

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | <mm/dd/yy> | <Your-Name> | <Brief description of changes in this revision> |

**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)

Our client “The Gaming Room” is a gaming company seeking to have their game “Draw It or Lose It”

developed as a web-based game so it can operate on multiple platforms. The game is currently only available on the Android platform. The game needs to be able to host one or more uniquely named teams consisting of multiple players, and only one instance of the game can exist at a time.

## [Design Constraints](#_2et92p0)

* The game requires the ability to create multiple teams. Therefore, we need to code the game so that the user can create multiple instances of a team.
* Each team must be able to have multiple players assigned to it. Each team class must be able to include multiple players.
* Each team and game need to have unique names. We must run an iterator through both to ensure that only one of a given name can exist at a time.
* A game can only have one instance in memory. We must use a Singleton design pattern to ensure only one instance of a game can exist at a time.

## [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, Entity is a superclass; it holds information that the subclasses “Game”, “Team”, and “Player” need to function, such as “name” and “id”. Instead of each class having to have its own onjects created, the developer used inheritance so that the subclasses could just pull information from the superclass. This is useful in the context of our requirements because we will be needing to define IDs and names for the games, teams, and players in this game.

We also see aggregation taking place in this model. When we look at the relationships between the classes “GameService”, “Game”, “Team”, and “Player”, we can observe aggregate relationships. Aggregation is when an instance of one class refers to an instance of another class. In this model GameService has one or more Games, Game has one of more Teams, and Team has one or more players. As required, we need the system to have a game, we need the game to contain teams, and we need the teams to contain players, so that is why we use aggregation.

**"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** | The advantages of hosting with Mac OS include the fact that their products are constantly being updated so everyhting will always be the best available, and their security is second to none.  Mac is well known for developing software that is exlusive to their own products. Additionally an apple product is almost always far more expensive than their Windows-based counterparts. This applies to their software and hardware. Mac is also often limited when it comes to windows-based applications such as Microsoft Access, a large database platform. | Linux has big advantages over Mac and Windows when it comes to ease of use. Linux is compatible with many different clients whereas Windows and Mac only function on their respective devices. Linux is also typically the cheaper option as the software is free, it doesn’t cost money to push updated versions out, and the harware requirements are usually much less. Linux servers have been known to last years without update.  The downside is that Linux is very basic. You won’t have acess to the best applications and security that Mac or Windows have. | Windows is somewhere in between Mac and Linux when it comes to Server quality. Windows has better applications than Linux like Access, MSSQL, etc., but the quality isn’t quite as good as Mac. On the other hand, hosting a server on Windows is typically cheaper than on Mac OS. | Developing an application for a mobile device can be very difficult because very little of the process for developing for a traditional OS will translate over. The development requires a specific set of skills, especially if it needs a lot of customization.  If you are able to have complete the development to host a web-based program on a mobile OS, it will likely run smoothly and be secure as mobile devices push out frequent updates. |
| **Client Side** | You need to consider that developing for a MAC device is completely unique to developing for a windows-based device. You will require specifically trained mac developers who use mac-specific products in order to create an application for Mac OS. This will likely cost more than a developer who is trained in more common languages. | Getting development work done for a Linux device is typically very easy and efficient. The software is free and the platform can use mostly any type of code, so many more developers are available. | Windows is also a relatively easy platform to build a program for. You have access to more programs than Linux, but it will cost a bit more money. At the same time though, Windows will be able to support most coding languages. | Coding for a mobile client can be very complicated. Firstly, the process is going to be completely unique to the code used for a traditional OS. The UI will need to be completely different so you will need to start from scratch. You also need to consider the need to develop for both Apple and Android devices as these processes will be different as well.  Developing a mobile version for your application will require a lot of time and resources but having a mobile app will be valuable as it will extend your reach to many more consumers. |
| **Development Tools** | Mac OS provides a suite of developer tools called The Apple Developer Tools. These can be a great set of developing tools for Mac programmers but you can also go with other programs such as Visual Studio.  The most commonly used language for Mac OS programming is Objective C. | Linux is massively developer-friendly in that it supports most programming languages and environments. A developer trained in any language should be able to effectively code in Linux with the IDE of their choosing. | C++ is known to be the most common coding language for Windows applications. However, many languages are applicable to Windows applications so a number of different languages will work fine.  Visual Studio and Eclipse are both great options for coding Windows applications. | Swift is the most commonly used programming language for Apple IOS apps, while Java is the most common for Android apps. Other languages are available but these are typically considered the best.  Visual Studio is can be used to develop IOS and Android applications. I would be a good idea to use the same IDE if developing the app for both systems. |

## 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 best platform for this job is Windows. You will have an easier time developing the application using a windows platform and the application will be able to run on various clients. Windows is the most accessible platform for most clients. Windows is also the most common platform used among developers, so you won’t have to invest as much into developing costs.
2. **Operating Systems Architectures**: Windows architecture contains all of the products and services that you will need to facilitate developing a web-based app. When comparing different major operating systems, Windows-based applications are accessible across more client-side platforms than any other.
3. **Storage Management**: Our storage challenges include storing data required to run the game as well as user data. To solve these, we should invest in a cloud-based storage system. We can ensure the security of our users’ data and we can modify how much storage we need as required. I would suggest using Windows cloud-based storage as it is a reputable service.
4. **Memory Management**: This game contains a collection of images that must be able to be shared across many different game instances. In order to display these images to different instances at one time, we should use a cache memory system. This will allow the game server to pull the images from the game’s memory as quick as possible.
5. **Distributed Systems and Networks**: For the game itself to run on multiple different platforms, I would recommend implementing a framework for cross-platform development. A cross-platform framework such as React Native will take you code via an API and manipulate it so that it runs on different platforms. I recommend React Native because it is commonly used for developing apps for different mobile platforms.

As stated in the section about storage, I would recommend implementing a shared database such as a cloud system in order to distribute the game software to multiple users

1. **Security**: With online games, you often have to ask the user for sensitive personal information in order to provide them with an account. It is your responsibility to keep this information safe. In the even that your system leaked a user’s information you could wind up in legal trouble. You should implement security measure on the server side of the system so that all information has to pass through it before reaching or leaving the server, before it branches out to the different clients. This will be the most efficient way to implement security software. I would recommend using Windows security software because its features are reputable, it is constantly updated, and it will be included if you are using other Windows applications.