

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 3.0

## 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 | 05/24/2022 | ShiAnn Oliver | Initial Design |
| 2.0 | 06/04/2022 | ShiAnn Oliver | Updated design analysis-expansion of platforms |
| 3.0 | 06/15/2022 | ShiAnn Oliver | Updated recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

Draw It or Lost It would like to build a web-based version of their game application. The game application produces a random drawing from a large library of stock drawings and the players must guess what is being drawn in under the time allotted. They would like to have the ability to have one or multiple teams in the game with multiple players on each team. Each team name and game must be unique and cannot be repeated. Web software will be built to meet all design constraints and validate names of each game and team names to ensure only one instance of a game can exist in memory at any given time.

## [Design Constraints](#_2et92p0)

* Must be web based in order to be available on all operating systems
* It will be modeled after the existing game applications
* Each game will have one or multiple teams playing
* Each team will have multiple players
* Game and team names must be unique
* Program must check and verify to ensure only one instance of each game/team exists in memory at any time.

## [Domain Model](#_8h2ehzxfam4o)

This diagram represents the relationship between the different classes in the the com.gamingroom package. The Game, Team, and Player classes inherit from the Entity class. The diagram displays the multiplicity between these classes. The ProgramDriver class contains the main and uses SingletonTester. These classes together allow for abstraction in this program. The entire package works together to run a smooth object-oriented gaming program.

**"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 has some limited development tools but can run smoother and more streamlined applications. Intuitive controls allow multitasking. | Linux has more complex interactions with the server and requires a higher user interaction. Security issues are less common and open source means more flexibility in configuration and keeps the software low cost. | Windows has broader development tool options but ultimately can run slower. Allows a wide range of third-party applications. Windows is much more vulnerable to malware and security issues are common | Mobile devices are a very limited platform but they often run in the cloud rather than on the device, making the program run smoother and faster. Development for mobile devices usually happens on non-mobile platforms. |
| **Client Side** | Mac OS runs on Mac hardware, making it a higher cost use. However, Mac runs fast and has a very simple interface, making it quick and user friendly requiring less expertise and time to run. In order to make software run on many different clients, it has to meet the stricter and very different guidelines on Mac OS. | Linux requires a higher level of expertise to run programs on but is the least expensive operating system to run and open source. Which allows more flexibility, but may be unreliable at times. | Windows is the more inexpensive option as it can be run on a wide range of hardware. It also requires less expertise to run on. It is the more flexible option for operating systems. Uses master database to store settings/device information on computers. | Mobile devices are the most common platform in todays society. Because they are so wide spread and they can often use the same development across the platforms, aside from iOS. This can keep the costs low and the expertise lower. |
| **Development Tools** | Development tools included on Mac, XCode, primarily use Objective-C and C++ to develop programs. However, vast amount of IDEs that use other languages are available for download and can be run cloud based. Using Xcode is free but further licensing may be required in order to bring software to market. | Linux programs are usually developed with C or C++. However, Java is also a common option and very powerful to run in the Linux OS and can be used cross platform. Open source operating system allows inexpensive development with less licensing. | Windows can commonly use C# and C++ for programs run on its operating system but Java is also a very common and powerful language and can be used across platforms. Microsoft Visual studios is a great IDE for developing Windows applications. | The programming languages are different depending on OS. Java for android, Swift for iOS, and C# for a Windows phone. The IDEs are different depending on the OS such as Xcode for iOS and Android Studio for Android |

## Recommendations

1. **Operating Platform**: I would recommend a Windows based platform first, since it has the most opportunity for a large market and is so versatile. The alternative would be to release it as a web-based program allowing it to be used across platforms and reach a larger audience.
2. **Operating Systems Architectures**: The architecture of the Windows platform is a layered design, containing the user mode and kernel mode. Microsoft makes both a 64-bit and a 32-bit version of the operating system. In kernel mode, the executing code has full access to the hardware, working at a low level to access the CPU and memory. User mode is where the programs are. The executing code does not have low level access like kernel mode does. When code runs in user mode, it delegates these needs to access the hardware.
3. **Storage Management**: The most reliable storage management system would be to use a cloud based rather than a local hard disk. This prevents loss of data and makes recovery in the event of a crash more reliable.
4. **Memory Management**: The memory management in Windows is used to control and maintain the primary memory and transfer processes. It keeps track of the memory locations and determines the amount of memory needed for each process. Windows automatically allocates a virtual address space for each process, allowing a large amount of memory to be viewed and accessed quickly.
5. **Distributed Systems and Networks**: The best way to allow the program to communicate between platforms would be to make the program web based and allow storage on the local platform only temporarily until it is stored long term in a cloud-based storage system. This can be achieved using local network connectivity and local storage as a temporary solution. The purpose of storing memory locally on the device is to avoid losing data during crashes and outages. When the device reconnects, the data can then transfer to a long-term storage.
6. **Security**: Information for the client will be kept private by using secure storage options only and ensuring the coding is maintained to be up to date with the most current security practices.