found and repaired. One of the most notable security features is restricted access, which gives users only the access they need to complete their tasks. Along with this, password and IDs are used to authenticate users. To protect information, Linux encrypts all information on the system.