

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 | 02/17/2021 | Chantel Williams | Recommendations |

**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 looking to develop a web-based game that serves multiple platforms based on their current game Draw it or Lose It. This game is currently only available on Android, but they are looking to expand so that it can become accessible to everyone. To win the game your team must guess the puzzle within a minute before their time expires. If they fail to do so, the other team then gets one chance to guess within 15 seconds to win the round. The team with the most won round will win the game.

As discussed, the following requirements have been asking for the gaming room application.

* We should establish a game system that will allow one or more teams to participate.
* Establish a system where teams will allow multiple players to be assigned to it.
* Establish a database that will restrict game and team names to be unique.
* Create another database where unique identifiers for each instance for game, team and player are recorded.

## [Design Constraints](#_2et92p0)

The following constraints have been identified based on the requirements discussed:

* The capacity of the server to host a certain number of players. We should anticipate the expected number of players initially using this service to ensure our server can handle the capacity. If the server needs to be upgraded to handle more users/players, we can reevaluate based on collected data.
* We will need enough storage on the server for stock drawings and other art assets.
* We will need to develop a secure log in for all players to allow the application to differentiate each player from all other players.
* To accommodate player login, we will need a database that can store this information and can expanded in the future to accommodate more players.

## [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)

Game, Team & Player classes all inherit information from the Entity class. Since this uses inheritance, we know that each of these classes will share common information like Id and name. The UML also shows us that GameService, Game, Team & Player are all references of each other because they share common game information such as how to get the players’ names and Id. GameService has a List data structure so that the GameService class has a collection of Game. The Game class will have a similar data structure so that each instance of Game has a List of Team. Furthermore, Team will have a list of Player that is a part of that Team. For GameService you cannot get the values of nextGameId,nextPlayerId, and nextTeamId unless you use the public functions such as getGamename, & getNextPlayerId, and getNextTeamId. Games cannot be added directly to the collection of games unless we use the function addGame. This encapsulation is also repeated in Game class where you cannot add Team directly to the collection without using the addTeam function. The Game Service Class has two functions that are named getGame. That is called method overloading and represents a static form of polymorphism. Method overloading is when you implement multiple functions within the same class that have the same name but different sets of parameters.

****

## [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** | **Pros:**  Mac is built off a Unix based platform, so most Unix based web administration capabilities would work on this.  **Cons:**  Mac is more expensive than Linux or Windows.  There could also be some diagnostic tools that are not available for Mac OS due to less people developing for Apple.  It can be hard to find tech support for Mac due to it not being a commonly used server platform. | **Pros:**  Free operating system.  **Cons:**  Since the OS is free, there it little to no support when problems occur & you might have to pay for additional support. | **Pros:**  Windows will be cheaper than Mac but more expensive than Linux.  Because of its availability, they have a lot of services and tools that would help with the maintenance of the server.  **Cons:**  Maintenance would still cost more than Linux but less or equal to Mac because of product availability. | **Pros:**  Cost effective for smaller projects and gives you mobility.  **Cons:**  May not be able to handle all the need and requirements of the company.  Bigger projects may not run as efficiently. Overall, other options have better specifications that would meet the needs of the company. |
| **Client Side** | **Pros:**  Supports most browsers and default is Safari.  Web layout will be easier to design and test because monitor aspect ratio can be standardized.  **Cons:**  Most Mac users know how to use safari but people who don’t own a Mac may have trouble with initial navigation. | **Pros:**  Web layout will be easier to design and test because monitor aspect ratio can be standardized.  **Cons:**  Linux has to many browsers to support so this will be hard to maintain.  Not many people know how to use Linux so they will need to be trained how to use Linux. | **Pros:**  Supports most web browsers and default is internet explorer.  Most people know how to use windows more because of its availability.  Web layout will be easier to design and test because monitor aspect ratio can be standardized. | **Pros:**  Most smart phones have a default web browser that will be supported.  **Cons:**  Screen size varies so it will take more time to test and develop the client interface to work on several different resolution sizes.  You will be limited to smart devices only, but the default web browser will depend on the brand. |
| **Development Tools** | **Cons:**  You must test how to web page looks on different browsers.  Basic HTML will work.  **Pros:**  Java script can be used for graphic assets. | **Cons:**  You must test how to web page looks on different browsers.  Basic HTML will work.  **Pros:**  Java script can be used for graphic assets. | **Cons:**  You must test how to web page looks on different browsers.  Basic HTML will work.  **Pros:**  Java script can be used for graphic assets. | **Cons:**  You must test how to web page looks on different browsers.  You will need to develop for desktop and mobile devices.  Java script isn’t available.  **Pros:**  Basic HTML will work. |

## 

## 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**

The recommended environment would be Windows operating platform. Windows will be more affordable and more available than other operating platforms. Its availability allows more services and tools that would help with the maintenance of the server. Windows 10 Enterprise is recommended because it has been used by those in the industry.

1. **Operating Systems Architectures**

The core of an operating system is the kernel in which Microsoft uses a hybrid kernel. The hybrid kernel is much like microkernels as they use a server application to provide more functionality, except they include some additional code in kernel space to increase performance. The hybrid model has scheduler activation, allowing the thread library to make smarter scheduling decisions.

1. **Storage Management**:

Windows 10 comes with a built-in store management feature. Storage sense allows you manage files on your hard drive while automatically freeing up disk space when needed. It is compatible with other built-in cloud storage systems such a one drive which allows you to save and manage data on the cloud. Windows also allows you to add more storage to your system by adding extra drives, this will improve performance and better protect your data.

1. **Memory Management**:

Since Draw it or Lose it will require a large database of pictures, we will need to look into the built-in memory allocation services that Windows provides. We will be using hard drives in the server to store our pictures and other account information, then the pictures will be loaded to RAM when it is needed. We can expand our hard drives on an as needed basis.

1. **Distributed Systems and Networks**

Unity would be the program of choice to make Draw it or Lose it accessible across all platforms.

Unity’s cross platform feature allows you to build once and then deploy your project to any other platform or device as needed. Its built-in network discovery feature allows you to connect to a game on a LAN or through another built-in third-party feature known as MatchMaker.

1. **Security**:

Windows comes with a built-in security feature that continually scans for viruses, malicious software, and any security threats. We could bump our security up a notch by adding other programs such as McAfee and Norton Antivirus to our pipeline to better protect and encrypt user data. To protect our data, we should ensure that user information is encrypted. We also want to protect our servers by having firewalls set up to prevent unauthorized access of the server.