

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## 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 | 01/16/25 | Jas King | Revisions were made to the Executive Summary and Design Constraints sections. |
| 1.1 | 01/26/25 | Jas King | Revisions were made to the Executive Summary, Design Constraints, and Domain Model sections. |
| 1.2 | 02/09/25 | Jas King | Revisions were made to the Evaluation section for Server Side, Client Side, and Development Tools. |
| 1.3 | 02/22/25 | Jas King | Revisions were made to the Recommendations section. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room would like to facilitate the development of a web-based gaming app that serves numerous platforms called “Draw it or Lose it”, loosely based on the 1980’s game show “Win, Lose, or Draw”. The application is currently only available to android devices. The client wants the app to be designed to show images from a large library of stock drawings that teams will compete to guess the phrase, title, or thing being shown.

## 

## Requirements

The client requests that the following requirements be met during the development of this app:

* The game will have the ability to have one or more teams involved
* Each team will have multiple players assigned to it
* The game must consist of four rounds, each lasting one minute
* Drawings must be shown at a steady rate and are fully complete at the 30-second mark
* The game must offer remaining teams 1 guess with a 15-second time limit if the team whose turn it is does not guess the puzzle before the time expires

## [Design Constraints](#_2et92p0)

The development of the “Draw It or Lose It” web application involves the following design constraints identified below:

* The game application must be developed for a web-based platform, only available to android devices, that is compatible with the network connection and security constraints for this project
* The game and team names, chosen by the users, must be unique to allow users to check whether a name choice already exists and can be used
* The application should allow only one instance of the game existing in the memory at any given time by creating unique identifiers for each instance of a game, team, or player

## [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 UML class diagram provided below depict the classes that exist within the “Draw It or Lose It” game application system. The diagram shows Entity establishing a relationship between the Game, Team, and Player class and highlighting one of the pillar of object-oriented programming, inheritance, where Entity is the base superclass. The Entity class contains attributes that are shared by all subclasses like ‘id’ or ‘name’. When looking at the relationships established between the classes below, we see that the Game, Team, and Player classes are subclasses that represent the main entities within the game application, and therefore, extend to the Entity class. Another pillar of object-oriented programming, encapsulation, is established through the GameService class as it encapsulates its attributes. Finally, by ensuring only one instance gets created promotes the final pillar, abstraction, by managing the games, teams, and players through only showing interactions that are essential to the user.

**"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.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | The Mac operating system offers tools for developers and everyday users, including a built-in Apache web server for easy web hosting and command line tools like Python for scripting and app development.  Setting up a Mac is generally simpler than on Windows, providing a better experience for IT staff and users due to its user-friendly design. However, macOS primarily supports Apple devices, limiting hardware options for those seeking customization or cost-effective alternatives. | Linux features a powerful command system and flexible command-line interface for system management, task automation, and issue resolution. Customizable open-source web servers like Apache make it a cost-effective option for hosting.  Linux servers provide stability and reliability for heavy workloads, ideal for businesses requiring high availability.  Linux enhances security through user permissions and firewall settings protecting against unauthorized access. Regular updates and strong support further bolster its security. | Windows provides a diverse array of software applications, including productivity and creative tools that are hard to virtualize on other platforms.  Additionally, Windows supports web extensions for better browser functionality, security, and accessibility, along with strong SQL support for effective database management.  Windows' extensive software ecosystem, user-friendly interface, web extension support, and robust SQL capabilities make it a good choice for productivity and user comfort. | Coding the backend and hosting the mobile app independently is impractical compared to using established operating systems like Linux, Mac, or Windows. This approach can waste time and resources on setup and maintenance, taking focus away from core app functionalities.  Moreover, if it doesn't match our team's current expertise, necessitating additional training in server management could reduce efficiency and product quality. Cross-app development also complicates coordination and may lead to compatibility issues and a fragmented user experience.  In contrast, established operating systems allow us to utilize existing tools and community support, streamlining development and enabling our team to focus on building a high-quality, user-friendly mobile app. |
| **Client Side** | Mac systems generally come at a higher price compared to Windows systems. In terms of user-friendliness, Mac is on par with Windows, though users may need a short to moderate adjustment period because of its intuitive interface. To ensure compatibility across different browsers and devices, implementing a responsive design is crucial. This may involve additional time spent on testing and optimization to ensure consistent user experience across various platforms. | Linux is beneficial for software and web developers due to its low cost and the abundance of open-source applications. However, it usually requires more time to master than Mac and Windows. To ensure compatibility across browsers and devices, applications must be designed responsively, considering various screen sizes and browser functionalities. This process demands a good grasp of responsive design principles and extensive testing across different browsers. | Unlike Linux and Mac operating systems, Windows offers a variety of distinctive tools that can only be emulated on other platforms. Furthermore, it provides comprehensive support for the development of web applications and websites and to virtualize other operating systems. Implementing a responsive design is essential for guaranteeing compatibility across diverse browsers and devices, leading to extra development time to ensure that the application operates effectively on all intended browsers. | Clients can access the application anytime and anywhere, requiring adjustments in development for various screen sizes to consider all potential dimensions for tablets, smartphones, and web browsers and optimal functionality on iOS and Android devices. This may require expertise in mobile web development and extensive testing across different devices. |
| **Development Tools** | Visual Studio Code provides features like syntax highlighting and code previews. Homebrew is used to install Unix and Mac utilities. Important languages that work on any platform are JavaScript, HTML, CSS, and SQL, along with Chrome's developer tools. Java and Python are also common languages, with popular IDEs such as Eclipse and IntelliJ IDEA. Large projects might need several development teams. | Essential tools for software development include Visual Studio Code, Atom, Vim, the bash command line, Git, Node.js, and Flask. Common programming languages are Java, Python, and JavaScript, with popular IDEs like Eclipse and IntelliJ IDEA. Tool and language choices vary by platform. | Modern software development relies on important tools like Visual Studio Code, Gvim, Git for Windows, Git Bash, Node.js, npm, and Yarn. Common programming languages are Java, Python, and JavaScript, and popular IDEs are Eclipse and Visual Studio. The choice of tools and languages depends on the target platform. | \*\*Web Browsers:\*\* The website must be compatible with mobile browsers like Firefox, Opera, Samsung Browser, Chrome, and Metro Browser. Testing on all is essential.  \*\*Application:\*\* JavaScript must be enabled on iOS and Android devices to access the app, available on Google Play and the Apple App Store. Android development typically uses Java with Android Studio, while iOS uses Swift with Xcode, with various development approaches based on the specific platforms. |

