

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 | 01/19/2023 | Beth Campbell | Updated information for project recommendations and overview of potential starting platform. |
| 1.1 | 01/31/2023 | Beth Campbell | Updated Evaluation to include recommendations and deciding factors |
| 1.2 | 02/19/2023 | Beth Campbell | Updated Recommendations to include new information for Operating Platforms, Operating System Architecture, Storage Management, Memory Management, Distributed Systems and Network, and Security |

**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 Gaming Room would like to create a multi-platform game called “Draw It or Lose it” and is currently only available on Android. The application will pull images from a library of stock drawings. The game consists of four rounds lasting one minute each. Drawings need to be rendered at a steady rate and be completed on the 30-second mark. If a team doesn’t guess the puzzle correctly before time is up, the remaining teams have to offer one guess each within a 15-second time limit.

## Requirements

* The game must be able to have at least one team involved.
* Each team will consist of multiple players.
* Team and Game Names will be checked for uniqueness so there is only allow one instance of each at any time.
* Only one instance of the game can exist in memory at any time.
* Unique identifiers of each instance of a game, team, or player must exist.

## [Design Constraints](#_2et92p0)

* Must run on multiple servers, platforms, and networks.
* Must be coded for multi-platform use.
* Consumers must have a minimum internet speed of 3Mbps for download and 0.5 to 1 Mbps for upload.
* Must account for only one instance of the game at any time.

## [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)

ProgramDriver class includes main and is a singleton format to ensure only one instance of the game exists at any time. Game class extends Entity and is related to Team. This checks a private list of teams for instances of team names. Team class extends Entity and is related to Player. This checks a private list of players for instances of player names. Player class extends Entity. GameService checks a list of games

**"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.**

**11**

## [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 OS has the ability to handle servers in mixed platform settings, therefore covering the capacity we need. It offers adaptable commands to design with. Mac OS provides amazing graphical user interface. Mac offers open source and easy-to-use web hosting. While offering all of this, Mac OS is typically much more expensive.  Mac server will cost around $500 up to $1000 per year. | The Linux platform is difficult to navigate. Although difficult, it is cost friendly and provides an uncomplicated server structure and accessibility. With a shared hosting plan, Linux users don’t choose which servers they run on. Linux server will cost between $250 to $1300 per year. | Windows can be an expensive server. With the cost comes more readily available software and the ease of use. Typically, you get a SQL Server Database with windows.  Windows server can cost from $972 to around $6,200 per year. | With mobile devices we run into the issue of constraints varying depending on the device. An Apple iPhone won’t run the same as a Samsung Galaxy and so forth. Mobile device servers typically cost $8 to $15 per phone. |
| **Client Side** | Mac system is typically one of the most expensive. The cost for a PC starts at $1,299 and only goes dramatically up. Clients need an average amount of time to develop but require a certain skillset to navigate the system. Compatibility test would be necessary to determine cross platform use. | Linux requires the minimum time and cost to use. These typically cost around $350. This is typically more user friendly than Windows. Compatibility test would be necessary to determine cross platform use. | Windows is more expensive than Linux but due to widespread availability, users have more access to Windows. Typically, users pay about $140 for Windows. The concepts on Windows are easy to grasp so minimum skill and time are required. Compatibility test would be necessary to determine cross platform use. | Mobile devices provide clients with flexibility and the capability to use the product anywhere they have service. With that in mind, it is more difficult to execute the program due to the different types of devices.  Prices for mobile devices with the capability to run a game start around $210 and can dramatically increase. Compatibility test would be necessary to determine cross platform use. |
| **Development Tools** | Languages for Mac consist of HTML, CSS, and JavaScript. Its IDEs have the capability of running VisualStudio, Eclipse, and PyCharm. A team of developers could make the process go quicker but typically one developer could create this. Due to this being medium complexity, prices would range from $32,000 to $48,000. | Linux languages consist of HTML, CSS, and JavaScript. The IDEs available are VisualStudio, Eclipse, Sublime and Atom. Since Linux is user friendly with an uncomplicated server, one developer could create this project. Due to this being simple complexity, prices would range from $16,000 to $32,000. | Windows languages consist of HTML, CSS, and JavaScript. The IDEs available are VisualStudio, Eclipse, and PyCharm. Since Windows is straight-forward, two developers could create this project. Due to this being medium complexity, prices would range from $32,000 to $48,000. | Mobile device languages consist of HTML, CSS, and JavaScript. IDEs can be VisualStudio, PyCharm, and Eclipse. Due to the complexity of this project, we would need a team of developers. Due to this being a complex app, prices for development would exceed $72,000. |

