

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 | 11/7/24 | Hunter Prince | Changes were made to cover page, document revision history, executive summary, design constraints, system architecture view, and domain model and 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](#_sbfa50wo7nsh)

The Gaming Room project is to develop a web-based game that can serve multiple platforms based on the current game draw it, or lose it, which currently is only available to android users. The purpose of the game is to have multiple teams consisting of upwards to seven people, going four rounds that are a minute each. When a picture is pulled from the library of images one team guesses till the time runs out, if not answered correctly, opposing members get 15 seconds to try and answer until time runs out.

## Requirements

The requirements for such a thing aren’t over the rough, you will require the technical knowledge to host The Gaming Room, and the ability to service multiple platforms. The game is also not stressful and should run on minimum requirements and software to function across various devices, though PC will likely give best results.

## [Design Constraints](#_2et92p0)

1. Must run on multiple platforms.
2. Each team requires multiple players, upwards to seven or so.
3. Only one instance of the game can exist at any time.
4. Game and team names must be unique and allow users to check whether a name is already being used by another.

## [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 create a relationship between Game, Team, and Player classes, this means they will inherit or get information from Entity. UML let’s us show this inheritance, making Entity the superclass, so that when we look at their relationships, we’ll see that Team and Player is a “has a” type. While Game has a Team and GameService has Game. Here, you have aggregation (HAS-A), when a user “has a” it’s an instance of one class and has a reference to an instance to another class. Then, when we look at this diagram, we see GameService has reference of Games, Games then is referenced by Teams, and then Team to Players.

**"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** | Mac has easier accessibility settings, and server config. It also has an extremely user-friendly GUI, and flexible terminal commands. | Cost Friendly, yet difficult for more people to navigate. However, the command shell makes for easy server access and setting configuration. | Server side is expensive, with extremely friendly user settings and GUI, and has a built-in command prompt. | Specifications are better in some devices, and this varies from user to user in how they user the device. |
| **Client Side** | Expensive for most users, the cost of the brand is high. User also needs some expertise in navigating the device, and it’s OS. | There is a lot of expertise in using Linux, as it’s extremely niche. Linux data is also required to use the OS, and it can have a high cost for entry. | More expensive than Linux, less expensive than Mac systems. Easier to learn than most, and widely used while also requiring minimum knowledge to set up. | Extremely flexible with clients, and developers to see any update that is put out. However, requires a certain skill set to develop for and implement things across devices. |
| **Development Tools** | Languages like HTML, CSS, and Java. Has libraries to help frontend development, and other dev tools like PyCharm, GitHub, Visual Studio, and more. | Languages like HTML, CSS, and Java. Libraries to support frontend, and languages. Linux systems include Java, Ruby, PHP, and Python. | Language like HTML, CSS, and Java. Libraries that support frontend and languages. Developer tools like Eclipse, command prompt, Pycharm, Visual Studio, and more. | Language that consists of HTML, CSS, and Java. Libraries that support frontend and languages. IDE’s for programming, can use things like HTML, php, C++, and Python. |

## 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 most appropriate operating platform for the Gaming Room to expand Draw it or lose it to other environments is Windows, as Windows is both widely used across many devices, and has many IDE’s.
2. **Operating Systems Architectures**: Windows is a graphical operating system, developed and published by Microsoft, and provides everything needed from storing files, to playing games, and connecting to the internet, and more.
3. **Storage Management**: Windows 10 comes with a feature called storage sense, this allows you to manage files on your hard drive, and see how much space these files take up, along with what else is on the hard drive.
4. **Memory Management**: Windows 10 storage sense would allow for storage and management of Draw it or Lose it photos, and game players, it also would let them all be stored in a single place.
5. **Distributed Systems and Networks**: Network based multiple players, or user interaction often have a database shared amongst the players that are distributed and then interact with one another over the same network. As it stands, making a game on a network requires you to implement a shared database and interlayer communication from scratch.
6. **Security**: Windows comes with great built-in security software; windows defender is more commonly used these days compared to some more well-known cyber security software names. However securing user data and information, it would be recommended to ensure files can be encrypted, and that public data is protected.