

**The Gaming Room: Draw It or Lose It**

# CS 230 Project Software Design Template

Version 4.0

Zheen Suseyi

08/10/2022

CS 230

SNHU

## Table of Contents

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

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

[**Document Revision History**](#_3znysh7)2

[**Executive Summary**](#_2et92p0)3

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

[**System Architecture View**](#_3dy6vkm)4

[**Domain Model**](#_1t3h5sf)4

[**Evaluation**](#_2s8eyo1)5

[**Recommendations**](#_3rdcrjn)7

## [Document Revision History](#_3znysh7)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0  2.0  3.0  4.0 | 07/15/2022  07/17/2022  07/31/2022  08/10/2022 | Zheen Suseyi  Zheen Suseyi  Zheen Suseyi  Zheen Suseyi | 1st Revision 07/15/2022  2nd Revision 07/17/2022  3rd Revision 07/31/2022  4th Revision 08/10/2022 |

**Instructions**

## [Executive Summary](#_2et92p0)

The Gaming Room is looking to branch their android platform only game out to more platforms. The game must have the following criteria:

* **A game will have the ability to have one or more teams involved.**
* **Each team will have multiple players assigned to it.**
* **Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.**
* **Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.**

The problem is, the staff at The Gaming Room does not know how to set up the environment. We were hired in order to prepare a software design document and start development on the game application. This is something that will be accomplished by research and hard work by Zheen Suseyi.

## [Design Constraints](#_tyjcwt)

Some design constraints that may arise would include turning an android mobile app into an application that works on many very different platforms. Another design constraint may include the ability to have multiple teams and multiple people on each team which may be tricky to implement.

## [System Architecture View](#_3dy6vkm)

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](#_1t3h5sf)

The Entity class inherits from the Game, Team, and the Player class. Meanwhile, the GameService, Game, Team, and Player class all show Association with each other. Lastly, the ProgramDriver and SingletonTester class show association and the ProgramDriver class directly uses the SingletonTester class. All of these classes are necessary for fulfilling this project successfully and are all pieces of the puzzle.

**"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](#_2s8eyo1)

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.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | **Mac:** Mac is a popular platform for web hosting with lots of different options when it comes to hosting. The Software is very secure but the downside is that Mac OS is not open source and free. This operating system supports hosting for a web based software application. | **Linux:** Very cost efficient and secure for web hosting. Advantages include Linux being very secure and practically free compared to Mac and Windows. The downside is that it may be difficult to learn how to use and find the correct applications needed. This operating system is the best for a web based software application. | **Windows:** The advantages of Windows include ease of use to most people and has some of the best software available along with hardware working the best on Windows.. Disadvantages include having the weakest security compared to Mac and Linux and having a large amount of spyware and bloatware pre-installed. This operating system supports hosting for a web based software application. | **Mobile:** Advantages include being the most popular platform on earth and being extremely portable. Disadvantages include poor security, and performance may vary depending on what phone is being used. |
| **Client Side** | Gaming is not normally done on Mac, so this may be a challenge for consumers to enjoy and for developers to develop. | Games are almost never played or made on Linux so this may be very tough for consumers to find and play the game on a Linux distro. | The majority of PC Gamers are on Windows, so this would be a great option for development and client enjoyment. | Mobile Gaming is extremely popular and only growing. This is a great platform for consumers and easy to develop with Java. |
| **Development Tools** | Mac programming is very popular in the industry because macs support many different languages and libraries. | Nearly all languages can be used in Linux with large libraries. Development tools are a major advantage of Linux. | Programming on Windows is very simple and many applications like visual studio and eclipse are available. | Very easy to create apps on mobile devices. May be difficult to port an Android app to an iPhone app store. |

## 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**: I would recommend Windows because the majority of people are familiar with programming in Windows and Windows has all of the necessary tools and IDEs for programming. Linux is superior for web-hosting because it is open source but overall the ease of use puts windows over the edge. Considering that this project is planned for a multi-platform release, Windows offers the appropriate tools in order to program in any language.
2. **Server Side**: Every operating system offers server-based deployment methods that could host this application. Linux is free and open source which is why it is preferred for server application hosting but I recommend just hosting the game on the cloud using AWS or something similar to save time and stress. There are definitely some licensing costs involved especially if they want to host their own server.
3. **Client Side:** In order to host the application on multiple different platforms, there will need to be a lot more time allocated making sure the game is optimized for all platforms. The game may run well on a high-end PC but it would need adjustments to run well on a laptop or an iPhone. This task will need a lot of hours and testing to successfully pull this off.
4. **Development Tools:** Programming languages that will be most likely used will be C++ for desktop development and Java for mobile development. Eclipse will be used for Java and Visual Studio can be used for C++. Both of these programs are the industry standard when it comes to programming. Eclipse is free and open source but Visual Studio has licensing costs.
5. **Operating Systems Architectures**: Windows allows applications to broadcast a GUI (Graphical User Interface) and has the widest range of support for computer hardware compared to other options. The fastest and best performing computers will always be Windows Systems due to the lack of GPU and CPU support Linux, Mac, and Mobile devices offer.
6. **Storage Management**: Due to ease of use with Windows, it would be very simple to add external hard-drives for additional storage. Accessing and using the cloud is also very easy to do on Windows granting nearly unlimited storage for projects. Not using a cloud based storage system just means relying on computer HDDs and SSDs to store all information.
7. **Memory Management**: Windows memory allocation may not be the best but for rendering and processing game files it is superior compared to other OS’es. An entire project can be done on a single computer provided that the hardware isn’t outdated. On a different OS like Linux or MAC, they may not have the support for a game on their systems.
8. **Distributed Systems and Networks**: Windows is clearly superior for publishing and distributing games because the main game marketplaces are almost exclusively Windows-based games. There are very few games on Linux and Mac so the majority of gamers are all on Windows. Porting a Windows game to a phone would also be much easier then Linux or Mac game to a phone. It is up to the company itself The Gaming Room to communicate with the various different platform corporations in order to allow crossplay.
9. **Security**: Windows has a variety of anti-malware but it really is up to each user to practice discretion and not install a virus or click on a phishing link in order to be completely secure. Even major billion dollar companies get hacked so employees must at all times be alert and be mindful of what they do on work systems. As for the players' security, The Gaming Room needs to make sure that their information is secured safely and by having smart employees, this can be achieved.