

< 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 | 09/24/2023 | Antoinique Curry | <Brief description of changes in this revision> |

**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 goal is to design and develop a user-friendly, interactive, and engaging web-based gaming application that faithfully recreates the "Draw It or Lose It" experience! Teams can compete and enjoy a digital version of the popular guessing game. This involves not only implementing the drawing mechanics and timer functionality but also creating a user interface that ensures an enjoyable gaming experience for players. Additionally, it is crucial to keep scalability and potential future hardware requirements in mind as the project progresses.

## 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](#_2et92p0)

## Game involves addressing real-time interaction, handling a large library of drawings, managing game rounds and timing, rendering images efficiently, and implementing features for player guessing and scalability. Additionally, ensuring security, cross platform compatibility, a excellent user experience, and regulatory compliance are IMPORTANT aspects of successful application development.

## [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 shows a structured relationship between entities in the system, highlighting inheritance, composition (aggregation), and association (dependency) as key object-oriented programming principles. These principles contribute to the design and organization of software, making it easier to model and manage more complicated relationships in the application.

**"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** | <Evaluate Mac for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Linux for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Windows for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> |
| **Client Side** | Supporting multiple types of clients on the Mac platform involves considering costs associated with hardware, software, and licenses, managing development time based on familiarity and project complexity, and ensuring your development team possesses the necessary programming, design, security, and optimization expertise for macOS | Multiple types of clients on the Linux platform offers cost advantages due to open source development tools and very cheap licensing fees. BUT it may include additional time for testing across Linux distributions and desktops. | Windows may involve costs associated with development tools and licensing, as well as hardware and testing expenses. Development time can vary based on project complexity and testing requirements. | Clients on mobile devices have specific costs related to development tools, app store fees, and device testing. Development time can vary based on how difficult of the app and the need for the testing and debugging. Expertise in platform-specific languages, APIs, UI design, performance optimization, security, and compliance with app store guidelines is crucial for successful mobile app development. |
| **Development Tools** | GitHub..These programming languages and development tools are essential for building and deploying software on the Mac platform. Depending on your project requirements, you may also need extra libraries, frameworks, and tools to address features or functionalities in your macOS application. | Java, Python, JavaScript just to name a few. These programming languages and development tools are essential for building and deploying software on the Linux platform. | Java | IoS and Android and Java development!  These programming languages and development tools are essential for building and deploying software on mobile devices. |

## 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**: Since the game is already on Android(mobile devices), I would recommend expanding the game to the Windows server which is also known as Microsoft Windows platform. Windows is another popular OS for clients to have a excellent, user-friendly gaming experience.
2. **Operating Systems Architectures**: Microsoft Windows has a rich history of operating platform architectures that have evolved over the years to meet changing technology needs and accommodate various hardware platforms. Starting with Windows 1.0 to the newest version Windows 11. For this game we recommend using the latest and newest most secure Windows OS which is Windows 11. Windows 11 is designed for a more streamlined, modern user experience. It includes a centered Start menu, enhanced gaming features, and improved compatibility with Android apps.
3. **Storage Management**: Windows 11 has some built in tools for storage management and optimization. Here are a few of the storage management features:

**Disk Cleanup**- A traditional utility in Windows that allows you to remove unnecessary files, such as temporary files and system files that are no longer needed.

**Storage Settings**- It allows you to view and manage storage across different categories, such as Apps, Documents, and Media. allows you to view and manage storage across different categories, such as Apps, Documents, and Media.

**External Storage Management**- Windows 11 includes tools for managing external storage devices, such as USB drives and external hard drives. You can format, eject, and optimize these devices.

**App Uninstallation-** You can uninstall applications and programs that you no longer need to free up storage space.

1. **Memory Management**: Windows uses a **Virtual Memory system** to manage memory. It extends the system's physical RAM with disk space to accommodate running applications. When physical RAM is exhausted, data is temporarily swapped to the hard drive, allowing more programs to run.

**Physical RAM:** The amount of physical RAM in your computer directly affects system performance. More RAM allows for smoother multitasking and faster application response times.

**Memory Protection:** Windows enforces memory protection to prevent one application from accessing or modifying another application's memory space. This feature right here is VERY CRUICAL!

1. **Distributed Systems and Networks**: Cross-Platform Compatibility- By doing this it will ensure compatibility with various devices and platforms, including Windows, macOS, Linux, Android, and iOS.

**Client-Server Architecture:** Implement a client-server model where clients (players) interact with a central server responsible for managing the game, hosting game sessions, and facilitating communication.

**Multi-Tier Architecture:** Break down the software into multiple tiers, including presentation (UI), application game logic, and databases.

**Data Synchronization**: Implement data synchronization mechanisms to ensure that game data and user progress are consistent across platforms. This includes ensuring that drawings and guesses are synchronized in real-time.

By implementing just these few elements to “Draw It or Lose It" game, you can create a robust and cross-platform experience that can adapt to the various challenges posed by network connectivity and outages. Regular monitoring and updates will be necessary to ensure the system's ongoing performance and reliability.

1. **Security**: Security is a critical consideration for Microsoft Windows systems, below are some of the security features and practices:

**Keep Windows Updated**- Make sure to regularly install Windows updates and security patches. Microsoft releases updates to address vulnerabilities and improve system security.

**Enable Firewall-** Make sure to activate the Windows Firewall to control incoming and outgoing network traffic. Configure firewall rules to allow only necessary services and applications.

**Antivirus Software**- Install and keep antivirus software up to date to protect against malware, including viruses, Trojans, and ransomware.

**Windows Defender-** Windows Defender is Microsoft's built-in antivirus and anti-malware software. Ensure it is active and regularly updated.