

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 |
| --- | --- | --- | --- |
| 3.0 | 04/21/21 | Gayle Sakiewicz | Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room currently has their game, Draw It or Lose It, as an Android application but wants to make it a web-based game. The web-based game will need to be able to work with multiple web browsers. The game needs to be able to have multiple teams with multiple players and one game instance to not be duplicated. Creative Technology Solutions can create an application with a web-based language that can work on multiple platforms and web browsers. A dedicated game server can be used to run the game and communicate with the players and teams over a network.

## [Design Constraints](#_2et92p0)

Internet connection speed will need to be considered when choosing a network provider as the game needs to be consistent with time as the game itself times users. The language for the application is narrowed as it will need to be able to work on the internet and across browsers. Only one game instance for the teams/players can be running at a time which involves programming and memory. The web-based game will need bi-directional communication between the application and the users/gamers, which will affect the decisions of hardware and networking.

## [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 diagram below, the parent Entity class is primarily used for its variables and methods that are common and are inherited among its child classes; Game, Team, and Player. The private variables and methods, or attributes, are denoted with a minus sign (-) so that they cannot be changed or manipulated by any class, whereas the public variables and methods are available and may be used among the other classes. The Game, Team, and Player classes are associated, also showing cardinality of all three can be as little as 0 games, teams, and players at any point in time to many. The GameService class is associated with the Game class as it will keep a list of the games in play in order to ensure that the application won’t duplicate a game. The singleton GameService class also keeps private variables as to assign Id numbers to games, players, and teams. The Team class creates a list of players for each team, as the Game class keeps a list of all teams. The ProgramDriver class uses the SingletonTester to make sure the application is not creating any duplicates of a game from the GameService.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Using Mac OS for the server can be done but is less common. A Mac operating system for an internet server would be ideal if it were being used strictly for Apple device applications. | Linux is the most used operating system for internet servers. Linux servers are more stable, last longer, and have less issues with viruses and malware which is better for security and uptime. Linux servers can run a large number of processes at a time which is good for a multi-player game environment. Linux servers are also more scalable and can be scaled without downtime. Linux does not have a dedicated support system to call but it is open-sourced so there is a lot of public support. | Windows servers are more expensive to run and secure as they are more prone to viruses and malware. Windows OS is more user friendly for those without a dedicated IT staff. Windows servers tend to be slower and less stable because of its proneness to viruses which means more potential downtime for a web-application.  Windows servers tend to need more upkeep and doesn’t last long, it would need to be maintained and monitored more closely. | Using a mobile device for a web application is possible but not ideal for this kind of application. Mobile devices used as servers don’t have the kind of memory as other dedicated servers and would need to always be on and with battery. |
| **Client Side** | People with Mac expertise would be needed for coding the software, Mac also has licensing fees so there are added costs. The use of IDE’s would be the same as it would with Windows or Linux to program an application to run on multiple web browsers. | As Linux is an open-source OS no licensing fees are required and if they are, they’re cheaper. Linux has free IDE’s and have multiple web server software programs. Running a Linux system is not for the average employee, a person who knows the system would be required. Linux servers are faster and have less upkeep so they have less downtime for maintaining a web-application. | More expensive buying everything from Microsoft and licensing fees  A Windows system would require more attention from an IT staff to keep them updated and secure. The tools and licensing fees for Windows are costly. Windows would have a more limited software choice for the application. Windows also requires a lot of memory, which would put more cost on hardware, it could also slow the system down. | Mobile devices have less software applications. Coding and debugging on a mobile device would be time consuming and difficult. |
| **Development Tools** | Objective-C and Swift are used to build software to deploy on Mac. There are multiple IDE’s that can be used on Macs to create the application including IntelliJ, Visual Studio, Eclipse, and NetBeans. | Linux will require standard C programming which provides more portability to other platforms. Linux has many free IDE’s including Eclipse, PyCharm, Spyder, NetBeans, etc. that support many languages.  Some application programming languages for Linux include C++, C, Perl, Java, and JavaScript. | Windows requires Objective C which has less portability. Windows has many IDE’s, some not free, including Eclipse, Microsoft Visual Studio, NetBeans, Atom, PyCharm, IDLE, Microsoft Visual C++, and more.  Application programming languages include C#, Swift, Java, Python, C++, PHP, Perl, and JavaScript. | Depending on the mobile device (i.e. Andriod or Apple) IDEs can include Android Studio, Codea, Pythonista, and AIDE. Languages are limited for mobile devices and they depend on the IDE that is being used. |

## Recommendations

1. **Operating Platform**: Linux is recommended as the operating platform for The Gaming Room.Linux is very portable when it comes to working on other platforms and hardware.
2. **Operating Systems Architectures**: Linux supports multiple architecture including 32-bit, 64-bit, and multiprocessor architectures. Linux has a user mode and kernel mode; the kernel also contains an NPTL thread library that can support hundreds of thousands of threads (processes/tasks) tasks that the application can manage. The Linux operating system architecture layers are comprised of the kernel, system libraries, and system utilities. “All kernel code and data structures are kept in a single address space…this space contains not only the core scheduling and virtual memory code but all kernel code, including device drivers, file systems, and networking code” (2008, Ch. 21, Gagne et al.).
3. **Storage Management**: Linux has two components of memory management, one dealing with physical memory and the other is virtual memory. The physical memory is separated into three zones and the primary memory manager in the Linux kernel is the page allocator (2008, Ch. 21, Gagne et al.). The virtual memory system maintains the address space visible to each process, “creates pages of VM on demand, and manages loading those pages from disk and swapping them back out to disk” (2008, Ch. 21, Gagne et al.).

“Linux also reserves for its own internal use a constant, architecture-dependent region of the virtual address space of every process. The page-table entries that map to these kernel pages are marked as protected, so that the pages are not visible or modifiable when the processor is running in user mode” (2008, Ch. 21, Gagne et al.).

1. **Memory Management**: Linux uses segmentation and paging for memory management and two levels of protection, user and kernel mode. “Linux uses a three-level paging strategy that works for both 32 and 64-bit architectures” (2008, Ch. 21, Gagne et al.). Linux uses a standard UNIX file-system where files can be anything capable of handling I/O. Linux’s virtual file system also “enables us to implement a file system that does not store data persistently at all but provides an interface to another functionality” like the process file system in which the contents are not “stored anywhere but are computed on demand according to user file I/O requests” (2008, Ch. 21, Gagne et al.).
2. **Distributed Systems and Networks**: With a distributed network, The Gaming Room can have separate sites connected in communication with nodes to coordinate the application software and share the workload. If The Gaming Room has a server on one coast and a server on the other via a WAN, this would enhance timing for gamers and also defend against outages. If an outage were to occur where one of The Gaming Room’s site is, the other will still be up and running so the application has better reliability. A distributed system can be implemented so the web-application can be fast, with devices or databases can share the workload of multiple games/gamers.
3. **Security**:

Data encryption can be implemented to protect stored user information such as usernames and passwords. Such securities should include encrypting data being sent between the client and the server and user information being stored on the server. HTTPS and SSL protocols can be used to encrypt communications and an encryption such as 128-bit can be used for the contents in memory and storage for user’s personal data. User authentication should be used to ensure the users’ information cannot be accessed by malicious persons. Security protocols including authentication will be necessary for the company’s assets that run the application so no unauthorized personnel can access the private data. The separation of user and kernel mode add a layer of protection as well as implementing access controls and least privilege.

References

Gagne, G., Galvin, P., & Silberschatz, A. (2008). Operating System Concepts (8th. ed.). Wiley Publishing.