

<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)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 06/03/2022 | Victor Tomasini | Added information related to software design. |

**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)

The client would like to expand the gaming application to various operating system platforms with different software patterns in a distributed environment. For solutions to the design problem, my team and I will evaluate the different operating systems that are being considered in order to find the best platform to run the gaming program.

## [Design Constraints](#_2et92p0)

The design constraints for the game application include developing the application in a way that allows the users to utilize its services from various operating platforms. The client also wants user authentication methods implemented into the application so that the user access and information remain secure from malicious activity.

## [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 UML class diagram contains classes implemented into “The Gaming Room” application in relation to the game “Draw It or Lose It”. Object oriented principles encompass inheritance as class attributes and methods are inherited from the entity class. Also, the “Game, Team, and Player” classes below demonstrate a “has a” relationship with the entity class. Encapsulation is also demonstrated in the entity class as it helps to secure information by restricting access to sensitive methods within the application as to protect user/ administrative data.

**"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** | Good graphical user interface offering ease of use. Mac has easy accessibility and server configuration.  Mac does have Mac OS X server available for use. Per Apple’s website, Mac OS X Server is only $20, so it would be inexpensive to implement. Mac, however, is not as popular as Linux or Windows for performing these tasks.  **Architecture is a hybrid of layered and modular enabling typical functionality to be scaled with less performance impact than fully layered architecture. Implementation technologies are more diverse, imposing fewer technical constraints than Windows. Expensive scalability as the OS carries more features than needed to host the application and licensing costs increase with additional servers.** | Difficulty navigating the platform. Like Mac, Linux has a command shell for simple server configuration and accessibility. Cost friendly.  Linux is interesting here as it has many distributions that have server capabilities. Linux server would be low-cost, and open source (which provides a lot of resources). Not many users are savvy with Linux, so it would need someone who is familiar with Linux running the server.  **Modular architecture exposes only cored functionality through the kernel with scalability supported through modules. Best OS processing power and stability. Open-source application availability yields lowest of technical constraint for needed solutions and tools.** | Like Mac servers, Windows server side is expensive. User friendly GUI.  Plenty of software options for developers. Command prompt.  Windows offers Windows Server. Looking at Microsoft’s website, it might be costly to implement, but it is fully functioning. Windows is likely the most used operating system, so finding users to operate Windows servers would be a lot easier than say, Linux.  **Least privilege access tends to require additional configuration vs. Linux. Process execution is more granular vs. other OS’s and tends to get blocked less. Supports a robust service API that integrates easily with Java. Layered architecture that is heavyweight and subject to performance and availability issues.**  **Costly scalability.** | Mobile device specifications vary from user to user.  Cheap.  Andriod, iOS, WP.  Challenges in creating the game that is compatible with most/all mobile platforms.  Given that mobile devices do not necessarily have the power that computers do, hosting a fully-fledged server on one may not be the best option compared to computers. Running servers on mobile devices is the most advantageous in terms of cost, as there is little to none to get one started.  **Deployment solutions can be leveraged that are independent of the OS.** |
| **Client Side** | Mac users require average amount of time. Accurate skills needed to navigate os. Expensive option for users.  Cost would be like a Windows setup, as these operating systems are not open source. Time would depend on expertise, as someone who has experience with Mac would need less time and someone who does not have as much experience with Mac would need more time.  **When evaluating Mac, we see that we need moderate expertise and the cost of creating the application is in similarity to creating an application on Windows. The amount of time is moderate, depending on the expertise because of the limited of software available on Mac.** | Linux users need maximum time and proficiency to support Linux systems. Minimum cost for Linux users. Linux knowledge is required to user this operating system.  Cost would be low (if there even is a cost) with Linux, as it is open source. Maximum time and experience would be necessary as Linux is not commonly used and you would need someone who is apt with Linux and allow them time to work, as Linux can be difficult even for someone with experience.  **The main advantage of Linux is the cost because mostly everything is free. However, the disadvantage of Linux is the difficulty and expertise you need compared to something like Windows. Also the time needed is more as well.** | User does not need a lot of time to understand how to support a Windows setup.  Cost is more expensive than that of Linux systems.  The cost would be like a Windows setup, as these operating systems are not open source. Time would depend on expertise, as someone who has experience with Windows would need less time and someone who does not have as much experience with Windows would need more time.  **The main advantage of Windows is that you need the minimum expertise to operate and that goes the same for time. However, where Windows fails is in the cost as it is a more expensive OS compared to Mac and Linux.** | Users need to commit a lot of time and skill to support mobile devices. Mobile platforms are difficult to perform applications created for other platforms.  Cost would not be too much of an issue with mobile devices. Experience may not be too much of an issue, as mobile devices can be easier to work with. More time would be needed, as there are multiple operating systems and multiple mobile devices that would need to be worked on.  **When evaluating Mobile devices, a big advantage is having the application on your phone and it being convenient if you have an internet connection. However, if your application is web based and not native than you can run into slower run times and not user-friendly environments. Also, the cost of creating the application is much higher than usual.** |
| **Development Tools** | IDEs for Mac systems include JavaScript, HTML, CSS, Python, etc.  Tools within Mac systems encompass Visual Studio, Eclipse, and online developing tools.  Swift would be the more common language used to write applications for Mac. There are multiple IDEs that can be used for Swift, such as Atom.  **Mac is typically used in video and graphic editing. Generally, not used for mainstream development due to their high cost and proprietary nature.** | IDEs for Linux systems include  Java, Python, Ruby on Rails, HTML, etc.  Development tools for these systems are Github, Repl.it, nodejs, Visual Studio, etc.  Eclipse and Atom are commonly used IDEs on Linux. Eclipse is primarily used for Java, although it can support other languages like C+.  **Includes a native C compiler and is capable of closely emulating a server environment. Includes native support for Apache, a very common web and app tier server adding additional APIs and ease of development.** | Languages utilized for Windows OS include Python, C++ (as well as other C-like languages), HTML, etc.  Developer tools include Eclipse, command prompt, Repl.it (many of the tools that are supported on other OS).  Eclipse and Visual Studio are popular IDEs for Windows. Visual Studio can be used for developing in HTML, C# and JavaScript.  **Windows can emulate a server environment. Similar to Linux and Apache includes support for .NET, which provides an integrated web/ app development tier with additional APIs included.** | Programming languages consist of HTML, php, C++, Python, etc.  Development tools include Github, Visual Studio, command prompt, Repl.it, etc.  For iPhones, the development tools are like those for Mac and iOS apps are typically written in Swift.  **Limited hardware resources along with a general lack of support for development tools make this platform undesirable for mainstream development. Some tablet devices do allow for some development tasks for interpreter-based applications.** |

