

<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 | <11/12/2022> | Isabella Cobak | <Brief description of changes in this revision> |

**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, currently the client’s game is only available as an Android app. The application they want to make is called “Draw it, or Lose It” which is loosely based a 1980s television game “Win, Lose, or Draw”. The clients wish to have their application available on all platforms.

## [Design Constraints](#_2et92p0)

* The application needs to be web based.
* The game needs to have the ability to have one or more teams involved.
* Only one instance of the game can exist in the games memory.
* Game and team names need to be unique.
* The ability to create teams with multiple players on them.
* The game consists of four rounds, each lasting one minute.

## [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 the handler for the Game, Team, and Player classes. These three classes inherit characteristics from the Entity class. The four classes GameService, Game, Team, and player all have references to each other. With the ProgramDriver class we can access and execute all of the classes that we’ve created. The SingletonTester class, allows that for each class the project can run according to the design constraints, ex. It allows for there to be more than one team while still only having one instance of the game running.

**"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** | Having a MacOS server offers a lot of advantages for mac clients on a network. There’d be full support of Mac applications and easy general administration. On the other side having a Mac server can be expensive to maintain and it isn’t optimal for larger companies that rely on using third-party programs. | Linux offers a lot of advantages with web hosting. The biggest plus with Linux is that it is free and open source. Another advantage is that Linux is very customizable so companies can customize things like security however they want to. Alternatively, there is a large learning curve for Linux, so if you aren’t familiar with it, it could be difficult to get used to. | Windows is a well rounded a popular operating system. Windows has a lot of support for third party software and applications. Another plus is that the patch updates are very easy. However, Windows is not free and you would have to pay for licensing. | Using mobile devices for a web server is incredibly impractical. However, Oracle is a company that offers mobile server-side implementation. It is able to manage applications, users, devices, and data on large deployments of mobile devices. Oracle database offers support for iOS and android development tools. |
| **Client Side** | Some of the pros for using Mac for the client side of things include a wide range of supported web browsers that feature many development tools. Along with easy cross-browser testing and an the time it takes to develop and deploy is rather efficient. However, using Mac would require using Apple products for it to function properly. | Linux is open source, so the cost of development would be very low. It also has a lot of open source software available so it would work well with most web browsers. However, Linux is not widely used so it could be difficult to find people who are knowledgeable on it. | Windows it widely used so there is a lot of support from other web browsers. It is also convenient for cross platform testing. The major downside is that it you have to pay for Windows licensing. | This allows for flexibility to be able use the software anywhere you want as long as you have a mobile device. There is also a wide range of resources available for android application development. However, it would be difficult to test in other environments and browsers. |
| **Development Tools** | HTML, CSS, JavaScript, and Python | Java, Ruby, Python, CSS | Ruby, Python, C++, and Java | C++, Python, Java, CSS, HTML |

## 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**: The Windows OS would be most effective at expanding the game application. From the three options that we have Windows is the most widely used. While there is a licensing fee with Windows it is well worth it because of its flexibility and widespread compatibility. Since this application is web based, Windows is best suited for handling user requests and troubleshooting.
2. **Operating Systems Architectures**: The design of Windows is very easy to understand and incredibly user friendly. Windows architecture also allows for applications to use the platform’s kernel process without directly affecting those processes. This allows applications to access memory and other important processes without directly affecting the application in a harmful way. Windows also has a wide selection of IDE’s that can be utilized for application development.
3. **Storage Management**: There is already Disk Management and Storage Sense built into the Windows OS. Windows also has a super helpful Disk Cleanup tool. Disk Management is mainly used for advanced storage tasks, whereas Disk Cleanup and Storage Sense helps to maintain storage by deleting unnecessary files.
4. **Memory Management**: For memory management with Windows you can either use physical memory or virtual memory. For The Gaming Room it would be most beneficial to use a virtual memory. For virtual memory Windows utilizes OneDrive. Virtual storage offers a lot more flexibility with how much data you can store. You will have to pay for additional storage if you exceed the limit, but it usually comes at a very low cost.
5. **Distributed Systems and Networks**: For dealing with this we can utilize a client-server distributing system, since we are going to have each client application depend on the single server application for our game, so each client application can be developed to that client’s system’s strengths. We will also require a strong server network because this game depends on multiple clients connecting to a single server to be able to play the game together.
6. **Security**: Windows has a built-in anti-virus called Windows Defender. It also offers VPN protection services. We would still need to implement routine security checks to ensure that the users are safe and that their data is protected. We’d need to encrypt all of the data that is being sent back and forth and we’d need to make sure that the developers working on the game are educated on how to secure and encrypt user information.