

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 | 07/16/2023 | Mohamed Jaddour | The web development of Draw it or Lose it |

**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 to develop a web-based program game, the game can be run on a multiple platform. The game name is “Draw it or Lose it”, currently the game is only available on Android. In this game, multiple teams composed of multiple players participate in four rounds, with each round lasting one minute. The game involves pulling pictures from a library of images, and the team takes turns guessing the answer within the assigned time. If the answer is not correct and not guessing the right answer within the time limit each member of the opposite team has a chance to answer within 15 second time window.

## Requirements

Needs one or more teams playing.

Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.

Each team has multiple players.

Must run on multiple platforms.

Only one instance of the game exists at any time.

## [Design Constraints](#_2et92p0)

Needs one or more teams playing.

Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.

Each team has multiple players.

Must run on multiple platforms.

Only one instance of the game exists at any time.

These are the guidelines for the code and software development need to be followed. While working on the game aspect, we must consider the application development. The goal is to ensure that the game runs seamlessly on all devices. Currently, is available on android, but we must adapt it for other mobile devices. The game will have to be compatible with the platforms such as Windows, Linux, and Apple.

Firstly, we must rewrite the code in Swift for Apple devices. This would involve adapting the existing codebase to work with the Swift programming language. Secondly, we can explore the possibility of utilizing the existing code and extending it to run on other devices leveraging different programming languages.

## [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 Entity Class creates a relationship between Game, Team, and Player classes. This relationship is achieved through inheritance, as the classes inherit information from Entity class. In UML notation, this inheritance relationship can be represented using the appropriate notation. By making Entity the superclass, common attributes such as “name” and “id’ are shared among all the classes. That allows for the code reusability and promotes a consistent structure across the derived classes. We observe that Team and Player have a “has a” type of relationship, team class contains references to instance of the Player class. The Game class has a reference to the Team class, and the GameService class has references to instances of the Game class. In UML, this type of relationship is aggregation. The concept of “has a,” it stands that one class instance contains a reference to another class instance.

Considering the diagram, the GameService class has a reference to instances of Games, Games has a reference to Team, and Team has a reference to Player. These relationships can be accurately represented using UML notation.

**"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 not well-known for hosting web applications on a large scale; thus, it can host websites.  Using Mac OS, developed by Apple Inc., on the servers host websites or applications, there will be a licensing cost involved. Those costs are usually associated with obtaining the legal right to use the software in a specific manner and for a specific number of installations.  **Advantages:**  Easy to use, integration with the Apple ecosystem will provide a seamless integration with other Apple devices, stability, and reliability, as Apple is well known for uptime and smooth operation.  **Weaknesses:**  Hardware limitations, macOS is designed to run on Apple hardware only, where it can be expensive. Cost, The licensing costs for macOS can be higher than other OS. | Linux is the most popular and preferred server-side development, offering server-based deployment methods where websites can be hosted.  The potential licensing costs, such as Ubuntu Server, CentOS, and Debian, are open source and freely available, and they can be used without any licensing costs.  **Advantages:**  The cost-effectiveness, most Linux distributions are open-source and free, which helps reduce the cost of acquiring the operation system. Security, Linux is designed with security in mind. The open-source nature of the operation system allows constant security.  **Weaknesses:**  Learning curve, Transition to Linux may involve a learning curve, system configurations, and troubleshooting can be challenging. Gaming support, The gaming library on Linux is still less extensive than on Windows. | Windows server developed by Microsoft is mainly designed for server deployments. It includes features and tools tailored for hosting websites and web applications.  Microsoft’s Windows server does come with licensing costs. Furthermore, it will be based on the number of processor cores in the server.  **Advantages:**  User-friendly and familiar interface, and the GUI is intuitive and easy to navigate. Extensive software support, where Windows has a large library with a wide range of applications, software availability, and hardware compatibility, is designed to work with a wide range of hardware. Gaming Windows is the dominant platform for gaming, with an extensive library of games and firm support.  **Weaknesses:**  The cost can be a significant factor, especially for organizations with large numbers of devices. Security Vulnerabilities: It is a target of malware and security threats due to its widespread usage. Privacy concern, Some users have expressed concerns about the data issues. | Mobile operation systems are designed for running applications on mobile devices and are not typically used as server operating systems for hosting websites,     however, there are ways to deploy web services or host websites on mobile devices.  Since mobile operation systems are designed to run on smartphones, tablets, and other mobile devices, there is no typical use for a traditional server platform for hosting websites.  **Advantages:**  User-Friendly Mobile operation system offers intuitive and touch-friendly interfaces that are easy to navigate and use. Hardware Optimization, Mobile OS platforms are optimized to work efficiently with the specific hardware of each device—security measures, where the OS incorporates the security features. Regular Updates, Mobile OS developers regularly release updates.  **Weaknesses:**  App Store restriction, Mobile OS platforms typically have strict app store guidelines and review processes—limited Customization, where they are such limited personalization.  Limited file management, Mobile OS does have limited management capabilities. |
| **Client Side** | Achieving cross-browser and cross-device compatibility demands a moderate level of expertise and time commitment during the application development process. The cost involved is comparable to that of developing for Windows. To ensure compatibility with all web browser platforms and mobile devices, developers must focus on responsive design, conduct comprehensive testing, and employ user-agent detection and feature detection techniques. Additionally, prioritizing accessibility considerations and allocating resources for regular updates and maintenance are vital to delivering a seamless user experience across diverse platforms and devices. | Achieving cross-browser and cross-device compatibility requires the highest level of expertise and a significant time investment during the application development process. However, the associated cost is kept at a minimum. To ensure compatibility with all web browser platforms and mobile devices, developers must meticulously implement responsive design principles, conduct extensive testing across various browsers and devices, and employ user-agent detection and feature detection techniques. Prioritizing accessibility considerations is crucial to providing a seamless user experience across diverse platforms and devices. Regular updates and maintenance are also essential to address compatibility issues and ensure long-term application performance. | Achieving cross-browser and cross-device compatibility demands minimal expertise and a relatively short development timeline. The cost involved is comparable to that of developing for Mac. To ensure compatibility with all web browser platforms and mobile devices, developers must focus on implementing responsive design, conducting thorough testing across various browsers and devices, and utilizing user-agent detection and feature detection techniques. Prioritizing accessibility considerations is essential to deliver a seamless user experience across different platforms and devices. Regular updates and maintenance are also important to address any compatibility issues that may arise and to keep the application performing optimally. | The application provides a remarkable level of flexibility, empowering both clients and developers to access updates and stay connected from any location. Whether it's clients wanting to view real-time progress or developers seeking to make modifications on the go, the application ensures seamless access to the latest information and changes from any place with an internet connection. |
| **Development Tools** | On Macs, Swift is popular for development. Notepad++ enhances the coding experience. Macs support various languages, including HTML, CSS, JavaScript, Java, Python, PHP, and Ruby, enabling diverse application and project creation. | Linux offers a diverse array of user-friendly development tools like Visual Studio, Eclipse, and Notepad++. It seamlessly integrates with various programming languages, including HTML, CSS, JavaScript, Java, Python, PHP, and Ruby, empowering developers to explore endless possibilities for creating innovative applications and projects. | User-friendly with Linux compatibility. Supports Visual Studio, Eclipse, and more. Simple tools like Notepad++. Languages: HTML, CSS, JavaScript, Java, Python, PHP, Ruby. | Create countless apps with Android and Swift. Both languages and software work on all three machines (Mac, Windows, Linux). Supported languages include HTML, CSS, JavaScript, as well as libraries for frontend and general-purpose use, such as Java, Python, PHP, and Ruby. |

