

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115639452)

[Table of Contents 2](#_Toc115639453)

[Document Revision History 2](#_Toc115639454)

[Executive Summary 3](#_Toc115639455)

[Design Constraints 3](#_Toc115639456)

[System Architecture View 3](#_Toc115639457)

[Domain Model 3](#_Toc115639458)

[Evaluation 4](#_Toc115639459)

[Recommendations 7](#_Toc115639460)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 09/17/22 | David Vega | Initial Draft |
| 1.1 | 10/02/22 | David Vega | Added Evaluation |
| 1.2 | 10/16/2022 | David Vega | Added Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room, a client of CTS, has asked us to develop a web-based version of their game called “Draw It or Lose It.” This game is akin to the 1980’s television game show where contestants draw pictures for their teammates to guess. Rather than drawing the pictures, players will be presented with partially rendered images that are completed at the 30-second mark. The team will have to guess the image before the image is fully rendered. If the time expires then the opposing team will have 15-seconds to guess the image. The game consists of four rounds of play that last one minute each.

The Gaming Room has advised us that the web-based game should only have one instance of the game at a time. In order to do this, we will implement a singleton pattern in our design to ensure that there is only one instantiation of the game utilizing unique id’s.

## [Design Constraints](#_2et92p0)

It is critical to note that this project is not a redesign or new design of the game but rather an implementation of an existing product. Therefore, in order maintain the integrity of the existing application we will need to determine if the technologies used to operate this game is compatible with many browsers. Cross-browser testing would be required to test the components of the game. The old adage “Time is Money” is also a constraint as with any design project, cost is a major factor in whether the project goals can be achieved or not. Quality, scope and time to implement a project are all impacted by budgetary constraints.

## [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)

Diagram

Description automatically generated

The above diagram illustrates the application’s logical implementation. The entry point to the game application is through the ProgramDriver class which initiates the GameService class that is responsible for the creation of the games, teams, and players. This class utilizes the singleton pattern by setting the GameService() constructor as a private member in the class. This ensures that there is only one instantiation of the game at any given time.

There are a variety of object-oriented programming principles within the design of this application. For example, the Game, Team, and Player classes are subclasses of the super class Entity. The Entity class inherits from the subclasses Game, Team, and Player. There is also the principle of encapsulation where private data members can only be used by other classes with direct access.

## [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** | As of April 21, 2022, Apple discontinued OS X Server and migrated its services to macOS. The latest version of OS X server is available for download with operating systems running Big Sur. There were some features that were removed since the migration but still allow current customers to take advantage of some of its more popular features such as Caching Server, File Sharing Server and Time Machine Server which are bundled with every install of macOS High Sierra and later. | Linux is one of the most popular operating systems for web hosting. The OS is open-source, and the public is free to use, modify, and share it. Once more, the OS is free. The zero-price tag is certainly a benefit when implementing or configuring a web hosting server. There are tons of resources and support for configuring this type of setup. Developers often prefer Linux because of its affordability, flexibility and reliability. | Windows servers are good for database driven pages because of its compatibility with MS Access. MS Access is GUI based and easy to use since mapping is virtually a drag and drop feature. This makes for an easy-to-use platform for non-technical users. Aside from its user-friendly features, the price to implement a Windows server can range between $500 - $6000+. The Windows server out-of-the-box is convenient for those with less experience with command line tools. | Mobile devices can host a basic web application that doesn’t require multi-user access or multiple resources running simultaneously. It is a great way to remove hosting costs. However, scalability is not something that would be expected from this type of setup. |
| **Client Side** | Mac development requires apple hardware. Xcode comes built into macOS. However, development is limited to Objective-C and Swift. There is an open-source framework that makes use of the Python language called Kivy. | Linux development can be done with C/C++, Python or Java.  There is support for multi-users in the GNU/Linux platform. However, it is not as widely used. | Windows is the preferred platform for software development. It supports multiple programming languages and tools.  It is easier to test applications than on other operating systems. | Mobile devices are intended to be single user clients.  Apple devices are limited to the use of proprietary IDE Xcode and languages such as Objective-C and Swift, whereas, Android SDK can supports a commonly used programming language such as Java. |
| **Development Tools** | macOS supports C, C++, Objective-C and Swift.  XCode is the proprietary IDE used in iOS development. However, there are other text editors that can support development for macOS applications, such as Visual Studio Code.  Xcode is free to download on mac systems but a fee of ~$99 is required for its cloud-based subscription. | Linux development is primarily in C.  There are multiple IDE’s that support the C language and compatible with Linux operating systems, one of which is Visual Studio. An alternative to Visual Studio is Eclipse IDE.  Both IDE’s are free to download. | Windows operating systems can support multiple programming languages but is primarily written in C and .NET.  There are multiple IDE’s that can be used but the most popular is Visual Studio.  Visual Studio community package is free to download but there are more robust versions of this software for a fee that ranges anywhere from $500 - $6000k depending on your needs. | Android development is done with the Google proprietary IDE Android Studio. This IDE is free to download.  iOS is exclusively developed in XCode land in the Objective-C and Swift programming languages. |

