

Draw It or Lose It Game App

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 05/18/23 | Oscar Rosa | Executive summary and Design Constraints added. Domain Model has been added with explanation.  Additional recommendations added. |
| 1.1 | 06/01/23 | Oscar Rosa | Revised Evaluation Table and added additional information per client specifications. |
| 1.2 | 06/16/23 | Oscar Rosa | Revised Recommendations to fit the needs of the client and meet their specifications. Added additional information into 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)

The Client The Game Room wants to develop a web-based game that can be used on multiple platforms. Currently the gaming application Draw it or Lose It is only available on Android, and the game’s main purpose is to have many teams with many players guess puzzles through drawings which goes on for four rounds and each round lasting a minute each. The game must have the capability to support multiple teams with multiple players in each team. Game names and team names must all be unique and only one instance of the game can exist in the memory at a time.

## 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)

* Each game must have the ability and capacity to have one or more teams that can play.
* From the teams playing the game, each team should have multiple players assigned to it.
* The game name and team names must have unique identifiers and IDs to allow the system to check if they are in use.
* There should only be one instance of the game at a time.
* The game application should be able to run on multiple platforms.

While writing the code for the gaming application these are the requirements that the client wants incorporated into their game. The client The Gaming Room would also prefer that their gaming application be available on multiple platforms, and the game application is currently available on Android. In order to have the game application available on multiple platforms, either the entire code needs to be rewritten which would take more time and resources from both our side and the client’s side or we find the means to salvage the code we already have and find means to adapt it so that the game application can be available on multiple platforms, which can be done if different computer languages are incorporated into the available code to make the game application stronger and be supported by other 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)

From the UML diagram below we are provided with a visual representation of how this game application will be developed. The Entity class shown in the UML diagram below is a base class that holds both common attributes and behaviors. In addition, the Entity class is also a parent class for the classes of Game, Team, and Player which is an example of inheritance and can be seen on the UML diagram as an open arrow. The Game, Team, Player, and the GameService class all share the same multiplicity, meaning that these classes contain either zero or many objects within them. The GameService class has the majority of the original methods that make up the game application, because of this it allows us to add multiple players to each team, as well as adding multiple teams to the game. These additions are then identified by unique names and ID. Also included in the UML diagram are two separate classes which are the ProgramDriver and SingletonTester. The ProgramDriver consists of the main method, which also calls upon the SingletonTester to test the code for the game application. The SingletonTester checks the program to see if a single instance of the game is running at a certain time. The SingletonTester fulfills the requirements of the application testing if one game instance exists within the game’s memory. Specifically, this is an example of Abstraction to test the functionality of the program. In addition, Polymorphism is the understanding that all teams, players, and game names are the same considering the attributes that are available to them. What makes this unique is the personal identifiers which create objects that these are assigned to.

**"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** | Easy to setup web hosting on Mac. Macs have upgradeable capabilities and have a variety of options for different web hosting requirements. Macs also have diverse implementation technologies which imposes fewer technical restraints. A disadvantage that Macs have is that it has expensive scalability as well as licensing. | Linux is characterized as being the most secure of the available options. Linux has the capabilities of noticing and catching security flaws before they become an issue. Linux share similar flexible terminal benefits as Mac while also being a cost-effective option compared to other platforms. Linux has cheaper scalability and licensing costs due to being an open-source OS. Linux has less support for Windows based products. | The major advantage Windows has is the amount of readily available software it can use for support. Windows-based products such as ASP.net, ASP, MS SQL Server, and VBScript are better supported. Windows is a more dominant OS platform than others and comes with more support and user friendliness compared to other OS platforms, and its licensing is cheaper compared to Mac and Linux. A disadvantage of Windows OS is that it is highly susceptible to viruses and other malicious attacks. | Has high scalability for webhosting, however it requires internet connectivity with said device in order to support DNS conversion, which adds extra relay steps. Mobile OS also have inexpensive licensing and mobile devices are extremely portable while also being cost effective. The disadvantage to mobile is that it is not made with server hosting in mind and has less functionality and power compared to other OS platforms, as well as having weaker security. |
| **Client Side** | High expertise required to develop software for clients, as well as being very expensive to maintain. Will take time to become familiar with OS and devices utilizing OS if unfamiliar with the functionality and tools. | A considerable amount of time is needed in order to understand and operate the OS without any issues as Linux is less commonly used. Has low costs as the OS is accessible to the public for free. Linux supports a minimal number of file structures that limits media type options that can be delivered to the client without programming. Not all devices can operate Linux, and approximately 67% of all web servers runs on Linux. | Has the least time requirement to learn and gain expertise in. Cost is higher than Linux but is much cheaper than Mac. There are some technical restraints due to targeting Microsoft web framework technologies. | Mobile devices are easier to use due to the simplistic design of the OS and the user friendliness the devices provide. There are still time constraints in learning the varying frameworks of mobile devices due to the many variations mobile devices can have. Mobile is slightly harder to implement than other devices but have frameworks in place such as HTML5 and CSS3 that help to work on a wide array of smartphones and tablets. |
| **Development Tools** | Most of the development is done on the Mac OS using Xcode IDE and the language swift. Xcode is free to download on Mac OS. | Java is a commonly used programming language for Linux. Linux supports Eclipse IDE for programming development. | Eclipse and Visual Studio are quite popular on Windows. Visual Studio houses a wide range of capabilities when it comes to programming languages. C or C++ most commonly used language, and Visual Studio is free to download. | Android has Xamarin and Kotlin. Also, mobile also supports languages such as HTML, CSS, and JavaScript. In addition, other languages such as Java, Runy, PHP, and Python can also be used for developing software and programs for mobile. Countless applications can be created with Android and IOS and the listed programming languages can be used by the other operating systems for computers. |

