

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 | 7/15/2023 | Jayden Cooley | Changes had been made to the cover page, document revision history, executive summary, design constraints, system architectural view, domain model, and recommendation. |
| 1.1 | 7/29/2023 | Jayden Cooley | Evaluation and other updates to have an accurate draft. |
| 1.2 | 8/12/2023 | Jayden Cooley | Added Recommendations to the Design Template. |

**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 a web-based game that serves multiple platforms based on the current game “Draw it or Lose it”, which is only available on android at this time. The premise of the game has multiple teams consisting of several people going four rounds with a minute each. When a picture is pulled from the library of images, one team guesses until the time runs out. If not answered, the other teams get to answer until the timer runs out.

## [Design Constraints](#_2et92p0)

* It must run on multiple platforms.
* Each team should have multiple players.
* Only one instance of the game can exist at any time.
* Game and team names must be unique to allow users to check whether a name is in use when choosing an online team name.

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

Entity creates a relationship between the Game, Team, and Player classes. This means they all inherit or get information from the Entity. With UML, we can show this with inheritance and make Entity a superclass. When looking at the relationship, we see the Team and Player classes are “has a” types while the Game class has Team and GameService has Game. When we use UML, we call it aggregation (HAS-A). When we look at this diagram, we see GameService has a reference of Games, Games a reference of Teams, and Team a reference of 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** | When we use UML, we call it aggregation (HAS-A). When a user “has a” I mean  it's an instance of one class and has a reference to an instance to another class. When we look at this  diagram, we see GameService has a reference of Games, Games a reference of Teams, and Team a  reference of Players  Mac has easy  accessibility and  server  configurability.  Easy to use  graphical user  interface.  Flexible terminal  commands.  Mac has easy accessibility and server configurability. Easy to use graphical user interface. Flexible terminal commands. | Cost friendly, but difficult to navigate. Command shell for simple server configuration and accessibility. | Server side is expensive but has user friendly GUI and command prompts. | Specifications are better in other devices and they differ between user. |
| **Client Side** | Expensive for users. Moderate time and expertise is required. Accurate skills needed to navigate OS. | There is a lot of expertise and time required. Linux data is required to use the operating system. Maximum cost on Linux users. | More expensive than Linux but easy to learn and understand how to support Windows. Minimum expertise needed. | Provide flexibility to clients and developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | Languages that consist of HTML, CSS, and JavaScript. Libraries to support frontend development. Other development tools include PyCharm, GitHub, Visual Studios, and more. | Languages that consist of HTML, CSS and JavaScript. Libraries to support frontend and languages. Other. Linux systems include JavaScript, Ruby, PHP and Python. | Languages that consist of HTML, CSS and JavaScript. Libraries to support frontend and languages. Developer tools include Eclipse, command prompt, PyCharm, Eclipse, and more. | Languages that consist of HTML, CSS and JavaScript. Libraries to support frontend and languages. IDE’s for programming languages consist of 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 that will allow The Gaming Room to expand Draw It or Lose It to other computer environments is Windows. Windows has the biggest install base and won’t run into a shortage of IDE’s to utilize.
2. **Operating Systems Architectures**: Microsoft Windows is a graphical operating system developed and published by Microsoft It provides ways to store files, run software, play games, watch videos, and connect to the internet easily and quickly.
3. **Storage Management**: Windows 10 comes with a feature called “Storage Sense”, essentially sensing and seeing how much storage is used up by duplicate items and junk.
4. **Memory Management**: Windows 10 Storage Sense would allow for storage management of Draw It or Lose It’s photos and players, allowing you to easily keep them secure and tracked.
5. **Distributed Systems and Networks**: Network based multiplayer such as network games typically include a database shared among players that are physically distributed and interact with one another over the network, currently having developers implement the shared database and the inter player communications from scratch.
6. **Security**: Windows comes with built-in security protection software. Though to secure user data and information it would be recommended to use another source.