

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 | 07/14/2021 | Charles Haines | Added executive summary, design constraints, model description, evaluation, and recommendation. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants to bring their popular game, Draw It or Lose It, to multiple platforms. Draw it or Lose It is currently limited to running in an Android App. I propose that in order to bring the game to as many platforms as possible, a web-based version of the game is the result of this new software project. The Gaming Room will need to host this game on a server. The server will be running the game logic and code which means clients playing the game will only be required to have a modern web browser since the game itself is running purely on the server. The client’s web browser will handle displaying the game and sending the server any input from the player.

## [Design Constraints](#_2et92p0)

* A server (or servers) must be selected to run the game code.
* The server(s) must be able to support many teams playing the game at the same time.
* The operating system of the server should be selected before beginning development of the application.
* A web server application needs to be selected (Apache, windows server, etc.)
* The web server must support the programming language selected to write the game code

It is important that the server OS and Web server technology is selected prior to beginning development because web servers may have limited support for various programming languages and web technologies. Making these selections prior to starting development will help the project stay under budget and avoid costly time delays and to avoid having to rewrite code that is not compatible with the server software.

## [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)

In the UML diagram of the proposed Draw It or Lose It, object-oriented design patterns are used to align with software development industry best-practices. The GameService class is designed using a singleton design pattern to ensure that only one instance of the GameService is running at once. This will keep memory and CPU usage at a minimum since running multiple threads of the GameService will heavily tax the server running the code. The Game, Team and Player classes are all derived from the Entity base class. The Game, Team, and Player therefor inherit all the attributes of the Entity base class. The Game, Team, and Player classes are all in zero to many relationships, meaning that at any given time zero or many of these class instances may exist. The variables associated with the Game and Team class are private in order to encapsulate the inner workings of the classes.

By using these various object-oriented principles, the code created will be reusable, easier to maintain, and more portable than it would be possible if the software were designed without the use of object-oriented design patterns and techniques.

****

## [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 must 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 OSX is based on Unix and is a strong candidate for hosting a web server. Mac hardware is generally more expensive and because of this it is not the best choice for hosting the Draw It or Lose it server. Unless the company has an existing OS X based webserver, I would not recommend a Mac to host the game server. A Unix /Linux specialist will be required to set up a server in an OS X environment. | Linux is an excellent choice to run a web server. Linux is the industry standard for hosting websites and other various servers. Linux is either free or low-cost depending on the distribution selected. The company will require someone who can set up, troubleshoot, and maintain the server. Anyone with advanced knowledge of Linux/Unix should be able to handle setting up and maintaining the server. | Windows would be a strong choice to host the server since windows server environments typically include a web server, however, due to licensing in a Windows server environment, this may prove to be the most-costly option. Running a server that supports large numbers of users will require expensive software licenses. A specialist will be required to deploy and maintain the Windows server. | While mobile devices offer users many freedoms and portability that cannot be matched by desktop platforms, mobile devices do not have the computational power to handle running a large server supporting many users. Mobile devices are designed to use less power, and work with hardware less powerful than what is found in typical desktop computers. Because of these reasons mobile devices are not an ideal host for the game server. |
| **Client Side** | Mac OSX is an ideal client for Draw It or Lose It. OS X has multiple browsers, including Safari, Chrome, Firefox, etc. which are modern web-browsers capable of displaying the game. Since the game is running on the server, there are no additional costs or time associated with making the game available to OS X via a web application. | Linux is an ideal client for Draw It or Lose it. Most modern Linux distributions come with a modern web-browser such as Chromium, Chrome, or Firefox. Since the game code is running purely on the server, no additional considerations are needed to make the game available for users running Linux. | Windows is an ideal client for running the Draw It or Lose It game. Windows runs many modern browsers such as Firefox, Edge, and Chrome. Since the game code is running on the server, no additional considerations are needed to make the game available to users running Windows with the exception that the game may not run on older versions of Windows that are no longer supported due to the lack of a modern browser. | While mobile devices would ideally be supported, mobile web browser technology is not as uniform or consistent as web-browsers created for desktop computers. Because of this, additional consideration will be required to ensure the game is properly displayed on mobile devices. Mobile devices potentially have very large and very small screens. This may pose a challenge if the web client is not designed with mobile devices and browsers in mind. IF the client is delivered as an application, additional consideration needs to be made for the differences between IOS and Android devices. IOS and Android use different programming languages and tools specific to each platform. |
| **Development Tools** | OS X has numerous tools available for software development. Eclipse is a powerful IDE that can be used for many programming languages, including Java and C++. X-Code is another powerful IDE available for OS X but is typically only used to build applications to run natively on Apple products. | Linux is a great choice for developing software is it offers many tools, including two great C/C++ compilers (CLANG, and GCC), java compilers, and various IDE’s such as Visual Studio Code and Eclipse. Linux has a powerful command line that can be used to develop software rapidly even on low budget hardware. | Microsoft Windows is another strong choice for developing software as it can create software from many programming languages such as C/C++, Java, .NET, etc. Microsoft Windows has many powerful IDE’s and editors available such as Eclipse, Visual Studio, and Visual Studio Code) | Applications developed for mobile devices are typically created using a desktop PC. Mac OS X and Windows are the most popular platforms for developing software for mobile devices. Generally, a framework is used in order to ensure software written for mobile devices is compatible with both Android and IOS devices. The various tools used are the same as those listed under Mac OSX and Windows since these same tools can target mobile devices. |