## 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 Windows operating system is the best platform choice for this project for several strong reasons. It is widely used in many fields, which means many users are already comfortable with it. This familiarity is important because it makes it easier for new users to learn and improves their overall experience, allowing them to use the application more effectively. The Windows platform integrates well with the Android app "Draw It or Lose It," enabling seamless device switching and enhancing the app's functionality and appeal. Leveraging features from the Android version would simplify development and ensure a consistent user experience across multiple platforms. Windows provides key resources for cross-platform application development, including IDEs that support various languages and frameworks. These IDEs enhance productivity with features like code completion, debugging tools, and version control integration, facilitating smoother teamwork. Running a project on Windows typically costs less than other operating systems due to the availability of free or low-cost development tools and the familiarity many developers have with it, which reduces training costs and accelerates development.

1. **Operating Systems Architectures**:

Windows architecture promotes strong interaction between applications and the kernel while without interfering with critical processes. This design allows applications to fully utilize the Windows platform, effective memory management, and access to essential system processes. By keeping user applications separate from the kernel, Windows ensures stability and security, protecting the system from issues caused by any faulty applications. With Windows 10, the Universal Windows Platform (UWP) was launched, enhancing Windows’ Runtime model. UWP provides a unified framework for developing applications across devices such as PCs, tablets, smartphones, and IoT, supporting multiple APIs that enable developers to use existing code and libraries while accessing new features. Developers can use a dual stack approach to create a single application that provides a consistent user experience across multiple device types. This method simplifies development by reducing codebases and maintenance, making it ideal for projects targeting a broad audience. Windows' design and UWP improvements enable developers to create flexible applications that utilize the operating system's features while ensuring security. Which is the best approach for this project.

1. **Storage Management**:

Server-based storage enhances data management and access with failover clustering by automatically redirecting requests to a backup server during hardware failures, redundancy, and advanced tools. It centralizes file access, automates tasks, boosts collaboration, ensures high availability, and protects data.

The Windows operating system provides tools like Disk Management and Storage Sense for efficient storage management. Disk Management allows users to create partitions, format disks, and manage volumes, while Disk Cleanup and Storage Sense are used for managing storage. Disk Cleanup removes unnecessary files and Storage Sense automates file removal and moves data to the cloud when local space is low. Together, they optimize storage, allowing users to worry less about space issues, ideal for this project’s storage needs.

1. **Memory Management**:

Windows includes Memory Management, essential for optimizing application performance, particularly in game applications. A comprehensive database of game-related image files allows for quick access to graphics, reducing load times and improving overall performance. The Windows operating system also offers Azure Storage, a cloud service for flexible and secure storage of images, videos, and game assets, allowing developers to store large data volumes with reliable availability and backup for smooth gameplay. Windows has virtual and physical address spaces for efficient memory management, enhanced by Azure Storage. The virtual memory system enables applications to utilize more memory than physically available, benefiting memory-intensive games and ensuring smooth performance on devices with limited RAM. Windows offers tools for managing file and application versions, including OneDrive for cloud storage with version control and Visual Studio for code management and collaboration. Azure Cloud services enhance Windows with tools for developing, deploying, and managing scalable applications, improving performance and reliability for efficient, responsive gaming experiences.

1. **Distributed Systems and Networks**:

A cross-platform development environment streamlines app creation by minimizing required skills, enabling developers to build apps for multiple operating systems with minimal changes. This allows teams to focus more on enhancing features and user experience. To minimize connectivity issues, servers should be well-planned and tested for client demands, ensuring suitable specifications for a smooth gaming experience during potential peak times. Based on this game, a client-server distribution system should be employed where each client app relies on a single server app. This centralizes game data for real-time synchronization and simplifies updates. Each client can also optimize performance based on its system's strengths for various screen sizes. A strong server network is also crucial for the game's success, enabling seamless player connections with reliable load balancing, backup systems, and security measures to ensure smooth gameplay and protect user data.

1. **Security**:

To minimize data breach risks, strong security measures are essential for protecting sensitive information. Individuals and organizations must actively safeguard their data from unauthorized access and cyber threats through advanced technologies and awareness of evolving vulnerabilities. Windows Defender is a built-in Windows security tool offering basic malware protection, real-time defense, a firewall, and regular updates. However, it may not match specialized services like Aura, which use advanced threat detection, machine learning, and dedicated support to combat a broader range of threats, including phishing and identity theft. To enhance data security, strong encryption methods should be used for all shared data. Encryption protects sensitive information from unauthorized access by converting it into a coded form readable only by authorized users. End-to-end encryption further safeguards data during transmission, minimizing exposure risks. Organizations should implement multi-factor authentication (MFA) with encryption to enhance security. MFA requires users to verify their identity through multiple methods, making unauthorized access harder and promoting data protection awareness. Regular security audits and vulnerability assessments are essential for organizations to identify and address weaknesses in their security systems, ensuring they stay updated against new threats.