

# **Draw It or Lose It**

# **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 | 06/15/23 | Joseph McBride | Recommendation of operating system and evaluation of said operating system. |

**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)

Application “Draw It or Lose It” is currently an Android application only. Client wants to develop a web-based game that serves multiple platforms based on their current game. Application is only compatible with the Android operating system meaning it will need to be developed to be compatible with Mac, Linux, and Windows operating systems if they want a web-based application. Web Server will have to be able to sustain two teams of people playing at the same time from separate devices in a rapid-fire answer setting while rendering images pulled from a database.

## Requirements

* Application needs to be transformed from Android compatible to Mac, Linux, and Windows compatible.
* Application needs to be able to run efficiently on the Web Server with multiple devices streaming to it at the same time.
* The game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time.

## [Design Constraints](#_2et92p0)

* Transforming an Android application into an application that runs on a web server and is supported by Mac, Linux, and Windows operating systems will prove timely and costly.
* Quality of the web server will play an important role in the efficiency of the application and how the end users experience is.
* Application will have to check that each game and team name is unique before proceeding.
* Will need to ensure that only one instance of the game exists in memory at any given time.
* The application will need to pull from a database of stock drawings and render them slowly.
* Must be able to have multiple devices connected to the same instance of the game at one time. This will test the application as well as the web server.

## [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)

In the UML Diagram we see that Entity creates a relationship between the Game, Team and Player classes. What this means is that they all inherit or get information from the Entity class. UML Diagrams can show inheritance, and this indicates that each class will share references like “name” and “id”. When we look at this diagram, we can see the GameService has a reference of Games. Games has a reference of Team, and Team has a reference of Player.

**"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** | Mac has a Mac OS X server that is available for use, and it is relatively cheap so using this would be relatively inexpensive to implement. Terminal commands to configure the server, make changes, or access are flexible. | Linux has many distributions that have server capabilities. A Linux Server would be low-cost and open source. Terminal commands to configure the server, make changes, or access are flexible. | A Windows Server might be costly to implement, but it is fully functioning. There is more software available to Windows compared to other operating systems. | Mobile devices don’t necessarily have the power that computers do so hosting a fully-fledged server on one is likely not going to be the best option compared to computers. It is better to have the server immobile and able to be tracked at a single location. |
| **Client Side** | A moderate level of expertise and time is required. The cost would be around the same range as Windows, which is on the higher end. | A high level of expertise and time would be required for Linux. However, it can be on the more inexpensive side. | Would require a small to moderate amount of expertise and time. Cost would be similar to Mac, on the higher end. | Provides some flexibility to clients and even developers. However, this is slightly more difficult to implement than other devices/operating systems. |
| **Development Tools** | Swift is the more popular option when running languages on Mac. Notepad++ is a good tool to use as well. Using languages like HTML/CSS/JavaScript on the front end. General purpose languages include Java, Python, Ruby, and PHP. | Linux works nicely with visual studio, eclipse, and notepad++. Also works with many more languages and tools. Using languages like HTML/CSS/JavaScript on the front end. General purpose languages include Java, Python, Ruby, and PHP. | Visual studio and eclipse are a couple of the many tools available to Windows. Has a wide variety of tools and languages available for use which in turn makes it a little easier. Using languages like HTML/CSS/JavaScript on the front end. General purpose languages include Java, Python, Ruby, and PHP. | Using android and swift, you can create many apps. Both languages and software have the ability to be run on all three machines. Using languages like HTML/CSS/JavaScript on the front end. General purpose languages include Java, Python, Ruby, and PHP. |

## 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**: Windows would be the operating platform that I suggest. I believe Windows to be the best and most efficient option. It is the most widely used operating platform and happens to be the one most people have a working knowledge of. There are very many IDEs that function with Windows as well as the cost to utilizing it is typically lower. Another plus side of Windows is that it has a protected supervisor mode, this helps ensure the computer’s reliability. Windows server OS supports several different server roles such as a web server, file server, database server, application server, and more.
2. **Operating Systems Architectures**: Windows architecture is setup it a way that applications can have access to memory, GUI/window setup, and more vital processes that make the application work without affecting the other process that make the operating system work. In plain terms, Windows architecture allows for applications to use the platform’s kernel process without it directing affecting those processes. Also, Powershell configuration can allow for regular maintenance across a company’s devices.
3. **Storage Management**: When it comes to storage management, Windows allows for a relocation of information to solid state storage in order to open more storage for user needs. Doing this allows the machines to have plenty of storage space in order to modify and save data without the potential problem of running out of storage. Disk Cleanup and Storage Sense can help manage storage by deleting files that are unnecessary.
4. **Memory Management**: Windows has a built-in memory management utility. A database would need to be built for all the game’s image files, that way they are able to be easily accessed by the application. Random access memory, and physical and virtual address space that allow 2-4 gigabytes of memory are a couple of memory management options that the Windows server OS offers.
5. **Distributed Systems and Networks**: A useful feature of distributed systems and networks includes the ability for users to communicate with data servers, web servers, etc. This allows for simple communication between servers and clients with a much lower chance of overloading. It also can allow for backups to be in place in the case of outages. Instead of having one server or area taking the full load of the entire application, this can instead be dispersed to help mitigate the damage of a potential outage.
6. **Security**: Windows server OS has many various security features that come with it. Provided security layers help prevent potential breaches of data or sensitive information such as client information. Another great feature offered with Windows server OS is the shielded virtual machines which prevent unauthorized access to protected data by the host administrator. Windows server OS also comes with Windows Defender Application Control to monitor and control which applications can run of the machine. The most common and well-known Windows security feature would be Windows Defender which detects and blocks any known malware that may possibly infect the system.