## 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**: Linux based servers seem to have more flexibility than other operating platforms. Since the platform itself is open source, it is easily accessible and can be customized to the end-user’s specifications.

Both front-end and back-end can communicate via API’s rather than through a web-service.

The API’s offer better security with keys and can also support encryption. The data can either be

Output in JSON or XML.

The implementation of a Linux operating platform is also not dependent on the development

language as all languages is supported since the data will be converted to either JSON or XML.

1. **Operating Systems Architectures**: There should be a back-end server that manages the logic for handling data while the client can handle front-end rendering. The advantage to this architecture is to reduce the number of calls to the API. Costs are determined by the number of calls made to the API. Isolating the front-end to the client can help reduce costs as well as provide users with an increased performance since the client can provide caching to help load images much quicker than having to make a call to the API.
2. **Storage Management**: There is no right or wrong when choosing a storage device. It is subjective and can vary with the needs of each end-user. With that said, Solid State Drives (SSD) are becoming the “go-to” storage device for modern computing systems. In comparison to Hard Disk Drives (HDD), the SSDs are less prone to malfunction because of the lack of mechanical components that the HDDs have and are portable. The SSD is a bit more expensive but well worth the investment to have peace of mind. That is not to say that SSD drives are indestructible, as they are prone to damage because they are in fact a physical device. More users are opting for SSD storage with cloud storage as a back-up. There are other cloud storage devices that can be purchased if users want the ability to manage the storage themselves or they can use a cloud storage provider.
3. **Memory Management**: The Linux distribution uses a technique called Page Cache. The main idea behind this is to store frequently used data into cache (RAM) so that the data is served more quickly than having to read from the disk (HDD or SDD). Cache is considered system memory and can be freed up because they can be reclaimed to meet the demands of a system. In other words, when data is read, it checks the page cache to see if the data has been requested before, if not a new entry is made into page cache and is filled with the data from the disk. In the case of the game “Draw It or Lose it”, the images would be placed into page cache once they’ve initially been retrieved so that in future instances of the game the images would load quickly.
4. **Distributed Systems and Networks**: In smaller projects a distributed system can help manage

shared resources on a network. This type of system can share resources such as hardware, software, or data. In larger projects, then configuration for this type of architecture can be more complex but still more efficient than a monolithic single server-based system. Scalability is often the reason for such a system since extending to multiple machines multiplies processing capabilities thereby improving performance and reducing failures. If one of those machines fail, the remaining machines can continue to operate without disrupting the overall computation. Since both the front-end and back-end will be accessing data through API’s, various operating platforms are supported with this architecture.

1. **Security**: As part of the development, there will be a strong focus on security with the APIs. Prioritizing security while in development will help to identify any potential vulnerabilities. The APIs will use Transfer Layer Security (“TLS”) encryption. This protocol is widely used in the transfer of sensitive data between applications over the internet. Implementing a web application firewall that can understand an API payload is also recommended as it can help filter HTTP traffic.