## 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**: For the Gaming Room’s “Draw It or Lose It” application, the Windows server operating system is recommended for expansion and the best result. With a protected and supervisor mode, Windows server OS ensures the computer’s reliability. This helps to serve the client their specific requests of computer running on that network. This feature ensures optimal performance from the computers and the users. The Windows server OS also supports many different server roles including web server, file server, application server, mail server, database server, print server and more. Windows server OS designed specifically to run on server hardware.
2. **Operating Systems Architectures**: Windows server operating system architectures encompass a variety of concepts including memory and file management, allowing the user to control and coordinate the computers memory to best suit their specific needs, i.e. allocating bits of memory to different programs and/ or free the space when it is no longer needed for later use. With multi-processor scheduling utilized, Windows server provides optimal application performance on machines with many processors. More specifically, multi-processor scheduling assigns a certain percentage of the processor’s power to distinct tasks from a ready state to a running state of process. Among these architectures, PowerShell configuration allows for regular maintenance across a company’s machines/ computers.
3. **Storage Management**: For storage management, Windows server allows a routine relocation of chunks of information to solid state storage to open more storage for the user’s needs. This ensures that the machines have plenty of storage space needed to modify and safe data to the system without running out of storage. Users can also read/ write company files on both personal and work-related devices through copying to servers in the data center.
4. **Memory Management**: Memory management options offered with Windows server OS contain random access memory and physical and virtual address space allowing between two to four gigabytes of memory. Supported page file enables the system to move pages to virtual address spaces to the hard drive of the system. This helps to free the random-access memory frame for additional needs/ uses.
5. **Distributed Systems and Networks**: Using networking support in distributed systems is a great way to implement and utilize software for those systems. The feature of distributed systems and networks offers simple communication with each other and various processors between many individual workstations. Another useful feature with distributed systems and networks includes the user’s ability to communicate with different servers, i.e. web servers, data servers, etc. With computers on the same network, they run with efficiency as tasks are divided up and process among both the user and the server.
6. **Security**: Provided security layers help to prevent breaches in data or sensitive information. This feature also blocks harmful attacks while improving overall security abilities. Shielded virtual machines in another feature offered with Windows server OS. With VMs unauthorized access to protected data is prevented by host administrators/ manager. Windows Defender Application Control ensure the control to which applications run on the machine. The feature does not require additional software or hardware needs. Additionally, Windows server OS users have built-in protection against memory corruption attacks. Windows server also features a Windows Defender which detects, and blocks known malware. Advanced threat analytics feature uses Active Directory network traffic as well as SIEM data to locate and notify the user of potential threats.