

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)4

[**Recommendations**](#_m8aleynsvzvc)6

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | <11/14/21> | Zavalla Huggins | Adding Entity class which will be the parent class of Game, Team and Player. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants to develop a web-based version of their gaming app, Draw It or Lose It. In the game, there are teams competing to guess what the images are of that the application is rendering, which are pulled from a large library of stock drawings. If the team that is guessing does not correctly guess the puzzle before time expired, the other teams will have the opportunity to put in one guess to solve the puzzle (with a 15-second time constraint). The Gaming Room wants the game to be web-based so that it can be available on multiple platforms. It is currently only available on Androids.

## [Design Constraints](#_2et92p0)

The Singleton Pattern is implemented in design since only one instance of the game should exist in the memory at a given time. One or more teams must be involved in each instance. The game and team names must be unique to allow users to check whether a name is in use when choosing a team name.

## [Domain Model](#_8h2ehzxfam4o)

Entity is the parent class to the Game, Team and Player class, exhibiting inheritance. The GameService, Game, Team, and Player class are all related and can all have zero to many instances. One Team can have zero to many players, each game can have zero to many teams, and each game service can have zero to many games. Abstraction is used in the Entity class since the “id” and the “name” variable are private. Also, the Game class has a private array of “teams” and the Team class has a private array of “players”. The Program Driver uses the SingletonTester class to ensure that only one instance of GameService exists in the memory at any given time.

**"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 is a popular choice in web hosting, although it is not the most preferred OS for web hosting. It is upgradeable and customizable. It also has less susceptibility to viruses. | Linux is very secure and the most customizable option but less support among most applications. | Windows is the most popular operating platform, meaning that it will be compatible with most users. It has the least steep learning curve among other OS’s. Unfortunately, Windows OS is more susceptible to viruses. | Mobile devices are the most accessible and will allow the game to be portable. It is also more cost effective. The customization will be limited, and mobile devices do not have high levels of security over personal information. |
| **Client Side** | Mac is one of the most expensive options and also has a longer loading time. Safari is the default web browser offered on Mac. It has privacy options such as built in password management system and is specially integrated for the Mac OS. | Linux has the steepest learning curve out of all the OS’s. It has a lower loading time than other options. It is the most expensive and one of the least popular options. You can only run 32-bit games, and not 64-bit games unfortunately. | The learning curve is higher than other OS’s. It is expensive because more resources are needed. It has a lower loading time than other options. The Chrome browser is the most compatible across platforms | Most of the general public has a mobile device, making the app more available to use. Loading time will be longer than other options. Most mobile devices do not allow in browser play, and makes you download an app specifically for the game. The user will have to submit to downloading the app. |
| **Development Tools** | PHP can be used to manage the online database. Javascript can be used to make the web page interactive. | PHP is commonly used in database management in browser. This will allow for easy storage of users’ login information and game statistics. | JavaScript and HTML: Each of these languages are popular in web development | Android Studio is a popular editor for Android apps. |

## Recommendations

1. **Operating Platform**

Windows OS is recommended since it is very common in web-based development. It can be highly secure, if developed properly, and has the shortest loading time with a lower price. It is one of the options with a high level of compatibility.

1. **Operating Systems Architectures**

Windows OS consists of two main components being the user mode and kernel mode. User mode manages processes for the system users. User mode has a lower privilege level and does not directly affect with the system level process. System processes have a separate memory from user applications so the two do not interfere. Because of the limited privileges of user mode, the risk for potential system corruption is minimized.

1. **Storage Management**:

A database management system is recommended for Draw It or Lose It because it can easily store player information and relate that to what team the player belongs to and then the team can be related to what game instance it belongs to.

1. **Memory Management:**

Windows OS uses the compression technique since it reduces the size of inactive data and will free up space in the RAM. This will be especially useful because only one instance of GameService can exists in one given moment. The compression technique will reduce the size of previous instances of GameService that are no longer being used.

1. **Distributed Systems and Networks**:

Using a WAN could be effective in implementing Draw It or Lose it since the network will not be tied to a single geographical location. Because the connection will global, more players will be able to be involved.

1. **Security:**

Because the game will be web-based, encryption will be needed to protect each users’ personal information. Each user can create their own login information so they can save their previous game results and to differentiate each user instance.