## 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 Gaming Room project should be initiated on Windows devices considering the software options, user-friendliness, and cost-effectiveness. Windows presents an extensive array of software choices for IDEs. Also, Windows is known for its user-friendly attributes.
2. **Operating Systems Architectures**: Windows provides fundamental tools utilized by all applications designed for the Windows platform, streamlining the development of Graphical User Interfaces, and enabling seamless access to system resources. This software ecosystem encompasses a diverse range of messaging and web services.
3. **Storage Management**: Windows 11 introduces an upgraded function called Storage Sense, presenting sophisticated file management capabilities tailored for your hard drive. Furthermore, Windows 11 grants the user the convenient ability to designate preferred storage locations for applications, ensuring effortless accessibility. Additionally, it facilitates secure data retention for businesses, featuring an integrated storage system that simplifies organization and file creation, ultimately promoting a structured and safeguarded approach to data management.
4. **Memory Management**: As the game initialization unfolds, creating a comprehensive image database becomes pivotal. The allocation of memory assumes a crucial role in enabling the storage of these images within the designated picture folder. This strategy contributes to the secure consolidation of the entire project on the computer, ensuring the cohesive presence of all game-related elements, including files accessed through the Integrated Development Environment.
5. **Distributed Systems and Networks**: Recognizing the disparities inherent in various operating systems, a thorough investigation was undertaken to delve into strategies for releasing the game in a manner that guarantees harmonious performance across an array of devices. In this endeavor, Develop 4, an Integrated Development Environment (IDE) tailor-made for cross-platform game creation, was conceived. Once the game takes shape within the Develop 4 ecosystem, the subsequent process of exporting the game file becomes notably straightforward. This streamlined mechanism facilitates effortless deployment across diverse platforms, encompassing web, iOS, and Android interfaces. By harnessing the capabilities of Develop 4 for cross-platform development and making strategic investments in a robust server infrastructure, the company can effectively tackle compatibility challenges while elevating the overall stability and dependability of the game.
6. **Security**: Windows offers inherent security safeguards to its users. It is advisable to contemplate the incorporation of supplementary security measures to fortify the protection of user data and sensitive information. The intrinsic security features entail routine scans aimed at identifying and eradicating malware, viruses, and various cyber hazards. Additionally, Windows autonomously updates the system to stay abreast of emerging threats and vulnerabilities.

**Sources:**

Balsom, J. (2018, August 6). *Understanding security in windows server environment*. Medium. https://netcomlearning.medium.com/understanding-security-in-windows-server-environment-2df4db6dc707

Wwlpublish. (n.d.). *Windows server file servers and storage management - training*. Training | Microsoft Learn. https://learn.microsoft.com/en-us/training/paths/windows-server-file-servers-storage-management/

Wikimedia Foundation. (2023, July 30). *Architecture of windows NT*. Wikipedia. https://en.wikipedia.org/wiki/Architecture\_of\_Windows\_NT