

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 | 05/20/2023 | Elana Masboob | The cover page was changed, document revision history added, new executive summary, design constraints, system architecture view, and new domain model with recommendation. |
| 1.2 | 6/1/2023 | Elana Masboob | Development requirements for MacOS, Linux, Windows, and Mobile Devices. |
| 1.3 | 6/17/2023 | Elana Masboob | New recommendations added |

**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 wants a web-based game that can run on multiple platforms including MacOS, Linux, Windows, and Mobile applications. Since the game, “Draw It or Lose It” is only available on Android systems, we are going to increase our player base.

The purpose of the game is for teams to guess images out of a library until the time runs out. There are multiple teams, consisting of several people working together to guess the images given.

## 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)

* There must be multiple teams involved, each with its own set of team members.
* Team members must be allowed to choose a unique name for their player as well as team. Team names in use will prompt the player to change the name to increase creativity and avoid duplicated team names.
* The game must run on MacOS, Linux, Windows, and all other mobile devices.
* Only one instance of the app should always exist.

## [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 diagram shows us how the classes are related to each other. Program Driver and Singleton Tester classes are aligned by a solid black arrow, the uses of “<<>>” lets us know that Program Driver employs Singleton Tester to let us know that there is only one instance during execution. Since Game Service and Game have a binary association, there is only one relationship between the two. A team and player also have a binary association, meaning a relationship between two classes. The classes of game, team, and player all show an inheritance relationship with the class entity.

**"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** | Mac is equipped with an Apache web server and a wide range of command line tools, such as Python, making it especially easy for configuration and access.  MacOS provides user-friendly options by creating a smooth user experience through perfect integration.  It is integral to note that Mac has specific hardware limitations making it a possibility of hardware configuration issues. This could become an issue for users looking for hardware requirements that Mac does not offer. | Linux provides a robust command system and web server that delivers similar functionality to MacOS and Windows. Linux servers are cost-effective, offering a more budget-friendly options (in comparison to MacOS and Windows)  Linux servers are renowned for their stability and reliability, allowing them to operate continuously for extended periods without encountering many issues. Linux servers can also establish high level security, effectively mitigating viruses. | Windows has several characteristics, advantages, and weaknesses for hosting web-based software applications.  Its user-friendly interface, wide software compatibility, integrated development tools, and active community contribute to its appeal.  Windows integrate best with Microsoft technologies, IIS web servers, regular updates, and user familiarity.  When using Windows as a server, it’s important to note that the costs of licenses compared to open-source alternatives is higher than Linux servers. | Coding backend and hosting the mobile app may not be as viable in comparison to Mac, Linux, and Windows.  It may not align well with the specific skillset of your team, which includes expertise in react-native-web development and Flask.  Implementing this approach would require building and hosting the sever from scratch, along with handling cross-app development challenges. |
| **Client Side** | In general, Mac tends to be more expensive compared to Windows and Linux.  However, both operating systems offer a similar level of ease of use, requiring a short to moderate amount of time to learn due to their intuitive interfaces. | Linux is highly regarded among software and web developers due to its cost-effectiveness and wide range of open-source programs that seamlessly integrate with the system.  However, it is important to note that Linux has a steeper learning curve compared to the default operating systems of Mac and Windows, requiring a significant amount of time to become proficient. | Windows stands out from Linux and Mac in terms of its wide array of exclusive tools that cannot be easily virtualized on other operating systems.  It offers extensive support for web application and website development, providing developers with a rich ecosystem of resources.  Additionally, Windows has the capability to virtualize other operating systems, further enhancing its versatility. | Mobile devices enable clients to access the app with flexibility, regardless of location or time.  Development adjustments are necessary to accommodate variations in screen real estate.  It is crucial to consider all possible screen sizes for tablets, smartphones, and browsers.  Mobile apps should prioritize an intuitive interface tailored to the small form factor of mobile devices. |
| **Development Tools** | VSCode offers syntax highlighting and code previews, while the Homebrew package manager facilitates the installation of Unix and Mac utilities. For development, Xcode serves as a comprehensive IDE, iTerm2 functions as a versatile emulator, and Tower acts as a dedicated git client. The Dash API browser provides a convenient resource for accessing documentation. Moreover, the platform-agnostic language set includes JavaScript, HTML, CSS, React, React Native, and React Native Web. Chrome development tools and SQL are also available for efficient web development and database management. | Visual Studio Code, Atom, Vim, and the bash command line are popular development tools, while Git is a widely used version control system. Node.js and Flask are frameworks that provide server-side functionality. In terms of programming languages, JavaScript, HTML, CSS, React, React Native, and React Native Web are frequently employed. Package managers such as npm and yarn facilitate dependency management. Chrome development tools aid in debugging and optimizing web applications, and MySQL is a widely adopted relational database management system. | Visual Studio Code, Gvim, Git for Windows, and Git Bash are popular tools for software development and version control. Node.js, npm, and yarn are commonly used for server-side JavaScript development and package management. In terms of programming languages, JavaScript, HTML, CSS, React, React Native, and React Native Web are widely employed. Chrome development tools provide debugging and optimization capabilities for web applications, while MySQL is a prevalent choice for relational database management. | The website should be designed to function properly on various mobile browsers, including Firefox, Opera, Samsung browser, Chrome, and the Metro browser. It is important to thoroughly test the website and the app itself on different mobile browsers. For the app, ensure that JavaScript is enabled on both iOS and Android devices to enable app access. Additionally, make the app available for download on the Google Play Store and Apple App Store. |

