

The Gaming Room

# **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 | 02/06/2022 | Nicholas Boodoo | Project Three |

**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 is a project that will require the development of a web based video game that will be able to be run on multiple different platforms. This game will be based on “Draw It or Lose It”. The game will be able to have one or more players playing for different teams. Each name in the game will be a unique username with a check for username originality. There can only be one instance of the game running at a time, therefore unique identifiers will be required for each instance of the game or player.

## [Design Constraints](#_2et92p0)

The player and team entities can only belong to a single game. If you would want to create another team, even if it has the same players on it, you’d need to create a new game. The game must be coded in Java and have a minimum network speeds of 3 Mbs.

## [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)

The Player, Team, and Game objects are all children descended from the Entity Object. The ID and Name properties of the Entity Object are all inherited, so we do not need to initialize them in their subclass. GameService is a singleton, and therefore we can only have one copy of it.

**"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** | Ability to utilize MaxOS X server.  MacOS X is a $20 investment. Not as popular as Linux or Windows so may have less support. | Linux servers are usually low cost and open source, which would mean that it would have a lot of support. Linux is kind of niche though so it would be harder to find talented people to run the servers. | Windows allows you to run off of the Windows servers, however it is much more expensive than the MacOS X servers to implement. Most widely used operating system, so finding talent to operate servers would be easiest. | Mobile devices so not really have the computing power to host servers. |
| **Client Side** | Same cost efficiency as Windows. Development time would depend on level of experience of the developer, which would mean that you’d want people experienced on Mac operating systems. | Cost, if any, would be low since Linux is open source. However, Linux is very niche, so maximum time and experience would be necessary. Experience levels would be the biggest bottleneck. | Same cost efficiency as MacOS. Windows it he most widely used operating system so it may be easy to find developers for Windows who are experienced in Windows. | Cost and experience level would not be that much of an issue since almost the same would apply to Windows, Mac and Linux. However, in order to streamline for mobile devices, more time might be required. |
| **Development Tools** | Swift is widely used to program for MacOS. Atom can be used as an IDE for Swift. | Eclipse and Atom are widely used IDE’s for Linux programming since a lot of Java and C++ applications will be utilized. | Eclipse, Visual Studio, Visual Studio Code, and IntelliJ are all common IDE’s for Windows applications since cover most of the C languages, Java and HTML. | Mobile iPhone applications can be developed in the same manner as MacOS applications. |

## 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 think that Windows would be the best operating platform to use because it is the most widely used platform and there are more people with advanced knowledge of Windows that the other operating platforms. There are many IDEs that will run on the Windows platform that we can develop on. Also, Windows utilization will contribute to a relatively low cost.
2. **Operating Systems Architectures**: Windows architecture allows for its applications to utilize the kernel processes without directly affecting those processes. Because of this, applications can utilize the ability of Windows to have a window setup and access to memory. This in turn, allows the application to avoid affecting processes vital operating system processes.
3. **Storage Management**: As far as storage is concerned, Windows has a few built in management applications: Disk Management, Disk Cleanup and Storage Sense. Disk Cleanup and Storage Sense help the operating platform by deleting unnecessary files. Disk Management offers advanced features such as partitioning and formatting drives, but can perform other advanced storage operations also.
4. **Memory Management**: Windows has a couple of memory management systems built into the operating platform. One such application is called the task manager, which shows all processes currently running within the operating system.
5. **Distributed Systems and Networks**: In order for this project to run according to the client’s specifications, a client-server distributing system will be necessary. Each client application will feed back into the single server application for the game. Each client application can be developed to suit the client’s system. A very strong server network will be required since we will have many clients connecting to a single server for each game session.
6. **Security**: Windows has its own built in security system called Windows Defender. Any data that will be sent to and from the server will be also be encrypted.