

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 2.0

## Table of Contents

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

[Table of Contents](#bookmark1)**2**

[Document Revision History](#bookmark2)**2**

[Executive Summary](#bookmark3)**3**

[Requirements](#bookmark4)**3**

[Design Constraints](#bookmark5)**3**

[System Architecture View](#bookmark6)**3**

[Domain Model](#bookmark7)**3**

[Evaluation](#bookmark8)**4**

[Recommendations](#bookmark9)**5**

## [Document Revision History](#bookmark10)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | <07/23/2024> | Jerome Talanquines | Sections edited  Executive Summary  Design Constraints  Domain Model |
| 2.0 | <08/04/2024> | Jerome Talanquines | Evaluation  ++Revised |
| 3.0 | <08/11/2024> | Jerome Talanquines | 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](#bookmark11)

The Gaming Room wants help in developing a web-based game that serves multiple platforms based on their current game, Draw It or Lose It. The client requested that the following software requirements for the game application to contain one or more teams involved, each team will have multiple players assigned, unique names for users and teams, lastly, only one instance of the game can exist in memory at any given time. The staff at The Gaming Room does not know how to set up the environment for their project and has asked us CTS in streamlining the development.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#bookmark12)

Support of multiple players - The application or game must be able to support multiple teams or players in a distributed environment. Without managing and proper planning, the intended project will not be able to function as intended.

Unique team names - The game application will contain unique names for users and teams. Without proper management of data, the application will reach conflict such as duplication of naming.

One instance of the game can exist at any given time - Only one instance of the game can be of existence at any given time. Ensuring that the application is not subject to memory conflict or leakage.

Cross platform compatibility - As requested for the framework the game needs to be compatible with various platforms or devices. Planning must be designed meticulously for this system to be implemented.

Live time interaction - Live time interaction needs to be present while the application is in use. Through communication technology. The system needs to be able to process speeds and data uploading while connected to the user's framework.

## [System Architecture View](#bookmark13)

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](#bookmark14)

* Entity Class model contains attributes and methods. Entity provides functionality to its subclasses.
* The GameService class is responsible for managing the games. The class contains the singleton design pattern using the getInstance() method. GameService is responsible for adding and setting games. GameService is associated with the Game class and has a 0 to many relationships.
* Game class inheriting from Entity includes methods to add teams and to provide a string representing of the Game class with a list of type Team. Game associates with the Team class and has a 0 to many relationship.
* Team class inherits from Entity includes methods to add players and to provide a string representation of the team with a list of type Player. Team associates with the Player class and has a 0 to many relationship.
* The Player class represents a player in a team and includes a string representation of the class with the toString() method.
* The ProgramDriver serves as the main method and is the entry point for the program. Program driver also has a dependency relationship with SingletonTester indicated by <<uses>>.
* SingletonTester class uses the singleton pattern and is used to test the pattern.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#bookmark15)

