

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 | 01/18/2023 | Chandni Singh | Currently only available as an Android Application, this company wants to expand their app based game into a web based game. |

1.0 02//5/2023 Chandni Singh Second update

1.0 02/19/2023 Chandni Singh Third Update

**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)

Draw It or Lose It is an Android based drawing game application that allows users to guess an image before a time runs outs. The Gaming Room wishes to expand their software towards a web based application with the same functionality.

## Requirements

*The clients business and technical requirements consist of a few things:*

* *The game has to have the functionality of one or more teams.*
* *Teams with have different players assigned to them*
* *Each playerID and game must be unique*
* *The application has to move from android based to web based.*

## [Design Constraints](#_2et92p0)

The design constraints for this application are few and concise.

* Application must be web based
* Each playedID must be unique
* Each game instance must occur once.

## [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)

At first glace of the UML design I see that the Entity Class is inherent from classes: Game, Team, and Player. Game, Team, and Player are encapsulated from the Game Service class, making each attribute and method more accessible. The Game, Team, and Player classes use abstraction allowing us to understanding how each method functions.

**"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 brings a lot more organization to the table than the other thee operating platforms. Mac is also cloud based so back-up is easier and more cost effective in the long run.  **Part 2:**  Mac’s OX server is available for use for server-based deployment. This server does have a cost benefit, this is that it offers an unlimited option – which would be more cost effective at $999 US, and the 10 client license which would cost $499 US. A plus to the 10-client license is that it can be upgraded to the unlimited at anytime for not additional fees. | Linux is a licensed as a free software and can be used by anyone. If offers security and privacy – although not as advanced as mac.  **Part 2:**  Linus servers are very popular amongst webhosting and due to Linux being opensource and virtually free, costs are exponentially cheaper compared to the other platforms. | One advantage that windows holdover all the other software is that it allows multiple applications to be run at the same time more effectively. It also offers backward compatibility with is a huge factor in my opinion.  **Part 2:**  Windows does also offer server options – they offer a never one every year. Potential licensing costs vary depending on the option you choose. For example, there is a $577.99 USD standard option and there is 2,999.99 server data center option. | Mobile devices are great because they allow you to work and play on the go. Each mobile device is different in respect to their operating platform such as the could for the iPad. There is a security risk for such devices as well.  **Part 2:**  Mobile devices would not be an option for server-based deployment for Draw it or Lose it, due to mobile devises not being equipped to handle such. Due to this, cost for server deployment are not known. |
| **Client Side** | Mac’s are more cost effective in the long run when developing applications due to features such as the cloud ect.  **Part 2:**  To develop an application under Mac you would need the following:   1. A Mac computer 2. Xcode   This could give in potential issues when running on other devices.  Factors required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices are:   1. Effective coding that can be updated if needed. 2. A proper a Doc-Type with clear definitions.   As for cost, time, and expertise – I believe that it would not take too long to develop, cost would be expensive but not to crazy, and for expertise you would need a team for someone with experience using Mac’s and Xcode or else time will increase. | Linux is free, so this would make for the most cost-effective option.  **Part 2:**  To develop Draw it of Lose it on Linux you would need a few things,   1. An IDE 2. Development language.   Development should be very straightforward and relatively easy on Linux. Another plus is that Multiuser support is available.  Factors required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices are:   1. Cross testing to make sure the application is working on all required devices.     For cost, time, and expertise to develop on Linux,  I believe development would be seamless as Linux is relatively simple to use, time would be dependent on team size and team expertise. I would advise finding a team of people that are familiar with Linux. | Most PC games are run on windows – which would make the game more accessible. The only downside would be less security.  **Part 2:**  Window’s applications user normally develop using :   1. .NET 2. C#   Factors required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices are:   1. A browser emulator that checks if the contents of the application is correct.   For cost, time, and expertise to develop on windows, cost would be moderate, nothing to expensive, for time, set up is relatively easy, and for expertise, windows is easier to use so I don’t think anyone with any specialty would be required. | There are two major mobile devise systems that are currently in use, windows and IOS. This would be expensive but financially beneficial in the long run.  **Part 2:**  Factors required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices are:   1. A could platform to make sure the application is running correctly and able to perform live testing.   For cost – would depend on the device- since there are so many of them, it would require a lot more time than the other platforms, and for expertise you would need someone experienced in mobile OS. |
| **Development Tools** | Mac does not support a lot of popular programming languages. Some that I’ve seen is React, Swift, and JavaScript. A lot of apple application are developed in Xcode – IDE  Part 2:  **Relevant programming languages and tools would be:**   1. **Eclipse – its free and open source** 2. **Cloud 9 – supports Python, Ruby, and PHP** 3. **Atom – flexible and fast and cross platform.**   **Impacts to the development team would be small. I don’t believe that multiple development teams would be needed. As for the development tools, more come with a free trail, but you can also pay for extra bonus that would help streamline the code.** | Linux supposed a maze of programming languages, some such as C, C++, Python, JavaScript and swift. The best IDE for Linux would be Visual Studios.  Part 2:  **Relevant programming languages and tools would be:**   1. **Eclipse – its free and open source** 2. **Cloud 9 – supports Python, Ruby, and PHP** 3. **Atom – flexible and fast and cross platform.**   **Impacts to the development team would be small. I don’t believe that multiple development teams would be needed. As for the development tools, more come with a free trail, but you can also pay for extra bonus that would help streamline the code.** | Windows supports a ton of IDE’s, such as Visual Studio, PyCharm, NetBeans, Eclipse, ect. Window supports a bunch of programming languages such as C++, Python, Java, JavaScript ect.  Part 2:  **Relevant programming languages and tools would be:**   1. **Eclipse – its free and open source** 2. **Cloud 9 – supports Python, Ruby, and PHP** 3. **Atom – flexible and fast and cross platform.**   **Impacts to the development team would be small. I don’t believe that multiple development teams would be needed. As for the development tools, more come with a free trail, but you can also pay for extra bonus that would help streamline the code.** | Examples of IDE’s that you can use to develop mobile based application is Android Studios, Qt Ide, Xcode, and Eclipse. Popular programming languages are JavaScript, Kotlin, C++, C#, Python, Swift, Ect.  Part 2:  **Relevant programming languages and tools would be:**   1. **Xcode – but only available for apple.** 2. **Eclipse – its free and open source and supports multiple languages.**   **Impacts to the development team would be dependent on which IDE they choose to use. Multiple development teams could be required depending on which IDE is used. For Cost, Eclipse is Free but you can pay for extras that could help make your code simpler and Xcode starts at $14.99 a month.** |

