

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 | 11/15/24 | Jaden Williams | Update the information needed for this project, ie the requirements, constraints, etc. |

**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 client wishes to make the “Draw it or Lose it” android game into a web-based multi-platform game. They are seeking help to streamline the development process, and to aid in the new platform development. Each game and team name will need to be unique, while only having one instance of the game at a time.

## Requirements

The client requires the game to have a single team, or multiple teams. Each team will have multiple players. Only one instance of the game must exist at a time. The system will check for unique names when the game is created to ensure a new game is running.

## [Design Constraints](#_2et92p0)

Meeting each of the client's design requirements is of utmost importance. One requirement is cross platform. This will prove exceptionally challenging making sure the web-based system works with every OS.

## [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)

**Entity is the Parent class of the system. It is parent to the Game, Team, and Player classes. These classes will inherit Entity's attributes, but each can have attributes of their own. The Game Service Class is there to ensure only a single instance of the game is running at a time, with a unique game, player and team name. Program Driver uses the Singleton Tester class and has the main statement within it.**

**"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** | MacOS is a well-managed and structured OS, ensuring maximum stability and consistency. Mac is expensive and has very limited hardware options. | Open sourced and has a wide range of options to choose from. Easy to customize and great for server use. Linux rarely comes in any prebuilt machines, so it is a custom build PC only | Integration with servers right out of the box. Windows does not have the best security. Most PCs come with this OS installed, so ease of use is a big factor. | Cannot host many users at a time. It is great for easy, rapid cloud development with little users. However mobile is more vulnerable than other platforms. |
| **Client Side** | MacOS is expensive due to it only being on Apple devices. This limits knowledgeable devs and may limit workflow ability. | Affordability is huge with Linux. The upkeep is generally easier, and many people know Linux well due to it being free. It may take some time to train new people how to use it though. | Windows is widely known and used. Nearly everyone who has used a computer has used windows, so a knowledge base is there. It would take little time or cost to use it. | Mobile lacks the power to utilize all the tools that a PC has access to. They are very cheap and accessible at low price points; however, OSs are specific in most scenarios making them each slightly different. |
| **Development Tools** | Swift and Xcode are the main language and IDE used with Apple. It makes for a quick and easy Apple geared development process. | Linux has many many different IDE’s available, each of which can offer a different experience. Docker Hub is one that comes to mind, it's a simple and easy to use environment that offers a ready to use platform. | Visual Studio is a free (In some cases) IDE. It is easy to use and was developed to be beginner friendly. It can be used for many different programming languages, from C, C++, Python, Java, ETC. | Java, C++, and Python are primary languages used in mobile development. Eclipse is the first IDE that comes to mind that supports mobile development. |

## 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**: Windows. Its ease of use, and widespread use makes it ideal. The current Android version of “Draw it Or Lose it” can be integrated into windows for ease of development. There will be many more developers and tools that can be used for the project.
2. **Operating Systems Architectures**: Windows 11 has a beautiful system called Universal Windows Platform. “UWP is one choice for creating apps that run on Windows 10 and Windows 11 devices and can be combined with other platforms. UWP apps can make use of Win32 APIs and .NET classes” (Microsoft, 2024)
3. **Storage Management**: Cloud/Server based storage will be critical to allow ease of access to files in a centralized format. It also adds redundancy to information storage to ensure no data is lost.
4. **Memory Management**: Memory is allocated in physical and virtual locations to ensure redundancy and no loss memory. OneDrive is a great, paid for, option for cloud bases memory storage as well.
5. **Distributed Systems and Networks**: If we consider what the client believes the projected size of the user base will be, then we can effectively create servers that will not reach capacity for some time as they will be “overbuilt” A cross-platform IDE will be needed to ensure maximum efficiency as well as easy cross-platform maintainability.
6. **Security**: Using a standard OS security will be inadequate in protecting the users’ information. Finding a third party, paid-for security service will be massively important for the information to remain secure.