

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 | <mm/dd/yy> | <Your-Name> | <Brief description of changes in this revision> |

**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 requested us to design a web-based game based on their current game Draw it or Lose it, which must run on multiple platforms. To facilitate the development process, we will streamline the development and design the environment of the game. Our solution is to split our development team into groups with each group developing the game for a specific platform then afterwards we will implement each other code to make the game run in multiple platforms. Now, we asked, what is the budget for this project? What is the deadline for this project? How many people do you want to involve in this project and their skills set?

## [Design Constraints](#_2et92p0)

The design constraints for this project are:

* A game will have the ability to have one or more teams involved. This means that we need to develop a class that creates several teams.
* Each team will have multiple players assigned to it. A class must add players to the teams.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name. We will use an iterator pattern to go through the list of team names and make sure that the same name will not be assign to other teams.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player. An entity class in the app will take care of creating a unique identifier for the game, team, or player classes.

## [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 GameService class is responsible for creating an instance of a game. That instance is passed to the Game class. They have a relationship of 0 to many meaning a game can have many games are playing in one instance of a game. Encapsulation is demonstrated because the Game class cannot access the instance variables of GameService. Game and Team classes show encapsulation as well because both attributes are private. They have a 0 to many relationships meaning a game can have multiple teams. The Team class has a 0 to many relationship with the Player class meaning that a team can have many players. The Entity class inherits the properties of Game, Team, and Player classes and holds their attributes and behavior. The ProgramDriver class is the main class, and it uses the SingletonTester class to test if there is only one instance of a game running.

****

## [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** | The hardware is top-of-the-line, the software is secure and Mac can host run Apache servers similar to Linux. The downside is that Mac web hosting is expensive. | The greatest advantage of utilizing Linux hosting rather than Mac is that is less expensive and Linux runs on Apache servers and can run basic web code, WordPress, forum software but, is not as secure as Mac. | Only Windows hosting can run applications that use ASP, .NET, Microsoft Access, or MSSQL databases because they own the rights to these apps. In the event that you need these technologies incorporated into your site, you must choose Windows. On the other hand while Windows servers can run applications like WordPress and vBulletin, it will regularly be slow and sloppy. | Web hosting on mobile devices allow for compatibility across different types of mobile devices and due to the multi-device support that responsive web design provides across various platforms, it's becoming easy to reach a broader audience than a mobile app can reach. One downside is the challenge of optimizing the website to run on mobile devices with low performance. |
| **Client Side** | For Mac the development takes an average amount time and it’s also expensive. In terms of expertise you’ll need an average skillset to navigate the OS. | The development time is substantial, and the expertise needed is high. On the bright side the cost is low compared to other OS’s. | The expertise and time required is minimal. The cost is more expensive than Linux. | The expertise required are slightly higher than other platforms and it requires a substantial amount of time to develop. The cost is also high. |
| **Development Tools** | We can run swift. We can also use tools like notepad++ . Programming languages for Macs include HTML, JavaScript, CSS. | You can run visual studio, eclipse, and notepad++. You can also use programming languages like HTML, JavaScript, CSS. | You can also run tools like visual studio, Eclipse, and notepad++. Along with languages like HTML, JavaScript, CSS. | You can use Android and Swift to create apps. Programming languages you we can use include JavaScript, Ruby, Scala, Python, etc. |

## 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**: My recommendation is Windows.
2. **Operating Systems Architectures**: The Windows architecture consists of a HAL (Hardware Abstraction Layer) which allows the operating system with a hardware device, a driver which allows an operating system and a device to communicate with each other, a microkernel which provides the mechanism to needed to implement an operating system, executive Services. In Kernel mode, the executing code has completed and unrestricted access to the underlying hardware.
3. **Storage Management**: I would use Storage Resource Management (SRC) which can help a storage administrator automate the backup and data recovery. The application can also help the administrator configure the management and performance monitoring, predict future storage expansion more accurately and understand where and how to use tiered storage, storage pools and thin provisioning.
4. **Memory Management**: In terms of memory management Windows uses a method called Paging to map the virtual address space to the physical address space, in such a way that all processes can get to run on the physical memory. If physical memory is big enough to execute all processes, then the virtual address space is mapped to physical address space at once. If physical memory is less than the memory demand, then the virtual memory manager will have to load each process sequentially on the physical memory, wait for the process to end, and then rewrite a new process on the physical memory and repeats this till all processes committed to virtual memory are executed.
5. **Distributed Systems and Networks**: Distributed applications often use a client-server architecture. Clients and servers share the work and cover certain application functions with the software installed on them. A product search is carried out using the following steps: The client acts as an input instance and a user interface that receives the user request and processes it so that it can be sent on to a server. The remote server then carries out the main part of the search function and searches a database. The search results are prepared on the server-side to be sent back to the client and are communicated to the client over the network. In the end, the results are displayed on the user’s screen.
6. **Security**: We can use User Account Control which protects users by preventing malware from damaging a machine, and helps organizations deploy a better managed desktop. When this feature is enabled, apps and tasks always run in the security context of a non-administrator account, unless an administrator specifically authorizes administrator level access to the system. Another security that can be use is Windows Defender Exploit Guard which includes exploit protection, attack surface reduction rules, network protection, and controlled folder access.