## 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 best recommended platform for this game would be Windows due to ease of access.

As previously mentioned, Windows is still the most suitable platform for The Gaming Room. When looking into the expansion of Draw It or Lose It, Windows would certainly outperform other platforms. Some reasons as to why Windows is most suitable for us are:

* + Windows is built within Android Operating Systems who frequently update their software to the best available.
  + Ease of access allows developers to explore creativity within the app; develop, design, and deploy apps and solutions for Windows.
  + Many different emulators for powershell, unbuntu, and cmd capabilities on Windows that allow for testing on every platform available.
  + Windows has the largest user base and most popular operating system for gaming.

1. **Operating Systems Architectures**: Windows offers a user-friendly operating system suitable for newcomers, enabling developers of all skill levels to work with it effortlessly. It furnishes essential services utilized by all Windows-based software, facilitating the display of Graphical User Interfaces (GUIs) while accessing system resources and offering numerous additional functionalities. These services encompass Graphics and Multimedia, messaging, and web-based features, which can be accessed through either a user account or a dedicated server.

Windows divides its operating system into User mode and Kernel mode. User mode processes directly interacting with users and significantly impact their interactive experience, while Kernel mode handles lower-level tasks such as input/output operations, memory management, networking, hardware management, and routines. This separation ensures an efficient system architecture.

Windows also utilizes a hierarchical directory structure for organizing and storing data, enabling easy management and retrieval of information within the system.

Additionally, Windows supports multiprocessing, allowing the execution of multiple tasks simultaneously, which enhances system performance by effectively utilizing computing resources. Windows also embraces hardware modularity, enabling users to customize and adapt their hardware components, providing flexibility and personalization options for their systems.

1. **Storage Management**: Windows 10 comes with a nice feature called storage sense. This allows you to scrutinize and manage files on your hard drive, along with how much space it takes up. Other features include being able to choose to save locations for apps making them easier to find. And just like other dives, you can also use the cloud to save data. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost or carelessly deleted.

We highly recommend Microsoft Azure for storage due to their competitive prices, excellent customer support, and continuous updates and support. Azure offers several noteworthy features, including the ability to deploy Docker containers in their Cloud computing environment for leveraging cloud storage instances. One of the advantages of Azure's cloud-based storage is the ease of scaling up or down based on user demand. This flexibility allows for seamless adjustment of storage capacity, especially during the initial launch when a high influx of users is expected.

