

Draw It Or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 01/25/2022 | Chris Cole | Edits to executive summary, design constraints, and domain model, recommendations completed. |

**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 is developing a web-based game that serves multiple platforms based on the current game, Draw It or Lose It. Currently only available on Android, the web-based version will be available for all OS platforms. The game will have the ability to have one or more teams involved. Each team will have multiple players assigned to it. Game and team names will be unique to allow users to check whether a name is in use when choosing a team name. Only one instance of the game will exist in memory at any given time. This will be accomplished by creating unique identifiers for each instance of a game, team, or player. Each round of the game will have set time parameters in which the starting team will be shown images and given the chance to guess what they are. The opposing team will get a chance to guess if the starting team is unsuccessful during their time period.

## [Design Constraints](#_2et92p0)

* Must run on multiple platforms.
* Architecture for 1 or more teams.
* Server space for game files.
* Each team is made up of multiple individuals.
* In order to ensure only one instance of the game is stored in memory unique game and team names are necessary.

These design constraints will affect the source code and the programming language. Java typically runs on the Java Virtual Machine (JVM), which can be integrated into multiple environments. The team and instance infrastructure will be implemented using Java. This software and the associated image files must be hosted on a server in order to give access to multiple clients.

## [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)

Game, Team, and Player classes all inherit from the Entity class. The Game class references the Team class, the Team class references the Player class. The GameService class has references to Game, Team, and Player classes. GameService features overloading, demonstrating polymorphism. There are many examples of encapsulation in the UML diagram, for example, splitting Game, Team, and Player into separate 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](#_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** | Offers a server based deployment method. Works very well within the Mac ecosystem. MacOS has the highest associated costs. User friendly, but not scalable. Least popular OS on the list, less software and tools available than other platforms. | Offers a server based deployment method. Linux is a low cost, open- source platform. Many popular software tools are available, scripts can be written in PHP, Python, and other Unix based languages. Offers high level of security. Linux is the most popular and most scalable option. Specialized expertise is required, Linux is generally less user friendly. | Offers a server based deployment method. High costs and licensing fees. Many popular software tools available, Windows supports ASP and .NET. Services are easy to use. Average security level, most frequently attacked. Frequent rebooting is necessary to run properly, generally not as scalable as Linux. | Offers a server based deployment method. Portability and scalability. As portable devices become more ubiquitous, software solutions will become more robust. Cloud cluster server use will only increase in the near future. Unfortunately, mobile devices will not be a viable choice for hosting this webapp. |
| **Client Side** | Low to moderate expertise needed. High cost, high level of available support. HTML/HTTP are both supported through browsers such as Safari, or Chrome. | High level of expertise needed. Low cost, less support than other platforms. HTML/HTTP are both supported through browsers such as Opera or Chromium. | Low to moderate level of expertise needed. High cost, high level of available support. HTML/HTTP are both supported through browsers such as Edge or Chrome. | Moderate level of expertise needed. Highly portable and scalable. Low cost, support available. HTML/HTTP are both supported through browser. Android support already available on many clients. |
| **Development Tools** | Xcode is the IDE used to code in Swift for Apple platforms. IntelliJ IDEA is a great Java IDE that also incorporates Apache 2. Licensing costs are high. Most modern programming languages are supported. | Eclipse, IntelliJ IDEA, and NetBeans are the big 3 Java IDEs for Linux. Apache will be very useful for this application. Open-source software limits costs. Most modern programming languages are supported. | Eclipse, IntelliJ IDEA, and NetBeans are also the big 3 IDEs for Windows. Licensing costs are high. Most modern programming languages are supported. | Eclipse, Xcode, and Android Studio are all available and supported on mobile platforms. Open-source software limits costs. Most modern programming languages are supported. The early stage of development on mobile makes this platform a poor choice for webapp 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**: I recommend the Linux OS for the Draw It Or Lose It application. Linux is open source and the licenses are free so scaling will be more cost effective. Linux provides the most flexible system architecture allowing for easy implementation on most hardware. It also provides the best security and most diverse developer tools of any platform.
2. **Operating Systems Architectures**: Linux has an open-source kernel that is very popular for servers. It uses a tree structure to store and organize files. It supports three types of users, regular, root, and service. Almost all utilities and libraries are free to use and edit. This system is very popular as a server because of its reliability, security, and uptime.
3. **Storage Management**: It is important to make sure the files are stored in the proper format and size for this application. The image files can be compressed to save storage space and the JPG format would be compatible with all platforms. The files can also be partitioned for higher security and access protocols. The storage could be provided by any of the big three providers of cloud storage, AWS, Azure, or Google for roughly the same price. Scaling would be efficient and cost effective.
4. **Memory Management**: Using the built-in memory management of Java will be key to developing a fast application. Each browser has different parameters for memory usage that must be accounted for. Properly storing, compressing, and delivering images will be the key to keeping the required resources minimized for each client.
5. **Distributed Systems and Networks**: One of the best features of Linux is its containerization and compatibility with Kubernetes. Container orchestration by Kubernetes can automate the deployment, networking, scalability, and availability of the application. The distribution of nodes makes the system more reliable and robust. Kubernetes has been adopted by all public clouds and is excellent for scaling up or down efficiently as needed.
6. **Security**: Linux provides full HTTPS support and native encryption. Strong passwords and multi-factor authorization are recommended. Basic update standards and testing will be followed. Strong access protections are provided by the OS. Additionally, the cloud storage solutions used provide security automation. Each client browser has its own security features that work to secure the total server client system.