

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 | 03/24/2025 | Darius Wilder | There were changes to the cover page, executive summary, constraints on design, system architecture view, and domain model. |

**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 objective of The Game Room project is to develop a game on the web that’s compatible with several platforms. The game’s concept is centered on the current game Draw It or Lose It, which is only available on Android. The premise of the game is to have multiple teams containing many players competing in four rounds for 60 seconds each. Whenever a picture is retrieved, a team must guess until time runs out. If unsuccessful, the other opposing teams each has an opportunity for 15 seconds.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

* Must be available and compatible with several platforms
* Each team has several players
* The game must be in use one instance at a time
* Names for teams and individual members must be distinct and allows players to check if name is already in use.

## [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)

A relationship between the Game, Team and Player classes is created by Entity, which means that there’s an inheritance of data from Entity as a superclass. Analyzing the UML, the Team and Player classes are a “has a” type, whereas the Game class learns from Team and GameService learns from Game.

**"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** | Ease of accessibility, server configurability, graphic user and interface  Flexibility in terminal commands | Affordability  Difficulty in platform navigation  Command shell for server configuration and accessibility | Expensive server  Easy GUI  Possesses a command prompt | More superior specification in other devices  Mobile device specifications differ from various users |
| **Client Side** | Cost-prohibitive for users  Requires time and expertise  Requires skills to navigate the operating system | Requires time and expertise  Operating system requires Linux data  Expensive for Linux users | More exorbitant than Linux systems  Easy in learning how to support a Windows setup  Very little expertise needed | Clients and developers can easily update  Slight difficulty to update versus other platforms |
| **Development Tools** | Languages include HTML, CSS, JavaScript  Libraries supporting front-end development  Tools including PyCharm, GitHub, Visual Studios | Languages include HTML, CSS, JavaScript  Libraries supporting front-end and languages  Systems such as JavaScript, Ruby, Python | Languages that include HTML, CSS, JavaScript  Libraries supporting front-end and languages  Development tools including Eclipse, PyCharm | Languages that include HTML, CSS, JavaScript  Libraries supporting front-end and languages  IDEs for programming languages such as HTML, C++, 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**: Windows is the best operating platform for *Draw It or Lose It* due to an abundance of IDEs to develop with.
2. **Operating Systems Architectures**: Windows is an operating system with optimal abilities to store files and data, launch and develop software, play games with Internet connection.
3. **Storage Management**: Windows has the capability to analyze and manage files in the hard drive and monitor storage space.
4. **Memory Management**: Windows can accommodate storage and management of Draw It or Lose It and the photos and player data within a singular storage space.
5. **Distributed Systems and Networks**: Network games with multi-user interaction systems typically possess a database for all players and data.
6. **Security**: Windows feature built-in security protection software, yet it is highly recommended to use another source to secure user data and information.