

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 |
| --- | --- | --- | --- |
| 3.0 | 02/20/2022 | Zachary Carper | 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)

The Gaming Room wants to implement a game that is web-based to run on a multitude of platforms. The game currently out is called “Draw It or Lose It” and as of now is only functional on Android devices. This is a team orientated game that has four rounds per minute. After a photo is display each team guesses to what it is essentially until time runs out. If the team hasn’t accurately guessed what it is then the other team gets a chance to figure it out.

## [Design Constraints](#_2et92p0)

* At minimum needs one team
* Each team for the game must consist of at least 2 individuals minimum
* Each Game and Team name must be unique and cannot be used more than once so that users can enter that or check if it is free
* Only one round of the game can be going at once
* Must work on a multitude of platforms.

These are big things we must overcome when looking at the development of the game itself. The company The Gaming Room would really prefer if this is compatible on all devices. It is already functional via android, but we need it to run all devices such as apple, windows, and Linux. To do this it needs to be re-wrote in languages such as Swift which is native to Apple devices or create a language that can be utilized in all aspects.

## [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)

Entity basically is making the relationship between the three of the following: Game, Team, and Player class in the UML diagram. This means that they are all pulling information from the main Entity. With UML diagram we can visually show the relationship and inheritance. In the diagram each one will show things like “name” and “id”. When looking at the UML diagram we see GameService has a reference of Games as well as game having a reference of team and so on.

**"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)

**Server Side:** For all the operating systems listed each one does indeed offer a server-based deployment method that is applicable to the to the hosting of the website. Linux is notorious for its open source-based tools so the utilization of Linux for a server would make this free. Linux is more advanced in terms of operating systems with developing but it is also the preferred method and can save you thousands. I highly recommend utilizing Linux (almost all major companies do). In terms of gauging which operating system you would like to host on there aren’t a huge variety of options. It really comes down to Windows and Linux. Linux would save you more money, but you are also sacrificing things such as security and support if doing so. Windows will be more money, but it honestly comes up to if your budget is good enough to host on there. Like I said, Windows obviously will be more expensive but in terms of security, support from the Microsoft team and just running a proprietary option vs open source it may be a better option. It comes down to how much you will need from the Operation System in terms of that. I still will recommend Linux at the end of the day.

**Client Side:** For the client side of this I will list the most efficient Developmental tools for each operating system with their corresponding languages.

For the Windows operating system the most common IDE and Tool is known as Visual Studio and Visual Studio Code. They are extremely reliable, and you can utilize a multitude of different languages on them. For the Mac operating system the most popular choice is Xcode which is a native Mac application (VS and VSCode from Microsoft can be utilized on Mac and Linux). The main use of XCode is Swift which is used for almost all MacOS and iOS applications for development. The last operating system is Linux and the best developmental tools to utilize on Linux is probably Eclipse. You can also utilize Visual Studio Code from Microsoft on it but in terms of the “smoothest” running IDE I’d recommend Eclipse as it is mainly utilized for Java and you C#/C++. I’ve personally used all of these, and I highly recommend XCode and Visual Studio / Visual Studio Code.

**Web Applications and Mobile Applications:** For web applications you can just utilize the browser or the server. The languages you mainly utilize is HTML, CSS, JS (JavaScript). You can utilize frameworks for these such as one of the most popular and my favorite being react. Doing this via the web server is extremely simple and effective as it constantly updates for everyone utilizing it immediately. The only downside of utilizing the web applications is just the internet. Overall, I highly recommend using React for this. For Mobile applications the best option for Android would be Android studio as it is literally made for android applications. Android studio supports Java and C++. Other languages can be utilized on it for extensions but the recommended is Java and C++. For MacOS the best option is the same as the client-side option which is XCode. As I said before the programming language for that is Swift. These are the best IDE’s that I could recommend utilizing throughout this process for the most effective and easiest processes to create your applications.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Easy to utilize terminal commands to make changes  It is upgradable which poses an advantage in terms of web hosting  It is not the most preferred in terms of web hosting | Cost friendly  Most preferred in terms of characteristics  Security issues are usually shown before they become an issue  It is more difficult to find web hosting applications compared to other | Has the most available software options compared to others  Most dominant operating system  Most comfortable to utilize with minimal loading times  Easy to get a virus compared to others | Doesn’t have the best specs compared to the other options  Easier to obtain, portable  Wider reach as it is more affordable for people  Can be more selective in terms of what mobile devices can handle it and can’t |
| **Client Side** | Cost is going to be relatively like what it will cost on Windows. What is needed to ensure it is compatible on all browsers that this operating system has? Minimum expertise will be needed for this. | What is going to be needed to ensure all browser can run this game effectively that is on this operating system? We will need an expert for this. | Cost is going to be relatively like what it will cost on Mac. What is needed to ensure it is compatible on all browsers that this operating system has? Minimum expertise will be needed for this. | Little more flexible than the other operating systems but can be more difficult for implementation compared to others. |
| **Development Tools** | Swift is always the most popular option on Mac for development. Mac can run all languages which is extremely beneficial for the frontend side of things. | Linux is versatile in terms of programs and languages it can utilize. It can be very beneficial when it comes to the developmental process and doesn’t tie us down to specific languages. It is fairly difficult to use on Linux but that is on. | Easier to use than Linux but easier to utilize than Linux. Windows can run all languages which is extremely beneficial for the frontend side of things. | Endless possibility on mobile. Can run swift and a multitude of other languages across all types. |

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

**Operating Platform:** I would start with creating this on Windows just for the sake of cost and all the availability in terms of IDE’s. You will find more people that utilize it and it would most likely be cheaper than others initially.

 **Operating Systems Architectures:** Windows has a multitude of services that provides things such as Graphical User Interface (GUI). You can utilize these servers via your home PC or even server sided.

 **Storage Management:**  Windows makes things easier in terms of finding applications and files compared to others. It can effectively manage your storage. You can also utilize the cloud for storage. The storage system is user friendly, so you don’t accidentally lose your project or delete something.

 **Memory Management:** The memory allocation from windows will make it easier to manage your memory as well as the photos and everything that is included in this project. It can keep certain things in the IDE’s, so they are not lost or misplaced.

 **Distributed Systems and Networks:** . Unity is a cross platform IDE that could be utilize for this development. Unity can be utilized so that everything can be cross platformed from Windows, Apple, Android, and so on. To prevent any issues the company will need to ensure they have everything in place to negate things such as too many players on at once, power outages and so on.

 **Security:** Windows comes with built in protection for malware, viruses, and so on. Additional precautions should take place beforehand to ensure maximum security such as cloud hosting that has security protocols and active support put in place.