

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 | 3/16/23 | Josh Cantu | Create a game that can be accessed via the web, iOS, and Android |

**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 paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

Draw It or Lose It is a game like a show in the ‘80’s called Win, Lose, or Draw. Teams compete to guess what the person is drawing. The application is meant to render images from a large stock image library drawing as clues. The game will consist of four 1-minute rounds. The images are rendered at a steady pace and fully revealed in 30 seconds. If the puzzle is not guessed correctly before the time expires, the opposing team will be allowed to take one guess to solve the puzzle with a 15-second timer.

## Requirements

Draw It or Lose It will be a group game developed on the web. Their current game is only available for Android OS. It will be accessible to PC, iOS, and Android OS people. The game will allow one or more teams to play. Each team will be able to assign multiple players. Games and team names will be allowed to allow users to check if a name is in use when choosing team names. Only one instance of the game can exist in the memory at any time. Each game, team, or player will use a unique identifier.

## [Design Constraints](#_2et92p0)

Draw It or Lose It must be created on the web and accessed in the play store with the Android OS, iOS, and the web. The app will require internet access. The app will require a stable environment in most versions of Android and iOS. The budget must be respected and fit the project. The web server must be stable for multiple game sessions happening at once. Version from the Play Store and iOS must be stable when updated and deployed.

## [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)

The “ProgramDriver” class will be the main method. It will utilize the Directed Association with SingletonTester to test any game instances of GameServices. Game, Team, and Player classes fall under Entity. This makes the Entity the parent class. This also means the Game, Team, and Player get their required attributes from the Entity. This will have those classes separate from each other. The player cannot have a team, but a team can have a player. The game service will only have one instance of it running. There cannot be more than one instance running. Each game can only have one unique Team at any time. This also applies to each Team. They can only have one individual player at a time for each Team.

**"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 paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Works great with multiple iOS and macOS devices. Easy administration and great support. Unlimited license. Not cost-effective. Not many 3rd party applications. | Linux is less resource heavy. Low cost or free. The learning curve with the use of CLI. Flexible and customizable. | Wide distribution means more security threats. Ease of use. Requires a license. Closed source. Serval standard versions. | Not recommended for hosting web services. Phones do not have a proper NIC. The CPU on the device is not powerful enough to support multiple clients. |
| **Client Side** | Mac is optimized when used with other Mac devices. Since this is an optimization, it could rack up costs. Mac has a friendly interface. | Cost and time may be higher since it will require skilled personnel to optimize using code. | There is a security risk since this is a widely used OS. Since it is widely used most people feel comfortable using this OS. | Required development to display correctly on a phone. Optimization is required when there is a huge screen difference. |
| **Development Tools** | Mac utilized iCode. | Most code can be installed or already preinstalled on most distros. | Visual Studio would be best when coding in Windows. Most languages can be installed and utilized. | Both phone OS requires to be developed for their perspective OS. Android uses Android Studio and iPhones use iCode. C++ and Java are universal languages but not optimized. |

## 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**: Linux, I believe, would be better in this situation. With the ease of access and customization, it would handle most game sessions at once with no issues.
2. **Operating Systems Architectures**: Linux distro is stable and secure with a big community of support.
3. **Storage Management**: SSD are the latest and greatest. It is faster acting when gathering the data required. It can be more expensive, but HDDs are mechanical and can fail.
4. **Memory Management**: A watcher can be set up to monitor and utilize resources during peak sessions. This could optimize the user experience to allow little to no downtime.
5. **Distributed Systems and Networks**: Having redundancies would be best. Utilizing at least two servers to host the game would be best in this situation. If one server goes down, you can utilize all games on one server until the other is repaired or replaced. This can include upgrades for the servers. The load can go onto one server until the service is done. This could be done after peak hours.
6. **Security**: Role-based security systems will be best. Separating the roles will ensure the players cannot access any internal game files. This also ensures trusted admins are making changes to the game system. It could allow accountability during any game changes.