

**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 | 02/25/2024 | Md Islam | Added Information related to the software design |

**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 that can be accessed by different platforms based on their current game functionality. The concept of the game is users are getting asked a series of questions.

## Requirements

* A game must be accommodated with multiple teams and players.
* Game must be run on all listed platforms.
* Provide validation against current players and team names.
* Only one instance of game must exist once.

## [Design Constraints](#_2et92p0)

The Game Room wants their games to be accessed by the iOS and Android platforms. There may be some issues with existing code for accessing different platforms, that’s why need additional development time and personnel if needed.

## [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 Entity class is a super class, while the Game, Player, and Team classes are inherited from the Entity class. The gameService class reference to the Game class, the Game class reference to the Team

class, and the Team class references the Player class. ProgramDriver class execute the

program and uses the Singleton class. The SingletonTester class also uses the ProgramTester class and

working together

**"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 to use terminal for server configuration. Mac has its own OSX Server, and it only cost $20, so it will be inexpensive to use but it’s not popular as Linux and windows OS for performing these tasks. It has more software than other platform. | Flexible to uses terminal for server configuration. Linux is an open source and multi-user OS, and it also has extended features to use. Not all the users are used to Linux. So, we need someone who has knowledge about Linux OS. | It also has a few commands for server configuration. Most popular OS compared to other OS. It can run more software than other OS. Windows offers their own Windows server. It's more costly than others but fully functioning. It will be lot easier to find users those has knowledge about Windows Server. | The mobile device doesn’t have the power that computers do, hosting a fully-fledged server on one may not be the best option compared to the computer. So, it is cost effective, and security can be weak. Need specific software for configuration. |
| **Client Side** | Cost can be almost like a window. We need expertise and knowledgeable resources. Time would depend on expertise. It will take less time if someone has knowledge about the Mac. | Low cost and Open Source. Need expertise to understand the functionality. Maximum time would be necessary, and Time would depend upon the experience of the team. Multiuser platform. | Windows development would require less time and expertise than other platforms. Cost might be same as mac and Linux platform. | The cost would not be too much of an  issue with mobile devices. Experience  may not be too much of an issue, as  mobile devices can  be easier to work with. So, Lowest cost would be required for development. Time is depending on available resources. |
| **Development Tools** | Swift is commonly used tool for Mac Application and can be used some other developmental tools like atom, etc. | Eclipse and Atom are commonly used tool for Linux. | Eclipse and Visual Studio are the most popular tools for Windows OS. VS is commonly used cross-platform tool. | For iPhones, the  development tools  are like those for  Mac, and iOS apps  are typically written in Swift. Also, Eclipse is popular development tool for mobile application. |

**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 the windows OS because of the cost and available resources. This environment will develop with future collaboration opportunities and would easily be ported to mobile platforms.
2. **Operating Systems Architectures**: In window-based applications uses GUI that helps to accessing system resources. These applications also referred to multimedia, messaging, webservices and so on. These services can be accessed by using a user account or server.
3. **Storage Management**: Windows OS allows users to manage files and maintain storage management system and GUI allows navigation and visualization of this hierarchy.
4. **Memory Management**: The gaming room might need to create a database for maintaining different libraries. The memory allocation allows easy storage and helps to keep the whole project together in remote system and local computer. The developers also can take advantage of the game engine to help provide the utilization of many libraries which already exist.
5. **Distributed Systems and Networks**: It will help to collaborate with different teams. It also keeps our code secure. We can manage versioning. If anything happens, it allows us to revert our code. It also helps to run our game on multiple platforms engaging in a single instance of the game.
6. **Security**: Windows comes with built-in virus protection and has additional security software that keeps this platform for data safety. This database safety would extend to any cross-platform functionality.