

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 | 03/20/2021 | Lynn McCargar | Cross platform development expansion and upgrade. |

**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 has a need to expand their game “Draw It or Lose It” to multiple platforms. Currently the game is only running on an Android platform. This project will expand the game to a web base application, with one or more teams, multiple players per team, game and player name check, and only one game instance at a time. This project will include images that the game will use, an image repository is part of the design. This project software design that will be streamline to offset the staffs lack of technology experience.

## [Design Constraints](#_2et92p0)

The project constraints during this development is outlined in this section of the software design. Identifying these constraints allows us to mitigate some of the problems during the design phase of project. Here we have listed most if not all of the constraints during the project development.

* Porting or rewriting existing code. Here is we need to review existing code for reusability. If a rewrite is needed, then this will add additional time to the project.
* One instance of the game in memory. To meet the constraint of a single instantiation of the game, we will implement the Singleton pattern. This software pattern is designed to only allow a single instance of the game in memory.
* Web based environment constraints. Since this game is going from only one platform to many, there will be a considerable development effort with different software development core competencies needed to successfully implement the project.
* Game and team names uniqueness of the project. Here is where we will need to build in means to check for duplicate names into the project.

## [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 entity class creates relationship with game, team, and player classes. The all inherit information from the entity class. The UML, demonstrates the inheritance with each class will get common references like, “name” and “Id”. The entity is a superclass in our UML diagram. Looking at the relationship, we see “team” and “player” is a “has a” type. “Game” has a “Team” and “GamesServices” has “Game”. UML identifies it aggregation (HAS A). When we use a “has a” its an instance of one class and has a reference to an instance to another class. Looking at the UML diagram, we see GameService has a reference of Game, Game a reference of Team, and Team a reference of Player.

****

## [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 10.7.3 (Lion) or later. * Administrator privileges for installation. * Java JDK and JRE 8 install as a system wide for all users. * 64 bit browser support. * No 32 bit browser on Java version 7 and up. * Xcode source version control, Git * Apache and PHP servers built in. | * Many Linux distributions available * Built in file, storage, Web server, DHCP, DNS * Git, Repos, etc. * Scalable and cost effective. * Less hardware requirements * No GUI option, faster response | * Windows Server 2016 or later. * Java JDK and JRE 9 supported. * Team Foundation version control, Git, Repos, etc. * File, DHCP, DNS, Web server built in. * Heavy hardware requirements with memory * GUI controls, slower server | * No industry standards for device servers * Third party solutions to convert an Android into a server * Devices with minimal hardware resources like storage and memory * Would not scale easily if needed. |
| **Client Side** | * Integrates well with other Apple devices such as, iPhone, iPad, etc. * Does not support 32 bit browsers in Java development * One of the more expensive option. | * Open source OS * Less expensive option * A little harder to learn as compared to other options. * Not as widely used as a client | * Good backwards compatibility with 32 bit browsers in Java development * Most used in companies. * An abondance of programs to chose from. * Not open source. * More venerable to hacking. | * Widely used * Good for testing between iOS and Android * Very portable * Less hackable * Not used in development historically |
| **Development Tools** | * Macs use swift for development. * Tools like notepad++ are useful. * Macs can run most development languages. But not limited to HTML, CSS, Javascript. | * Linux can be used with Eclipse, Visual Studio. * Can use notepad++. Supporting most languages consist * but not limited to HTML, CSS, Javascript. | * Less of a learning curve compared to other options. * Can run Eclipse, Visual Studio, and most of the IDEs available. * A lot tools like notepad++. | * Swift and Android can ne used to create a multitude of applications. * Both languages and software can be used on the other systems. * But with limitations to development tools and IDEs. |

## 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**: After careful consideration the Windows server is recommended for The Gaming Room expansion project for “Draw It or Lose It” game. The Windows server offers easy to use graphical user interface (GUI) for administration of the operating platform. The interoperability of this platform allows for the client to expand across different computer environments. Easy and optional automatized system updates for secure and easy administration of the platform. The Windows server is integrated with a web, dns, dhcp, etc, services. Along with easy integration of many 3rd party application to help expand their, as well as other opportunities.
2. **Operating Systems Architectures**: The Windows server system is a layered architecture design that consists of two main components, user mode and kernel mode. It is a preemptive, reentrant operating system, which has been designed to work with uniprocessor and symmetrical multi-processor SMP based computers. Windows Admin Center is a locally deployed, browser-based app for managing servers, clusters, hyper-converged infrastructure. System Insights brings local predictive analytics capabilities natively to Windows Server. These predictive capabilities, each backed by a machine-learning model, locally analyze Windows Server system data, such as performance counters and events, providing insight into the functioning of your servers and helping you reduce the operational expenses associated with reactively managing issues in your Windows Server deployments.
3. **Storage Management**: The Windows server platforms offers, Storage Migration Service makes it easier to migrate servers to a newer version of Windows Server. This would aid the client expansion efforts. It provides a graphical tool that inventories data on servers and then transfers the data and configuration to newer servers—all without apps or users having to change anything. System Insights locally analyzes Windows Server system data and provides insight into the functioning of the server. Disk anomaly detection is a capability that highlights when disks are behaving differently than usual. Seeing these anomalous moments can be helpful when troubleshooting issues on your systems. The ReFS filesystem stores up to ten times more data on the same volume with deduplication and compression. The variable-size chunk store with optional compression maximizes savings rates, while the multi-threaded post-processing architecture keeps performance impact minimal. Supports volumes up to 64 TB and will deduplicate the first 4 TB of each file.
4. **Memory Management**: Performance Monitor is the main tool for monitoring system performance and for identifying the location of the bottleneck on the Windows platform. But RAM is a limited resource, whereas for most practical purposes, virtual memory is unlimited. There can be many processes, and each process has its own 2 GB of private virtual address space. When the memory being used by all the existing processes exceeds the available RAM, the Windows operating system moves pages (4-KB pieces) of one or more virtual address spaces to the computer's hard disk. This will ensure the game “Draw It or Lose It” will have plenty memory for their expansion efforts.
5. **Distributed Systems and Networks**: Networking and distributed systems provide the infrastructure for computation, communication and storage involving a diverse and potentially large number of people, devices, and software processes. Some issues of concern include performance, scalability, functionality, and manageability. The primary aims is developing a solution that meets these concerns and utilizes the system architecture, and underlying principles for these networking and distributed systems. These network and distributed areas include protocol, wireless and mobile networks, disruption tolerant networks, sensor networks, WWW protocols and content distribution networks, real-time and multimedia systems, network algorithmics, performance modeling and analysis, network measurement, virtualization, storage, and file systems.
6. **Security**: The security across multiple platforms takes a layered approach. We have the web services that need protect; Firewalls, Secure Socket Layer (SSL) connections, and using security filters passing header sessions, helps to keep authorized calls to and from the user sessions. Hardening the OS and Transport Layer Security (TLS) registry settings. Control Flow Guard (CFG) is a highly-optimized platform security feature that was created to combat memory corruption vulnerabilities. Windows Defender protects from viruses and malware. Much of these concerns are features in the Windows server operating platform.