## 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**: I would recommend using the Windows operating system. It allows for a common language and programs throughout all platforms while still providing a user-friendly environment. Windows not only offers virtual serves but has extensive software support. Therefore, Windows would be the best choice for the OS platform.
2. **Operating Systems Architectures**: Windows users are more common, and Windows offers applications to show a Graphical User Interface. Windows also allows access to system resources, audio incorporation and more (Marcho, 2019). The process for Windows must include an executable program, a private virtual address space, system resources accessible to all threads, a unique identifier, at least one execution thread, and an access token. Threads are vital to this process and must contain the contents representing the state of the processor, two stacks (one for kernel-mode and one for user-mode), a private storage area, and a unique identifier. With kernel-mode, all code shares a virtual address pace. With user-mode, Windows creates a process for the application. All virtual address space is private and can’t alter data that belongs to another application (User Mode and Kernel Mode, 2022).

Administrators can organize application files and transfer information between servers with ease. Web servers may be installed and configured for gaming access, as well. Data is easily transferable through SQL servers and Windows.

1. **Storage Management**: Windows allows for the application to be stored in the main memory after downloading. With this, users get faster processing. There is also option of Cloud server storage which allows developers more than enough space. (Windows Architecture, n.d.) With Windows 10, there is the ability to see how much memory is being taken up on the hard drive once the game is installed.

ShareArchiver (Top 5 Storage Management Software to Consider in 2022, n.d.)is my recommendation for a storage management solution. It provides flexible storage management that can handle large amounts of data. To save on storage capacity, the manager detects and replace less frequently used files while compressing and transferring data to network storage. It offers data integration and migration, disaster recovery, data security, and permission and rights management.

1. **Memory Management**: There must be room for a pre-set library to store the stock drawings. Windows has its own virtual address space. This allows 4 gigabytes of memory to be viewed for each 32-bit process. On 64-bit windows, each process is allotted 8-terabytes of address space (Windows Memory Management, 2022). Memory is allocated using stack memory, which allows for less memory consumed and uploads processes efficiently (Stack-based Memory Allocation, n.d.).
2. **Distributed Systems and Networks**: Windows Operating System offers simple communication between different platforms. This will allow users to use something as simple as an email log-in or social media log-in to sign in on different devices. With the use of Virtual Memory, it’s just a matter of simply transferring the memory from one device to another. Windows uses interprocess communications to divide labor among computers on a network. Network support is a great way to use the software for distributed systems. There will always be issues like congestion or routing but that is manageable. (Interprocess Communications, 2021) This will all depend on the IDE chosen to be used. For security and transferability, I would recommend Eclipse if using Java language. There is also the option to use VisualStudio if coding in multiple languages.
3. **Security**: With user account control settings, Windows protects data coming and going from the system. VPN services protects client’s accounts and data from being stolen or used criminally. Windows also comes with anti-spyware built in to prevent viruses, therefore protecting the client from outside threats. While there is the pre-installed Windows security, outside security is always recommended with computers. The functionality of a password, OTP, and two factor authentication would reinforce security measures. Security questions can be implemented to reset passwords if they are forgotten.

# References

*Interprocess Communications*. (2021, January 07). Retrieved from Microsoft: https://learn.microsoft.com/en-us/windows/win32/ipc/interprocess-communications

Marcho, C. (2019, March 15). *Windows Architecture- The Basics*. Retrieved from Microsoft: https://techcommunity.microsoft.com/t5/ask-the-performance-team/windows-architecture-the-basics/ba-p/372345

*Stack-based Memory Allocation*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Stack-based\_memory\_allocation

*Top 5 Storage Management Software to Consider in 2022*. (n.d.). Retrieved from ShareArchiver: https://sharearchiver.com/best-storage-management-software/

*User Mode and Kernel Mode*. (2022, November 04). Retrieved from Microsoft: https://learn.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode

*Windows Architecture*. (n.d.). Retrieved from V Skills Certified: https://www.vskills.in/certification/tutorial/windows-architecture/

*Windows Memory Management*. (2022, February 01). Retrieved from Geeks For Geeks: https://www.geeksforgeeks.org/windows-memory-managment/