

Draw It or Lose It

# **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 | 07/14/2022 | Daniel Anderson | Added Teams, and Players. Update checks to ensure no copies of team name or player name. |

**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 game currently is only in mobile application form and development is happening to turn it into a web-based game. Based on the mobile application as a starting point we will be adding the ability to make teams and name them. There will also be the ability to name each player playing the game on each team.

## [Design Constraints](#_2et92p0)

One design constraint is having the game check team names to ensure there is only one copy of the team’s name per game to ensure there is not any issues with the teams. Another constraint is when adding players to ensure no name is copied within each team and game. The last big constraint is to ensure that only one game is going at a time by tagging each game with its own unique identifier so there is no overlap.

## [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 UML diagram shows the Program Driver and Singleton Tester communicate to test different variables to ensure they are working properly. Game Service, Game, Team, and Player all communicate within each other to share information. Entity is the master class for all the other classes to pull information from by giving each its own unique identifier.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac boosts a great amount of creativity for their servers. The servers work a lot better with mac equipment. A major downside is that Mac is very limited on its harder which can cause some restrictions | Linux is great for stable and powerful servers. With no upfront cost to using the system is easy to get access to. The downside is just understanding how the actual system works which creates a learning curve. | Windows is great at web hosting and sharing files in an effective manner. The biggest down fall is they are not that popular but are gaining popularity. | Mobile offers and great deal of portability which can be very important. Mobile devices are complex to create applications for due the extra amount of work needed to keep the running. |
| **Client Side** | Mac is extremely expensive compared to the other options and does not have a high level of compatibility. | Linux is very helpful because it runs completely on open source. | Windows allows a great deal of emulation which really helps with the creativity you can have as an owner. | Lost cost for mobile device and become steep due to that fact that programming for a tablet has different challenges as compared to a phone. |
| **Development Tools** | All languages can be used on Mac, there is one language that stands out about the rest and that is Objective C. | Linux has the most flexibility when it comes to using different languages. With the ability to use any language really helps to show how flexible the system is. | C++ is the most common and probably the most used programming language on window systems. | Two different devices run on two different languages. Java is the core system for any android OS. iOS on the other hand uses swift in order to develop applications. |

## 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 Operating platform that I would recommend is Windows. There are many reasons that I would advise using windows from the UI to the ability to use different components and systems to expand on the development of the application. I do realize that Linux is open source and has great accessibility, however there is a learning curve behind using it when compared to the Windows OS.
2. **Operating Systems Architectures**: The biggest architecture that will be used by Windows in X86. This architecture has a lot of flexibility with hardware and has great developmental programs such as Unity and Unreal Engines. Using those development engines mixed with the power of X86 allows for easier development of an application on the windows platform.
3. **Storage Management**: Managing the storage is always a tricky thing, because do you go for the more expensive and faster disk or the cheaper disk but holds more information. The difference between HDD and SDD is greater than just the beginning letter, HDD boosts a lot more storage at a cheaper price it however is not as fast as the SDD. With all the being said the SDD is the best option because the speed that it offers is more reliable when compared to the HDD and that alone makes it worth the price.
4. **Memory Management**: Memory is important when balancing multiple tasks at one time and this is where RAM comes into play, however windows does have virtual space on both its 32- and 64-bit systems at 4gb and 8gb respectively. Having the virtual space and the RAM to assist it allows windows to balance more tasks efficiently and allowing you CPU and GPU to work more effectively.
5. **Distributed Systems and Networks**: Having a stable network and the systems that support is very important since this will be an online application. The database will need to be maintained to allow every user to have access to it when they are playing the game. Using the already established database from the mobile game, bringing it to windows will allow you to expand on it to increases its stability. By increase the network performance will not only help the PC version of the game but also assist the mobile application as well since it can all become a shared database.
6. **Security**: Windows does have many security features already built into the OS that allow protection for the servers and clients. By using the already built-in features on windows and implementing other industry standards such as authentication and training to help protect from outside threats. The training may seem mute when considering the size of the company but protecting from SPAM and social engineering is a good practice to protect your company. Looking into anti cheat software should also be considered since it is an online game.