

<Name-of-Software-Application>

# **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 | 11/10/2022 | Jonathan Borntreger | Added an executive summary and design constraints. |
| 1.1 | 11/27/2022 | Jonathan Borntreger | Added an Evaluation |
| 1.2 | 12/8/2022 | Jonathan Borntreger | Added my personal recommendations. |

**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)

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

The Gaming Room wants a program to organize their games. It must be able to track multiple teams with multiple players. All names must be unique per game. Also, there can only be one instance of a game at any one time. I need to know how many games they plan on running at a time so I can make sure the code can handle it.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

Time is a big constraint as we have many other projects to work on also. There is also a lack of Java experience in our crew. Websites can run slow so memory management will have to be considered. Maybe games can be deleted after use so they aren’t stored in memory.

## [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)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

The classes Game, Team, and Player all inherit functions from the class Entity. This shows inheritance which is an important aspect of object-oriented programming. These classes are also all associated with one another. For each game there can be many teams and each team can have many players. This fits perfectly with what the Gaming Room is asking since they want everything to be uniquely named. Program Driver contains our main code and the singletonTester tests to make sure names are unique.

**"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 has a graphical interface that makes server changes easy. Also has a lot of software compatibility to increase ease of use. | Linux is completely free and has a built-in terminal to navigate. Completely private and secure. However, a lot of software isn’t compatible with Linux compared to windows. | Most 3rd party software is supported on Windows so compatibility should not be an issue. | Mobile Apps are very accessible for users. Also pretty cheap to set up. |
| **Client Side** | Macs typically tend to be a more expensive option for users. However, they are very user-friendly easily understood. | Linux is more difficult to understand and use. Users must be proficient in operating the OS to be able to get to the application. | Like Mac, Windows is also easy to use and very user-friendly. Browser is built right into Windows to easily access the application. | As long as the mobile device has a browser it should be able to access the application since it is web-based. Most people are familiar with cellphones. Web app can look strange if not optimized for each particular phone. |
| **Development Tools** | Some Common Programming languages include JavaScript, Java, Python, C#, and TypeScript. You can use GitHub, visual studio, eclipse, and more to help develop your software application. | Some Common Programming languages include JavaScript, Java, Python, C#, and TypeScript. You can use Eclipse, visual studio, GitHub, and more to help develop your program. | Some Common Programming languages include JavaScript, Java, Python, C#, and TypeScript. You can use Eclipse, Notepad, visual studio, Github, and more to help develop your program. | Some Common Programming languages include JavaScript, Java, Python, C#, and TypeScript. Types of development tools you can use include swiftic, GitHub, and visual studio, and more. |

## 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**: <Recommend an appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.>
   1. I think all 3 operating systems have their ups and downs, but I personally recommend Windows for web-based applications. Windows is relatively cheap compared to Apple and most software works well on Windows. It also has the largest selection of compatible software to choose from compared to Linux or apple.
2. **Operating Systems Architectures**: <Describe the details of the chosen operating platform architectures.>
   1. Windows has a user mode and a kernel mode to run the OS effectively. The kernel mode has full access to all parts of the computer and makes sure that nothing bad happens when in user mode. Windows also comes with a file management system to track and manage files.
3. **Storage Management**: <Identify an appropriate storage management system to be used with the recommended operating platform.>
   1. I would recommend using any cloud-based storage. These can be used to store all necessary files and can be accessed on any computer that has access to the cloud. Alternatively, you could also store everything on a HDD or an SSD on a main central computer.
4. **Memory Management**: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>
   1. Windows uses virtual addresses and physical addresses to manage memory usage. This is to keep track of every process so memory can be managed effectively.
5. **Distributed Systems and Networks**: <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>
   1. The game would have to be hosted on a server that is accessible by every operating system that they want it to be compatible with. Luckily, windows it pretty good for hosting servers so there should only be some minor glitches here and there. Like server congestion or memory issues. If it is configured for to work on a specific browser I don’t see many issues popping up.
6. **Security**: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.>
   1. Of the three windows is probably the least secure in terms of security. However, this can be easily fixed by taking appropriate counter measures. These include making your own program to protect your data, or you could hire a security specialist to do it for you. Windows also has built in security software to help as well.
7. <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.>
   1. The most prevalent programming languages and tools include Python, Java, GitHub, Visual studio, and many more. Windows especially has many options since most software is made to be available on windows first.