

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01-22-2023 | Blake Kemp | Prototype for the design of the software |
| 1.1 | 02-03-2023 | Blake Kemp | Second Draft |
| 1.2 | 02-15-2023 | Blake Kemp | Final Draft |

**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 currently in the process of developing a web-based version of their popular Android game, Draw It or Lose It. The new game will be designed to be cross-platform compatible, allowing players to access it from a variety of devices. fixed game will feature multiple games and teams, with each team comprised of multiple players. For the game to function properly, it will be necessary to implement a system for ensuring the uniqueness of the game and team names. What's more, the game will utilize a large library of stock drawings, which will be incorporated into the gameplay. However, the development team at The Gaming Room lacks the necessary knowledge and experience to set up the necessary environment and infrastructure for the game.

## Requirements

**Business Requirements:**

* Development of a web-based version of the current Android game, Draw It or Lose It, to be cross-platform compatible
* Implementation of a system for ensuring uniqueness of game and team names
* Utilization of a large library of stock drawings

**Technical Requirements:**

* Setting up the necessary environment and infrastructure for the game
* Cross-platform compatibility for the game to be accessible on a variety of devices
* A system for ensuring the uniqueness of game and team names
* Integration of the large library of stock drawings into the gameplay
* Knowledge and experience in web-based game development.

## [Design Constraints](#_2et92p0)

## When developing the web-based version of Draw It or Lose It, several design constraints must be taken into consideration. One of the main constraints is the need to create an API that is tailored to work with 3 different platforms, specifically Android, IOS and the web. The API must be able to allow 1 or more teams from any of the platforms to participate in the game. Additionally, the game and team names must be unique, and the API should have the ability to alert the team captain if a team name already exists, allowing them to choose another one. To further limit instances of the game to one, the use of unique IDs for each instance of a game, team, and player should be implemented. These design constraints must be considered to ensure the proper functionality and security of the game across all platforms.

## [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 ProgramDriver Class serves as the entry point of the program and contains the main method. It uses the Directed Association and SingletonTester classes to check for an existing instance of GameService. The Entity class acts as the parent class for the Game, Team, and Player classes, with these classes inheriting the required attributes from Entity. In terms of relationships, a Player cannot belong to multiple Teams, a Team cannot belong to multiple Games, but a Game can have multiple teams, and a GameService can only have one instance of each game running at any time. Also, each Game can only have one unique Team at any time and each Team can only have one of Each Player at one time. These constraints are implemented to ensure that the game runs smoothly and to avoid any conflicts or duplicates.

**"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](#_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** | Macs are known for their reliability, stability, built-in security features and compatibility with a wide range of software and development tools, but they are more expensive and less customizable than other types of computers.  The cost of OS X Server is $499 for 10 clients, and $999 for unlimited client support. | Linux is an open-source operating system known for its flexibility, security, and cost-effectiveness, but its lack of a unified user interface and lack of compatibility with some proprietary software can be a disadvantage for hosting a web-based software application.  Additionally, cloud providers such as Google and Amazon generally prefer to offer Linux over Windows. | Windows is a widely used and familiar operating system known for its compatibility with a wide range of software and devices, but it can be prone to security vulnerabilities and requires regular updates and maintenance, making it less suitable for hosting web-based software applications.  The cost of licensing for Windows Server can be substantial, especially when compared to Linux. Windows Server licenses can range from $6,200 for up to 16 core licenses to $500 for up to 50 clients per installation per year. It's worth noting that the hosting options for Windows may be more limited compared to Linux. | Mobile devices, such as smartphones and tablets, offer convenience and mobility, but their limited processing power, storage capacity, and screen size can make them less suitable for hosting web-based software applications that require heavy computing resources and large amount of data storage.  Another disadvantage of Windows Server is that the hardware requirements tend to be more limited, with limitations on factors such as RAM, and the servers are not as easily scalable as blade servers. The cost of hosting is uncertain, as it may require in-house design and development of the necessary tools. |
| **Client Side** | When supporting multiple types of clients on Mac, software development considerations such as cost, time, expertise for maintaining separate versions of the software, testing for compatibility, addressing potential security vulnerabilities, and having access to Mac hardware and a developer proficient in MacOS software development kits (SDKs) should be considered.  Finally, it's important to consider that Windows has a much larger user base, with a market share of 75%, compared to macOS at 16%. This means that the market opportunity for macOS is smaller. | In terms of cost the most expensive part of this would be the development time, Additionally, using a Linux based OS would be slightly challenging from a learning standpoint.  In addition, multi-user support is a feature of the GNU/Linux platform. However, the development of GNU/Linux may have limited value due to its relatively low market share. | Minimum time and skill needed.  Like mac when looking at price. The most important need for Windows is undoubtedly competence. Would strongly suggest utilizing the.NET framework for its capability and security.  Windows has been a native multi-user operating system since Windows XP. Furthermore, with a market share of 75%, it is the preferred choice for most computer users, making it a stronger option from a business perspective. | Allows clients or even developers the freedom to access updates wherever they are.  a little more challenging to use than other gadgets.  The Android SDK is based on Java, so code developed for Windows and Linux could serve as a starting point. On the other hand, iOS uses SWIFT, meaning that the same requirements for Mac apply, including hardware needs. |
| **Development Tools** | To build software for deploying on Mac, relevant programming languages such as Swift and Objective-C, tools like Xcode IDE and Cocoa application development framework, and additional tools like Git for version control, Core Data, Core Animation, and Core Location are commonly used.  XCode, the integrated development environment for iOS, is priced at $99 per year per developer. | To build software for deploying on Linux, relevant programming languages such as C/C++, Java, Python, and PHP, tools like Eclipse, IntelliJ IDEA, and GCC, and additional tools like Git for version control, Linux libraries and frameworks like GTK and QT, are commonly used. | To build software for deploying on Windows, relevant programming languages such as C#, C++, Visual Basic, and Python, tools like Visual Studio, Windows SDK, and MSBuild, and additional tools like Git for version control, Microsoft libraries and frameworks like the .NET framework, are commonly used.  The cost of Visual Studio, a development environment for Windows, ranges from $45 to $250 per user per year, depending on the features included. | To build software for deploying on mobile devices, relevant programming languages such as Java and Kotlin for android and Swift and Objective-C for iOS, tools like Android Studio and Xcode, and additional tools like Git for version control, mobile development frameworks like React Native and Xamarin, are commonly used.  XCode, the integrated development environment for iOS, is priced at $99 per year per developer. |

