

<Draw It or Lose It>

# **CS 230 Project Software Design Template**

Version 1.0

## 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 | 01/26/25 | Marshall Cullars | Within this updated revision changes have been made to the executive summary, requirements, design constraints, evaluations, and 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 wants a web-based game styled after the 1980’s television game “Draw it or Lose it”. Instead of players drawing imagines on an easel for the hints the application will pull stock imagines from library for the clues. The game must also support multiple players with unique team names, with not empty teams or duplicate names.

## Requirements

*The finished game must be able to support more than one team at a time. Each team will have multiple players assigned to them. Their team names and games titles must be unique so that players can check if a game name is in use. Only one instance of a game can exists in memory at a time.*

## [Design Constraints](#_2et92p0)

1.**Web based application**: The game will be in a web-based environment so it must be developed to be compatible with all the popular web browsers and platforms.

2. **Server Size**: The game must be prepared for increased player traffic. To prevent performance issues and customer complaints, server optimization and data management should be a priority.

3. **Uniqueness**: The application will have to have a running database of active usernames. That way the random generator can check existing names and produce a statement notifying players if the name is in user.

4. **Security**: The game must be secure and withstand malicious attacks for hackers.

## [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)

Program Driver: This is the starting point of the application.

Singleton Tester: This is the tester for GameService.

Entity: Parent class for Game, Team, and Player

GameService: This class handles general game operations. This class has a relationship with Game, Team, and Player. GameService utilizes a Singleton Pattern so that only one GameService exists at a time.

Game: This class manages the game. It also has the ability to manipulate the teams list.

Team: This class manages the teams. It also has the ability to manipulate the player list.

Player: Holds the player ID to be used by its parent class.

OOP Principles

Inheritance: Entity demonstrates this principle being that is the base class for multiple other classes.

Encapsulation: GameServices demonstrates this principle. This is shown by the private and public variables of the class. The data in this class is changed by setters and getters but cannot be accessed or changed by other classes.

Polymorphism: This is shown by the classes of Game, Team, and Player. These classes have to string override to complete it purposes.

**"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 servers have high ratings for reliable and are generally stable. It is easy to use and flexible so scaling it up in the future will take minimal effort. One drawback of Mac is it can be more expensive than similar configurations from other companies. Also, it has combability issues with less common software. | Linux will be a stable and reliable environment. The Linux platform is rarely hacked so security should not be an issue. Linux is also a more complex system. Updates have the potential to be difficult and time consuming. | Windows is widely used. It is compatible with most software and web hosts. Technical support is readily assessable, and up-dates are quick and easy to perform. Windows is not suitable for multiple user systems. Also is highly susceptible to malware. Security may be a concern. | Mobile devices offer full control over the server and application. Also, it is easily scalable, so it leaves room for the company to grow. With mobile devices the performance of the application will depend on cell reception. |
| **Client Side** | The most popular browser used on MAC computers is Safari. It is a built-in browser that has come standard with the PC since 2003. Since this is the default browser most users it out of convenience. With development of a web based application focusing solely on safari could limit our audition. Safari is not a cross-platform browser and generally will only appear on apple products. Also some websites will load normally in other browser but fail in safari. Leaving our core audition to only be apple users. | Linux users most installed web browser is Mozilla FireFox. FireFox is an open-source browser which aligns with mindset of Linux. FireFox has a strong privacy focus and is equipped with “Enhanced Tracking Protection”. This feature blocks third-party trackers and cookies. Compared to a browser like Chrome is can experience longer loading times. Firefox has trouble handling complex web sites or multiple tabs open at once. This could hurt the performance of our game. | Windows most popular web browser is Google Chrome. Google Chrome has high ratings for speed and efficient. This will lend itself to our application being as there is a time based aspect to the game. It is also cross-platform being that it works on many devices. It does fall short on privacy and reliability. With a lot of users claiming that it crashes often and tracks user data for Ads. | Mobile applications most popular browser is Google Chrome. Chrome has made itself available for Android and IOS devices two of the most popular most device platforms on the market. The benefits and drawbacks will be the same as listed in the Windows column. |
| **Development Tools** | **Programming Languages**: C/C++, Python, JavaScript, Object-C, and Swift(Apples preferred)  **Tools:** SwiftUI, UIKit, Xcode, and Homebrew  Apple offers an Apple Developer Program. This program is 99 USD per year will assist in the development of the application. Apple has strict standard for applications and all software must be reviewed before launch. | **Programming Languages:** Python, Java, GO, Ruby, Scala, and C/C++  **Tools:** Git and Node.js  The most popular development tool for Linux is Git. Git is a free open-source VCS that will assist the team in managing and tracking changes to source code. | **Programming Languages:** Python, JavaScript, Java, and PHP.  **Tools:** WinUI, WPF, RAD Studio, Microsoft Visual Studio, and Apache NetBeans  With the popularity of Windows there is extensive documentation for deploying on Windows. Microsoft Visual Studio is the preferred tool for developers where it can run JavaScript. There will be no licensing cost attached. | **Programming Languages:** Python, C#, and C++.  **Tools**: Ionic, Flutter, Xcode, React Native, Xamarin, Dart, and Kotilin.  The Apple Developer Program will be required for mobile devices as well to be able to reach Iphone users. Xcode would be the go to tool to use for this process to deploy on IOS systems. |

## 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 recommend that the company uses Linux. Linux is open source and mostly free. It will involve some extra training to the development team being that Linux is the most complex of our operating option, but the company has already developed a Android based game. Android operating system is based on Linux kernel. Meaning the team can draw upon similarities and experience to learn Linux quickly. The small investment will yield us great returns in the future by using Linux.
2. **Operating Systems Architectures**: Linux is an open-source operating system. The main components of Linux are the kernel, hardware, and shell. The kernel is the core-operating system that handles hardware interactions. The hardware architecture consists of memory, CPU, I/O, and other physical devices. The shell is a piece of software that gives the user an interface to operate.
3. **Storage Management**: For the storage I recommend adapting a cloud-based storage system. Google cloud seems to be the best fit for our purposes. With using a cloud storage system, the company will not have to allocate resources to acquiring or maintaining physical storage devices. Also cloud storage are easily scalable up or down depending on our needs.
4. **Memory Management**: Linux memory management is handled by the kernel. The kernel will ensure that system resources are allocated appropriately. Memory management also has many configuration settings so that we can customize the operation to fit our needs.
5. **Distributed Systems and Networks**: WebSocket communication is supported across most modern web browsers. It is a communication protocol I recommend using. With this both the client and server can send information at the same. It can handle multiple data steams at once which would be ideal for our application since it will be multiple teams on the server at one time. Also, WebSocket communication is fast and good for real-time applications. Also recommend the use of Google Chrome as our primary browser. It has high cross-platform capabilities and available to a vast majority of people.
6. **Security**: Linux in general has high ratings for security. The built-in systems line Linux Kernal Lockdown will help to keep our information safe. For added protection Google cloud also comes with a host of security features. Encryption, security monitoring, and identity access are included. These features can be customized to fit our standard and security needs.