## 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 platform that I would recommend is Windows due to low cost and flexibility. Windows is able to support multiple IDE’s and languages which in turn would support their need to expand their game to multiple platforms and clientele. Windows servers also provide storage management which is a great tool especially when it comes to the pictures that have to be utilized for the game. It also allows remote access – which is a good aspect in our current time.
2. **Operating Systems Architectures:** Windows has a ton of flexibility. Windows also supports a lot of IDE’s and programming languages which is useful when programming. Windows gives a lot of room for advanced storage tasks without making the application nonfunctional. Windows also has backwards compatibility which is a major plus towards game development. The architecture that windows uses is NT – which has two major parts (user mode and kernel mode). Kernal mode works to ensure different applications work on different operation systems. User mode works by ensuring that code runs on the right security level. It also uses memory management and thread prioritization – which is useful when creating a game that runs on multiple platforms.
3. **Storage Management**: Something great about Windows is that is it has storage management built into it. Disk Management and Storage Sense are both be useful tools to manage memory for the application. Storage sense is great because it also helps with erasing unnecessary data.
4. **Memory Management**: Windows also has its own virtual address space- which is a memory allocation technique that helps to store data and manage it process as fast as possible. For my personal recommendation I would go with Windows virtual address space for my Memory Management.
5. **Distributed Systems and Networks**: I believe the most effect way the client would be able to communication with the server would be through an outside source called Unity. Unity supports game development across multiple platforms. In unity the client opens communication to the server by sending a request, the server then processes the request, and if applicable creates a response. Unity can handle a high volume of clients and is very cost effective as well.
6. **Security**: Security is very important and having your customers information at risk could risk lawsuits etc. Although Windows isn’t the most secure operating system, there are measures that the company can take to keep private data away from the public. Microsoft defender is a great option that is cost effective, and rapidly responds and detects threats asap. There are also outside options available to purchase as well such as, Norton and McAfee, that help to detect viruses and alert the system immediately.