## Recommendations

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

**Operating Platform**:

To expand Draw It or Lose It to different computer environments, we advise using Windows above all other operating systems. These factors make this platform a serious contender:

* + Works with the most recent Android build. Install Android Studio or select a cross platform program such as Xamarin, React, or Cordova to set up your windows development environment.
  + Windows makes it simple for developers to create, develop, and distribute software for Windows PCs by utilizing several Microsoft products.
  + Numerous cmd/PowerShell/Ubuntu emulators are available for Windows, enabling testing on all platforms.
  + Most widely used operating system and largest user base for game development.

1. **Operating Systems Architectures**: Windows operating system consists of two main components, User mode and Kernel mode, that perform distinct functions. User mode processes influence most of the user's interactions with the system, while Kernel mode deals with system-level functions such as input/output operations, memory management, networking, hardware management, and system routines. Windows has a directory structure for organizing and storing data, enabling easy navigation and location of files. Additionally, Windows allows hardware modularity and multiprocessing, which facilitates system customization according to the user's specific needs. Understanding these components and customization options can help users effectively utilize and optimize the Windows operating system.
2. **Storage Management**: I strongly advise using Microsoft Azure for storage because of its affordable costs, first-rate customer service, and ongoing improvements and support. Azure also provides the following extra features: To use cloud storage instances, Docker containers may be installed in the Azure Cloud computing environment. -Depending on the user base, cloud-based storage may be simply scaled up or down. The storage might be expanded during the initial launch when it is expected that there would be a lot of users.

* -Azure offers the Azure File System, Azure Storage Containers, and AzureBlob Storage as storage alternatives. 200 8MB Base Game Images for "Draw It or Lose It" would be kept in Azure File Share, which offers 1.6GB of storage for each user.

1. **Memory Management**: The latest version of the Windows operating system, Windows 10, has once again enhanced memory management for faster and more efficient loading from memory. The main methods for accomplishing this are disk paging and demand paging, which serve as an addition to the computer's RAM and physical memory. Disk paging does this by setting aside space on the hard drive for additional RAM. Demand paging divides processes into smaller tasks that are only put into memory when a job is urgently needed for processing. The virtual memory address space can be fully used by every process in Windows 10, which is more than enough for applications.
2. **Distributed Systems and Networks**: Azure's accessibility while working with networks and dispersed systems is another reason, we advise choosing them as your cloud service provider. With cloud-based email alerts, Azure App insights Logging, and monitoring tools available, Azure guarantees optimal uptime. This will be critical when scaling to allow 1,000 simultaneous matches with 4 players each. - By shifting the network burden to Azure, you can put more of your attention on the application's features.
3. **Security**: Azure makes the logistics of user information and personal data security simple. The connected devices will establish a connection with the "Azure App Service" that is active on an App Service Plan and uses Azure Active Directory to authenticate users. Azure offers several supplementary features, including

* IP settings for whitelisting certain access to resources (like a player or personal information) or the entire app.
* A cloud storage solution that includes a VPN for increased security
* A database might be password-protected, only allow access from IP Whitelists, and demand SSL communication to safeguard user data.
* Possibilities to obscure user data to safeguard private information in case of a breach.

Note: To communicate with the cloud, you must have an internet connection.