

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 | 01/24/2021 | Drew Townsend | Initial document creation |
| 2.0 | 02/07/2021 | Drew Townsend | Added information for development requirements |
| 3.0 | 02/21/2021 | Drew Townsend | Added **Recommendations** to template |

**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 is seeking to develop a web application of their Android application, called Draw It or Lose it. Draw It or Lose It allows multiple teams, with multiple players on each team, to guess what image is slowly being rendered over 30 seconds. If the current team can’t figure it out, the other teams have 15 seconds to guess as well.

The development team will need to ensure the application only has one instance of the game running at one time. The game and team names must be unique, and the application will check these parameters. The development team will need to find the best way to host the application, and how to develop the game for that type of server.

## [Design Constraints](#_2et92p0)

* One design constraint that will need to be examined is where to host the application. Hosting is essential to allowing people on the internet to access the web application. There are a number of ways to host an application, and each has its own pros and cons. Hosting via the cloud, like Amazon Web Services, is a cheap and reliable way to host versus purchasing a server to host the application.
* Another design constraint is what operating system to host the application on. Each operating system (Windows, Mac, Linux, etc…) also has its own pros and cons, and the development team will need to research each of these to evaluate the best OS to use server side.
* After finding the best server operating system to host the web application, the development team will need to evaluate what programming language(s) to use, and which database to use. Picking the right programming language from a web application is crucial for performance and useability. Picking the right database can affect the speeds of saving and calling for data, and the right database for the job can lower the chances of corruption of data.
* The web application will need to be accessed on different operating systems and browsers. For example, a user on Safari on Mac OS should be able to access the game just as easily as a user on Google Chrome on Windows 10.
* The users will also need to have internet access to the game. They will need to be able to view the image as it renders and submit their guesses. Users shouldn’t require a high upload and download speed to access the website.
* After finding the right development tools, we will need to ensure the right developers are working on this project. For example, if we decide to use Java (a common choice for building web apps) we need to have Java developers on the team.
* The client already has a working product for Android, which should speed up the design and development process. We will most likely follow the design on the Android, but we will need to adapt it for larger screens and keyboard and mouse.
* The development of the web application will be done in-house. We will then need to decide whether the app is hosted in the cloud or on-site at the client’s location. The client will either need to pay a monthly fee to host the app, or they will need to setup their own server to run the application.

## [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 below UML class diagram shows the planned classes and their relationships to one another. At the top of the diagram we can see the Entity class. The Entity class is the parent class of the Game, Team, and Player classes. These classes inherit Entity’s variables and methods, and they specify the necessary variables and methods needed for each individual class. Entity contains the variables and methods to store and get IDs and names, and then print that data to a string. Game has a list of Teams, and the ability to add another team and print the game ID and name to a string. There is a zero to many relationship between Game and Team, respectively. Team has a list of Players, and the ability to add players and print that team’s name and ID to a string. There is a zero to many relationship between Team and Player. Lastly, Player has the ability to print to a string. We can then view GameService, which contains a Singleton pattern. This ensures there is only one GameService running at a time. GameService has a 0 to many relationship with Game. GameService allows us to add and retrieve games, and get player and team IDs. Lastly, there the ProgramDriver allows us to create these objects and run the program. SingletonTester tests that there is only one GameService instance running, and ProgramDriver uses SingletonTester to do this.

****

## [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 X Server comes with an Apache web server that allows it to host web applications. Apache is commonly used and offers encrypted traffic to and from the server. Licensing is cheap, at around $20, but the cost of Mac hardware tends to be more expensive. The server can scale up and down, based on the resources from the hardware. | Linux is an open-source operating system. It offers a lot of freedom for the administrators. There are different servers, some free and some a little costly. Red Hat can scale well, depending on resources from hardware, but it can cost between $300-$8000. Many cloud providers offer Linux servers with scalability. The website can be hosted on an Apache server running on Linux. | Windows Server is a bit more intuitive for admins to use. Windows is a common OS to host web applications, via programs like Apache. Licensing can be expensive, $500-$6000. Windows comes with a built-in web server but offers many other options. Windows can scale as well, especially when using a cloud service provider. | Mobile devices aren’t generally known to host web applications. They typically won’t have enough resources to host a web application and trying to host an app on a mobile device would be more of a workaround, rather than a solution. Mobile devices also don’t support as many programs to host web applications like Windows or Linux. |
| **Client Side** | Macs come with a web browser by default, called Safari. Safari can read languages such as HTML, CSS, JavaScript, etc… Mac users can also install common browsers like Chrome and Firefox. | Linux users will likely use common browsers like Chrome and Firefox. These browsers can also read HTML, CSS, JavaScript, and more. | Windows comes preinstalled with Internet Explorer and Edge. Internet Explorer is currently being phased out, as it has security issues and is falling behind other common browsers. Edge is a Chromium browser. Similar to Chrome, Edge can open sites using HTML, CSS, JavaScript, etc… | When thinking of mobile devices, screen size can be a limiting factor. We need to think about creating a dynamic website that can adjust and still look great on mobile devices. Mobile devices shouldn’t have trouble opening most websites that are developed using HTML, CSS, JavaScript, etc… |
| **Development Tools** | When developing for Mac, developers can use Eclipse, or similar IDE’s, to develop web apps. Eclipse can be used to build web applications coded in Java on the backend and HTML/CSS/JavaScript on the front end, for example. An SQL database could be used as well. Eclipse is also free to use. | Eclipse can also run on Linux, just like on Mac. Developers can also use Microsoft Visual Studio as an IDE. Visual Studio costs anywhere between $45-$250 per month for companies. Again, HTML/CSS/JavsScript can be used for the front-end, and Java could be used on the back-end. | Developers on Windows tend to use Visual Studio, but many other IDE’s can be used. Developers should use languages like HTML/CSS/JavaScript on the front end. The backend could utilize Java, C#, or C for example. SQL databases are also common. | One thing to be aware of is Java is not supported on Safari for iOS. However, common languages like JavaScript, Python, HTML, and more can be used and interpreted by most mobile devices. IDE’s like Eclipse and Visual Studio can create dynamic websites that function well on mobile devices. |

