

The gaming room (CTS)

# **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 | 12/5/21 | Ronald S | Outlined software requirements. |

**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)

To achieve cross platform compatibility, it will take special care and attention to detail as not all operating systems behave the same. This will also require more testing on other operating systems to ensure that it works as we want before fully releasing it.

## [Design Constraints](#_2et92p0)

The game itself is pretty straight forward, one constraint we will face will be finding out how it behaves on the different operating systems we will use.

## [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 game, team, and player class inherit the Entity class. The GameService, Game, Team, and Player classes all have a multiplicity of many. The top of the boxes indicated with a “-“ are the private attributes and the bottom indicated with “+” are public methods.

**"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** | Mac is known for its flexible terminal commands. | Like Mac, Linux is known for its flexible terminal commands but more cost efficient. | Compared to other systems windows is the most readily available. | Mobile does not seem like the best OS for this application, especially if the app grows too much. |
| **Client Side** | Moderate expertise needed.  Costs around the same as windows.  Time dependent on application development. | Takes high expertise.  Usually cheaper than Windows or Mac.  Time dependent on application development. | Low to moderate expertise needed.  Costs around the same as Mac.  Time dependent on application development. | Offers flexibility to clients to see updates at any place.  Slightly more difficult to implement. |
| **Development Tools** | Swift, notepad++, HTML, CSS, Javascript, Python, Ruby, Java | Visual studio, Eclipse, notepad++, HTML, CSS, Javascript, Java, Python, C++ | Visual studio, Eclipse, notepad++, HTML, CSS, Javascript, Java, Python, C++ | Depends on which mobile OS. Possibly: Visual studio, Eclipse, notepad++, HTML, CSS, Javascript, Java, Python, C++ |

## 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 using a Windows operating system to run the application, it requires minimum expertise, and it has plenty of different tools that can be utilized.
2. **Operating Systems Architectures**: Windows operating system has file management that allows the user to control and manage the memory. Also, developers can use multiple programming languages.
3. **Storage Management**: Windows uses a good storage system where you can download the application and it will be stored in the main memory which will allow for faster processes.
4. **Memory Management**: Windows offers virtual and physical address space.
5. **Distributed Systems and Networks**: Windows offers simple communication between many simple workspaces. If you use networking support in distributed systems, it is a great way to use the software for distributed systems. These systems offer easy communication between each other.
6. **Security**: Windows offers user account control that offers security to protect data going in and out of the system. Windows has a built-in anti-spyware system that prevents unwanted software on the client’s system.