## 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 Windows for the client The Gaming Room because of how widely supported the operating system is. Windows is the most widely used operating platform currently with most individuals having a general working knowledge of the operating system. In this instance it would be safe to say that the Windows OS would provide the most flexibility in programming options with minimal expertise that would be sufficient enough for the client to operate and work with in order to complete their game application. In addition, the Windows operating system platform also supports an abundance of IDEs with a typical cost to utilize these tools being considerably lower than other operating systems. Furthermore, the Windows server operating environment is specifically designed and tailored to run on server hardware and provides support for multiple server roles that can be utilized for the client’s game application.
2. **Operating Systems Architectures**: Windows Operating System offers services that are utilized by all Windows based applications which enables them to show a Graphical User Interface or GUI, which can also access system resources and further offer other options as well. Additionally, applications such as messaging, web services, graphics and multimedia can also be used on specific servers or user accounts when needed. Furthermore, the architecture of the Windows operating system utilizes a layered architecture that is divided into two main components which are the user mode and kernel mode. Depending on what type of code is running the processor can switch between the two modes. In user mode the processor will load applications as they are prompted, meaning that while the processor is in user mode it cannot access virtual addresses that are specifically reserved for the operating system. By limiting this access to the virtual address space of a user-mode application, it can prevent the application from altering or even damaging potentially critical operating system information. In the event of an application crashing the system other applications would remain unaffected because while in user mode applications are run in isolation. The other side of the Windows layered architecture is the kernel mode. Kernel mode is the operating system’s main components, and all the code running in kernel mode share the same virtual address space, meaning that all the system drivers running in kernel mode are run collectively instead of in isolation. This is the system’s core essentially, and in the event of a crash this would also crash the entire system.
3. **Storage Management**: Windows 10 has a feature known as storage sense in which it can help in managing files on a computer’s hard drive as well as seeing how much memory is being used on the computer. Storage Sense does not run automatically, but there is an option for users to enable this feature. Storage Sense runs when hard disk space is low, and will delete all junk, temp, and recycled files to clear space on the disk. Additionally, cloud storage is also available for use and makes it easy to backup work or projects with ease and can be accessed from anywhere. The built-in storage system aids with easy file creation and placement of large projects, making it easy to sort and prevent accidental deletion. In addition, device storage can also be utilized and allows for system files to be stored on various apps. In doing so, load times for these file and applications become much faster as well as providing a means to create system backups on devices in case of a system recovery being needed. All in all, Windows storage options are highly customizable and can be tailored to user specific wants and needs.
4. **Memory Management**: A database or library of the pictures used in the game will need to be created. Memory allocation allows for easy storage and allows for entire projects to be kept together in a more secured location on the computer. Developers can also take advantage of the game’s engine to help provide the utilization of many libraries, which are already built into the game’s engine. In addition, Windows also uses a PageFile system which is a memory management technique, that moves files from RAM to the hard disk to free up RAM. These files are kept on the hard disk until needed, and the files are stored on the hard disk as temp files for faster retrieval. Another technique that Windows utilizes is the Prefetch memory management technique which is a data usage anticipation calculated and orchestrated by the cache manager. The cache manager creates maps and directories for every program running on the system, and when an application is about to load Prefetch will utilize the maps and directories created by the cache manager to locate the application and preload them. The Windows 10 operating system utilizes a technique known as memory compression, and in order to combat the volatility of the RAM the memory manager monitors the amount of RAM being used and when a predetermined amount has been used the memory manager compresses the files and moves them to a compression store where these files are kept until needed.
5. **Distributed Systems and Networks**: I have researched the different ways to have a game published to run on multiple different kinds of devices and operating systems. From my research I have found that Develop 4 has the capabilities to enable cross-platform functionality. This is an IDE that can be run on any device. When the game application is completed, the application can be deployed onto the web, Android, IOS, and others that have cross-play capabilities. This will help the game run on multiple platforms engaging with a single instance of the game as well as dealing with other dependencies. In addition, choosing a dedicated gaming server would be the best option for handling high traffic as well as providing the game application with more uptime and bandwidth while also being low cost to implement.
6. **Security**: The Windows operating system comes with built-in virus and security protection, as well as having a robust market for additional security protection software that will make this platform ideal for data safety. The built-in protection allows the system to scan for any viruses or malware and isolates or removes them from the system and is done in real time. In addition, the built-in protection is constantly being updated to automatically keep the systems, servers, and user information safe from any malicious intent. Furthermore, Microsoft hosts all Windows cloud server-based storage and has security measures in place to ensure a secure cloud server, with such measures being encryption, validation, and network protection to name a few.