

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 1/21/2023 | Charlie Obonaga | <Brief description of changes in this revision> |

**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)

Our client, The Gaming Room, wants to develop a web-based game that can run on multiple platforms. This game will be based on their already existing “Draw It or Lose It” game which is currently only available on android. This game will have multiple teams consisting of several players going for four rounds that last a minute each. When a picture is pulled from a library of images, one team will guess the picture before time runs out. If the team fails to answer before time expires, each opposing team member will get to answer until their allotted 15 seconds runs out.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

The design constraints for the game application are as follows:

* All games must have one or more teams involved.
* Each team will have multiple players assigned.
* Game and team names must be unique to allow users to check whether the name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time.
* Game must be able to run on multiple platforms.

## [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 Entity class creates an inheritance relationship between the Game, Team, and Player class. Entity inherits or retrieves information from the Game, Team, and Player classes. It does so by retrieving information from each class that shares common references like “id” and “name”. Since Entity extends to Game, Team, and Player classes, this makes Entity a superclass. Looking at this diagram below, we see a few “has a” relationship between classes. It starts with the GameService class which shows a “has a” relationship with the Game class. Game, in turn, shows a “has a” relationship with Team while Team shows the same relationship with Player. While this aggregation of “has a” relationships may seem confusing, it merely demonstrates an instance of one class having a reference to an instance of another class. Taking another look at the diagram below, we can see GameService has a reference of Game, while Game has a reference of Team, and Team has a reference of Player. All classes therefore moves about and retrieves information which is then to be inherited by the superclass Entity. The class ProgramDriver extends to SingletonTester and ensures that the main() code remains usable in all instances.

**"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)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Flexible terminal commands to access, configure and make changes to the server.  OS is easily upgradeable and has various options for web hosting platforms.  Unfortunately, the OS is not a huge preference for web hosting services. | Has same terminal configuration options as MacOS but is a cheaper alternative.  Security is a big area of expertise for Linux.  Security flaws are caught before they become an issue.  However, it is more difficult to find applications that support the web hosting services needs on Linux. | Terminal configuration is not as flexible as in previous OS but Windows has more software available.  Considered the dominant and most preferred for web hosting services.  Offers high comfortability and low loading times for web hosting services.  Virus and malware protection leaves a lot to be desired, however. | Mobile device specifications will vary from user to user.  Specifications may be better on other devices.  Recommended for server to be immobile so that it can be tracked to a single place. |
| **Client Side** | Moderate time and expertise required for supporting clients on MacOS.  Cost required is similar to Windows. | Maximum time and expertise required for supporting clients on Linux.  Cost is the cheapest option out of all OS. | Minimum time and expertise required for supporting clients on Windows.  Cost required is similar to MacOS. | OS on certain devices provide flexibility to clients and developers to set updates.  Slightly more difficult to implement than on other OS, however. |
| **Development Tools** | MacOS can run several programming languages such as Python, Java, C++ and others.  The most commonly preferred by programmers is Objective-C. | Linux supports most programming languages including Python, Java, C++ and many others.  Linux OS is very security-oriented which means few programming and security issues should arise. | Windows support virtually all languages but the most popular is C++.  This programming language is widely used in companies all around the world. | Different mobile OS will have different language capabilities.  Some brands, such as Android and iOS, will have preferred languages that best works with the OS. For Android, the language most used is Java while iOS mainly uses Swiftic. |

## 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**: An appropriate operating platform to start with for the game “Draw It or Lose It” should be on the Windows OS as it has more software to work with than other OS’s. Windows OS is also relatively easier to understand so software programmers should be feeling more comfortable with the setup. The entire cost for expertise should also be minimal.
2. **Operating Systems Architectures**: For operating system architectures, the use of Windows would be preferable. The architectures of Windows can be based on any services provided by Windows-based applications. These applications can range from multimedia and graphics to messaging services.
3. **Storage Management**: As with the last two recommendations, I would recommend using Windows. A particular feature called storage sense allows you to take control of your files on your hard drive and manage them with ease. You can check how much space files take up on your hard drive while also being able to save locations of applications for future use. Lastly, storage sense also allows the use of the cloud to store files.
4. **Memory Management**: For our recommended OS Windows, memory management can be seen in the storage of pictures. For the game Draw It or Lose It, it will be necessary to have a library filled with large amounts of pictures for the game to reference back to. For easy memory management, Windows does allow for large collections of pictures to be stored in a single default picture file.
5. **Distributed Systems and Networks**: For the game “Draw It or Lose It” to communicate between various platforms, we need to use an IDE that can run on multiple platforms. Once the game is created, we can export the game file into another OS such as Android, iOS, Linux, etc. with ease. This will allow players to cross-play amongst multiple platforms. This tactic will also help with dependencies such as outages and connectivity. To ensure the dependencies do not become a problem in the future, the Gaming Room will need to ensure their servers are strong enough to support large player volumes as well as having adequate backup power in case of power outages.
6. **Security**: Although Windows does not rank highly in terms of security, it does have its own built-in security protection software. This software should be deemed adequate enough to protect users from common viruses and malware but they are encouraged to buy additional programming security software should users feel the need for advanced protection.