

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## Table of Contents

[CS 230 Project Software Design Template 1](#_heading=h.gjdgxs)

[Table of Contents 1](#_heading=h.30j0zll)

[Document Revision History 2](#_heading=h.3znysh7)

[Executive Summary 2](#_heading=h.tyjcwt)

[Requirements 3](#_heading=h.3dy6vkm)

[Design Constraints 3](#_heading=h.1t3h5sf)

[System Architecture View 3](#_heading=h.4d34og8)

[Domain Model 3](#_heading=h.2s8eyo1)

[Evaluation 4](#_heading=h.17dp8vu)

[Recommendations 5](#_heading=h.26in1rg)

## [Document Revision History](#_heading=h.lnxbz9)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 5/28/2024 | Jarvis Brown | Design template updated to cover changes and recommendations to client for Draw It or Lose It. Updates covered are Executive Summary, Design Constraints, Requirements, Evaluation and Recommendations. |
| 1.1 | 6/9/2024 | Jarvis Brown | Updated the evaluated the inputs and added more details of the information provided. |
| 1.2 | 6/23/2024 | Jarvis Brown | Reviewed recommendations, based on previous guidance for clear instructions details for 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.35nkun2)

Creative Technology Solutions is seeking to expand a popular game that is currently only available on android devices. The company wants to develop a web-based game that will be available to multiple platforms. The game consists of four rounds that last one minute. Drawings showcase at a steady pace and teams must guess the puzzle before time expires. As the client wishes the game to be able to be cross-platform and evaluation of different operating systems will be provided with recommendation based on current understanding of the clients requirements.

## Requirements

*\*The game will allow multiple teams to participate*

*\*Teams will have multiple players assigned*

*\*System should limit names to not allow the same name multiple times*

*\*One instance of the game should exist in memory at any given time. Using unique identifiers can accomplish this.*

## [Design Constraints](#_heading=h.1ksv4uv)

\*Must run on multiple Operating systems. This can be accomplished by incorporating web-based computer language(s).

\*Teams should have multiple players but will need to cap the team members based on the number of players per game.

\*Server Capacity will need to determine what are the limitations of the server.

\*Naming conventions should be unique to avoid duplicate names of Games and team members only. As Games will be started from multiple locations, the Game’s Name for the group using that specific game should limit but recommend incorporating invitation codes to ensure people join the correct game.

## [System Architecture View](#_heading=h.44sinio)

## [Domain Model](#_heading=h.2jxsxqh)

The class Entity is created with several attributes and methods. This class is a parent class to the GameService, Game, Team and Player classes as these classes all inherit the public methods from the Entity class. The diagram shows how multiplicity is reflected. The classes that have “0...\*” between them tells that there are zero too many relationships between those classes.

**"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.z337ya)

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** | Platform great for web hosting, is user friendly and popular among many new developers. Limited hardware and flexible options when compared to windows and linux. Multiple teams will be needed to understand IOS coding practices and general coding practices like Java when operating on IOS devices. If using MAC services generally you can find free hosting but if going through third parties may incur more cost. Some programs are not available to use on IOS and may require implementation through Virtual machines. | Easier use for open source and customization. Cost can be lower than Mac and Windows. Due to the vast flexibility, the learning curve can be more of an issue than dealing with windows or Mac. Next to IOS platforms, the security functions are still considered top notch and always being updated. Linux is more developer friendly and you can generally find adequate open source services that can rival that of IOS platforms | Commonly used for windows servers. Strong support for Microsoft technologies. Easy use of third-party software. Higher target for malware. Easier to find a team that understands windows systems as long as basic coding understanding is under their library of knowledge. Windows offers some limited free host services such as Microsoft Azure which can be anywhere from 30 days to 1 year depending. Extra cost may be incurred depending on the technical requirements. | Portability allows for mobile testing, but due to limited capabilities such as RAM and memory, it will not have continued operations like the others. The cost will vary depending on the type of device used such as IOS or Android but other than general free software like IOS’ Xcode or Android Studio other software will incur extra cost. |
| **Client Side** | Popular with many developers but the cost, limitations and restrictions may not allow the proper implementation or changes to the platform. Certain software and programs needed to run the program may not be accessible. If willing to forgo some capabilities but for tighter security will be a better option. | Will require dedicated personnel to learn the software but will have the best longer-term results in terms of future customizations. Will offer best development for personnel and cost for the company. Can be branched into windows to some degree. | Will not require much effort to train personnel but the security and vetting of third-party software may incur unpredictable cost to ensure privacy and system integrity concerns are addressed. | Unless simple tasks are needed. This will provide more difficulty to implement as mobile devices do not have the memory, or processing power. Would not be valuable to rely on. |
| **Development Tools** | Uses Swift, Java, python, and C++ as some of the programming languages. Some of the IDES supported are Eclipse, Xcode and Visual Studio Code and other development tools like AWS and AZURE. | Programming languages include Java, Python, C++, Kolin, etc. IDES include Eclipse, android studio, visual studio code, intellij IDEA and other tools like Android SDK, Apache Cordova.. | Programming languages include Java, Python, C++, Kolin, etc. IDES include Eclipse, android studio, visual studio code, and other tools like Windows SDK, Android SDK. | Programming languages differ depending on the mobile device. IOs can use some MacOs languages while Android can use some of the common Windows languages like Java. IDES Android studio for droids and Xcode for iOS. Other tools such as google play store or App 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**: The recommended platform to operate The Draw It or Lose It game would be Linux. Linux may have a learning curve for some but due to its flexibility and customization it has better terms of security with less restrictions than the other operating systems.
2. **Operating Systems Architectures**: Linux is a layered structure where is main system is its kernel that supports Symmetric multiprocessing with each processor performing every task in the operating System. Linux is lower in cost since it is an open source based operating system. With its versatility, Linux can adapt to almost any environment be it hosting through virtual machines or desktop server hosting.
3. **Storage Management**: Linux uses logical volume management or LVM. It's a robust storage management technology and it allows virtual storage. The storage devices gather and allocate storage resources.
4. **Memory Management**: Linux memory management allocates physical memory and handles virtual memory. Linux employs swapping and paging for memory management which will reduce the amount of memory needed for Draw It or Lose It so only chunks of memory are needed for the application.
5. **Distributed Systems and Networks**: Linux uses Microservices, a software architecture for complex web applications. Like the cloud, this would benefit the client's Draw It or Lose It game as it would be able to communicate on different platforms that can access networks as microservices are network-based services. Microservices allows individual services to be managed independently. Using lightweight protocols this ensures that regardless of the platform all services can interact with no issues. Linux’s adaptability allows you design the system to avoid single points of network failures or outages.
6. **Security**: Security is a must-have for the client. Linux uses SELinux as a security feature that is ingrained in the Linux Kernel. This uses the Mandatory Access Control (MAC) system which gives administrators control to access system resources and isolate issues. Being an open source OS, security features for Linux are constantly improving thanks to developers. Having encryptions will ensure data can’t be taken from unauthorized users. Implementing OAuth2 is a secure authorization that can limit access and make sure users can’t access others information.