## 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 a Windows server to host Draw It or Lose it. Windows Server is widely known and used as a web server by administrators. Windows allows admins to easily manage user and file access controls. Cloud providers, like AWS, offer Windows Server as a scalable solution which would allow thousands of users to play Draw It or Lose It at the same time.
2. **Operating Systems Architectures**: With Windows, administrators can easily organize the application files and transmit data between servers via SMB protocol. Administrators can also easily install web servers, like Apache, and configure them for public access. Admins can also setup SQL servers on Windows so that the application can receive and transmit data.
3. **Storage Management**: Windows Server offers a wide array of storage management options. Admins can use one hard drive, or hundreds of drives. NTFS, Window’s file system, supports up to 8 petabyte volumes and allows admins to manage access via Access Control Lists. Volumes can be increased by adding unallocated space, and HDD’s can be automatically defragmented. Windows also offers many backup options so that application and user data is redundant.
4. **Memory Management**: Windows supports both 32-bit and 64-bit operating systems with a maximum of 24 TB memory when using 64-bit. Memory on Windows is both physical and virtual. Windows also makes use of memory pages. Processes are stored in these pages, and when a page is not being used it is moved from memory onto the disk partition. When that page is needed again, it is swapped for another page. This typically occurs when the system is low on physical RAM.
5. **Distributed Systems and Networks**: As the client has stated, they want to be able to have Draw It or Lose It accessed from multiple operating systems on the Internet. This can be accomplished in a few ways. Firstly, I would recommend using a cloud provider, such as AWS or Microsoft Azure, to host the application. These providers offer a wide array of solutions to admins. Admins can host the application on a Windows Server in the cloud, and this application can be reached by clients from all over the world via the Internet. We also need to find a common language that the operating systems and Internet browsers can communicate through. This will likely be HTML. The application’s front end will be coded in HTML via interfaces so that all Internet browsers can send and receive data. This should allow Windows, Mac, Linux, and mobile devices to access the application. This HTML will then be used by the backend to send or save data in a database. The web clients won’t need to know any of the backend code.
6. **Security**: As mentioned, NTFS offers Access Control Lists so admins can manage permissions for specific files and folders. Many browsers work with HTTPS which provides a more secure, encrypted connection for sensitive user information. The application itself should offer many security features, like user roles so that guests, users, and admins all have different access. The application can also offer 2-factor authentication and automatic password changes.

Resources

Buy red hat enterprise linux subscriptions. (n.d.). Retrieved February 07, 2021, from https://www.redhat.com/en/store/linux-platforms

MacOS server - tech specs. (n.d.). Retrieved February 07, 2021, from <https://www.apple.com/macos/server/specs/>

Pricing and purchasing options. (2021, January 27). Retrieved February 07, 2021, from <https://visualstudio.microsoft.com/vs/pricing/>

Lavieri, E. D. (2019). In *Hands-on design patterns with Java: Learn design patterns that enable the building of large-scale software architectures*. Birmingham: Packt Publishing.

Microsoft. (n.d.). NTFS overview. Retrieved February 22, 2021, from https://docs.microsoft.com/en-us/windows-server/storage/file-server/ntfs-overview

Microsoft. (n.d.). Retrieved February 22, 2021, from https://support.microsoft.com/en-us/topic/ram-virtual-memory-pagefile-and-memory-management-in-windows-b7573aef-3410-18e2-5c65-20a8de889265