

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 | 5/22/2025 | Taylor James | Edits were made to all sections; Executive summary, Requirements etc.. |
| 2.0 | 6/5/2025 | Taylor James | Change info in Evaluation table |
| 3/0 | 6/16/2025 | Taylor James | Complete Recommendations 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 goal of this project is to create a web-based game for the client, The Gaming Room that is based on the current game of draw it or lose it. The client wants this app to be able to run on multiple platforms because the current iteration of the game is only found in an android app. The purpose of the game is to have multiple teams that have a group of people, take turns pulling a picture from a list and try to guess what it is. They have until time runs out before an opposing team gets to guess until 15 seconds goes by.

## Requirements

Accessible through a web based platform allowing compatible to multiple devices

Team names are unique to distinguish and prevent confusion/conflicts

App should only allow of a single instance of the game at one time

## [Design Constraints](#_2et92p0)

App must be able to run on multiple platforms

Teams should be made up of multiple players

Teams have unique names

Single Instance

## [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 creates the relationship between Game, Team, and player class. Using the UML diagram shows that these classes get their information from Entity. From the diagram we can see that Team and player is a has type, while Game has team and Gameservice has games. This means that they have an instance to another 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** | Flexible terminal to configure the server or make changes  Characteristics:  Advantages: Upgradeable  Various options for web hosting  Disadvantage: | More cost friendly  Characteristics: Secure  Advantages: Security flaws are caught before things get worse. Most preferred for web hosting  Disadvantage: Difficult to find application to support web hosting requirements | Broad support and wide range of software  Characteristics: Closed platform. Dominant to other OS  Advantages: High resource requirements, less loading times, high comfortability.  Disadvantage: Susceptible to viruses, poor tech support | Varied hardware capabilities  Better if server is immobile and tacked.  Characteristics: More popular, high portability  Advantage: Wide reach, better compatibility  Disadvantage: High selective to various smart mobile devices. Low security |
| **Client Side** | Moderate expertise is required. | Steep learning curve and diverse expertise needed  Max expertise and time required. Minimum cost | Expensive but easier to learn  Min expertise and time compared to other OS | Connection limitations  Client flexibility  More difficult to implement |
| **Development Tools** | When running mac languages, Swift is the more popular option. Though macs can run most languages HTML/CSS/JAVAscript while having libraries that support general purpose languages such as python or ruby | Linux can work with visual studio. eclipse and notepad++.  Languages consists of HTML/CSS/Javascript while supporting libraries of basic languages such as python | Dev tools consists eclipse, command prompt etc..  Easier to use than linux but can run the same languages and applications.  Languages consists of HTML/CSS/Javascript while supporting libraries of basic languages such as python | IDE’s for programming consists of HTML, PHP and C++  Countless app creation. All 3 software can run on all devices.  Most languages consists of TML/CSS/Javascript  While support basic programming languages such as 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**: After researching the available operating platforms, I recommend using Linux OS. Linux provides stable and secure performance, and it is open ended. Meaning a developer can make the system based on the needed requirements. Because Draw it or lose it is a web based app, Linux provides a good platform to start from.
2. **Operating Systems Architectures:** Based of off the system we are going to use, a multi tier architecture would be best. Multi tier takes the presentation, application and data and splits them into different tiers. This allows for more efficient scaling.
3. **Storage Management**: Because this application requires rendering of images, it would be best for a dedicated drive (SSD) to be used to store the large library of images. These drives are typically faster than older styled drives and are what are commonly found in high tech computers for things like gaming or art-studio creation.
4. **Memory Management**: Linux, the chosen platform we are using provides support for memory management. It uses a paging system, by sorting unused data and moving it to a disk, leading to optimized memory usage.

**Distributed Systems and Networks**: API’s are needed for the platforms to communicate effectively and over HTTP. Linux provides support for RESTful API’s.

6. Security: Another advantage of using Linux operating system, is that it provides excellent support for security features like access control or firewalls. I would also recommend SSL/TLS encryption for all communication.