

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 | 01/22/23 | Matt McCormick | Updated Executive Summary, Requirements, Design Constraints, System Architecture View, Domain Model, Evaluation and recommendations for the Draw It or Loose It application. |

**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 has requested for a web-based game called “Draw It or Lose It” to be developed. “Draw it or Lose It’ will be a multiplayer game that can be accessed from multiple platforms. The web-based game will be developed in Java so that multiple operating software can access it. The Java file will be set up utilizing OOP (object-oriented programming) to ensure efficient and secure coding.

## Requirements

The Gaming Room “Draw it or Lose It” web-based application will be accessed from multiple operation systems. The game will allow a multiplayer team system. The game will go four rounds of play lasting one minute each. The program will render images from a library of stock drawings. Players will then have to guess the image within 30 seconds. If the team does not guess within 30 second, the other team has the option to guess the image within 15 seconds. The game and team will be unique for each user.

## [Design Constraints](#_2et92p0)

The following design constraints have been identified for the “Draw it or Lose it” web-base game :

* Application to be developed for multiple operating systems.
* Game will allow multiplier use.
* Game and team names will have to be unique to a user. This will allow them to check if a name is in use when picking a team.
* Only one instance of the game can exist in the stored memory at a given time. Code will have to create new identifiers for each game instance, player and team.

## [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 Gaming Room “Draw it or Lose It” web-based application will utilize OOP principles. The Entity class is a superclass the builds a relationship between the Game, Team, and Player class. Game, Team, and Player will generally inherit the information within the Entity class. These variables that will be shared are the name and id variables. Player, Team and Game, are a “Has A” type of relationship. These are shared zero to many times. The “Has A” relationship shares the instances between the classes to account for the game, team and player. The Game class references the Team array list to verify instances. The team class references the Player array list to verify instances. The ProgramDriver Class inherits the testSingleton function from the SingletonTester class.

**"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 web hosting to clients. It is a secure web hosting system but is more costly than others. Has ability to be upgrade if needed. Settings on server can also be changed easy if needed. Mac has less security issues compared to windows. One down fall it is a less preferred web hosting service based on cost and less software’s work with mac.  Server needs are that it must have ability to send universal language back to different operation systems. Needs to be able to handle large player base. | Offers web hosting to clients. It has a good security system and usually catches flaws before them become an issue. It is one of the more popular web hosting OS systems. It tends to be cheaper than Mac but not Windows. It can be difficult to find applications that support the requirements for web hosting.  Server needs are that it must have ability to send universal language back to different operation systems. Needs to be able to handle large player base. | Offers web hosting to clients. It has a poor security system. It tends to be a cheaper option compared to Mac and Linux. Can be easily compromised and have virus spread onto system. It has more software compatibility compared to Linux and Mac. Requiring high resource requirement which allows low loading time to a smooth playing experience.  Server needs are that it must have ability to send universal language back to different operation systems. Needs to be able to handle large player base. | Does not offer web hosting. Servers are best if they are in one location and not mobile. Other OS systems tend to be better as those require better hardware to run. Will not be able to support a large client base. |
| **Client Side** | Moderate expertise is required to developing the client side of the program. Coding will have to accept HTTP language from the server since the server will be set for multiple OS systems. Program will have to send inputs to server side. Time and resources would be based on client deadlines and software specifications. Cost effective since server side needs the hardware power to complete the program. | Maximum expertise is required to developing the client side of the program. Coding will have to accept HTTP language from the server since the server will be set for multiple OS systems. Program will have to send inputs to server side. Time and resources would be based on client deadlines and software specifications. Cost effective since server side needs the hardware power to complete the program. | Minimum expertise is required to developing the client side of the program. Coding will have to accept HTTP language from the server since the server will be set for multiple OS systems. Program will have to send inputs to server side. Time and resources would be based on client deadlines and software specifications. Cost effective since server side needs the hardware power to complete the program. | Moderate expertise required. Will need ability to read HTTP language from server and be able to send inputs back. Needs little hardware since server side will complete the program. |
| **Development Tools** | Macs can run most major programming languages like Python, java and C++. Additionally it can support HTML, CSS and Javasctipt. It allows for IDE’s to have supporting libraries that are useful for programming. Most of the major IDE’s like Pycharm, Visual Studio and Eclipse are compatible with Mac. Depending on level of complexity of a project and timeline, additional resources may or may not need to be brought into complete a project. Python and C++ have large libraries to help support front end work while Java is all class and object building. | Like Macs, Linux can run most major programming languages like Python, java and C++. It can also support HTML, CSS and Javasctipt. It allows for IDE’s to have supporting libraries that are useful for programming. Most of the major IDE’s like Pycharm, Visual Studio and Eclipse are compatible with Mac. Depending on level of complexity of a project and timeline, additional resources may or may not need to be brought into complete a project. Python and C++ have large libraries to help support front end work while Java is all class and object building. | Like Macs and Linux, Windows can run most major programming languages like Python, java and C++. It can also support HTML, CSS and Javasctipt. It allows for IDE’s to have supporting libraries that are useful for programming. Most of the major IDE’s like Pycharm, Visual Studio and Eclipse are compatible with Mac. Depending on level of complexity of a project and timeline, additional resources may or may not need to be brought into complete a project. Python and C++ have large libraries to help support front end work while Java is all class and object building. | Android and swift can both be used for mobile apps. Both of these languages can be ran on Windows, Mac and Linux. It can also support Java, Python and C++. |

## 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 best operation platform for the server would be Linux. This isn’t the most popular operating platform for personal computers but would work good for a server operating platform. This tends to be cheaper than MAC but has a better securing system than windows. With server language being transferred back to the client operation system. It will allow multiple users operation systems to uses this server.
2. **Operating Systems Architectures**: The Linux kernel is stable and secure. The Kubernetes clusters allow easy separation of system and even hardware needs.
3. **Storage Management**: The server can be set up with a SSD or cloud storage. A hard drive in this application can be slow an cause game lag. The cloud storage is not ideal either but can be cost saving depending on how large the game is. The SSD is the best choice based on speed. This will allow a small amount of load time between pictures/game sessions. This will also be best for a seamless gaming experience.
4. **Memory Management**: Linux will allow the operation system to change its required memory based on the needed usage. This allows the system to prioritize the game as needed to allow for the best gaming experience. If there is multiple software’s running at once, the system will prioritize the software that is set to the highest priority.
5. **Distributed Systems and Networks**: With the host server being ran with Linux, the server would have to be able to transfer the data to multiple operating systems. This can be accomplished with REST (Representation State Transfer) approach. This allows a universal language between multiple operating systems. The Host server will send back a JSON files that contain dynamic data to the different operating systems. The client side will send in a input which the host server will be able to interpret based on what type of programming language that the gaming was created in.
6. **Security**: The security system for the host server for “Draw it or Lose it” should be a user based security system. This will allow users who are accessed to the server only allow modifications to the servers based on their role. If they don’t have admit rights, they won’t be able to make modifications. The server settings can also be set up with two set authentications to verify the admin accessing the server.