Azure provides multiple storage options, such as Azure File system, Azure Storage Containers, and Azure Blob Storage. Each option caters to specific storage needs and offers diverse functionalities. For example, in the case of storing 200 8MB Base Game Images for the "Draw It or Lose It" application, we recommend utilizing Azure File Share, which offers 1.6GB of storage per user. Cloud storage is rapidly gaining popularity as a storage solution due to its cost efficiency and scalability, making it an increasingly favored choice for businesses and individuals alike.

1. **Memory Management**: When developing this game, it is necessary to establish a database or library containing numerous pictures. Memory allocation enables convenient storage of these pictures outside the default picture folder. This ensures that your entire project remains securely organized in a designated area on your computer. This applies even when working with your integrated development environment (IDE) and accessing files from it to create the game.

The latest version of the Windows operating system, Windows 10, has made notable progress in memory management to enhance the speed and efficiency of data loading from memory. These advancements primarily rely on two techniques: disc paging and demand paging, which extend the computer's physical memory or RAM. Disc paging involves setting aside a portion of the hard disk to function as extra RAM. This enables Windows 10 to utilize the disk space as an extension of physical memory, resulting in faster retrieval and processing of data.

In contrast, demand paging divides processes into smaller tasks and loads them into memory only when they are needed for immediate execution. This strategy optimizes memory usage by loading specific components on-demand, leading to more efficient utilization of system resources. Furthermore, Windows 10 allows each process to access the entire virtual memory address space, providing ample memory capacity to meet the requirements of applications. This ensures that applications have enough memory space to operate effectively without encountering limitations. Overall, the memory management improvements in Windows 10 significantly enhance the speed and efficiency of data loading by employing disc paging, demand paging, and maximizing the available virtual memory address space.

1. **Distributed Systems and Networks**: Because each operating system being different I investigated ways to publish the game to run on all dives. I found Develop 4 which enables cross-platform game creation. It’s an IDE that can be run on any device. Once the game is created you can simply export the game file onto the web, iOS, Android, and many more options that will allow cross-play. This will help with dependencies. To prevent other problems like outages or connectivity, the company will need to make sure their servers are strong enough to support large player volumes along with backup power for power outages.

Azure stands out as a preferred cloud service provider due to its seamless accessibility for distributed systems and networks. It offers excellent uptime through cloud-based email alerts, Azure App insights logging, and monitoring solutions. These features are particularly valuable when scaling up to support 1000 concurrent games with 4 players each, guaranteeing reliable and uninterrupted performance for smooth gameplay experiences.

Moreover, Azure's network capabilities allow you to offload the network load, freeing up your focus to concentrate on developing and enhancing your application's features. This enables you to prioritize the core functionalities of your application while Azure takes care of the network infrastructure, ensuring efficient and dependable network operations.

1. **Security**: Windows comes with built-in security protection software. To secure user data and information, it would be recommended to use another source. Though if we are talking about what is on the machine, windows come pre-equipped with protection. This system scans for malware (malicious software), viruses, and security threats. This all happens in real-time, and because threats change the system updates automatically to keep the system and user information safe.

Azure streamlines the security of user information and personal data. When using Azure, devices connect to the "Azure App Service" hosted on an App Service Plan, with secure login authentication managed by Azure Active Directory. Azure offers additional security features, including:

- IP configurations that allow for whitelisting specific access to resources, such as player or personal information, or granting access to the entire app based on designated IP addresses.

- The option to store data in a Virtual Private Network (VPN) within the cloud, providing an extra layer of security.

- Databases can be configured to permit access only from specified IP addresses (IP Whitelist), requiring passwords and enforcing SSL connectivity to safeguard user data.

- Azure provides options to obfuscate user data, ensuring the protection of personal information in the event of a security breach.

It's important to note that an active internet connection is necessary to communicate with Azure's cloud services.