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 must 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 hosting a web-based software application offers high performance processors and efficient storage devices. Mac also being a Unix-based system provides a stable and modern environment for hosting web-applications. MacOS is also known for its intuitive interface and ease of use which enhances productivity and makes developing tasks smoother. A weakness of the operating system is not being typically used on production servers for high-traffic websites. As for large-scale applications, dedicated servers or cloud services, the servers show more appropriateness. Also, MacOS at times does not show compatibility with certain software and requires alternative solutions and patches. | Linux being an open-source license free operating system helps reduce software licensing costs. Linux distributions like Ubuntu Server, CentOS, and Debian are optimized for handling web traffic and application loads. Linux is also the foundation of modern cloud-native applications like Docker. Showing a Cloud-Ready OS. Linux, being an uncommon source, comes with advanced features and command line operations. The servers run without a graphical user interface. There are also potential software compatibility issues and administrative complexity. | Windows offers a friendly graphical user interface. Making it accessible for administrators and users who prefer a visual environment for managing server, application, and services. Windows servers include Microsoft SQL server, Exchange server, and SharePoint. Makes it a strong choice for organizations that rely on systems for their operations. Windows is also well integrated with the Microsoft ecosystem. Windows servers generally require more resources in terms of RAM, CPU, and storage in comparison to Linux servers. Windows servers, although possessing advanced security features, have been more vulnerable to attacks in comparison to Linux. | Mobile devices offer flexible platforms for hosting and accessing web-based software applications. Strong advantages in terms of connectivity, portability, and modern hardware capabilities. However, limitations with the devices include resolution, battery life, and security concerns that must be addressed depending on the application. Mentioning optimal performance for mobile users, it is good to consider, intended user base, device specifications, and the nature of the web application. |
| Client Side | MacOS is optimized for graphical performance. This shows usefulness when developing and testing web applications. MacOS supports a wide range of frameworks Node.js, React, and Angular showing versatility for building web applications. Although MacOS hardware tends to be on the more expensive side. Safari, the main browser and especially the older versions may not support older web technologies. | Supporting multiple types of clients on Linux based systems involves consideration across time, cost, and expertise. Multiple clients call the need for expensing for specialized tools, testing infrastructure, and additional maintenance efforts. Since there are considerations for this uncommon system. Additional challenges will be involved with time like compatibility and performance. Expertise must be shown in Linux. Cross-platform design, security project management, and UX design help successfully manage the complexities of this application. | Developing software and support to multiple clients of Windows considerations. As for costs, the servers for Windows can be expensive, in particular enterprises that utilize multiple servers and advanced features. Time considerations such as testing, debugging, integration and deployment must be managed properly. Finally, as for expertise, considerations need to be made when understanding the Windows ecosystem. Including tools, testing, and security. | Software development considerations with mobile devices involved with cost would be analyzing development tools and frameworks. React native or flutter can reduce costs by allowing developers to write code but at the cost of additional costs, time, and maintenance. Ensuring usability across multiple devices and operating systems can reveal time consuming practices. To fix this issue, effective test strategies, such as automated tests and using cloud-based device farms will improve the process.  Mobile devices require a different type of expertise when considering the cross-platform approach and frameworks. Native development programming and teams can improve time constraints and integrations. |
| Development Tools | Several programming languages can support the MacOS. Languages such as JavaScript, HTML/CSS, Python, and Java. And IDE’s such as Visual Studio, PyCharm, JetBrains, and Eclipse also support Mac. Showing versatility in development tools helps with front and backend development. | When building deployable software for Linux several languages and tools come to mind. Front-end and back-end performance languages like JavaScript can assist deployment. For desktop development a language like C++ commonly used with the .NET Core framework can assist with application. IDE’s such as CLion, NetBeans, IntelliJ. Lastly, text editors like emacs or sublime text can improve the programming experience for Linux users. | When using Windows certain tools will show improvement in the system when deploying software. Programming languages such as VB.NET, SQL, TypeScript as well as several other popular programming languages improve the quality that Windows can produce. IDEs including Visual Studio, Xamarin, and Atom can be useful for the OS.  You also can identify several tools and frameworks such as Nuget, Azure DevOps server and SQL server management studio which gives Windows more flexibility. | Relevant programming languages such as tools and IDEs can be used when deploying software on mobile devices. Native development for iOS languages such as Swift and Objective-C or for Android including Kotlin and Java improve the development of these systems. Additional tools and services such as testing tools like Appium and XCTest help with automated testing and applications. IDEs and cross-platform development names like Expo, Dartpad, and Android Studio come to mind when studying the deployment of software. |

## Recommendations

Analyze the characteristics of techniques specific to various systems architectures and recommend to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: <Recommend an appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.>

An appropriate server that I would recommend would be anything that supports cloud-based computing. As requested by Gaming Room, I feel it would be right to choice to use an operating platform that supports this concept. Therefore, I feel Windows would be an appropriate choice for this question. The system offers scalability potential with widespread usage and familiarity and easy integration of Microsoft services like azure.

1. **Operating Systems Architectures**: <Describe the details of the chosen operating platform architectures.>

Windows platforms offer advanced security features, customer support, new software, legacy systems,

and modern systems. Security features that include consistent updating of the system's policies and including enterprise security features such as antivirus protection. Customer support online chat, phone services and cross-platform assistance. Windows shows compatibility with older systems and continuously updates their system ensuring libraries are and software’s are functional.

1. **Storage Management**: <Identify an appropriate storage management system to be used with the recommended operating platform.>

New technology file system, which is the default system for windows, would be a good recommendation for the platform's storage system. Provides a full set of features including security descriptors, encryption, disk quotas, and rich metadata. It can be used with Cluster Shared Volumes (CSV) to provide continuously available volumes that can be accessed simultaneously from multiple nodes of a failover cluster.

1. **Memory Management**: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>

Memory is used when data is stored for a short time. When a computer loses power, the data is lost. Memory is an electronic component that can store data and information on a temporary basis. The software Draw It or Lose It uses a combination of dynamic allocation and cache management to ensure that the application runs smoothly.

1. **Distributed Systems and Networks**: <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>

To ensure that the application Draw It or Lose It allows communication between various platforms the system must ensure that the system is operating on the web with back-end and front-end services containing source code that creates compatibility. To accomplish this, it is vital to make sure that your APIs facilitate cross-platform techniques, this would be demonstratable through programming practices like RESTful. Programming languages and libraries are useful for creating this architecture.

1. **Security**: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.>

User information across various platforms will contain security through the encryption, individual platforms and as requested by the client the cloud. The capabilities and user protection from the recommended operating platform Windows will show similar design in comparison to the other compatible clients. The demonstration of passwords in the form of encryption and built in security features demonstrated by the cloud will provide the web application on the browser the system it requires. It is also worth mentioning that most operating platforms will demonstrate their own personal security.