

Gaming Room

# **CS 230 Project Software Design Template**

Version 1.0

Project 3

## 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 |
| --- | --- | --- | --- |
| 3.0 | 12/15/2024 | Babatope Ayeni | The Game Room |

## [Executive Summary](#_sbfa50wo7nsh)

The demand for mobile applications and web-based games has surged significantly, presenting both opportunities and challenges for developers. Creative Technology Solutions (CTS) has partnered with The Gaming Room to address one such challenge. Their existing Android-only game, *Draw It or Lose It*, is popular but constrained by its limited platform availability. Additionally, similar games have faced issues with gameplay mechanics, such as requiring players to manually draw images on an easel, which often leads to delays and detracts from user enjoyment.

To overcome these limitations, CTS is spearheading the development of a web-based version of *Draw It or Lose It* that can run seamlessly across multiple platforms. This version will replace the manual drawing feature with a large library of stock images to serve as visual clues, ensuring smoother and more engaging gameplay. The goal is to create an intuitive, scalable, and versatile application that meets the growing demand for cross-platform compatibility and delivers a seamless experience for a diverse user base.

## [Design Constraints](#_2et92p0)

In creating web-based software, certain design constraints must be taken into account. The design phase is one of the most crucial stages in the development of an application. Examples of these constraints include UML diagrams, class diagrams, and ESS diagrams. These non-functional components provide developers with a visual representation of the application’s requirements, helping them better understand the system. Such constraints define key elements like artifacts, actors, actions, classes, and roles, which contribute to clearer documentation and comprehension of the software being built. Additionally, they equip developers with the necessary programming tools, technology specifications, and other critical requirements provided by the clients.

## [Domain Model](#_8h2ehzxfam4o)

The primary object-oriented programming principle demonstrated in the UML diagram below is inheritance. This principle allows the SingletonTester class to carry out activities and responsibilities inherited from the main class. Specifically, the SingletonTester class derives from the ProgramDriver class, which is identified as the main class in the diagram. The UML diagram consists of seven classes: ProgramDriver, GameService, Team, SingletonTester, Entity, Player, and Game. The Entity class, serving as a parent interface, is connected to four child classes. Additionally, GameService, Game, Team, and Player are interrelated through an association relationship, where each entity depends on the others.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | | **Mobile Devices** |
| --- | --- | --- | --- | --- | --- |
| **Server Side** | **Characteristics**: It is popular in web hosting. **Advantages**: It is upgradeable and offers various options for different web hosting requirements.  **Disadvantages**: It is less preferred for web hosting services. | **Characteristics**: Secured, most preferred.  **Advantages**: Security flaws are often caught before they become an issue, making it the most preferred choice for web hosting services.  **Disadvantages**: It is more difficult to find applications that support the specific web hosting needs. | **Characteristics**: It is dominant compared to other platforms.  **Advantages**: Close platform, high resource requirements, less loading time, high comfortability.  **Disadvantages**: Easy virus susceptibility, poor technical support. | | **Characteristics**: More popular, high portability.  **Advantages**: Has a wider reach, better compatibility, and is cost-effective.  **Disadvantages**: Highly selective to various smart mobile devices, poor security. |
| **Client Side** | Developing software for Mac clients requires a high level of expertise. It is an expensive option, as clients are typically charged monthly. Additionally, accessing the software can take a considerable amount of time. | High expertise is needed due to the limited availability of compatible applications. Although loading times are reduced, it remains a costly option because it is less widely used. | Development for Windows clients requires significant expertise due to high resource demands. While it offers faster loading times, the high resource requirements make it an expensive option. | | Development for mobile devices is cost-effective, with minimal loading time for pages. It is a popular choice with widespread use, providing high-quality technical support for clients. |
| **Development Tools** | PHP programming language, JavaScript | PHP programming language | | Java programming, HTML/CSS, NetBeans | Android Studio, Android programming |

**Recommendations**

#### Operating Platform

The recommended operating platform for the development and deployment of Draw It or Lose It is the Windows operating system. Windows is a widely used platform that offers high compatibility with web-based software development, making it an ideal choice for expanding the application to multiple operating systems. Its key advantages include high security, affordability, and efficient performance, ensuring the application runs seamlessly across diverse environments. Additionally, Windows supports portability, allowing the software to adapt easily to different devices and platforms.

#### Operating System Architecture

The Windows operating system architecture is designed with two primary components: user mode and kernel mode.

1. **User Mode**: Applications and programs operate in this mode with restricted access to system resources. This ensures that any malfunctioning application does not compromise the entire system.
2. **Kernel Mode**: The kernel has unrestricted access to system memory and hardware, enabling it to manage essential system functions, such as memory, processes, and device drivers.

Windows uses a preemptive multitasking model and is compatible with both symmetric multiprocessing (SMP) and uniprocessor configurations. It processes input and output (I/O) requests efficiently through a packet-driven model, which ensures high-speed data processing and responsiveness critical for a game like Draw It or Lose It.

#### Storage Management

For storage management, a Database Management System (DBMS) is recommended. The DBMS integrates seamlessly with the Windows platform and provides:

* **User-Friendliness**: Simplified management of the game’s large image library and user data.
* **Scalability**: The ability to handle future expansions or increased player demand.
* **Cross-Platform Compatibility**: Ensures smooth operation across other operating systems as the game expands.

The DBMS will efficiently manage the 200 high-definition image files, each approximately 8MB in size, required for the game’s library, while maintaining fast retrieval times and reliable data integrity.

#### Memory Management

Windows’ memory management techniques are well-suited to handle the resource-intensive requirements of Draw It or Lose It:

1. **Memory Compression**: Optimizes RAM usage by compressing less frequently accessed data, improving responsiveness during gameplay.
2. **Paging System**: Implements a page file system to move excess memory pages from RAM to the hard disk, ensuring consistent performance even when memory demands exceed physical RAM capacity.

These techniques maintain a smooth gaming experience by minimizing delays and ensuring that system resources are efficiently allocated, especially during peak usage.

#### Distributed Systems and Networks

To enable communication across various platforms, a distributed system using a Local Area Network (LAN) is recommended. This setup will:

* **Support Multiple Instances**: Allow multiple instances of the game to operate simultaneously.
* **Ensure Reliability**: Use hubs or switches to enhance connectivity and minimize outages. Hubs amplify signals to maintain network stability over long distances.
* **Simplify Maintenance**: Provide an easy-to-manage infrastructure that ensures minimal downtime.

This distributed network will allow seamless interaction between players on different platforms, ensuring that gameplay is uninterrupted even if one node in the system experiences issues.

#### Security

To safeguard user data and ensure secure gameplay, robust security measures are essential. Windows offers advanced security features that will enhance data protection, including:

* **Encryption**: Encrypt sensitive user data to prevent unauthorized access during transmission between platforms.
* **Access Controls**: Implement role-based access to restrict sensitive system areas to authorized personnel only.
* **Regular Updates**: Leverage Windows’ regular security updates to mitigate vulnerabilities and keep the system secure against emerging threats.

These measures will ensure that user information remains confidential and secure, building trust with the client and end-users alike.

### Conclusion

By leveraging the Windows platform’s advanced architecture, robust memory and storage management techniques, reliable distributed systems, and top-tier security features, Draw It or Lose It can successfully expand into a web-based application. This approach ensures scalability, reliability, and an engaging user experience, meeting the client’s requirements for cross-platform functionality and enhanced security.