## Recommendations

* **Server Recommended Operating Platform**: Linux (Arch or Ubuntu based distribution)
  + Secure
  + Stable
  + Runs on almost any computer hardware
  + Low cost
  + Arch Linux is only recommended for use by experienced system administrators because it is highly customizable and requires the administrator to choose which parts of the system will be used and/or installed.
  + Ubuntu is a safer choice because it comes with a complete desktop environment out of the box and has support for Docker.
  + Support for virtual and shared memory.
  + Supports many types of database technologies for use in Draw It or Lose It
* **Server Storage Mechanism:** 
  + Locally attached RAID
    - A RAID setup allows for redundancy. Redundancy allows for the system to continue functioning in case one of the disks in the array fails.
* **Network:** High speed broadband connection
  + Draw It or Lose It is a REST style API which the server and client communicate via HTTP requests.
  + HTTP requests can contain parameters that can specify a maximum image size that the client can support.
  + Application can be deployed in a container such as Docker for redundancy, avoiding potential downtime due to unforeseen network problems.
* **Security:** HTTP via SSL/TLS
  + All communications between the server and client will be encrypted using SSL/TLS over HTTP to protect intercepted communications from being read as plain text.
  + IPV6 support will ensure the application is ready for the future of the IP protocol.
  + Server and client should communicate to ensure the same version of software is being used to prevent clients from connecting to old or incompatible versions of the game running on the server.
* **Client:** Firefox / Chrome
  + Most if not all modern operating systems support Firefox or Chrome
  + Since a REST API is used, no specific client application needs to be designed since both web browsers natively support HTTP and secure HTTP (HTTPS) natively.

I recommend running the game server on a Linux server. Linux servers are highly configurable, dependable, and the most cost-effective choice when comparing to other platforms such as Microsoft Windows, and the Unix based OS X from Apple. Most modern Linux distributions can run both 32 bit and 64-bit code. A 64-bit operating system such as Linux will have access to a large amount of memory due to the nature of 64-bit software having a much larger number of memory addresses the system can take advantage of. The Linux server should be outfitted with a large amount of storage in the form of magnetic hard disks. The hard drives can be configured in what is called RAID, which in general terms means that the failure of one or potentially more hard disks will not result in a loss of data or functionality. Magnetic hard disks are slightly slower than solid state drives, however, mechanical hard disks are more cost-effective.

Serving the game from a Linux server ensures that most modern devices will be able to connect via a web-browser and run the game without worrying about compatibility between various devices. Since the game is running entirely on the server, the result of a server outage would mean clients are not able to connect to the game. Running multiple servers can alleviate this issue by allowing the game to run on a backup server in the event of an outage or natural disaster. The server can also make use of containers for application deployment. Containers allow for quick deployment to another machine in the event of a system or network failure.

Linux offers best in class security when compared to other operating platforms. Most Linux software, including web server applications, are open source. Open-source software means that many eyes are viewing the source code allowing security holes to be found and patched rapidly. Linux also offers basic security features such as file permissions, user and kernel mode separation, and password protected user accounts.