

<Draw It or Lose It>

# **CS 230 Project Software Design Template**

Version 3.0

## 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)

game

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/19/2023 | MIKEL PANARITI | The Gaming Room Requirements |
| 2.0 | 4/2/2023 | MIKEL PANARITI | The Gaming Room Evaluation for a web-based game. |
| 3.0 | 4/16/2023 | MIKEL PANARITI | The Gaming Room Recommendations for a web-based game. |

**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 develop a web-based game to run on multiple platforms. It will be based on their current game “Draw It or Lose It”, which is currently available in an Android app only. The game will have multiple teams with several team members, and it has four rounds that last 1 minute each. The application will render images at a steady rate and will be fully completed after 30 seconds. The team should guess the puzzle before the time ends, or the other teams will have the chance to offer one guess each to solve the puzzle within 15 seconds.

## Requirements

1. The game should have One or More team involved.
2. Each team will have multiple players assigned to it.
3. Game and team names must be unique to allow user to check if a name is in use.
4. Only one instance of the game can exist at any given time.
5. The program must run on different platforms.

## [Design Constraints](#_2et92p0)

1. The game should have one or multiple teams that are assigned with multiple players.
2. The game and team names should be unique and the program should allow users to check if a name is in use or if they can use that name.
3. The program is available in an Android app, but it should work on other platforms too, including macOS, Linux, Windows, and mobile devices.
4. The program should be user friendly and offer seemingly design in all Operating Systems.
5. The speed of the program should be optimized.

## [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)

In the UML diagram below the Entity Class is the super class, where Game, Team, and Player classes inherit the properties. There is also aggregation between Player and Team, Team and Game, and Game and GameService. This means that these classes have a “ has-a” relationship and the aggregate class will contain a reference to another class and has ownership of the class. The main reason we are using aggregation is to maintain code reusability.

**"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** | It is a popular web hosting OS that can be easily accessed and configured. It offers very high security features, but it comes with higher costs and limited software support. It requires some expertise to use. | It is an open-source server that can be easily configured. It offers very high security and if is cost friendly. However, it offers less support and is not preferred for Web hosting services. It is also less commonly used so it requires higher expertise to run. | It has a better range of software and offers better support and is very easy to access and configure making it one of the most preferred web hosting services.  It has more security threats, but detecting the problems in an efficient way without allowing to risk the data security makes it a secure web hosting service.  It has higher cost, but it is widely used so it is easier to find someone to work on the server. | This option is very cost friendly and has higher portability.  The server is shared, what leads to more security threats, and lower security. |
| **Client Side** | Moderate cost, time and expertise required.  Users need more experience with Mac to use the application easily. | Maximum expertise and time required but with minimum costs. Linux is not widely used so it would require more time to learn and run the application. | Windows is widely used and is easily accessible. It is cost moderate and takes less time and expertise. | This will be the most flexible option and very cost friendly. It requires more time to implement in different mobile OS and higher expertise. |
| **Development Tools** | You can use different IDE including Eclipse, Visual Studio, XCode, etc. Mac can run Java, C++, Python, HTML, etc. but the most popular option is Swift. | You can use different IDE including Eclipse, Visual Studio etc. Linux can run Java, C++, Python, JavaScript, HTML, etc. | You can use different IDE including Eclipse, Visual Studio, PyCharm, NetBeans etc.. Windows can run Java, C++, Python, HTML, JavaScript, C, etc.. | You can use Visual Studio, Android Studio, Eclipse, JetBrains, etc.  Java, Python, Swift or JavaScript will be good options for mobile devices. |

## 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 operating system recommended will be Windows. It is cost effective and requires minimal time and expertise levels.
2. **Operating Systems Architectures**: All Windows based applications use the Windows Application Programming Interface (API) that enables them to access system resources.
3. **Storage Management**: The game is a web-based game so it is important to have a cloud-based storage that can give users easy access and can be used by different platforms. The cloud-based storage that will be chosen should be secure, reliable and offer enough storage for the expected number of users.
4. **Memory Management**: The game should not use pictures with large memory usage. Reducing the size of pictures will allow the game to run without errors and optimize speed.
5. **Distributed Systems and Networks**: To make sure that the game runs smoothly between different platforms the nearest server with the user should be used. In case of outage from a server a different server should be used automatically. By using Content Delivery Network and Load Balancing Systems, the game will have a good communication between servers.
6. **Security**: There should be updates in a regular basis to fix bugs and ensure a secure program. The program should be secure and should prevent data breaches and have up to date security protocols.

**References**

Kristoffer Bonheur. “Advantages and Disadvantages of MacOS | Profolus.” *Profolus*, 23 Apr. 2018, [www.profolus.com/topics/advantages-and-disadvantages-of-macos/](http://www.profolus.com/topics/advantages-and-disadvantages-of-macos/).

‌

Cloudflare. “What Is a CDN? | How Do CDNs Work? | Cloudflare.” *Cloudflare*, [www.cloudflare.com/learning/cdn/what-is-a-cdn/](http://www.cloudflare.com/learning/cdn/what-is-a-cdn/).

‌“Windows Architecture - the Basics.” *TECHCOMMUNITY.MICROSOFT.COM*, 16 Mar. 2019, techcommunity.microsoft.com/t5/ask-the-performance-team/windows-architecture-the-basics/ba-p/372345.

‌

Splunk. (2021, February 1). *What Are Distributed Systems?* Splunk. <https://www.splunk.com/en_us/data-insider/what-are-distributed-systems.html>

‌