

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 | 03/20/2022 | Mitchell Cabral | Evaluating different operating systems for the Gaming Room |
| 2.0 | 04/03/2022 | Mitchell Cabral | Evaluation of characteristics, advantages, and weaknesses of different OS platforms |
| 3.0 | 04/17/2022 | Mitchell Cabral | Recommendations |

**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)

My team and I were asked by the client to investigate multiple operating systems to find the best ones for their game “Draw It or Lose It” to be available on. It is currently running off the android OS and the client would like to increase their reach.

## [Design Constraints](#_2et92p0)

Some design constraints could be developing the game in such a way that it is able to run off multiple operating systems. The client wants each instance to be authentic to each user, keeping their access to their information secure.

## [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)

When we look at the UML diagram below, we can see that all classes are connecting to entity except for the drive and tester class. This means that Entity is the parent class, and the others are its child classes. The child classes will inherit certain information from Entity and that can be shown by each class sharing references to such like “name” or “id”.

**"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.**

**(Project 2: Module 5-2)**

## [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 good graphical user interface making it easy to use with good accessibility.  **Characteristics:**  This platform is known for its web hosting capabilities.  **Advantages:**  Rarely susceptible to viruses, free OS upgrades to keep users performing at the highest capabilities and high stability to allow the user less time interrupted.  **Weaknesses:**  Not cost efficient, can be very expensive and this platform is not typically used for games. | Similar to what Mac has to offer but on a cheaper side.  **Characteristics:**  High security features and is a popular option for users.  **Advantages:**  Better security features than the leading platforms, has great stability allowing the user to spend less time interrupted and can handle higher volume of traffic than other OS.  **Weaknesses:**  Lacks applications that support its needs for web hosting. | Windows is Similar to Mac in the way that it is on the more expensive side but it very user friendly and offer many software options for developers such as command prompt  **Characteristics:**  Considered to be a more dominant platform compared to other OS and easier to maneuver.  **Advantages:**  Faster speeds than other operating systems, ease of use and comfortability and does not require high external knowledge for user.  **Weaknesses:**  Can easily be corrupted by viruses. | Expertise needed to vary between mobile devices, it is very cost effective meaning it is cheap.  **Characteristics:**  Compared to other platforms this is the most portable and is most used throughout the world.  **Advantages:**  This is the most cost efficient compared to the other platforms and has a far reach of users since its most used.  **Weaknesses:**  Lacks security features and platform is not consistent between different smartphones. |
| **Client Side** | Mac requires moderate amount of time and expertise to use because it is not commonly used for gaming applications. Is more expensive than the other OS options. | Linux requires more expertise and time to get this application up and running. Will not be cost efficient for the client. Can be difficult to ensure the platform is compatible with mobile platforms like the other OS. | Minimum user expertise and time is required to get this operating system up and running for the application, the cost will be efficient for the client. May be difficult to ensure that his platform will be compatible with mobile platforms. | Less skill and time are needed to implement the gaming application on this platform because the client is already running on android OS. Will be very cost efficient for the client. Can be difficult to implement apps made for other OS |
| **Development Tools** | Mac allows us to use these IDEs such as Python, HTML, JavaScript, etc. For tool within Mac system, we got Encompass visual studio, Eclipse, Swift, etc.  Licensing costs may be required. Multiple teams should be considered for faster and smoother transitioning of the gaming application. | IDEs that are available on Linux include Java, Python, Ruby, etc. And tools that are available are Visual studios, Github, etc.  Licensing costs may be required.  Multiple teams should be considered for smoother development of the gaming application and to make compatible with mobile platform. | IDEs that are available for windows include Python, Java, C++, C, etc. And tools that can be used are Eclipse, command prompt, Visual Studios, Etc.  Licensing costs may be required. The use of multiple teams should be considered for faster and smoother transitioning | IDEs that are available for mobile users HTML, C++, Python, etc. And tools that can be used are Github, Visual Studios, etc. Licensing costs may be required. Multiple teams should be considered for faster and smoother development of the gaming application. |

## 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**: The operating system I would recommend would be Windows since it a very versatile OS and has many software options available along with most languages for the Gaming Room team to use. Windows also features a protected and supervisor mode which allows the user to run on any network of their choosing. This will ensure the Gaming Room team will have the most optimal performance.
2. **Operating Systems Architectures**: By choosing Windows you have services available to use that cross between any Windows device that enable applications to show a GUI while accessing system resources and such. These services also include things such as web services, multimedia, messaging, etc. Also, by using windows the user is able to have access to their file and memory management of the system, they will be able to allocate data to different programs in order to free up space of their choosing.
3. **Storage Management**: Windows has very effective storage management such as constant relocation of information being sent to solid state storage making sure the user doesn’t run out of storage quickly. The users will also be able to read and write company files to personal or work-related devices more easily by just copying to servers.
4. **Memory Management**: Memory management offered by Windows include random access memory and virtual access space giving us up to 4 gigabytes of free memory. The system can move pages of the virtual access space to the hard drive. This helps free the random-access memory frame for additional needs.
5. **Distributed Systems and Networks**: Using networking support between distributed systems is a good way to implement software for those systems. This feature of distributed systems offers simple communication between each other and various processors. Another cool feature of distributed systems and networks is that this allows the user to communicate with different servers such as web servers. Connectivity and outages should not be much of an issue since work will be distributed between the user and the server.
6. **Security**: When it comes to security windows comes with a built-in security program that will protect the user from malicious attacks or malware. This security program works very well and will keep the user information safe. Windows defender prevents attacks in multiple ways, one being by having a host/administrator to give access to protected files.