

**The Gaming Room**

# **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 | 03/19/2025 | Zachary Lecroy | Updates were made to the cover page, document revision history, summary, and other important areas. |
| 1.1 | 04/04/2025 | Zachary Lecroy | Updates made to evaluation and server / client-side chart. |
| 1.2 | 4/17/2025 | Zachary Lecroy | Updates made to the recommendation section. |

**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 wants to expand their Android game, “Draw it or Lose it,” into a web application making it available to multiple platforms. This is in hopes to broaden the games audience and make it more accessible. The purpose of the game is to have one or more teams with several players. The app will create images and give the teams one minute to guess before the puzzle expires. If they are unable to guess the image, then the other teams have a chance to steal with less time.

## [Design Constraints](#_2et92p0)

* *Teams must allow multiple players*
* *Must run on multiple platforms to increase traffic*
* *One instance of the game at a time*
* *Game and team names must be unique*
* *Allow players to check what team names are in use*
* *Optimize allowing multiple game sessions to run smoothly*

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

The base class or Entity is the building block and creates a relationship between Game, Team, and the Player classes. They all inherit information from Entity. With this UML diagram we can see the inheritance which makes Entity a superclass. The Game class incorporates the full game setup with everything needed to play a game. The Team class takes care of all team tracking such as names and players on the team. The Player keeps track of all players. In this UML diagram we see the GameService has a reference of Game, Game has a reference of Team, and Team has a reference of Player.

**"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** | Easy accessibility and server configurations.  It has different options for web hosting and is upgradable.  Good graphical user interface.  The weakness would be that it is less desirable for web hosting. | Basically, all the same as mac but even more cost friendly.  Has less security flaws and it is a preferred choice for web hosting services.  The weakness would be it is more difficult to find applications to support the web hosting. | There is more software available compared to the other OS’s. Some advantages include less loading time and high comfortability with users.  Weaknesses include virus susceptibility, expensive, and bad tech support. | Mobile device specifications vary from user to user. These are highly portable and popular. This has a wider reach and better cost effectiveness.  A disadvantage would be the poor security. |
| **Client Side** | This is expensive for users and usability can differ compared to most OS. Moderate time and expertise required. | This has moderate to high expertise required. Extremely cost-effective option. | This is minimum expertise and time required as this is the most familiar option for the masses. Similar cost to mac with some lower end options. | This option provides flexibility for clients and developers. This makes it so they can see updates anywhere with high portability. Expertise and time vary by device. |
| **Development Tools** | We can run swift on mac devices, but mac can run all languages including Java, CSS, Python, C++. Some tools used to develop would be Clion, VScode, Pycharm, and Sublimetext. | Linux can work with eclipse, and VScode. Linux can run most languages without much issue but in my experience, it is slightly more difficult to setup. | The windows option is the same as Linux and mac but is an easier setup. | Using android and swift you can create many apps. Both languages can be run on all other machines which makes it easy to switch development locations. |

## 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**: For The Gaming Room to effectively expand Draw It or Lose It across multiple computing environments, we recommend adopting a cloud-based server platform using Linux hosted on Amazon Web Services. This combination offers scalability, cost-efficiency, cross-platform support, and wide compatibility with other systems and development frameworks.
2. **Operating Systems Architectures**: Linux is a strong and flexible system that powers many major websites and apps. It runs well on modern hardware and is ideal for web based games. It allows The Gaming Room to easily adjust system resources as the game grows and supports modern tools for software development and security.
3. **Storage Management**: For storing game data, user information, and media (like images or drawings), we suggest using AWS storage services. These tools automatically grow as needed and keep data safe and backed up. This makes sure the game can run smoothly and continue operating even as more users join.
4. **Memory Management**: Linux handles memory in a smart and efficient way. It keeps the game running fast by giving the right amount of memory to each part of the game as needed. If things get too busy, it knows how to manage that load and prevent crashes or slowdowns, helping Draw It or Lose It run reliably even when many people are playing at once.
5. **Distributed Systems and Networks**: To make sure Draw It or Lose It works across different devices like phones, tablets, and computers, we recommend using a system design where different parts of the game can communicate with each other over the internet. This design allows users to play together no matter what device they’re on. If something goes wrong in one part of the system, the rest of the game can keep running without interruption.
6. **Security**: To keep players’ information safe, we recommend using secure connections, encrypted data storage, and reliable login systems. The platform will also include built-in protections against common online threats and regular monitoring to quickly detect and respond to any issues. These steps will help ensure a safe and trustworthy experience for all users.