

The Gaming Room

# **CS 230 Project Software Design Template**

Version 3.0

## Table of Contents

[CS 230 Project Software Design Template](#bookmark)1

[Table of Contents](#bookmark1)2

[Document Revision History](#bookmark2)2

[Executive Summary](#bookmark3)3

[Design Constraints](#bookmark4)3

[System Architecture View](#bookmark5)3

[Domain Model](#bookmark6)3

[Evaluation](#bookmark7)3

[Recommendations](#bookmark8)5

## [D](#bookmark9)[ocument Revision History](#bookmark9)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 12/11/22 | Azra Khan | - Updates to Recommendations section. |

## [E](#bookmark10)[xecutive Summary](#bookmark10)

The client , The Gaming Room, wants to develop a web based game that serves multiple platforms based on a current Android game they have called, “Draw It or Lose It”. They want the game to allow teams of one or more. Each team should be multiple players. The game name and team names should be unique, and tell the user if their name is being used already so they can think of a new one. They asked that only one instance should exist at one time.

## [Design Constraints](#bookmark11)

Design constraints is that the staff does not know how to set up an environment to create a game like this. We have to work with an app for a phone and develop it into an app for the web, which is all different type of game controls as well as appearance. This game is with teams of multiple players, so there is no option for single player mode. This might limit an audience of people who like to play single player games. This could be a costly project with demands of the software requiring different hardware to test out various operating systems with this new games functionality.

## [System Architecture View](#bookmark12)

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.

## [D](#bookmark13)[omain Model](#bookmark13)

All of the sub classes are part of one large entity class. Game,Team, and Player, all get their information from GameService and share that to Entity. GameService gets all information from user input. That information is then divided into each unique identifier class for each instance of Game, Team, and Player. SingletonTester is used so this way only one instance exists at one time. ProgramDriver contains the main() method so that all things can run.

**"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.**

## [E](#bookmark14)[valuation](#bookmark14)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Most user friendly. Similar across all servers. More intuitive. Smooth fast and easy. Very secure and strict on security. | Very secure and strong security. Bugs and security flaw can easily be discovered. Powerful. Most preferred for web hosting. More difficult to find supporting web software. | Most used all over. Most easily accessible and customizable. More software available compared to other OS. Easier virus susceptibility. | Portable. Popular. Most difficult and lengthy process do to the amount of customizations you have to make. There is the additional cost of licensing that would need to be renewed as well once creating any software for a mobile app store. |
| **Client Side** | This can be costly because client will have to buy a system made by Apple. Moderate expertise needed | No upfront costs unless buying a linux computer, but this is not necessary, though many do prefer to purchase this. Maximum expertise needed. | Works great to develop websites and windows related apps, and can use Visual Studio, or Eclipse, etc. to create. Minimum expertise needed. | Limited screen space on tablets and phones. Provides flexibility to clients to see updates at any place and even developers too. Slightly more difficult to implement. |
| **Development Tools** | Limited hardware upgrade options. Swift or  Objective C is most commonly used language. Can run other/ all languages smoothly. Can use notepad++. Works with and not limited to ; HTML/CSS/JavaScript. | Supports a wide range of programming languages smoothly. Linux can work with visual studio, eclipse, along with notepad++.  Works with and not limited to ; HTML/CSS/JavaScript. | C++ is used here. Can run all languages smoothly like Linux, more simpler. Can use visual studio, eclipse, notepad++.  Works with and not limited to ; HTML/CSS/JavaScript. | Java is most commonly used here. Can create using Swift and Android.Works with and not limited to; HTML/CSS/JavaScript. |

## Recommendations

1. **Operating Platform**:

I would recommend The Gaming Room start on Windows devices, since it has more available software, while also requiring minimum expertise and costs. It is adaptive, and easily available to learn and use and grow with for multiple programs and languages.

1. **Operating Systems Architectures**:

Windows provides services for windows based applications that can show a GUI while accessing system resources. These can also refer to graphics, media, and web services. This can be used using an account or server specifically.

1. **Storage Management**:

Windows 10 has a storage feature called “storage sense”. This allows you to manage files on your hard drive, and see how much space it takes up so you are able to discard junk that takes up unnecessary space. You can chose locations and save them for apps. This makes it easier to find. You can also use cloud to save data too. The built in storage just allows for easy creation and placement for larger projects so they wont get lost. This game will require you to save lots of pictures and graphics. The memory allows for easy storage of graphics outside of the default picture folder. This allows you to keep the whole project together in a more secure area. This is including when you’re working with IDE and opening files to create.

4. **Memory Management**:

Given that you are developing a game that displays pictures, this will require you to save lots of pictures. The memory on Windows 10 allows for easy storage of graphics outside of the default picture folder. This allows you to keep the whole project together in a more secure area. This is including when you’re working with IDE and opening files to create.

5. **Distributed Systems and Networks**:

Network games typically use a shared data base where the players that are distributed in real life and interact with one another over the same network. Currently network game developers have to use the shared database and the inter player communication from scratch.I found that Unity can be used to enable cross platform game creation. It can be used on a variety of platforms and can be used with Visual Studio IDE. It can create 2D and 3D development features. It can support desktop, mobile, and gaming consoles.

6. **Security**:

Windows comes pre quipped with built in security. This system can scan for malware, viruses, and security threats. It updates automatically to keep the system safe. Though, it is recommended to use another source for more protection against the multitude of threats. Data control is key and developers are obliged under the GDPR to minimize data collection and processing by using anonymisation or pseudonymisation of personal data where ever it is possible. This protects users from being able to be traced from their identity and personal information on the network so they are able to play freely without the fear of others knowing any information about them that they do not put out themselves.